

משרד החינוך
מינהל תקשוב, טכנולוגיה ומערכות מידע
הפיקוח על מדעי המחשב והנדסת תוכנה

תכנית ללימודי מדעי המחשב ורובוטיקה

בתי הספר היסודיים

טיוטה לעדכון משנת התשע״ז

אוקטובר 2016, תשרי התשע״ז

דבר ד״ר עופר רימון, מנהל המנהל לתקשוב, טכנולוגיה ומערכות מידע

החינוך הוא מרכיב מרכזי בחוסן הכלכלי, החברתי ואף הביטחוני של כל מדינה ובמיוחד מדינת ישראל. מערכת החינוך בישראל והחינוך המדעי טכנולוגי בפרט, מהווים את הבסיס להיווצרות 'STARTUP NATION'.

התפתחות תעשיית הידע בישראל הובילה להתפתחות תכניות לימוד עדכניות ומתקדמות בתיכון בתחום מדעי המחשב ובתחומי התמחות בהנדסת תוכנה. כך למשל פתחנו מגמה חדשה לגילוי ואיתור מידע דיגיטאלי; התמחות חדשה בהגנת סייבר; תכניות לימוד חדשות כמו מערכות גרפיות ומערכות מומחה בשירותי אינטרנט ובטלפונים ניידים.

במקביל להרחבה והעמקה של התכניות בחטיבה העליונה, פיתחנו תכנית ללימודי מדעי המחשב בחטיבת הביניים לתלמידים מצטיינים במסגרת **עתודה מדעית טכנולוגית**.

התכנית ב**מדעי המחשב ורובוטיקה לבית ספר היסודי** משלימה חלק נוסף ומרכזי בפאזל של תכניות הלימוד במדעי המחשב ובהנדסת תוכנה ויוצרת רצף מיסודי ועד לכיתה י״ב, מכלול תכניות אשר חשיבותן אינה רק בתכנים – חשובה יותר זו הכנת תלמידי ישראל לאתגרים של המאה ה- 21. לימודי הרובוטיקה מהווים את הצוהר הטוב ביותר ללימודי טכנולוגיה שכן הם משלבים לימודי מכניקה, אלקטרוניקה ותכנות שמהווים נדבך נוסף ואוריינות נדרשת לתלמידים במאה ה- 21.

אני מאמין ויתרה מזו בטוח שהתכניות שהמנהל למדע וטכנולוגיה מפתח ומיישם יתרמו תרומה משמעותית לחוסנה של המדינה. יש בהן את העוצמה והיכולת ליצור תהליכים שמובילים למוביליות חברתית; להנגשת חינוך מדעי-טכנולוגי איכותי לכל המגזרים; לגישור על פערים בין הפריפריה הגיאוגרפית חברתית למרכז. בזכותן ימומש הרעיון המרכזי בדמוקרטיה – מלבד הזכות להצביע – הזכות לשוויון הזדמנויות לכל תלמיד ותלמידה בישראל.

ד״ר עופר רימון

מנהל מינהל תקשוב טכנולוגיה ומערכות מידע

דבר פרופ׳ שלומי דולב, יו״ר ועדת המקצוע מדעי המחשב במשרד החינוך

כיום יש חשיבות רבה ללימוד מדעי המחשב בבית הספר היסודי. התלמידים שמגיעים לבית הספר היסודי נחשפים מימי תינוקתם למחשבים בצורות שונות, בפרט לטלפונים חכמים, והשימוש במחשבים והאינטראקציה איתם הופכים להיות טבעיים – לכן מערכת החינוך חייבת להתעדכן בהתאם.

לימוד מדעי המחשב ביסודי הופך למוחשי את הלימוד במקצועות המדעיים האחרים כולל מתמטיקה, פיסיקה, מדעי החיים וכימיה. אין צורך להסביר מדוע צריך לספור צעדים נכון כדי שרובוט שמריץ תכנית ינוע לעבר מטרה, והתגובה והבדיקה אינם של מורה אלא של הביצוע של המחשב במציאות. ההרגל שיוקנה לתלמידים – בתכנון לוגי של צעדים, בהפשטה של מצבים ומבני נתונים, ובתגובת המציאות לתכנון – ילווה אותם לאורך כל הלימודים והחיים המקצועיים.

כבר כשכותבים תכנית לימודים במדעי המחשב ועורכים אותה, ועוד שהיא בהדפסה, היא מתחילה להתיישן בקצב מהיר שהרי טכנולוגית המחשבים משתנה ומתפתחת כל יום. לכן במהלך תהליך כתיבת תכנית הלימודים הנחה אותנו הצורך לחשוף את התלמידים לעקרונות מדעי המחשב שמשמשים בסיס מדעי איתן לשנים. בכוונתנו לאסוף כל הזמן חומרים זמינים ברשת שיהיו מעודכנים ומשותפים על ידי כל העוסקים במקצוע: מפקחים, מורים ותלמידים.

החשיפה הראשונה צריכה להיות מהנה ומשעשעת גם לבנים וגם לבנות. צריך שיהיה להם טעם טוב מהמקצוע, ממש כמו תחביב, ורצון להמשיך להעמיק וללמוד, אחרת נוכל לגרום לחלקם להירתע ולא להמשיך ללמוד מדעי המחשב. במיוחד חשוב לאפשר לתלמידים עם יכולות התחלתיות שונות להתקדם בקצב שלהם.

בחרנו להתחיל ללמד בעזרת לימוד שפת סקרטץ׳, שפה עם ממשק בעברית צבעוני וידידותי לילדים. בנוסף כללנו רובוטיקה כחלק מהלימוד בשלבים הראשונים. לימוד רובוטיקה יכול לעזור לתלמידים לקשר בין רמת ההפשטה הנדרשת בתכנון אלגוריתם ובכתיבת התכנית לבין פעולה מוחשית בסביבה הפיזית, סביבה שנשלטת על ידיהם. זו גם התנסות מהנה שיכולה לתרום לאהבת המקצוע.

המורים הם מפתח ראשי בהצלחת התכנית וכאן התמקדו המאמצים. חשוב ביותר להכיר בקושי שמורי המקצוע חייבים להתמודד אתו יום יום לאורך שנים – לימוד ועדכון שעורים מתמיד. לכן היחס למורי מדעי המחשב צריך להיות מיוחד, עם הרבה תמיכה בהכשרה, בחומרי לימוד, בהתחשבות במאמץ הנוסף, בהערכת הביצועים ובתגמול.

תכנית הלימודים ביסודי מחייבת עדכון בחטיבת הביניים ובחטיבה העליונה. וועדת המקצוע מתחילה לגבש רעיונות לגבי תכניות ההמשך. מטרה אפשרית היא הכשרת בוגרי תיכון ברמה שמאפשרת המשך לימוד עצמי, שיפתח בפניהם אפשרות להשתכרות. כך אולי נוכל לבנות מערך לשכבות גדולות באוכלוסייה, שכבות שכיום לא נחשפות ללימודיים אקדמיים, שיאפשר להן להתפרנס בכבוד ללא לימודים אקדמיים או לממן את הלימוד האקדמי. מכך מדינת ישראל תרוויח חברתית וכלכלית.

בתודה לשר נפתלי בנט שפגש אותי בנושא; למנמ״ר דר׳ עופר רימון ולמפמ״ר דר׳ אבי כהן שיזמו ופעלו ומצאו משאבים לכתיבת התכנית; לרוני דיין שמפעיל את התכנית; לכל חברי וועדת המקצוע ובראש וראשונה לפרופ׳ יהודית גל עזר, קודמתי בתפקיד וידי הימנית בוועדת המקצוע; ולוויליאם פרג׳ון, שכתב את התכנית לפי רוח הדברים שנקבעה ובקצב הנדרש בכדי שנצליח לממש את החזון כבר בשנה הקרובה.

לבסוף אאחל הצלחה לדרך שאנו מתחילים.

פרופ׳ שלומי דולב

יו״ר ועדת המקצוע במשרד החינוך

דבר ד״ר אבי כהן, מפמ״ר מדעי המחשב וממונה מגמות עתירות ידע

ברור לכאורה ואין מחלוקת בין אנשי חינוך על יעדים רבים. כולם יסכימו להצהרות הבאות: תכניות לימוד חייבות לכלול רכישת מיומנויות המאה 21 ויישומן, מיומנויות הכוללות: למידת עמיתים; שיתוף פעולה; קבלת החלטות; חיפוש ארגון והרכבת ידע; חשיבה ביקורתית; פתרון בעיות ועוד. ברור לכולם שאת ההוראה יש להתאים ללומד – יש להתחשב במקורות הידע המרובים, להתאים את דרך ההוראה, הלמידה וההערכה בהתאם ליכולות של כל תלמיד, להעצים אינטליגנציות קיימות ולפתח אחרות.

למידה צריכה להתקיים בסביבת למידה פתוחה וגמישה בה ניתן לממש למידה מבוססת בעיה/ פרויקט. תהליכי הל״ה (הוראה, למידה, הערכה) צריכים להיות ספיראליים ולכלול הערכה חלופית ומעצבת, ונושאי התרגול והעבודה הפרויקטנטית צריכים להיות רב תחומיים.

אין ספק שלכלול את כל היעדים והשאיפות הללו זהו אתגר לא פשוט, והשגתם כמכלול היא המטרה המרכזית של מערכת החינוך.

תחום הוראת מדעי המחשב במערכת החינוך בעולם ובמיוחד בישראל יוצר הזדמנות להשגת חלק נכבד מהיעדים. לקיום תכנית לימודים במדעי המחשב בבית ספר יסודי ישנה חשיבות גדולה מאוד. שפת מדעי המחשב היא שפה מרכזית במרחב השפות אותן לומד תלמיד בבית ספר יסודי שכולל את השפה בה הוא קורא וכותב; את השפה הזרה; את שפת המדע-מתמטיקה; את שפת האמנות והגוף ואת שפת הטכנולוגיה – מדעי המחשב. החשיבות של מדעי המחשב כענף מרכזי גדלה והיא משפיעה על כל היבט של חיינו.

תלמיד שפותר בעיה באמצעות תכנית מחשב משתמש במיומנויות הפשטה, ביכולת לנתח ולממש פתרון בעיות/ משימות בעזרת כתיבה והפעלה של תכניות מחשב ובהצגת תכניות למימושן. הוא צריך להיות מסוגל לייצג בעיה/משימה מתחומים שונים (פיסיקה, מתמטיקה, כימיה, ניהול, בעיות/משימות ממדעי החברה והרוח) באמצעות ייצוג/מודל שניתן לתרגמו לתכנית מחשב.

כמיומנות מרכזית של מדעי המחשב ייצוג של בעיה/משימה באמצעות מחשב כולל כתיבת קוד הינה מיומנות חשובה שכל תלמיד הלומד בבית ספר בישראל צריך לרכוש ולדעת ליישם. כפי שמתמטיקה משפיעה על דרך המחשבה והגישה המדעית, מדעי המחשב ממחישים את החשיבות של המחשבה האנליטית ואת האפשרות המיידית ליישמה בחיי היום יום. למדעי המחשב יש השפעה עזה על כל היבט בחיינו ובמידה רבה שפת המחשב היא שפת המדע והטכנולוגיה.

תכנית הלימודים במדעי המחשב לבית הספר היסודי תעזור לפתח בוגרים בעלי אוריינות יצירתית מדעית וטכנולוגית. היא תתרום לגידול ילדים סקרנים ותאבי דעת שמסוגלים ללמוד באופן עצמאי ומשתמשים בידע מדעי וטכנולוגי ובמיומנויות חשיבה ועשייה בכדיי להבין מודלים בתחומים שונים: לייצג מודלים אלו למחשב באמצעות שפת תכנות; לכתוב פתרון תכנותי לבעיות; לבחון את הפתרון לבעיה שהוצגה ואת המענה לפתרון; להשוות בין פתרונות ויעילותם.

השגת מכלול היעדים שהצגתי בתחילת דברי אינה משימה קלה כלל ועיקר אבל תחום מדעי המחשב מהווה פלטפורמה להשגתם ומכלול היעדים שמניתי הם תוצר טבעי של תהליכי הל״ה במדעי המחשב.

תלמידים חוקרים ולומדים לבד, תלמידים משתפים פעולה עם חבריהם, כל תלמיד מתבלט בתחום כישוריו ובא לידי ביטוי בעזרת תהליכי ההוראה והלמידה. הבעיות והפרויקטים אותם אנו מיישמים במדעי המחשב, בהוראה ובלמידה, הם רב תחומיים וכל תלמיד מוצא בהם עניין.

תכנית לימודים זו משתלבת במכלול התכניות במדעי המחשב שבחטיבת הביניים ובתיכון, ובכך נוצר רצף ייחודי ברמה עולמית שאני בטוח שהוא יקדם את התלמידים מכל המגזרים, את מערכת החינוך ואת כלכלת המדינה ורווחתה. העובדה שהתכנית הזו מכוונת ומיועדת לכלל התלמידים תורמת לשוויון הזדמנויות בחינוך ולתהליכי מוביליות חברתית וכלכלית בקרב ילדי ישראל.

אבי כהן

מפמ״ר מדעי המחשב וממונה מגמות עתירות ידע

משרד החינוך התרבות והספורט
המינהל למדע ולטכנולוגיה

הפיקוח על הוראת מדעי המחשב וטכנולוגיות מידע

פרופ' שלומי דולב -	יושב ראש ועדת המקצוע למדעי המחשב, משרד החינוך
פרופ' יהודית גל-עזר -	מ"מ יו"ר ועדת המקצוע; אקדמיה: האוניברסיטה הפתוחה המחלקה למתמטיקה ומדעי המחשב
ד"ר אבי כהן -	מפמ"ר, משרד החינוך, מרכז הועדה
מר רוני דיין -	מנהל אגף טכנולוגיות מידע
ויליאם פרג'ון -	מרכז פרויקט, כתיבת תכנית הלימודים
ד"ר ראובן חוטובלי -	אקדמיה: אפקה – המכללה האקדמית להנדסה בת"א ואוניברסיטת תל אביב; יו"ר וועדת המקצוע להנדסת תכנה וסייבר – משרד החינוך
פרופ' בני שור -	אקדמיה: אוניברסיטת תל אביב, בית הספר למדעי המחשב
פרופ' אבי מנדלסון -	אקדמיה: טכניון חיפה, בית הספר למדעי המחשב
ד"ר ברוריה הברמן -	אקדמיה: מכון טכנולוגי חולון, המחלקה למדעי המחשב; מכון ויצמן
ד"ר מיכל ארמוני -	אקדמיה: מכון ויצמן
ד"ר תמי לפידות -	מנהלת מרכז המורים
גב' אלה מריאנובסקי -	מורה, רכזת מגמה, בית ספר תיכון למדעים 'העלה', לוד
מר שלומי אחנין-	מפמ"ר אלקטרוניקה, עמל פתח-תקווה; מכללת 'הכפר הירוק'
מר יוסף בלאן -	מורה, בית הספר האנגליקני, נצרת
מר ארז סיני -	מורה, בית הספר קבוצת יבנה
מר יגאל דור -	מפקח, משה"ח
גב' ריקי יפה -	מורה לחט"ב, בית הספר החקלאי, 'הכפר הירוק'
מר יריב דגן -	מורה לחט"ב, בית הספר מקיף אזורי 'עמי אסף', כפר סבא
מר רמי בורנשטיין -	מורה ליסודי ומפקח, משה"ח
מר נרי בר-און -	מורה ליסודי, בי"ס 'דורון', רמת השרון
מר ניסים יצחק -	מורה ליסודי, בי"ס 'מילקן', אריאל

משרד החינוך, המינהל למדע וטכנולוגיה, רחוב השלושה 2, יד-אליהו, תל-אביב 61092
טלפון ישיר: 03 -6896147 ● טלפון מזכירות: 03 -6896150 ● פקס: 03 – 6896199
HTTP://WWW.CSIT.ORG.IL

תוכן עניינים

מבוא 2

עקרונות 3

מטרות התכנית 4

מבנה התכנית 5

שער ראשון - חלק ראשון של מודול מדעי המחשב 7

פרק א' מבוא לאלגוריתמיקה 10

פרק ב' אלגוריתמיקה מרעיון לקוד – שלב בסיס 14

פרק ג' ביטויים לוגיים, תכנות מרובה דמויות, ריבוי תסריטים ועיבוד מקביל 20

פרק ד' משתנים 25

פרק ה' משתנים – הרחבה: משפטי תנאי 'אם'; 'אם ואם לא' 33

פרק ו' תכנות מרובה דמויות, תלבושות ותסריטים - הפשטה וייצוג מערכות .. 43

פרק ז' הגדרת משתנים – הרחבה: משתנה לדמות, משתנה לכל הדמויות, משתנה ענן .. 48

פרק ח' אופרטורים לוגים, קינון משפט תנאי 54

פרק ט' כתיבה של אלגוריתם מילולי, תכנון ויישום (מרעיון לקוד) 61

פרק י' מבוא למבנה בקרה לולאה, לולאת 'לעולמים' 69

מודול מדעי המחשב - פרויקטי העשרה לפי רמות 77

תכנית רמה א': 77

תכנית רמה ב' 90

תכנית רמה ג': 98

חלק שני של מודול מדעי המחשב 103

פרק יא' הכל מתחבר 103

פרק יב' לולאות – הרחבה 113

פרק יג' פרוייקטים ותרגול מסכם 135

שער שני - מודול אלגוריתמיקה ורובוטיקה 145

פרק א' מבוא למערכות מחשב 156

פרק ב' מבוא למערכות 158

פרק ג' סקירה מפורטת של מרכיבי הרובוט וכתיבת קטעי תכנית קצרים 165

פרק ד' תנועת הרובוט ופניות - הרחבה 173

פרק ה' כתיבה של אלגוריתם מילולי, תכנון ויישום (מרעיון לקוד) 186

פרק ו' בקרה בחוג פתוח; בקרה בחוג סגור - חיישנים, לולאת 'לעולמים' 194

פרק ז' משתנים , משפטי תנאי 'אם', 'אם ואם לא' 201

פרק ח' לולאות וחיישנים – הרחבה; קלט מהמקלדת 207

פרק ט' לוגיקה – תנאים מורכבים , לולאת 'חזור עד ש' 220

פרק י' מערכות בקרה בחוג סגור – הרחבה 229

פרק יא' שילוב מערכות ומבנה בקרה מתקדמים – פרויקטים 231

שער שלישי - תכנית חלופה המוצעת לכיתה ו' 236

פרק א' לבנות משתמש – תכנון מודולרי ומובנה 236

פרק ב' תכנית הסתברות, אלגוריתמיקה וכתיבת קוד – 10 שעות 244

פרק ג' שיבוט דמויות 250

שער רביעי - נספחים 255

נספח א' סיכום מקוצר של הוראות שפת סקרטץ' 255

נספח ב' סיכום מורחב של הוראות שפת סקרטץ' 272

נספח ג' מושגים מרכזיים בסביבת התכנות סקרטץ' 305

נספח ד' סביבת התכנות סקרטץ': תפריטים, תגיות, כלים 314

נספח ה' ידע מתמטי במתמטיקה של תלמידי כיתות ד'; ה' 327


תכנית ללימודי מדעי המחשב ורובוטיקה
בתי הספר היסודיים

מבוא

החשיבות של מדעי המחשב כענף מרכזי בין תחומים מדעיים וטכנולוגים מגוונים גדלה והיא משפיעה על כל היבט של חיינו. תלמיד בוגר של מערכת החינוך צריך שיהיו לו מיומנויות מעבר לשימוש בישומי מחשב – מיומנויות הפשטה, ניתוח ומימוש פתרון בעיות/משימות בעזרת כתיבת והפעלת תכניות מחשב והצגת תכניות למימושן. לדוגמה, להיות מסוגל להציג בעיה/משימה מתחומים שונים (פיסיקה, מתמטיקה, כימיה, ניהול, ובעיות/משימות ממדעי החברה והרוח) באמצעות ייצוג/מודל שניתן לתרגמו לתכנית מחשב שניתן להריץ אותה ולהגיע לתוצאות המתוכננות. תוצאות הלמידה והיישום אינן רק תוצאה מספרית. תוצאה יכולה להיות סימולציה שעוקבת אחרי ביצוע תהליך כימי, תיאור של מערכת אקולוגית והשתנותה עם הזמן, או משחק מחשב.

ייצוג מודלים ליישום המחשב, תכנון וכתיבת קוד, הינן מיומנויות חשובות שכל תלמיד הלומד בבית ספר בישראל צריך לרכוש ולדעת ליישם. כפי שמתמטיקה משפיעה על דרך המחשבה והגישה המדעית, מדעי המחשב ממחישים את החשיבות של המחשבה האנליטית ואת האפשרות המיידית ליישמה בחיי היום יום. למדעי המחשב יש השפעה מדהימה על כל היבט בחיינו ובמידה רבה שפת המחשב היא שפת הטכנולוגיה, שפה שניתן לתקשר באמצעותה בצורה מדויקת והיא ניתנת לבחינה מיידית.

אוריינות במדעי המחשב מתייחסת להיבטים הבאים:

- חשיבה אלגוריתמית – יכולת הפשטה; ניתוח; הגדרת פעולות מחשב למימוש משימה
- מימוש בתכנית מחשב – יכולת מימוש אלגוריתם באמצעות כתיבת קוד הלכה למעשה
- רקע תומך בתהליך בנית פתרון יעיל – מבנה המחשב; מבנה נתונים; כולל ידע ויישום בתחום הרובוטיקה שמשלב כתיבת קוד עם הבנה של מערכות טכנולוגיות מורכבות

תכנית הלימודים במדעי המחשב לבית הספר היסודי תעזור לפתח בוגרים בעלי אוריינות יצירתית מדעית וטכנולוגית. היא תתרום לגידול ילדים סקרנים ותאבי דעת שמסוגלים ללמוד באופן עצמאי ושמשתמשים בידע מדעי וטכנולוגי ובמיומנויות חשיבה ועשייה בכדי להבין מודלים בתחומים שונים: לייצג מודלים אלו למחשב באמצעות שפת תכנות; לכתוב פתרון תכנותי לבעיות; לבחון את הפתרון לבעיה שהוצגה ואת המענה לפתרון; להשוות בין פתרונות ויעילותם; לבנות ממשק ידידותי למשתמש בפתרון שנכתב. כל זאת באינטראקציה עצמאית עם מחשב שבשגרה מגיב ליצירה המופשטת שתוכנתה באופן מידי וחד משמעי.

התלמידים יכירו בכך שתחום מדעי המחשב הוא פרי המחשבה האנושית ויהיו מודעים לחשיבות התחום לקידום האנושות. בנוסף הם יבינו עד כמה משמעותי לחברה שילובן של מערכות תוכנה בכל תחום: תקשורת מחשבים שהופכת את העולם לכפר גלובלי; אגירה-חיפוש-ועיבוד ידע מעבר ליכולת המוח האנושי; שימוש ברובוטים בצורה יום יומית נרחבת כמכוניות הנוהגות בעצמן במשימות מסוכנות ומאתגרות, פשוטות או מורכבות; פיתוח מערכות תוכנה ובקרה המבקרות את מערכות המים, החשמל, הבריאות ועוד.

בדומה להוראת המדעים בבתי הספר, מתבססת תכנית הלימודים במדעי המחשב על גישת STS שמשלבת בין התחומים: מדע, טכנולוגיה וחברה (Science, Technology, Society). השילוב מתבטא בחיבור בין מדע וטכנולוגיה. לדוגמה בעיה מדעית/הנדסית שנפתרת באמצעות תכנית מחשב או מטלות חקר ואיסוף נתונים על פני מאדים שמתבצעות באמצעות רובוט. למדעי המחשב יש גם השפעה עצומה על החברה. למשל אפליקציות לטלפונים חכמים או פיתוח הרשתות החברתיות, שני תחומים שהם תוצרים של מדעי המחשב והשפעתם על החברה כבירה. הכלים שהתלמידים ירכשו יוכלו לפתח אצלם מחשבה יצירתית, אנליטית, ויישומית ולפתוח בפניהם אפשרויות ללימוד עצמי ולתעסוקה בעתיד ללא תלות במקום המגורים שלהם או במגזר ממנו באו, וכך לחזק את חוסננו החברתי והכלכלי.

עקרונות

מהות מקצוע 'מדעי המחשב'

- ✓ המורכבות של התמודדות עם משימה במדעי המחשב כוללת בניית מודל רעיוני, החלטה על אופן ייצוג הנתונים, כתיבה ויישום האלגוריתם הרעיוני לתכנית בשפת מחשב, כתיבת התכנית בסביבת העבודה, הרצת התכנית, ניפוי שגיאות, דיון באיכות הפתרון, חיפוש והשוואה עם פתרונות נוספים
- ✓ לימודי מדעי המחשב יכללו הבנייה של ידע מתמטי, ניתוח ופירוק בעיות/מטלות לבעיות ומטלות משנה, חיבור מרכיבי הפתרון המוצע לכלל תכנית מחשב מלאה
- ✓ פיתוח החשיבה יכלול הבנייה של מיומנויות חשיבה מסדר גבוה, חשיבה יצירתית וביקורתית, מיומנויות תקשורת ושיתופיות, וחשיבה מטה-קוגניטיבית
- ✓ תכנון ההוראה של מיומנויות חשיבה ותהליכי חשיבה יעשה באופן היררכי וספירלי במשולב עם התכנים הנלמדים ותוך התייחסות להוראה מפורשת של מיומנויות ולשימוש בהן בהקשרים חדשים
- ✓ לימודי מדעי המחשב יכללו תיאור בעיות ורעיונות באמצעות ייצוג תרשימי, טבלאי והיררכי, ועל-ידי אלגוריתם מילולי של פסיאודו קוד (קוד מדומה)
- ✓ הוראת מדעי המחשב כוללת חינוך לחשיבה שנובעת מאופי המקצוע על היבטיו התוכניים, החשיבתיים והמתודולוגיים, בכדיי לטפח בקרב הלומדים היבטים מחשבתיים הייחודיים למקצוע מדעי המחשב. בפרט תשוכלל חשיבה יצירתית על מבנים אבסטרקטים דמיוניים וברמות אבסטרקציה שונות - מעין בניית 'מגדלי אבסטרקציה באוויר', מימושם כמבני נתונים ואלגוריתמים והכרת הפלא במימושם במחשב ובהפעלתם ליצירת יחסי קלט/פלט נדרשים
- ✓ תהליכי הוראה-למידה והערכה של המקצוע יעשו על ידי יישום גישות קונסטרוקטיביסטיות תוך מתן מענה לשונות של הלומדים בכיתה ולשימוש בסביבות למידה ובייצוגים מגוונים לשם הצגת פתרון לבעיות ובכדיי לבצע את המטלות
- ✓ פיתוח אוריינות במדעי המחשב משפיעה או מושפעת מבקיאויות נוספות הכוללות: אוריינות לשונית, אוריינות מתמטית, אוריינות דיגיטלית ואוריינות מדעית. התייחסות אליהן תשולב בתהליכי הוראה-למידה-הערכה
- ✓ לימודי מדעי המחשב ורובוטיקה יתקיימו בסביבות למידה מגוונות ועשירות שיאפשרו חוויית למידה מעניינת, מאתגרת ומשמעותית. במהלך הלמידה ייושמו מגוון דרכי הוראה-למידה-הערכה הייחודיים למקצוע ושתורמים להבניית ידע ולפיתוח מיומנויות חשיבה ועשייה, לתקשורת ולשיתופיות

מטרות התוכנית

חשיפה להיבטים השונים של מדעי המחשב – הפשטה, פתרון אלגוריתמי ומימושו – דרך התנסויות מותאמות גיל מהנות, מגוונות, ומעוררות עניין ומוטיבציה:

1. היכרות עם עולם מדעי המחשב ובמרכזו פתרון בעיות ויישום הפתרונות באמצעות חשיבה אלגוריתמית.
2. הקניית מושגים בסיסיים בחשיבה לוגית ובחשיבה אלגוריתמית בדרך חווייתית ובשילוב פעילויות רבות ממוחשבת ולא ממוחשבות. יושם דגש על פעילויות חווייתיות יחד עם פעילויות תכנותיות עם מערכת רובוט. כל הפעילויות יהיו מותאמות לשכבת הגיל.
3. הבנת תהליכי פתרון בעיות ומימוש הפתרון (באמצעות שפת מחשב) תוך כדי מעבר בשלבים של תכנון; בנייה ובדיקת אלגוריתם כללי; כתיבת אלגוריתם מילולי; תכנון המרכיבים של היישום (באמצעות כלים שונים) ויישום הפתרון במערכת הרובוט.
4. שילוב בין-תחומי – מדעי המחשב ככלי בתחומי ידע אחרים.
5. פיתוח כישורי רפלקציה להבנת תהליך הפתרון וחיזוק חשיבה ביקורתית לצורך הערכת הפתרון.
6. פיתוח כישורים של עבודת צוות לפתירת בעיות מורכבות ולביצוע פרויקטים בהיקפים שונים.

הערכת התכנית

במהלך השנתיים הקרובות בזמן שהתכנית תיושם בשטח היא תלווה בהערכה אקדמית מדעית שתבחן את הצלחתה במדדים שיקבעו על ידי משרד החינוך והגוף המעריך.

קהל היעד

תכנית הלימודים במדעי המחשב ורובוטיקה לבית הספר היסודי מיועדת לכל תלמידי כיתות ד'-ה'-ו', בכל המגזרים במדינת ישראל ללא קשר למגדר, דת או לאום.

התכנית מותאמת לאוכלוסיית לומדים הטרוגנית. נושאי הלימוד בכל כיתה יילמדו בדרגות שונות של התמקדות, מהיקף בסיסי המחייב את כלל אוכלוסיית התלמידים ועד להיקף מורחב ומעמיק שנועד לתלמידים בעלי מוטיבציה ויכולת לעבודה עצמית. על בית הספר לחתור לקדם את כל תלמידיו על ידי חיזוק תוכני החומר הבסיסי בקרב התלמידים המתקשים, ועל ידי הצבת יעדים ואתגרים, מעבר לאלה שהוגדרו כהישגים בסיסיים, לתלמידים שיראו נכונות ועניין.

התכנית מגדירה תחומי תוכן, נושאים מרכזיים, מטרות ורעיונות מרכזיים שמתייחסים לתכנים ומיומנויות. עם זאת, בתי הספר יוכלו להעניק למרכיבים אלה ביטוי ייחודי בהתאם לצרכי האוכלוסייה וליעדיה (תלמידים, מורים, בית ספר וקהילה) – יוזמות מקומיות, הדגשת היבטים ייחודיים (כגון תעשייה, מדע, תקשורת ועוד), שימוש במשאבי הסביבה הקרובה (אתרים, הורים, חברות הי-טק, ארגונים ללא כוונות רווח), יאפשרו ביטוי ייחודי של הפעלת התכנית על ידי כל בית ספר.

התכנית מיועדת למורים המלמדים את מקצוע מדעי המחשב בבית הספר היסודי. התכנית מכוונת גם למפתחים וכותבים של חומרי למידה. ראוי שהתכנית תמצא את מקומה במסגרות שונות של הכשרה ופיתוח מקצועי של מורים המלמדים את המקצוע, או במסגרת תכניות הכשרה של המורים לעתיד במכללות ובאוניברסיטאות.

מבנה התכנית

התכנים של התכנית מאורגנים בתחומי תוכן מרכזיים. תכנים אלו יילמדו ברצף ספירלי מובנה בכל שכבות הגיל ד'-ו'.

להלן תחומי התוכן שבתכנית:

1. **חישוביות, אלגוריתמיקה ומיומנויות המאה 21.**
2. **מימוש בשפות וסביבות עבודה שונות – כתיבת קוד.**
3. **מבנה המחשב.**
4. **רובוטיקה.**
5. **תכלול (נושאים ומטלות היוצרים חיבור בין כל נושאי התכנית).**

כל תחום תוכן כולל נושאים מרכזיים העוסקים בהיבטים של טכנולוגיה ושל מדעי המחשב. בכל אחד מהם כלולים ציוני דרך שהם היגדים המפרטים את הנושא המרכזי. כל אחד מציוני הדרך בתכנית נשען על רעיון מרכזי אחד או יותר. הרעיונות המרכזיים מייצגים עקרונות חשובים הקשורים או נובעים מהנושא המרכזי.

מפרט התכנים מותאם למסגרות של כיתה הטרוגנית וכולל לימודי חובה המהווים את בסיס הידע לכלל התלמידים, ופרוט תכנים של העמקה והרחבה לתלמידים מתקדמים.

כדי לאפשר יישום של תכנית הלימודים נדרשות, בממוצע, שתי שעות הוראה שבועיות (2 ש"ש) לכל אחת מדרגות הכיתה.

מודול מדעי המחשב כיתה ד'	מודול בסיס 60 שעות מודול פרויקטי העשרה לפי רמות מודול מתקדם - 30 שעות
מודול אלגוריתמיקה ורובוטיקה כיתה ה'	מודול בסיס 60 שעות
תוכן למודול לכיתה ו' וחלופה ראשונה	חלופה ראשונה - 30 שעות

הסבר:

מדעי המחשב מודול בסיס

תכנית בהיקף של 60 שעות שתלמד בשכבה ד'.

מודול פרויקטי העשרה לפי רמות

מודול הכולל 30 פרויקטים של העשרה לפי רמות: רמה א' - 15 פרויקטים; רמה ב' - 10 פרויקטים; רמה ג' - 5 פרויקטים.מטרת המודול הזה היא לתמוך בתכנית לכיתה ד' או לכיתה ה' לפי שיקולי בית הספר. (בהתאם לרמת הכיתה בה מיושמת התכנית)

מודול מתקדם בהיקף 30 שעות

המודול נועד לתת מענה לאחד מהמצבים הבאים:

1. לבית ספר שאפשר לתלמידיו ללמוד עקרונות מדעי המחשב בכיתות א'-ג' במסגרת פרויקטים שונים (חוגים, שעורי תגבור וכו') וכתוצאה מכך תלמידים סיימו את תכנית 60 השעות הבסיסית לפני סיום שנת הלימודים – המודול הזה יכול להיות תכנית המשך.
2. לבית ספר שיחליט ללמד 3 שעות שבועיות המודול משלים את התכנית ל-90 שעות.
3. לבתי ספר שמסיבות שונות לא ילמדו בכיתה ה' רובוטיקה. תכנית זו תעמוד לרשותם כהמשך למודול 60 השעות בשילוב עם מודול פרויקטי העשרה.
4. לבתי ספר שמקיימים קורסי המשך ומעוניינים להשתתף בימי שיא ותחרויות כמו יום הסקרטצ' הבינלאומי או הארצי, תחרות קוד גורו וכדומה, מודול זה יכול לסייע ולשמש כמקור להשלמת ידע על הסביבה וכמקור לרעיונות לפרויקטים.

המודול כולל פרויקטים שונים שיכולים לשמש את המורים במתן מטלות סיום מחצית השנה כמטלות סיכום. הדוגמאות מפורטות עם הסברים ויסייעו למורה להנחות את התלמידים.

מודול אלגוריתמיקה ורובוטיקה

מודול בסיס בהיקף 60 שעות, מיועד לשכבה ה'.

מודול חלופות לשכבה ו' בהיקף 30 שעות

הצעה מקדמית לנושאים וחלופה ראשונה בתכנית שכבה ו'. בהמשך יפתחו מפתחים שונים חלופות לתכנית לשכבה ו' בהתאם להתפתחויות ובהנחיית ופיקוח הגופים המקצועיים והרלוונטיים במשרד.

שער ראשון

חלק ראשון של מודול מדעי המחשב

מסלולי התקדמות בלמידה לפי 3 רמות:

התכנית הכללית מתחילה מהוראות בסיס (ארגון סדרתי של 2-3 הוראות).

התכנית מסתיימת (מודול 60 שעות) ברמה של יישום הכולל: מספר דמויות, מספר תסריטים לדמות, משתנים, מבני תנאי, לולאת חזור לעולמים (לולאה אינסופית), לולאת חזור מספר פעמים, אינטראקציה בין דמויות וטיפול באירועים.

קיימת בתכנית גמישות וניתן לדחות נושאים מסוימים או לא ללמדם כלל. הבחירה תלויה במורים ובמרכזי התכנית והיא מותאמת לכל בית ספר ולכל קבוצת לימוד. בתכנית מוצעים רמות ומסלולי התקדמות ופירוט נושאים: רמת בסיס, רמת ביניים, רמה גבוהה.

התכנית הכללית והמלאה ל- 60 שעות כוללת:

א. פיתוח בעיה: מרעיון לאלגוריתם מילולי, מאלגוריתם מילולי לקוד.

ב. כתיבת תסריטים בסיסיים – מבנה סידרתי, הוראות מקבוצת תנועה, צלילים ועוד.

ג. עיבוד במקביל, ריבוי דמויות.

ד. קלט ממשתמש, הרצה תלויה בקלט.

ה. כלים ויישומם – מרעיון לקוד בתהליך מובנה.

ו. משתנים – שימוש בסיסי, שלב א' ללא תבנית מנייה.

ז. משפטי תנאי – משפט 'אם', משפט 'אם ואם לא'.

ח. שילוב קלט, משתנה, משפט תנאי.

ט. שילוב בין-תחומי: מדעי המחשב ככלי בתחומי ידע אחרים, בנייה של יישום בכל שלבי התהליך.

י. הרחבה למשתנים: משתנה לדמות, לכל הדמויות, משתנה ענן.

י"א. אופרטורים לוגיים ושילובם בהוראת תנאי ובלולאה מותנת

י"ב. ביצוע חוזר:

I. לולאה אינסופית – 'לולאת לעולמים'.

II. לולאה 'חזור מספר פעמים'.

III. לולאת 'חזור עד ש'.

כל התכניות בכל רמות הלמידה ילמדו בהיקף 60 שעות. ההבדל בין רמות הלמידה יתבטא בדחייה של נושאים או ויתור על נושאים.

לדוגמה ניתן לבחור לא ללמד את המשפט **'אם ואם לא'** ובכל יישום בו יידרש לבדוק שני מצבים יעשה שימוש בשני משפטי תנאי בסיסיים. שיקול דעת דומה יתכן בבחירה לא לעשות שימוש באופרטורים לוגיים ולהעדיף חלופה של מספר משפטי תנאי או קינון משפטי תנאי. שלוש הרמות המוצגות כמומלצות מנסות להתמודד עם קשיים בהבנת רעיונות מושגיים ויישומם, שצפויים בכיתות ד'-ה'. הן מתמודדות עם השאלה כיצד לטוות מסלולים שעוקפים חלק מהקשיים או שתומכים תמיכה נוספת בלמידה, בהתאם לקבוצת הלימוד.

מודול 'פרויקטי העשרה' המופיע בתכנית הלימודים מציג עבור הרמות השונות מבחר אתגרים המאפשרים המשך העמקה בנושאים ותמיכה פדגוגית דידקטית בתהליכי הלמידה. הוא נכתב למימוש תהליכים אלו: ויתור על נושאים מסוימים או מצב שבו נדרש חיזוק ותרגול חוויתי; תמיכה בלמידה של נושא מורכב שמצריך יותר תרגול ובמיוחד תרגול חוויתי שמאתגר, מעניין ודורש יישום של מיומנויות המאה 21.

תכנית לפי רמות:

רמה א'

בתכנית רמה א' שחלקה העיקרי מתוך התכנית המלאה הוא כ- 45 שעות, תעשה ההרחבה ל- 60 שעות בהגדלת מספר הדוגמאות; מתן אפשרויות והזדמנויות לכתיבת יישומים; הקצאת זמן רב יותר לתרגול; גיוון של דוגמאות "משחקיות"; יותר דמויות; יותר תסריטים; עבודה על גרפיקה אישית (המעוצבת על ידי התלמיד ולא מיובאת); שילוב צלילים. לתכנית יש 15 פרויקטים במודול פרויקטי העשרה.

רמה ב'

בתכנית רמה ב' שחלקה העיקרי מתוך התכנית המלאה הוא כ- 50 שעות, תעשה ההרחבה ל- 60 בהגדלת מספר הדוגמאות; מתן אפשרויות והזדמנויות לכתיבת יישומים; הקצאת זמן רב יותר לתרגול; גיוון של דוגמאות "משחקיות"; יותר דמויות; יותר תסריטים; עבודה על גרפיקה אישית (המעוצבת על ידי התלמיד ולא מיובאת); שילוב צלילים. לתכנית יש 10 פרויקטים במודול פרויקטי העשרה.

רמה ג'

בתכנית רמה ג' שחלקה העיקרי מתוך התכנית המלאה הוא כ- 60 שעות, מומלץ להמשיך ולהרחיב לפרק י"א; לפרק י"ב; לכלול לולאת **'חזור __ פעמים'** ולכלול שימוש מוגבל בלולאת **'חזור עד ש'**; להשתמש בגרפיקה אישית לדמויות ולרקעים; להרחיב את השימוש בדמויות ותסריטים; להעביר מסרים ופרויקטים שמשלבים ידע מדעי וטכנולוגיה ואפילו לשלב נושאים מתחומים רבי מלל. לתכנית ברמה ג' יש 5 פרויקטים במודול פרויקטי העשרה.

ההבדל בין הרמות אינו שאלה של מתן תכניות שונות לתלמידים בעלי יכולות שונות ויצירת 'הקבצות' וכדומה. שלושת הרמות נועדו לתת מענה לשונות ככל שתהיה.

פירוט סעיפי התוכנית לפי רמות יישום

		רמה א'	רמה ב'	רמה ג'
א	פיתוח בעיה: מרעיון לאלגוריתם מילולי, מאלגוריתם מילולי לקוד	כל הרמות		
ב	כתיבת תסריטים בסיסיים: מבנה סדרתי, הוראות מקבוצת תנועה, צלילים ועוד	כל הרמות		
ג	עיבוד במקביל, ריבוי דמויות	3 דמויות	5-3 דמויות	6-4 דמויות
	מספר תסריטים לדמות	3-2	5-2	6-4
	שליחת וקבלת מסרים	3-2	6-3	אין מגבלה
ד	קלט ממשתמש באמצעות ההוראה **'שאל'**	כל הרמות		
ה	כלים ויישומם – פיתוח מרעיון לקוד בתהליך מובנה	כל הרמות		
ו	משתנים – שימוש בסיסי משתנה מערכת 'תשובה'	כל הרמות		
ז	משפט תנאי	**'אם..'**	**'אם..'** **'אם.. ואם לא..'**	**'אם..'** **'אם.. ואם לא..'**
ח	שילוב בין-תחומי, מדעי המחשב ככלי בתחומי ידע אחרים, בנייה של יישום כל שלבי התהליך – לשלב בדוגמאות בסיס	כל הרמות		
ט	הרחבה בנושא משתנים: משתנה לדמות, לכל הדמויות, משתנה ענן	-	הבחנה בין משתנה לדמות, משתנה לכל הדמויות	כולל משתנה ענן
י	אופרטורים לוגיים ושילובם במשפטי תנאי	לא לכלול את האופרטורים הלוגיים בתכנית הבסיס, ליצור חלופה בקוד עם מספר משפטי תנאי בסיסיים ללא המשפט **'אם ואם לא'**.		אופרטור **'וגם'** אופרטור **'או'** לא מומלץ אופרטור **'לא'**
יא	לולאה אינסופית (**'לעולמים'**)	כל הרמות		
	לולאה מבוקרת (**'חזור מספר פעמים'**)	לא	כן	כן
	לולאה מותנת (**'חזור עד ש..'**)	לא	לא	כן
	תבנית מנייה	לא	תבנית מנייה רק בהקשר תוכן וצורך ברור	

פרק א' מבוא לאלגוריתמיקה

תחום תוכן: כתיבת קוד, חישוביות, אלגוריתמיקה ומיומנויות המאה 21



הוראה חד משמעית

הוראה ברורה למבצע. המבצע יבצע הוראה זו שוב ושוב ללא שינוי בביצוע (המבצע יכול להיות אדם, או מכונה-מחשב).

אלגוריתם

דרך שיטתית (כלומר כזו שצעדיה מוגדרים היטב) לביצוע של משימה מסוימת במספר סופי של צעדים.

שפת תכנות

שפת תכנות היא אוסף של חוקים תחביריים וסמנטיים המגדירים שפה פורמלית. באמצעות שפת התכנות אנו כותבים תכניות מחשב.

תכנית מחשב

רצף של הוראות בשפת תכנות כלשהי שנועדו לביצוע משימה מסוימת במחשב. תכנית מחשב היא מימוש של אלגוריתם באמצעות שפת תכנות.

ביטוי מתמטי

רצף סופי של סימנים בעלי משמעות מתמטית. הסימנים בביטוי כוללים לרוב מספרים, פונקציות, פעולות, אופרטורים, משתנים, יחסים לוגיים ואובייקטים מתמטיים אחרים.

תסריט

המילה תסריט היא מילה נרדפת לתכנית מחשב, בשפת תכנות סקרטץ' אנחנו יוצרים תסריטים השייכים לדמויות ולרקע. תסריטים אלו הם תכניות מחשב.

דרכי הערכה

1. מטלת מעבדה, בחירת הוראה מקבוצות ההוראות **'צלילים'** / **'תנועה'** והפעלתה עם הקלקה על ההוראה.
2. מטלה של כתיבת תסריט בסיסי עם הוראה מקבוצת ההוראות **'אירועים'** בשילוב הוראת **'אמור'** עם טקסט.
3. מטלה של כתיבת תסריט בסיסי עם הוראה מקבוצת ההוראות **'אירועים'** בשילוב הוראת **'אמור'** עם ביטוי מתמטי.
4. מטלת כתיבת תסריט כמענה למשימה כגון: תלמיד קיבל שני ציונים, ציון 60 וציון 90, כתוב תסריט לחישוב והצגת הציון הממוצע.

מטרות ביצועיות

1. התלמיד ידע לבחור הוראה מקבוצת ההוראות ולהפעילה עם הקלקה על ההוראה.
2. התלמיד ידע לבחור הוראה מקבוצות ההוראות בשפת סקרטץ', להעבירה למסך התסריטים ולהפעילה עם הקלקה על ההוראה.
3. התלמיד ידע לערוך ביטוי מתמטי באמצעות קבוצת ההוראות **'מפעילים'** שכולל אחת מפעולות החשבון הבסיסיות ושני מספרים קבועים, ויוכל לשלב את הביטוי בהוראות שונות.
4. התלמיד ידע לבחור הוראה מקבוצת ההוראות **'אירועים'** (המשמשת כהוראת פתיחה לתסריט) ולחבר את ההוראה כהוראה מקדימה להוראות ביצוע.
5. התלמיד ידע לשלב הוראה מקבוצת ההוראות **'אירועים'** עם ההוראה **'אמור'** בשילוב ביטויים מתמטיים, טקסטים והוראות צליל.

מטרות ביצועיות – דוגמאות

1. התלמיד ידע לבחור הוראה מקבוצות ההוראות בשפת סקרטץ', להעבירה למסך התסריטים ולהפעילה עם הקלקה על ההוראה.

ביצוע מטלה זו הוא בסיסי אך ישנן אפשרויות רבות וחשוב למקד את התלמידים. יש להתמקד בהוראה **'אמור'** או הוראת **'נגן צליל'** או הוראת **'תנועה'**

הוראת השמעת צליל:

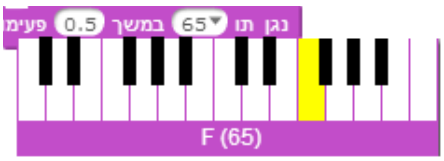


צריך להציג בפני התלמידים את האפשרות הקיימת בהוראות שונות בסקרטץ' של גלילה, ולהנחות אותם לבחור באחת מהאפשרויות בביצוע של ההוראה.

הוראת השמעת תו:

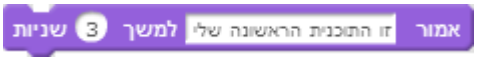


גלילה, וקבלת האפשרויות להשמעת תו שונה בהוראה **'נגן תו'**:



2. התלמיד ידע לערוך ביטוי מתמטי שכולל אחת מפעולות החשבון הבסיסיות ושני מספרים קבועים באמצעות קבוצת ההוראות 'מפעילים' ולשלב את הביטוי בהוראות שונות, ובפרק זה בהוראה 'אמור'.

הוראת **'אמור'**:



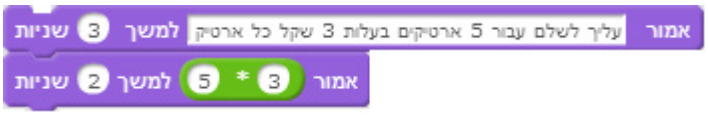
חשוב להציג את האפשרויות בהוראות מסוימות בסקרטץ' ולעדכן את התוכן של שדות המידע בהוראה. כאן בהוראה **'אמור'** אפשר לשנות את הטקסט (בהמשך לשלב ביטויים וכדומה), אפשר לשנות את פרק הזמן של הופעת הפלט (מה שהדמות אומרת)

3. התלמיד ידע לערוך ביטוי מתמטי, שכולל אחת מפעולות החשבון הבסיסיות ושני מספרים קבועים, באמצעות קבוצת ההוראות **'מפעילים'**, ולשלב את הביטוי בהוראות שונות. ובפרק זה בהוראה **'אמור'**.

ההוראה **'אמור'** – תוצאת ביטוי חשבוני:



שתי הוראות **'אמור'**:



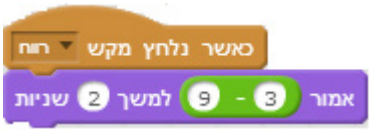
4. התלמיד ידע לבחור הוראה מקבוצת ההוראות **'אירועים'** (המשמשת כהוראת פתיחה לתסריט) ולחבר את ההוראה כהוראה מקדימה להוראות ביצוע.

הוראות קבוצת **'אירועים'** מאתחלות ביצוע של תסריט. בהצגת ההוראות חשוב להציג כמה סוגי הוראות וגם את האפשרות לשנות את זהות המפעיל של התסריט. לדוגמה בהוראה: **'כאשר נלחץ מקש'**

הוראת אירוע **'כאשר נלחץ דגל ירוק'**



הוראת אירוע **'כאשר נלחץ מקש'**



ההוראה **'כאשר נלחץ מקש'** מאפשרת בחירת כל המקשים שמאתחלים אירוע מסוים.



המשך גלילה עם מקשי חיצים יציג את כל המקשים על המקלדת.

5. התלמיד ידע לשלב הוראה מקבוצת ההוראות **'אירועים'** עם ההוראה **'אמור'** יחד עם ביטויים מתמטיים, טקסטים והוראות **'נגן צליל'**.

כאן אפשר להתחיל לדבר על המושג תכנית/תסריט, יש לנו קבוצת הוראות המתחילה להתבצע עם הוראת אירוע, יש מבנה סדרתי של ביצוע וחשוב לדון על רעיון של מבנה סדרתי של קבוצת פקודות שמבצע את ההוראות לפי הסדר. צריך לתרגל עם התלמידים שינוי בסדר ההוראות של התסריט ולבקש מהם לראות אלו הוראות התבצעו, באיזה סדר ומה השפעת השינוי על תוצאות התסריט.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
הוראה חד משמעית ביטוי מתמטי ומרכיביו	הדגשת הרעיון שאלגוריתם יכלול הוראות המובנות ל 'מכונה' המבצעת. ההוראה ממורה 'שב על הכיסא' ברורה לתלמיד. ההוראה 'תעביר לשולחני מספר מחברות' אינה ברורה. תלמיד מסוים יעביר 3 מחברות ואחר יעביר 2 מחברות. ביחידה זו התלמידים מתרגלים מתן הוראות למחשב וצפייה בביצוען. צריך לעודד את התלמידים לשנות מרכיבים בהוראות (כמו מה יש לומר בהוראה **'אמור'** ומשך השניות של ההוראה **'אמור'**)	✓		הוראה מהי **1 שעות** כתיבת ביטויים כתיבת הוראות פלט (סקרטץ' – ההוראה **'אמור'**) כתיבת הוראות מקבוצת 'צלילים', 'תנועה' תרגול הוראות בסיסיות נוספות **1 שעות**
תכנית - תסריט ארגון קבוצת הוראות	תרגול ביצירת אוסף סדור של הוראות וביצוע התסריט. התלמידים ישתמשו בכל אחת מההוראות לאתחול תסריט בקבוצה 'אירועים'			תכנית מהי סדר ורצף בביצוע תכנית **1 שעות**
תכנון הקוד לפני כתיבתו משמעות של הוראה	חשיבות הכתיבה האלגוריתמית תיושם במטלות שיתבצעו בשתי דרכים: פרק זמן מוגדר לכתיבת קוד לביצוע מטלה; פרק זמן הכולל תכנון של 3-5 דקות חשיבה. יש לבצע זאת על מטלות חישוב מתמטי כמו: ממוצע של שני מספרים או מכפלה של סכום שני מספרים במספר שלישי			חשיבות כתיבת אלגוריתם לפני כתיבת קוד כתיבת תוכנית עם התחלה וסוף **1 שעות**

פרק ב' אלגוריתמיקה מרעיון לקוד – שלב בסיס

תחום תוכן: כתיבת קוד, חישוביות, אלגוריתמיקה ומיומנויות המאה 21



נקודת מוצא

המצב בתחילת האלגוריתם ולפני ביצועו. בסביבת סקרטץ' המידע על נקודת המוצא יכול להיות מורכב ממצב של כל דמות, ממיקום הדמות וכיוונה, ממשתני דמויות ומשתנים לכל הדמויות, ממראה הדמות (תלבושת) ומנתונים התחלתיים שכלולים בתיאור המטלה.

נקודת יעד

התוצאה הסופית של האלגוריתם. בדומה להגדרת נקודת המוצא גם הגדרת נקודת היעד דורשת תוצאות מוגדרות.

דרכי הערכה

1. מתן יעד ודרישה לכתוב את השלבים לביצוע היעד לפי הסדר עם הוראות חד משמעיות (מטלות ללא מחשב).
2. מתן יעד למימוש באמצעות קוד, הכוונה לכתיבה בעברית חופשית של שלבי הביצוע.
3. מתן מטלות שמחברות הוראות חישוב, ביטויים עם מפעילים מתקדמים והצגת תוצאות.
4. מתן מטלות המשלבות הוראות 'תנועה', הוראות 'סיבוב', הוראת **'פנה לכיוון'**.
5. כתיבת תסריטים שמתבצעים על ידי שתי דמויות במקביל. בתסריטים ישולבו הוראות מקבוצות שונות.
6. שימוש נרחב יותר (מספר מקשים לדוגמה) לביצוע תסריטים שמתבצעים כתוצאה מאירוע (כגון לחיצה על מקש) – EVENT DRIVEN PROGRAMMING

מטרות ביצועיות

1. התלמיד ידע לנסח בעברית הוראות חד משמעיות לביצוע מטלה.
2. התלמיד יכתוב תסריט קצר לחישוב ביטוי מתמטי ויעביר מסר לדמות אחרת שתבצע פעולה כלשהי – העברת מסרים כמנגנון שיתוף בין מרכיבים בתכנית (בשפת סקרטץ' זו העברה מסרים בין דמויות ורקע).
3. התלמיד ידע לכתוב תסריט לחישוב ביטוי. התסריט יופעל עם לחיצה על הדמות.
4. התלמיד יכתוב קטע עם חישוב ביטוי שפותר בעיה הנדסית/מתמטית כשבסיום החישוב הביטוי שולח מסר לדמות שמבצעת איור גרפי.

5. התלמיד ידע לכתוב תסריט שיתבצע על ידי שתי דמויות במקביל.

6. התלמיד ידע לכתוב תסריטים שהאירוע להפעלתם יהיה 'כאשר נלחץ מקש'. התסריטים יכללו שימוש בארבעה מקשים.

מטרות ביצועיות - דוגמאות

1. כתוב אלגוריתם לתיקון תקר באופניים.

תיקון תקר באופניים:

- שחרר את בורג הגלגל והוצא את הגלגל ממקומו
- פרק את הצמיג החיצוני
- הוצא את הפנימית
- תקן את התקר
- הרכב את הפנימית
- הרכב את הצמיג
- הרכב את הגלגל
- סגור את הבורג

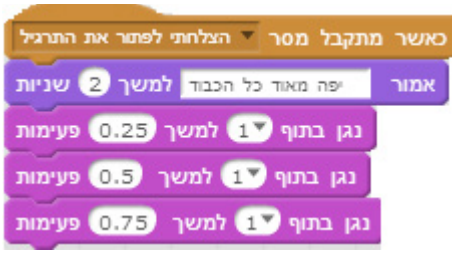
חשוב! סדר ההוראות. דון עם התלמידים על התוצאות של שינוי ההוראות באלגוריתם, שים לב שבשלב הזה אנחנו לא קוראים למה שכתבנו אלגוריתם מילולי – כלומר אין הקפדה על ניסוח קבוע ומספור הוראות וכדומה – הכתיבה חופשית וההוראות אינן ממוספרות.

2. כתוב תסריט לדמות א' לחישוב והצגת הכפולה 5X3. התסריט יופעל עם הקשה על מקש ארוך. דמות ב' תקבל מסר בכל סיום של חישוב ותשמיע צליל התפעלות (בחר כרצונך את הצליל)

התסריט של דמות **א'**



התסריט של דמות **ב'**



3. כתוב תסריט קטע להצגת כל הכפולות של מספרים שלמים לקבלת הערך 8 – לדוגמה 2X4. התסריט יופעל עם לחיצה על הדמות.

חשיבה מקדימה: מה הן כל הכפולות של מספרים שלמים לקבלת הערך 8.

1X8 (8X1)

2X4 (4X2)

אתגר: אם הדרישה הייתה להציג את כל הסכומים של שני מספרים שתוצאתם היא 8, אלו ביטויים יש לרשום?



4. כתוב תסריט שיחשב היקף של מלבן (בחר את אורך הצלעות). הדמות שמחשבת את ההיקף תשלח מסר לדמות שתצייר את המלבן

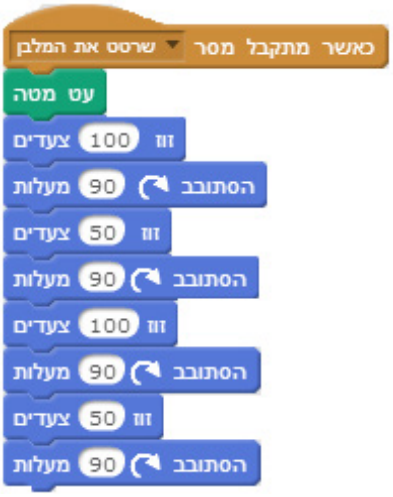
השתמש בהוראות:



דמות א' מבצעת את המשימה ושולחת מסר



דמות ב' מקבלת את המסר ומבצעת את התסריט 'ליצור מלבן'



5. כתוב תסריט בו שתי דמויות גולשות אחת לעבר רעותה.

מטלה זו מציגה את הרעיון של עיבוד במקביל כששתי הדמויות מבצעות יחד את התסריטים. אין להדגיש את המושג בפני התלמידים אולם רצוי לציין בפניהם את הביצוע – אנחנו רואים שתי דמויות נעות ומבצעות תסריט שונה באותו הזמן (כמעט).

התסריט של





התסריט של





6. כתוב תסריט שיאפשר יצירת 'צייר'. כדי ליצור את התסריט השתמש בהוראה **'עט מטה'**; הוראות 'פנייה' ו'תנועה'; ההוראה **'כאשר נלחץ מקש'**.

תכנון היישום:

פעולות לאתחול היישום יכללו:

ניקוי מסך

הורדת עט

ארבעה תסריטים לתנועה לכל אחד מ-4 הכיוונים : 90 ,-90, 180 ,0

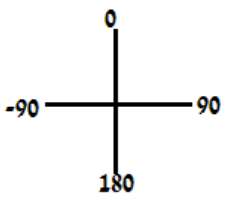
כל תסריט יכלול:

אירוע – לחיצת מקש לפי בחירה לתחילת פעולת התסריט

א. פנייה לכיוון הנכון

ב. תנועה בכיוון ובגודל מסוים

הכיוונים בזירה :



חשוב! הכיוונים שנבחרו ע"י יוצרי סביבת סקרטץ' אינם כיוונים 'טבעיים' שמובנים לתלמידי כיתה ד' או ה' מידע קודם. אין בכיוונים התייחסות לכיוונים הטבעיים: שמאלה, ימינה, למטה, למעלה. במידה רבה הסביבה מייצגת כיוונים בבחירה שרירותית ולכן זה לא טבעי ולא מתחבר לתלמידים עם ידע וניסיון קודמים. לכן חשוב לתרגל באמצעות דוגמאות להמחשה ולהבנת הכיוונים בסביבה.

מומלץ לתת לתלמידים לכתוב שתי הוראות, הוראה אחת '**פנה לכיוון**', ואחת '**זוז_צעדים**' ולתרגל את הכיוונים.



צייר בסיסי

הדמות שנבחר: עיפרון

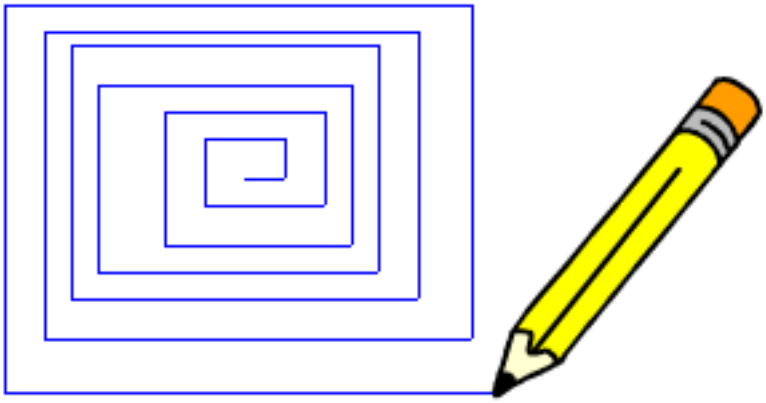
באמצעות אפשרות עריכת 'תלבושת' ניתן למרכז את הדמות כך שקווים הנכתבים על ידי העיפרון יכתבו 'מהקצה המחודד של העיפרון'.



התסריטים:



דוגמה לתוצאה של הרצת התסריטים:



תרגול בדיאלוג אדברסיאלי – על המשמעות של חד-משמעות של הוראה (מקור: ד"ר מאיר גולדברג)

התלמידים מתחלקים לזוגות. תלמיד אחד (הראשון) צריך לתאר מטלה כלשהי: בכתב; עם חלוקה לסעיפים; תוך פירוט מלא של מה צריך בן הזוג השני לעשות על מנת לבצע את המטלה (להכפיל שני מספרים דו-ספרתיים עם נייר ועפרון, למחוק את הלוח, לרוקן פח זבל, וכדומה). התלמיד השני צריך לבצע את המטלה תוך שהוא עוקב אחרי השלבים שתיאר התלמיד הראשון צעד אחר צעד, באופן מדויק – ואדברסיאלי – כלומר: מטרתו של התלמיד השני היא להכשיל את התלמיד הראשון. הוא ממלא אחר ההוראות כלשונן אבל בכל פעם שהפקודות אינן מספיק מדויקות מותר לו לפרש אותן באופן שיסתור את כוונת התלמיד הראשון. תפקיד המורה (או תלמיד אחר שסופח לקבוצה למטרה זו) יהיה לפקח ולוודא שאכן התלמיד השני פועל בהתאם לכללים. כשהתלמיד הראשון מסכים שהכללים שתיאר אינם מספיק ספציפיים, אינם מספיק חד-משמעיים ובאופן כללי אינם מספקים בכדיי לכפות על בן זוגו למלא את המטלה, התלמידים יתחלפו בתפקידים.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
נקודת מוצא ויעד סופי	להדגיש את רעיון התהליך, יש נקודת מוצא ויש יעד סופי. התהליך מוביל מנקודת המוצא ליעד הסופי	✓		כתיבת הדרך לפתרון מטלה מנקודת המוצא ליעד פתרון בכמה דרכים חשיבות סדר ההוראות **2 שעות**
מרעיון לקוד	כעת אנחנו מדברים במושגי תכנית וקטע של תכנית. בסביבת סקרטץ' אפשר לדבר במושגים של תסריט ובהמשך במושגים של ריבוי תסריטים ומקבילות	✓	✓	פתרון מטלות מתחום המתמטיקה **1 שעות** פתרון מטלות מתחום הגיאומטריה **1 שעות** בכל התרגול יש לעבוד על היכרות טובה של סביבת הלמידה והוראות ביצוע נוספות
יישומים מרובי דמויות מרובי תסריטים תכנות מונחה אירועים	אנחנו מנצלים את המאפיינים של סביבת סקרטץ', שפה בה תהליכים מופעלים כתוצאה מאירועים המתרחשים בעת הפעלת התכנית כגון: כשיתקבל מסר, כאשר נלחץ מקש, כאשר נלחץ דגל ירוק סביבה/פרדיגמה EVENT DRIVEN PROGRAMMING			התלמיד יבצע מטלות הכוללות מספר דמויות, לכל דמות מספר תסריטים (2-3 דמויות, 2-3 תסריטים לדמות, 1-2 יתבצעו במקביל) **3 שעות**

פרק ג' ביטויים לוגיים, תכנות מרובה דמויות,
ריבוי תסריטים ועיבוד מקביל

תחום תוכן: כתיבת קוד, חישוביות, אלגוריתמיקה ומיומנויות המאה 21



יחס לוגי; ביטוי לוגי

יחס לוגי הוא אופרטור של פעולה שמחבר בין שני ביטויים ויוצר ביטוי לוגי.

לדוגמה: 3<6; < הוא היחס הלוגי, קבלנו ביטוי לוגי שנוכל להגיד האם הוא אמת או שקר.

דוגמאות נוספות: d<=4, התוצאה של ביטוי לוגי היא 'אמת' או 'שקר'.

שפת סקרטץ' מזמנת לנו ביטויים לוגיים שקיימים בכל שפות תכנות:



ופסוקים לוגיים ייחודיים לסביבת סקרטץ':





ערך אמת; ערך שקר

כל ביטוי לוגי יקבל את אחד הערכים, ערך **'אמת'** או ערך **'שקר'**.

אם הביטוי הוא 'האם נוגע בדמות', ואכן דמות אחת נוגעת בדמות שנייה שצוינה, הערך של הביטוי יהיה **'אמת'**. אם אין נגיעה של שתי דמויות, ערך הביטוי יהיה **'שקר'**.

דרכי הערכה

1. מתן הגדרות לביטויים לוגיים בשפה טבעית למימוש באמצעות קוד.
2. מתן מטלות המחברות הוראות חישוב, ביטויים עם מפעילים מתקדמים, הצגת תוצאות ושילוב תסריטים.
3. מתן מטלות המשלבות הוראות 'תנועה', הוראות 'סיבוב', הוראת **'פנה לכיוון'** ושילוב ביטויים לוגיים.
4. מתן מטלות עם עיבוד מקבילי (תסריטים המבוצעים באותו הזמן – כמעט).
5. מתן מטלות עם זימון תלוי תנאי של התרחשות אירוע: 'כאשר לוחצים על דמות', 'כאשר התקבל מסר', 'כאשר לוחצים על מקש'.

מטרות ביצועיות

1. התלמיד ינסח טענה מתמטית שמייצגת ביטוי לוגי ויבחר את המפעיל המתאים לייצוג הביטוי. חשוב לעודד את כתיבת השלבים לביצוע בעברית חופשית.
2. התלמיד ידע לכתוב תסריט שיתבצע במקביל על ידי שתי דמויות. במסגרת יחידה זו צריך לשלב תרגול של ביטויים מתמטיים ושל ביטויים לוגיים.
3. התלמיד יכתוב תסריט הכולל כמה דמויות וכמה תסריטים. חלק מהתסריטים יופעלו במקביל. בתסריטים ישולבו ביטויים מתמטיים, ביטויים לוגיים והתניות לוגיות (כמו 'נוגע ב')

מטרות ביצועיות - דוגמאות

1. מתן יעד לכתיבת ביטויים לוגיים ומימושם באמצעות קוד. לעודד את כתיבת השלבים לביצוע בעברית חופשית.

האם המכפלה של 4 ב-2 גדולה מ-12



האם החלוקה של 15 ב-5 היא ללא שארית



האם הסכום של 3 והמכפלה של 5 ב-12 גדולה מ- 63









חשוב! בכתיבת ביטויים מתמטיים יש סדר פעולות ויש סוגריים. בתחביר של סקרטץ' הם באים לידי ביטוי באמצעות השילוב שעושים בין המפעילים, כך שהשילוב של המפעילים המתמטיים בתוך המפעיל הלוגי יהיה זהה לביטוי הלוגי שהתלמידים כותבים. המטלה הזו מחברת בין הבנה ושליטה בסדר הפעולות ובמשמעות של סימני היחס <,>,= לבין העברת הביטויים לשפת תכנות בהתאם לתחביר השפה.

יש להדגיש ולהפנים ש-**'שקר'** ו**'אמת'** (TRUE/FALSE) הם ערכים קבועים בשפה ולא מילים (כמו 'כן/לא' או 'נכון/לא נכון').

משימות תרגול

דמות 1 מחשבת ממוצע של שלושה מספרים. דמות 2 תקבל מסר ותנוע לכיוון דמות 1

תסריט דמות 1

חישוב ושליחת מסר



תסריט דמות 2

קבלת מסר וביצוע פעולות



2. א. האמודאי נע לכיוון הדג

דוגמה זו נבחרה בכדי לשנות את הרקע של הבמה. בתסריט האמודאי נע לעבר דמות הדג

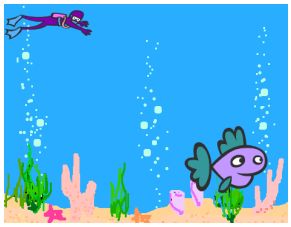
הדמויות בדוגמה זו:



Fish1	Diver1
דג	צוללן

השמות באנגלית הם השמות של סביבת סקרטץ' ואנחנו יכולים לשנות אותם. אפשר להשתמש בשמות ברירת המחדל או לבחור שמות אחרים בעברית

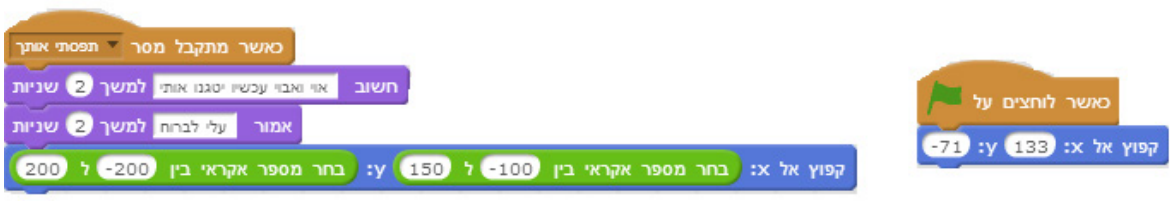
שינוי רקע הבמה והתוצאה של בחירת הדמויות והרקע:



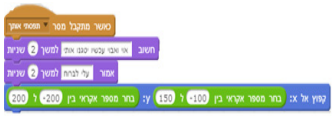
תסריטי דמות הצוללן



תסריטי דמות הדג



טבלת תהליכים

דמות	כאשר לוחצים על	כאשר לוחצים על	כאשר מתקבל מסר
			
			
			

ב. שועל מנסה לתפוס ברווז

השועל מנסה להקטין את המרחק בינו לבין הברווז. הברווז עוקב אחר סמן העכבר. האם יפגשו?

הדמויות :

הברווז 
Duck

השועל 
Fox

הבמה 

תסריטי הברווז





תסריטי השועל





אתגר: לחשוב על תסריטים שונים בהם המרחק בין הברווז לשועל יגדל והסיכוי של השועל לתפוס את הברווז יקטן (הבדיקה בהפעלת התסריטים) – ניתן להשתמש בהוראות תנועה שונות ולשנות ערכים במפעיל **'בחר מספר אקראי'**.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
ביטוי לוגי	דיון מהו ביטוי לוגי דוגמאות	✓	✓	התלמיד ידע מהו ביטוי לוגי ומהו ערך של ביטוי לוגי באמצעות שילובו בתסריט ודרך הצגת הערך עם ההוראה '**אמור**' 1 שעות כתיבת ביטויים לוגיים כתיבת קוד עם ביטוי לוגי 1 שעות
קוד, דמות וגרפיקה	חיבור לרעיונות של כתיבת תכנית בדרך של תכנון. תכנון מכלול התכנית, בחירת הדמויות, תכנון העברת מסרים בין דמויות ועיבוד מקבילי	✓	✓	מטלות כיתה עם הוראות תנועה: '**פנה**' ו'**סיבוב**'. שילוב הוראות התנועה עם פסוקים לוגיים (כמו: 'נוגע ב') 2 שעות

פרק ד' משתנים

תחום תוכן: כתיבת קוד, חישוביות, אלגוריתמיקה ומיומנויות המאה 21

מושגים

- ✓ משתנה
- ✓ מחרוזת
- ✓ ערך מספרי
- ✓ ערך לוגי
- ✓ אתחול ערך
- ✓ שינוי ערך

משתנה

כל משתנה הוא 'קופסה' שבה ניתן לאחסן ערכים. כל משתנה מאוחסן בזיכרון המחשב ולמקום האחסון יש כתובת בזיכרון. אולם בכדיי שנוכל להתייחס למשתנה בקלות עלינו לתת לו שם ייחודי. מותר לתת שם בעברית או באנגלית.



משתנה משנה את ערכו לפי ההוראות שהמתכנת משלב בתסריטים.
נניח שיש לנו משתנה הנקרא 'פרי', הוא מכיל עצמים מסוג פירות.

המשתנה יכול להכיל ערכים שונים בזמנים שונים בעת הרצת תסריט:



עתה נכתוב את 'ההוראה' –שים את העצם בננות במשתנה 'פרי' ונקבל:



אנחנו רואים שערך התוכן של המשתנה 'פרי' השתנה ועתה הוא מכיל בננות.

יש לשים לב שאם ניתן פקודה חדשה:

שים את העצם תפוח במשתנה 'פרי', אז התוכן של 'פרי' יתחלף והוא יכיל תפוח.



סוגי ערכים המאוחסנים במשתנים – DATA TYPE

מחרוזת - מילה

אוסף של תווים המורכב מאותיות, ספרות וסימנים נקרא מחרוזת או בשם שמוכר בסביבת סקרטץ' 'מילה'. ישנן פעולות מיוחדות שניתן לבצע על משתנה המכיל מחרוזת כמו לחשב את אורך המחרוזת-מילה, או לגשת לתו במיקום מסוים במחרוזת-מילה. ניתן לשרשר שתי מחרוזות או יותר.



ערך מספרי

ערך מספרי הוא ערך שמורכב מספרות אם מדובר בערך שלם. אם מדובר בערך ממשי הוא יכלול גם נקודה עשרונית. מספרים יכולים להיות גדולים מ-0, קטנים מ-0 או שווים ל-0. מספרים הם חלק מביטויים מתמטיים שאנחנו משתמשים בהם בתסריטים שאנחנו כותבים. מספר יכול להיות ערך המוגדר 'קבוע', לדוגמה 7, או 9.787, או ערך המאוחסן במשתנה ואז ניתן לשנות אותו.

ערך לוגי

ערך לוגי יכול לקבל אחד משני ערכים: אמת (true), שקר (false). ישנם גם ביטויים לוגיים שחישובם יהיה אחד מהערכים 'true' או 'false'. לדוגמה ביטוי לוגי: $8 < 5$, $x+3>8$, או לדוגמה החיישנים בסקרטץ': 'נוגע בצבע', 'נוגע ב' – כאשר הם מחושבים תוצאתם תהיה 'true' או 'false' (אמת או שקר)

משתנה, אתחול ערך, שינוי ערך

משתנה (variable) הוא תא זיכרון בתכנית המכיל נתון שיכול להשתנות בזמן הריצה (הפעלה) של התכנית. משתנה בשפת תכנות מקבל שם, כשבמרבית השפות שם המשתנה כולל תווים מסוג אותיות, ספרות וסימנים נוספים (כמו: _). בשפת סקרטץ' ניתן לתת שמות בעברית, לא מומלץ להתחיל שם משתנה בספרה; לתת שם כמו '1' או '23'. חלק מהשפות כוללות בהגדרה של המשתנה את סוג הערכים שהמשתנה יכול להכיל (שלמים, ממשיים, תווים, בוליאני). בשפת סקרטץ'

אין צורך להכריז על סוג הערכים כי הסוג נקבע לפי הערכים המושמים במשתנה ואפשר להשים ערכים שלמים, ערכים ממשיים, וטקסטים (מחרוזת – String).

אתחול ערך משתנה, שינוי ערך

אתחול ערך משתנה הוא לעיתים קרובות חלק מתחילת ביצוע אלגוריתם. מומלץ לאתחל משתנים ולא להניח שיש בהם ערך התחלתי הניתן באופן אוטומטי (לדוגמה בערך מספרי מאתחלים ב-0 או 1 וכו..), דוגמה: ההוראה אתחל ערך משתנה בשפת סקרטץ'



הערך עשוי להשתנות במהלך ריצה באמצעות הוראת השמה. הוראת **'ההשמה'** (השמה – קביעת ערך) משמעותה ערך חדש במשתנה. השמת הערך דורסת את הערך הקיים במשתנה.

דוגמאות למשפטי השמה בשפת סקרטץ':



קלט

רכיבי הקלט הם הרכיבים המאפשרים לנו להעביר מידע למחשב. בין רכיבים אלו נציין את המקלדת ואת העכבר. באמצעות המקלדת והעכבר אנחנו מעבירים מידע למחשב וכך אנחנו יכולים להשתמש בשפת סקרטץ'. בהוראה **'כאשר נלחץ מקש__'; 'כאשר לוחצים על דמות זו'** (עם עכבר) – אנחנו יכולים לקלוט גם ערכים שונים באמצעות המקלדת. הערכים האלו מושמים לתוך משתנים במהלך הרצת התסריט. בשפת סקרטץ' הוראה לביצוע קלט מהמקלדת היא:



חשוב לציין שקלטים ממקלדת ועכבר (ומרכיבי קלט נוספים) משולבים בפרדיגמה הכוללת של תכנות מונחה אירועים בסביבת סקרטץ'.

המשתנה 'תשובה' בסקרטץ'

בשפת סקרטץ' מוגדרים משתני מערכת. אלו משתנים שאין צורך להגדירם והם חלק משפת התכנות. משתנה מיוחד הוא 'תשובה' - 

משתנה זה מכיל קלט שמתקבל באמצעות הפקודה **'שאל'**. הערך ממשתנה 'תשובה' ניתן להעברה למשתנה אחר שהוגדר על ידי הכותב,לפי הצורך.

דרכי הערכה

1. מתן מטלות לביצוע קלט של ערכים למשתנה המערכת 'תשובה'. ביצוע פעולות מתמטיות עם הערך שהתקבל, שימוש בהוראת האתחול והוראת ההשמה.
2. מתן מטלות הדורשות כתיבת תיאור מילולי של תהליך עם נקודת מוצא ונקודת יעד ומימושו בסביבת העבודה. במטלה ייכלל שימוש במשתנים, בגרפיקה, במסרים בין דמויות ובעיבוד מקבילי.

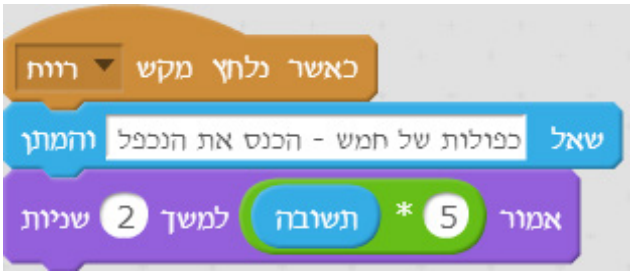
מטרות ביצועיות

1. התלמיד יכתוב תסריטים המשלבים קלט של ערכים עם ביצוע פעולות מתמטיות ועם הצגת התוצאה.
2. התלמיד יכתוב תסריט המשלב את ההוראה **'כאשר נלחץ מקש'**. הוא יבנה תסריטים שונים על פי תפריט ובכל תסריט יטפל באפשרות שנבחרה לפי התו שהוקש.

1. התלמיד יכתוב תסריטים המשלבים קלט של ערכים וביצוע פעולות מתמטיות יחד עם הצגת התוצאה.

בתרגול יש להשתמש בבעיות 'מילוליות' עם משתנה יחיד, וליצור רעיון לפתרון המטלה ולכתיבת תסריט שיש להריץ מספר פעמים עבור קלטים שונים.

חישוב מכפלה של – 5



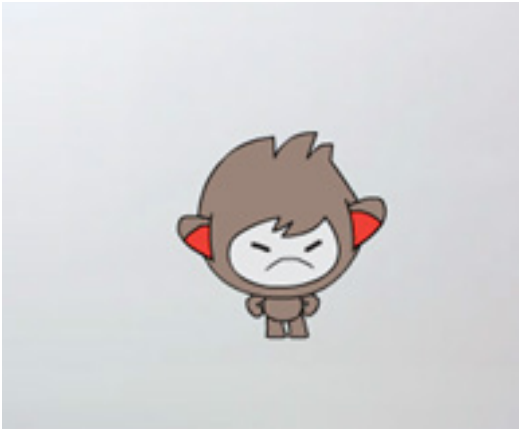
- חישוב שטח והיקף ריבוע
- צייר ריבוע לפי אורך צלע מבוקש
- צייר משולש שווה צלעות לפי אורך צלע מבוקש
- עלות מחברת 5 ₪, חשב עלות מספר מחברות לפי קלט
- חשב את המספר העוקב למספר שנקלט
- צור מספר אקראי וחשב את מכפלתו במספר מהקלט
- חשב מנה של מספר המתקבל מהקלט במספר 2

הערה: הדוגמה כוללת שימוש בשני משתנים, אם יש בכך קושי מול קבוצת התלמידים נא לא לעשות בה שימוש.

2. התלמיד יכתוב תסריט המשלב את ההוראה '**כאשר נלחץ מקש**', יבנה תסריטים שונים על פי תפריט ובכל תסריט יטפל באפשרות שנבחרה לפי התו שהוקש. בדוגמה זו: (x,/,+,-) ארבע פעולות חשבון.

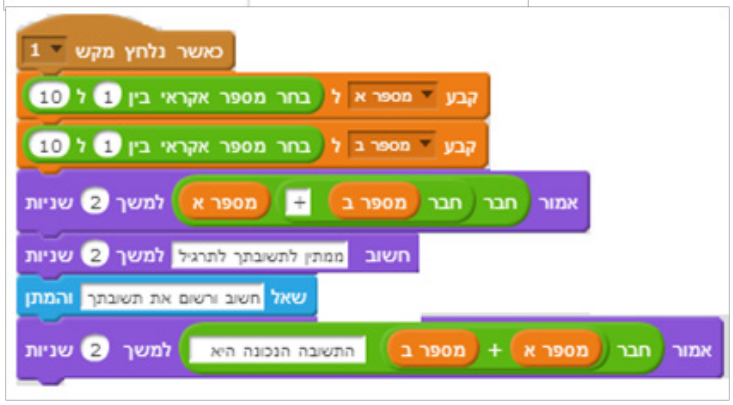
א. משחק המציג תרגיל באחת מפעולות החשבון

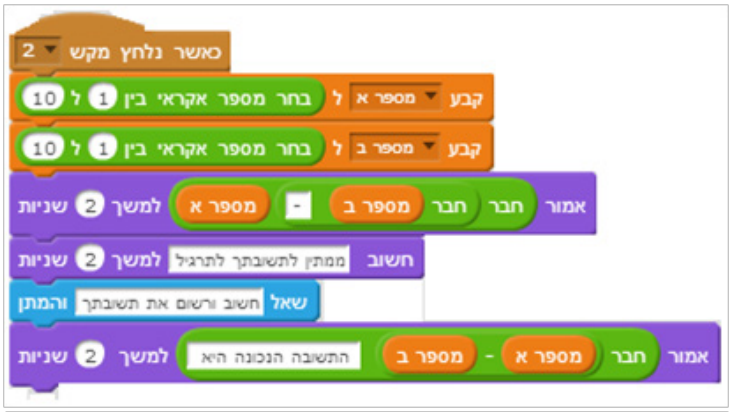
דוגמה: המורה לחשבון: הדמות Nano (שים לב הרקע שונה לתמונת 'לוח מחיק' בכיתה).

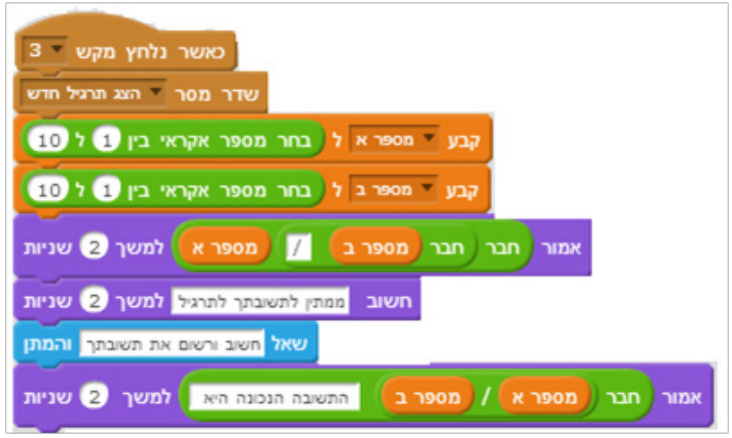


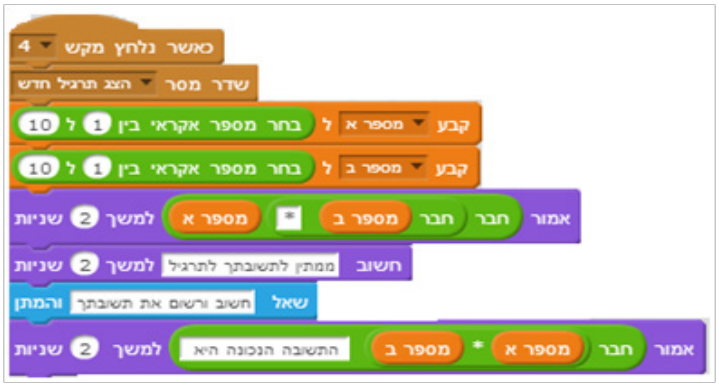
התסריטים של שיעור בחשבון בארבע פעולות החשבון:











חשוב! בפרויקט תלמידים יכולים להשתמש בהוראות שלא נלמדו בצורה ישירה בכיתה ושיעשירו את התוצר: מוסיקה, תנועה וסיבוב של הדמות, שינוי רקע, שינוי תלבושת. למידה עצמאית בדרך של גילוי תעודד את התלמידים לחקור. ניתן להיעזר בנספחי סיכום הוראות ומושגים בסביבת תכנות בשפת סקרטץ'.

מומלץ לחזור לפרויקט זה לאחר שנלמד את ההוראות: **'אם'**, **'אם ואם לא'** ואז נוכל לבדוק את תשובת המשתמש: האם ענה נכון? (אם ענה נכון תוצג הדמות של Nano מחייך ואם לא תוצג הדמות של Nano עצוב).

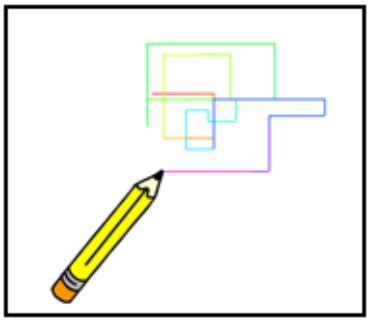
ב. שינוי בצבע העט – שיפור פרויקט הצייר

הפרויקט כולל את דמות העיפרון שנע על המסך ומשנה את צבעו לפי ערך משתנה בשם 'צבע'. חשוב להפנות את תשומת לב התלמידים לייצוג של ערך המשתנה 'צבע' בקצה העליון של הבמה:









שינויים שניתן להציג לתלמידים:

א. שינוי הצבע הוא בהוספת 1 לערך המשתנה 'צבע'. ניתן לקבוע ערך שינוי גדול יותר; ניתן להוסיף תסריט מבוסס אירוע של מקש אחר נניח המקש 'M' והקוד יקטין את ערך המשתנה 'צבע'.

ב. ניתן ליצור משתנה נוסף של גודל עט ולכתוב תסריט מבוסס הקשת מקש לדוגמה 'S' . וכך, עם שני האירועים – שינוי מספר הצבע של העט ושינוי גודל העט – הציורים שניצור יהיו מעניינים. התוספת של התסריט והציור שאנו מקבלים מהרצת ניסיון:

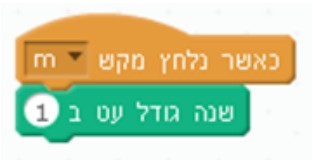
המשתנה שהוספנו: 

התסריט שהוספנו, וציור שנוצר בהרצה:





פתרון אפשרי המתאים לרמה א' עם שימוש במשתני מערכת והוראות מיוחדות של קבוצת ההוראות 'עט':



חשוב: מעניין וחשוב לדון בכיתה על שם המשתנה שבחרנו. שנו את התסריט רק בשם המשתנה ובחרו שם אקראי (נניח 'אאאא'). קיימו דיון על ההבדל, על השפעת שינוי השם על ביצוע הקוד ולמה חשוב לתת שמות משמעותיים.

ג. שינוי גודל דמות

הפרויקט הבא מציע שימוש נוסף במשתנה – משתנה הקובע את גודל הדמות. כל היישום מונחה אירועים, מעניין וממחיש לתלמידים שימוש מעניין במשתנה באמצעות הוראות מקבוצת ההוראות עט

כולל ההוראות: 

היישום, כולל תסריט אתחול (דגל ירוק), תסריט הגדלת הדמות, תסריט הקטנת הדמות, תסריט החתמת הדמות, תסריט תנועה ימינה של הדמות, תסריט תנועה שמאלה של הדמות.



הבמה לאחר הרצת דוגמה:



יישום של הרעיון ללא משתנים

היישום שהוצג עבור – יישום צייר מתקדם – ניתן לתכנון ויישום ללא שימוש במשתנים אלא באמצעות משתני המערכת של שפת סקרטץ' המתייחסים לצבע, גודל עט וגודל דמות.

לדוגמה שינוי גודל דמות ניתן לבצע גם כך:



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
קלט לתכנית	דיון ברעיון של הפעלת תסריט עם ערכים שונים, רעיון בסיסי של **נכונות התכנית** – יש לקיים דיון ברמה בסיסית בשאלה זו. הכנסת קלט מאפשרת יותר חקר והצגת פתרונות עבור מצבים שונים. הבנת הרעיון של קלט שמאפשר הפעלת תסריט עם ערכים שונים, וברמת משמעות יותר גבוהה – רעיון של תבנית מתמטית. המשתנה 'תשובה' מאפשר שימוש בשלבים ברעיון המשתנים בתכנית מחשב, אין צורך ליצור את המשתנה, את השם שלו 'תשובה' ניתן להסביר כ'תשובה' של המשתמש. זה נתיב שעוקף קשיים ראשוניים, בדרך לשימוש רחב יותר במשתנים.	✓	✓	הסבר מהי **'תשובה'** עם דוגמאות תוכניות פשוטות עם קלט **1 שעות**
קלט בעיות מילוליות - רב תחומיות כולל משתנים	עיון בנושאי לימוד במדע, טכנולוגיה ומתמטיקה והתאמת נושאים ובעיות לאלגוריתמים ותסריטים	✓	✓	קלט, בעיות מתמטיות וגיאומטריות וכתיבת קוד **1 שעות**
פרויקט	תלמידים יבחרו נושא מדעי – טכנולוגיה ומתמטיקה – ויכתבו תסריטים הקשורים לנושא.	✓	✓	תכנון מרכיבי הפתרון, נקודת מוצא, יעד ותהליך, הרצה עם קלטים שונים **3שעות** סיכום, הצגת תוצרים כיתתית משותפת **1 שעות**

פרק ה' משתנים - הרחבה: משפטי תנאי 'אם'; 'אם ואם לא'

תחומי תוכן: חישוביות אלגוריתמיקה ומיומנויות המאה 21, כתיבת קוד

מושגים

- ✓ משפט 'אם'
- ✓ משפט 'אם ואם לא'

משפט 'אם'

משפט **'אם'**, הוא מבנה בקרה על זרימת ביצוע תסריט.

כאשר המחשב מבצע קבוצת פקודות בכל שפת תכנות הוא מבצע את הפקודות באופן סדרתי, כלומר פקודה אחר פקודה.

אולם, לעיתים עלינו לבדוק האם התרחש מצב מסוים ולפעול בהתאם לתוצאת הבדיקה. משפט **'אם'** מאפשר לנו לבצע קבוצת פקודות ספציפית בשילוב ביטוי לוגי, רק אם התנאי שנבדק בביטוי הוא **'אמת'**.

משפט 'אם ואם לא'

בקרת זרימת ביצוע תכנית של משפט מסוג זה מאפשרת כתיבת קבוצת פקודות כאשר התנאי אכן מתקיים (**'אמת'**). במידה והתנאי אינו מתקיים, והערך הלוגי הוא **'שקר'**, מתבצעת קבוצה אחרת של פקודות.

מבנה **'אם'** ומבנה **'אם ואם לא'**; **'אם ואם לא'**, מאפשרים לנו לקבל תשובה **'אמת'** או תשובה **'שקר'** למצבים שנבדקים בהוראות אלו. בהתאם לתשובה שמתקבלת מתנאי הבדיקה אנו משלבים הוראות ביצוע.

דוגמה: תסריט בפרויקט הכולל שתי דמויות: כלב, חתול.

הוראות התנאי בודקת באמצעות המפעיל **'נוגע ב'**, האם דמות הכלב נוגעת בדמות החתול.

קטע דוגמה לשימוש בהוראה 'אם'



קטע דוגמה לשימוש בהוראה 'אם ואם לא'



חשוב! בקטע הראשון כלב ממשיך לנוע גם אם הוא נוגע בחתול וגם אם לא. בקטע השני כלב אומר "תפסתי אותך" אם הוא נוגע בחתול, ומתקדם לעבר החתול אם הוא לא נוגע.

דרכי הערכה

1. משימת מעבדה – כתיבת ביטויים.
2. משימת מעבדה – ציור מרובעים שונים לפי דרישה.
3. משימת מעבדה – פרויקט קטן. לדוגמה: הצגת תרגיל חישוב (כפל; חילוק; חיבור; חיסור), קבלת תשובה ומשוב.

מטרות ביצועיות

1. התלמיד יקבל מטלה מילולית הכרוכה בשימוש במשפט '**אם**' ויכתוב תסריט/ים למימוש המטלה.
2. התלמיד יקבל מטלה מילולית הכרוכה בשימוש במשפט '**אם ואם לא**' ויכתוב תסריט/ים למימוש המטלה.
3. התלמיד יישם משפטי '**אם**', '**אם ואם לא**' במשחק פשוט.
4. התלמיד יעצב את מבנה המטלה ואת מרכיבי המטלה וייצג עבור היישום משתנים לכלל הדמויות ומשתנים לדמויות מסוימות, בהתאם לצורך.

מטרות ביצועיות - דוגמאות

1.א. מי מנצח?

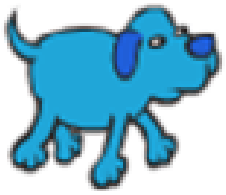
כתוב תסריט הקולט שני מספרים שונים (מספר נקודות שצברו חתול וכלב במשחק כדורסל) ואמור מי מנצח במשחק. אפשר להניח שמספרים שונים



ב. המרדף אחר החתול

כתוב תסריט שבו שתי דמויות 'ניצבות' בקצה הבמה; שתי דמויות פונות למרכז הבמה; באמצע הבמה ניצב חתול; הדמויות נעות עם הפקודה 'זוז' (ערך אקראי); הדמות שמגיעה ראשונה לחתול אומרת "אני הגעתי" ומעבירה לחתול מסר שישמיע "מיאו" ויאמר את שם הדמות שהגיעה אליו.

במטלה יש שילוב של: משפטי תנאי, העברת מסרים, עיבוד מקבילי

דמות	תסריט של דמות
 רקסי	
 ברק	
 חתול	 

סיכום הדוגמה: השתמשנו במשפט 'אם' כדי לבדוק איזו דמות נגעה בדמות החתול. שפת סקרטץ' מאפשרת לנו לנווט באמצעות הרעיון המרכזי של פרדיגמת תכנות מונחה אירועים, ולבחור אלו תסריטים יתבצעו לפי האירועים שמתרחשים ובאיזה סדר. כך השתמשנו בהוראות 'שדר מסר'; 'כאשר מתקבל מסר', כדי לטפל באופן שונה בהמשך הביצוע אם הכלב 'רקסי' הגיע ראשון או הכלב 'ברק' הגיע ראשון. האירוע דגל ירוק אתחל את תהליך תנועת הדמויות 'רקסי', ו'ברק' לעבר דמות החתול.

טבלת דמויות ותסריטים

דמות/ אירוע	כאשר לוחצים על	כאשר מתקבל מסר רקסי	כאשר מתקבל מסר ברק
 חתול	V	V	V
 ברק	V		
 רקסי	V		

טבלת דמויות ותסריטים

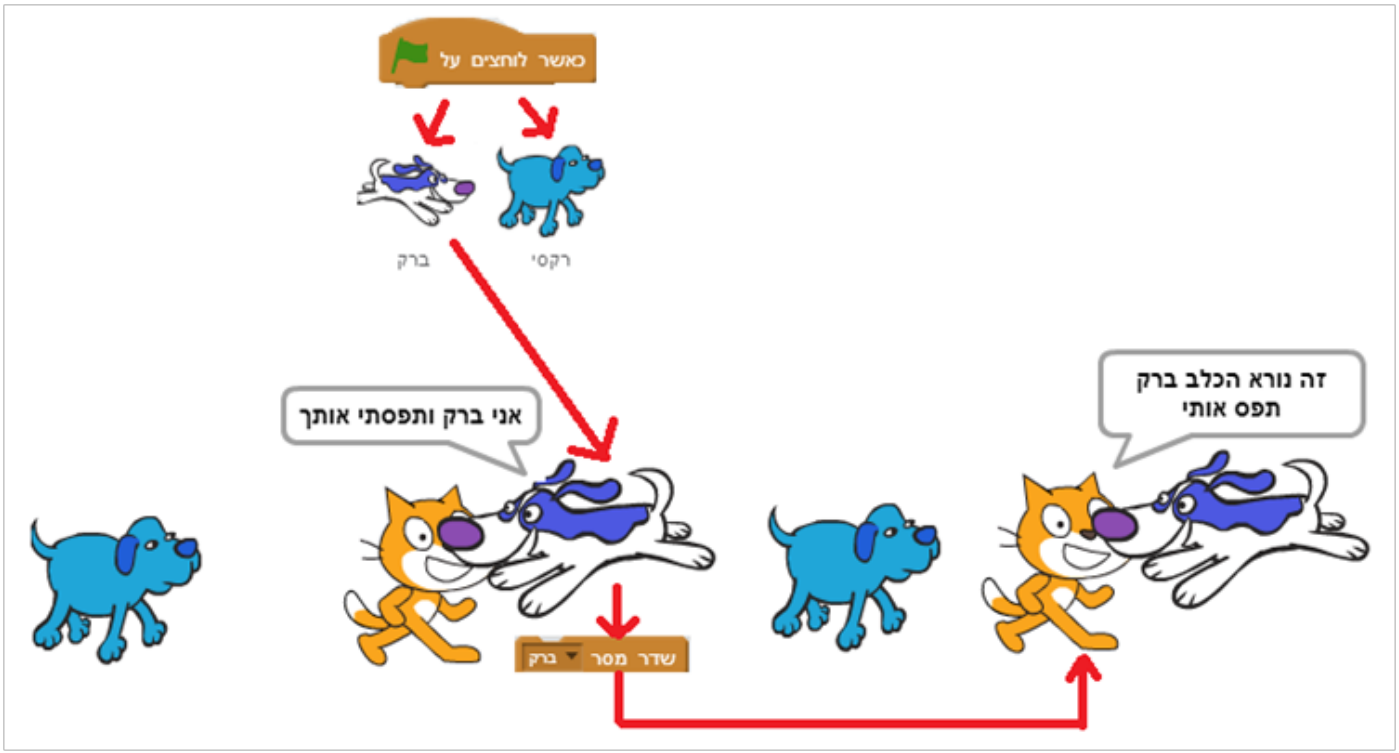
טבלת דמויות ותסריטים מציגה לנו את מבנה היישום, אלו אירועים מתרחשים במקביל ואת התסריטים שמופעלים בהתאם לאירועים שמתרחשים בעת ביצוע הקוד. בשלבים הראשונים של למידה וכתיבה של יישומים מרובי דמויות ומרובי תסריטים מומלץ שהמורה יציג את טבלת הדמויות והתסריטים עבור היישום המוצג. בהמשך התפתחות וההתקדמות בתכנית הלימודים מומלץ לאמץ את הטבלה ככלי מקדים לפני כתיבת קוד מפורט בעבודה בסביבת פרדיגמה מרובת תסריטים; תסריטים המופעלים על ידי אירועים; ותסריטים הפועלים במקביל. לא די לנו בכלים כגון אלגוריתם מילולי או, למי שמוצא לנכון, שימוש בתרשימי זרימה (לא מומלץ). אנחנו חייבים לפתח כלים חדשים המאפשרים תכנון בפרדיגמה של תכנות מונחה אירועים – טבלת דמויות ותסריטים היא כלי בסיסי בו נשתמש. כלי נוסף בו נעשה שימוש יהיה **תרשים תזמון האירועים**. האירוע **כאשר לוחצים על דגל ירוק**, מתבצע במקביל אצל כל הדמויות. האירוע מפעיל תסריט המאתחל את נקודת המוצא של ההדמיה.

תרשים תזמון ביצוע התסריטים:

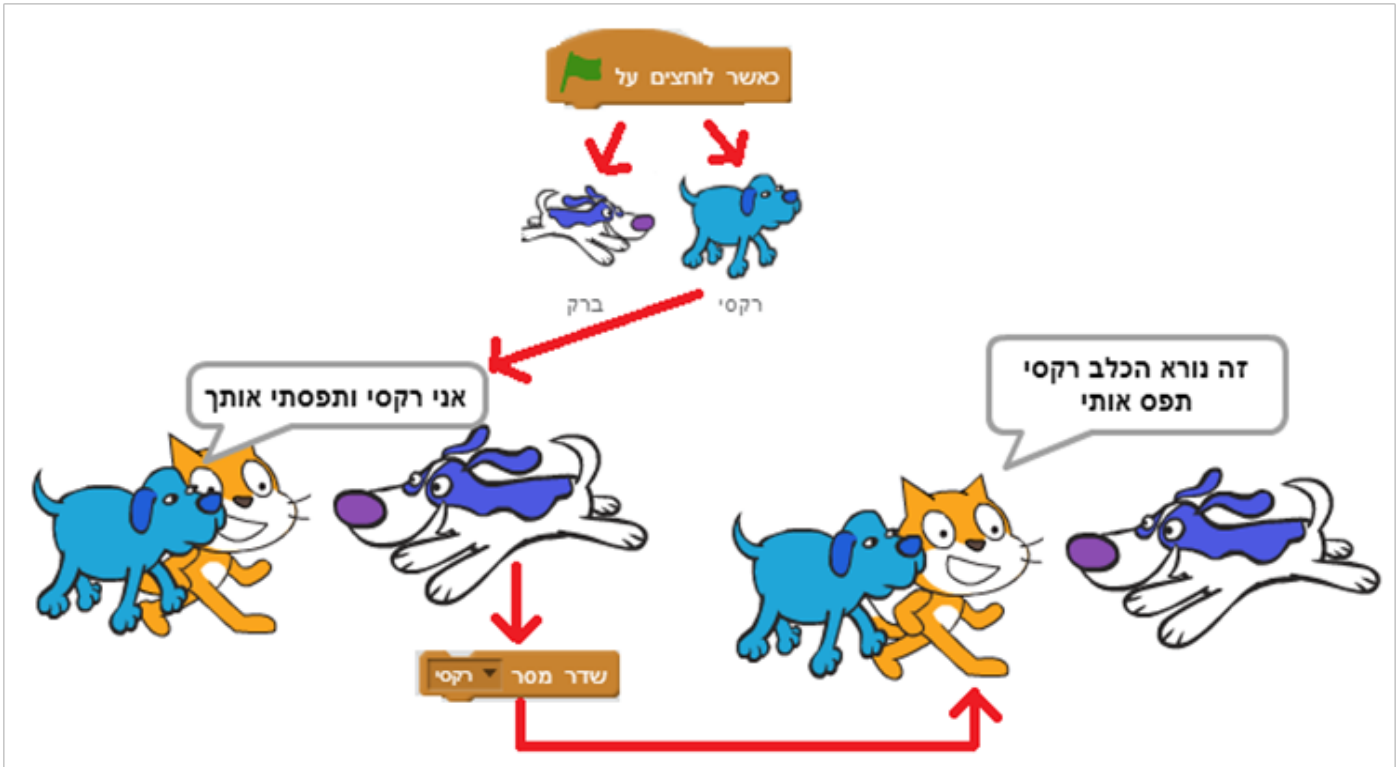
תרשים תזמון האירועים מסייע לנו להבין איך תכנות מונחה אירועים פועל. אנחנו יכולים לעקוב אחר ביצוע התסריטים באמצעות התרשים. בהמשך הלמידה ניתן להתחיל ללמד תלמידים לתכנן את התסריטים שברצונם לכתוב, זאת מאחר ואנו נמצאים בסביבה תכנותית מעט מורכבת. בממד אחד של התסריטים, כל תסריט פועל בצורה לינארית עם בקרת זרימה של סדרתיות, תנאים ולולאות, ומתרחש בסביבת תכנות פרוצדורלית. ואילו בממד שני של התסריטים הם מופעלים בתזמונים שונים בהתאם לאירועים המתרחשים ולעיתים קרובות גם במקביל. לכן הטבלה לעיל מציגה כל דמות והתסריטים שייכים לה, וכך אנו יכולים לראות את התהליכים המתבצעים במקביל ואת התסריטים המתבצעים בתגובה לאירועים השונים. תרשים תזמון האירועים מציג את 'התנהלות הסיפור שלנו' – רצף שבסיסו יהיה תמיד זהה אבל יהיו שינויים בתלות בזהות של הדמות שהגיעה ראשונה לחתול ובהתאם לכך הדמות שולחת מסר: חתול 'שמזהה' מי הגיע אליו ראשון ומגיב לאירוע כשהתקבל מסר.

תרשימי תזמון האירועים עבור שני המצבים: רקסי תפס את החתול ראשון, או ברק תפס את החתול ראשון.

תרשים תזמון אירועים – ברק תפס את החתול ראשון



תרשים תזמון אירועים – רקסי תפס את החתול ראשון



תרשים תזמון אירועים מביא לידי ביטוי את מימוש טבלת 'תסריטים ואירועים' בעת ביצוע, כלומר התרשים הוא מעין 'טבלת מעקב' אחר עולם הפרדיגמה הפרוצדורלית בעולם של הפרדיגמה EVENT-DRIVEN PROGRAMMING. כאן, בספר 'תכנית הלימודים', התרשים מוצג עם הדמויות ובצורה צבעונית אולם ניתן להציג את התרשים בצורה גרפית פשוטה עם כותרות בכתב יד וחצים, במיוחד אם רוצים כמה מעקבי ביצוע אחר התהליכים עבור 'הרצות' שונות של היישום.

2. מרוץ בעלי חיים

יוצג משחק בסיסי שבהמשך הלמידה, כאשר נלמד על לולאות, נוכל לחזור למשחק ולשפרו

איש - מזניק המרוץ

כלב, סוס, פיל – שלושת הדמויות המשתתפות במרוץ.

הדמויות שבחרנו:



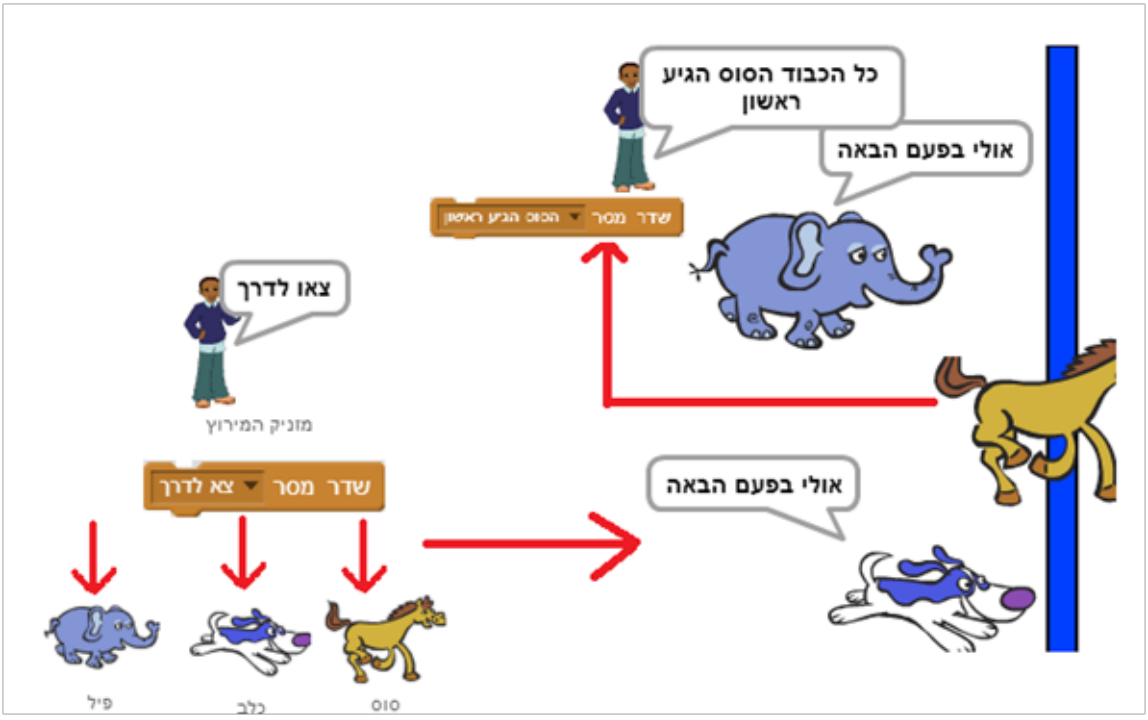
הרעיון המרכזי במשחק: המזניק מודיע על התחלת המרוץ. שלושת בעלי החיים קופצים. אם דמות נוגעת בצבע כחול – קו הגמר – הדמות משדרת מסר למזניק שהיא הגיעה לקו הגמר, ואם היא לא הגיעה לגמר הדמות אומרת "אולי בפעם הבאה".

טבלת דמויות ותסריטים

דמות/ אירוע	כאשר מתקבל מסר הגיע ראשון	כאשר לוחצים על	כאשר מתקבל מסר צא לדרך
מזניק המרוץ	כאשר מתקבל מסר הכלב הגיע ראשון כאשר מתקבל מסר הסוס הגיע ראשון כאשר מתקבל מסר הפיל הגיע ראשון	V	
פיל		V	V
כלב		V	V
סוס		V	V

האירוע **'צא לדרך'** מופעל על ידי המזניק ששולח מסר למשתתפי המרוץ: פיל, כלב, סוס.

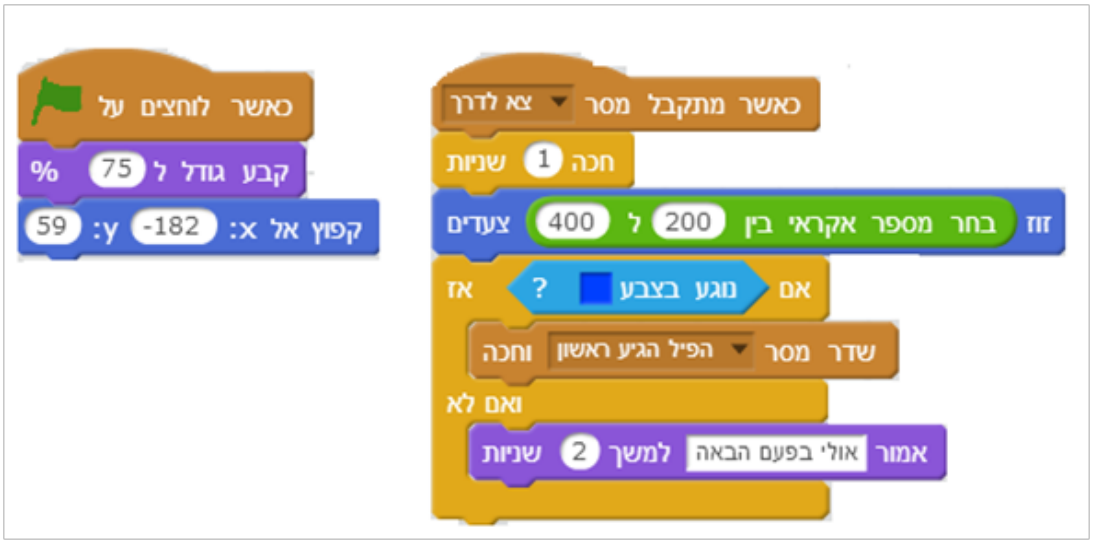
כל אחת מהדמויות: פיל; כלב; סוס, שמגיעה לקו הגמר, שולחת מסר למזניק שהיא הגיעה ראשונה, ולכן לדמות המזניק שייכים האירועים: מסר – **הכלב הגיע ראשון**; מסר – **הסוס הגיע ראשון**; מסר – **הפיל הגיע ראשון**



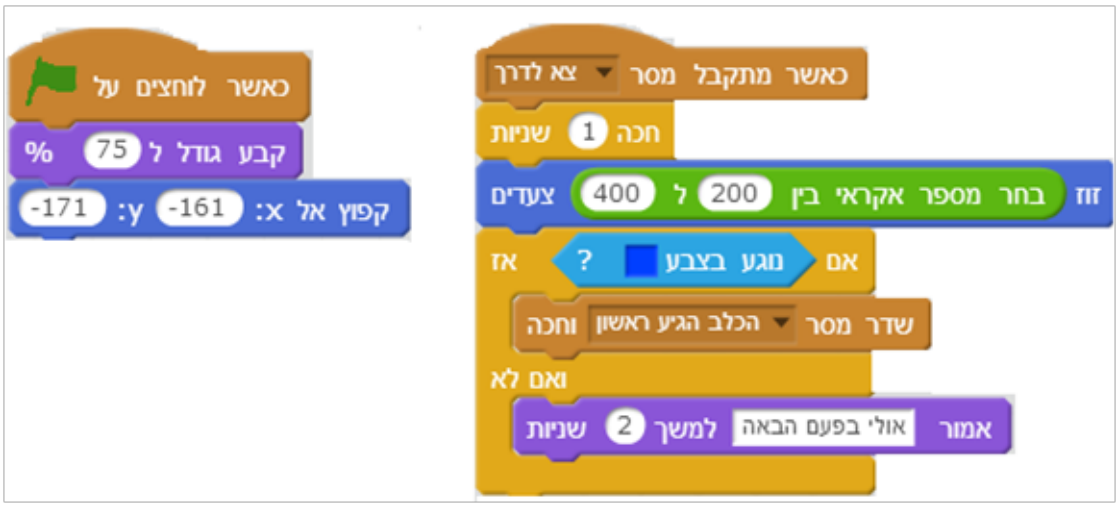
כפי שצויין בדוגמה שמוצגת למעלה **תרשים תזמון אירועים** מביא לידי ביטוי את מימוש טבלת 'תסריטים ואירועים' בעת ביצוע, כלומר התרשים הוא מעין 'טבלת מעקב' של עולם הפרדיגמה הפרוצדורלית בעולם של הפרדיגמה EVENT DRIVEN PROGRAMMING. כאן בספר 'תכנית הלימודים', הצגנו את תזמון האירועים עבור המצב בו דמות הסוס הגיעה ראשונה. כמובן שאפשר להציג את התרשים עבור שני המצבים הנוספים (כלב הגיע ראשון או פיל הגיע ראשון), אבל ניתן להציג תרשים אחד שתורם להבנת זרימת האירועים. כמו כן, אם בונים את התרשים לפני כתיבת היישום אפשר לתאר תרשים של הרצה מסוימת ולהכליל מתרשים זה לשאר המצבים.

פירוט התסריטים לפי דמויות

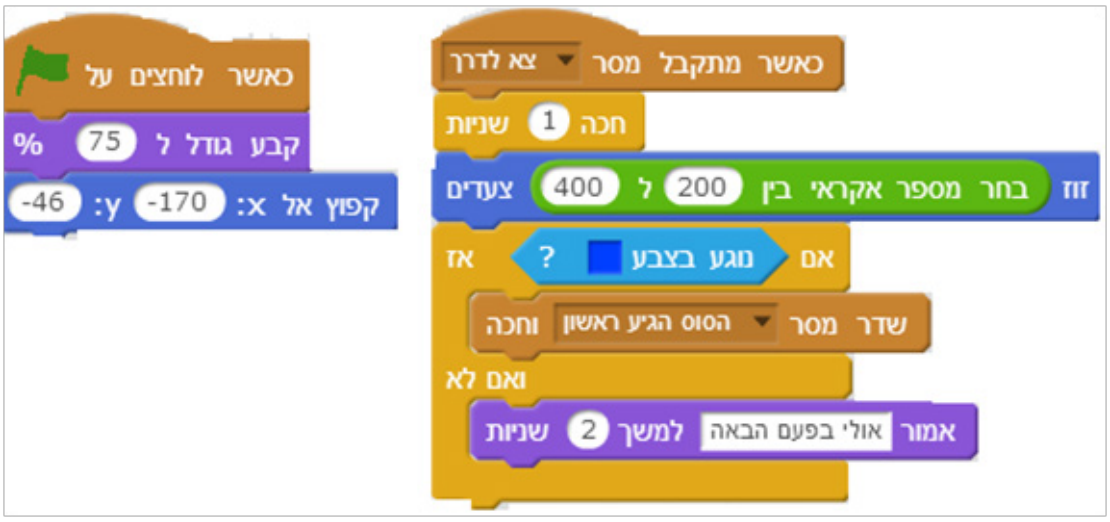



כאשר לוחצים על
קבע גודל ל 75 %
קפוץ אל x: -182 y: 59
כאשר מתקבל מסר צא לדרך
חכה 1 שניות
זוז בחר מספר אקראי בין 200 ל 400 צעדים
אם נוגע בצבע ? אז
שדר מסר הפיל הגיע ראשון וחכה
ואם לא
אמור אולי בפעם הבאה למשך 2 שניות




כאשר לוחצים על
קבע גודל ל 75 %
קפוץ אל x: -161 y: -171
כאשר מתקבל מסר צא לדרך
חכה 1 שניות
זוז בחר מספר אקראי בין 200 ל 400 צעדים
אם נוגע בצבע ? אז
שדר מסר הכלב הגיע ראשון וחכה
ואם לא
אמור אולי בפעם הבאה למשך 2 שניות




כאשר לוחצים על
קבע גודל ל 75 %
קפוץ אל x: -170 y: -46
כאשר מתקבל מסר צא לדרך
חכה 1 שניות
זוז בחר מספר אקראי בין 200 ל 400 צעדים
אם נוגע בצבע ? אז
שדר מסר הסוס הגיע ראשון וחכה
ואם לא
אמור אולי בפעם הבאה למשך 2 שניות



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
משתנים ריענון וחזרה	דיון כללי ודוגמאות	✓	✓	אתחול ערך של משתנה, שינוי בערך משתנה, עבודה עם כמה משתנים, פתרון בעיה אלגוריתמית עם ריבוי משתנים **2 שעות**
משפט 'אם', משפט 'אם ואם לא'	דיון כללי, דוגמאות, חשיבה על דוגמאות נוספות	✓	✓	פתרון בעיה אלגוריתמית או יצירת משחק קצר, אנימציה בסיסית. כל המטלות משולבות עם משתנים **2 שעות**
פתרון בעיות וביצוע מטלות	הצגת בעיה אלגוריתמית, דיון בהיבט המתמטי / מדעי / טכנולוגי של הבעיה, אלו שחקנים נכללים בבעיה ובפתרון (משתנים), תיאור שלבי הפתרון באמצעות ניסוח בעברית. יישום בקוד	✓	✓	הצגת בעיות אלגוריתמיות, ניתוח תכנון ויישום, ריבוי דמויות, ריבוי תסריטים, עיבוד במקביל **2 שעות**
משפט תנאי בסיסי	הצגת הנחיצות במשפט תנאי בדוגמאות מהחיים, מהקשרים מתמטיים, מדעיים וטכנולוגים	✓	✓	הבנה של משפט תנאי עם שילוב של ביטוי לוגי העברת רעיון ותהליך מותנה לתסריט עם משפט תנאי **1 שעות** מימוש משפטי תנאי עם כל האופרטורים הלוגים **קטן** < **גדול** > **שווה** = **1 שעות**

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
משפט תנאי מורכב	תנאי **'אם ואם לא'** דורש חשיבה ואנליזה של מצבים. כאשר התנאי הוא למשל ציון>0, התלמיד צריך להסיק מה המשמעות של **'ואם לא'** עבור התנאי. וזה נכון לכלל התנאים שיכתבו בהתאם למטלה	✓	✓	הצגה של ברירת שני מצבים ויישום הברירה במשפט **'אם ואם לא'** **1 שעות** שליטה ויישום שילוב אופרטורים לוגים שונים במשפטי **'אם ואם לא'** **1 שעות** פתרון בעיות מגוונות כולל מצבים לוגיים של יותר מדמות אחת – תנאים מקבוצת ההוראות 'חיישנים' (נוגע ב, נוגע בצבע, מקש_נלחץ וכדומה.) **1 שעות**
מעקב אחר תסריטים ותזמון הביצוע של התסריטים	הצגת הכלים: טבלת דמויות ותסריטים ותרשים תזמון תסריטים. הכלים חשובים להבנת אופן הפעולה של תכנות מונחה אירועים. תפקידם של כלים אלו לתמוך בפיתוח יישומים בפרדיגמת תכנות מונחה אירועים, וגם להבין יישומים של כותבים אחרים	✓	✓	כתיבת יישומים עם 2-3 דמויות ומספר תהליכים **2 שעות** שימוש ויישום כלים לפיתוח ומעקב אחר ביצוע, כולל: טבלת דמויות ואירועים, תרשים תזמון תסריטים **2 שעות**

פרק ו' תכנות מרובה דמויות, תלבושות ותסריטים -

הפשטה וייצוג מערכות 4-2 דמויות, 2-1 תלבושות לדמות , 2-1 תסריטים לדמות

תחומי תוכן: חישוביות, אלגוריתמיקה ומיומנויות המאה 21, כתיבת קוד



דמות שחקן

'דמות' היא מרכיב מרכזי בשפת סקרטץ'. עבור דמויות אנחנו כותבים תסריטים (יש גם תסריטים לרקע), ואומרים ש'הדמות מבצעת את התסריטים'. ההגדרה הזו אינה מדויקת כי דמות אינה עצם המבצע פעולה באופן בו אנו מגדירים בפרדיגמה מונחת עצמים. בהיבט המעשי ניתן להציג בפני התלמידים את הקשר בין 'דמות' ל'תסריטים לדמות', ולהראות שכאשר מוחקים דמות נמחקים גם התסריטים המשויכים לדמות (זהירות! המחיקה מתבצעת ללא אזהרה).

תלבושת

כל דמות בסביבת סקרטץ' מיוצגת בייצוג גרפי – לכל דמות ייצוג גרפי אחד ויותר, כלומר לדמות יתכנו מספר ייצוגים גרפיים. כל מראה גרפי של דמות נקרא **'תלבושת'**, ניתן להוסיף תלבושות לדמות, לערוך ולשנות תלבושות. בקבוצת ההוראות 'מראה' בסביבת סקרטץ' קיימת הוראה לביצוע פעולות שונות על התלבושות של הדמות.

עולם הבעיה

בהתאם להגדרת מושג שמופיע בהקשרים ובתחומים שונים כוונתנו במושג 'עולם הבעיה' היא: כל המאפיינים של הבעיה/מטלה/פרויקט שאנו רוצים לפתור או לפתח; אלו דמויות קשורות לפרויקט; אלו מאפיינים נדרשים לדמויות; אלו פעולות נדרש שהדמויות הללו יבצעו ועוד. את מכלול המאפיינים של המטלה אנו מכנים 'עולם הבעיה'.

מצב

בהתאם להגדרת מושג שמופיע בהקשרים ובתחומים שונים כוונתנו בתכנית הלימודים במושג 'מצב' היא ערכים מסוימים של דמות; מיקום הדמות על הזירה ועוד מצב מסוים של ההדמיה/משחק. כאשר אנחנו מגדירים מצבים ישנם תהליכים המעבירים את הדמות ממצב למצב או שהם מעבירים ממצב למצב את ההדמיה/משחק/יישום.

מימוש - כתיבת קוד

התהליך של העברת אלגוריתם לתכנית/ תסריט מחשב בסביבה בה אנו עובדים.

דרכי הערכה

1. כתיבת אלגוריתם למטלה
2. משימת הדמיה של תהליך ; משימת פתרון בעיה מתמטית-גיאומטרית – המשימה בזוגות, הערכה מעצבת (לאורך כל התהליך).

מטרות ביצועיות

1. התלמיד ידע להוסיף דמות מאוסף הדמויות בסביבת הפיתוח או מקובץ חיצוני.
2. התלמיד ידע ליצור ולעצב תלבושות נוספות לכל דמות לפי דרישת המטלה.
3. התלמיד ייתן שמות משמעותיים לדמויות, לתלבושות ולמשתנים.
4. התלמיד יבצע תהליך של העברת עולם הבעיה לייצוג נכון למחשב שעונה על דרישות המטלה.
5. התלמיד יכיר את הייצוג של 0;1 כייצוג של שני מצבים.
6. התלמיד יכיר ויעשה שימוש בהוראות.



מטרות ביצועיות –דוגמאות

1. התלמיד ידע להוסיף דמות מאוסף הדמויות בסביבת הפיתוח או מקובץ חיצוני.
2. התלמיד ידע ליצור ולעצב תלבושות נוספות לכל דמות לפי דרישת המטלה .
3. התלמיד ייתן שמות משמעותיים לדמויות, לתלבושות ולמשתנים.

הדמיה של מעגל חשמלי:

הדמויות: נורה חשמלית, חוט חשמל, מתג חשמלי, סוללה חשמלית

4. התלמיד יבצע תהליך של העברת עולם הבעיה לייצוג נכון למחשב שעונה על דרישות המטלה.

- נורה חשמלית : שני מצבים : דולקת / לא דולקת
- הגדרת משתנה לתיאור שני המצבים
- מתג חשמלי : שני מצבים : סוגר מעגל חשמלי / פותח מעגל חשמלי הגדרת משתנה לתיאור שני המצבים
- התלמיד יכיר את הייצוג של 0;1 כייצוג של שני מצבים

5. התלמיד יכיר ויעשה שימוש בהוראות :



נושא מרכזי	נושאי משנה
1. הגדרה וייצוג שחקנים	א. תיאור עולם הבעיה
	ב. בחירת השחקנים
2. הגדרת מצבים	א. הגדרה עבור כל שחקן: מצבים, פעולות, משתנים
	ב. כתיבה אלגוריתמית של התהליכים והקשר בין השחקנים
3. יישום מלא	א. כתיבת הקוד
	ב. 'הרצה' חוזרת של ההדמיה, ניתוח התוצאות, חיבור בין התוצאות לעולם הבעיה המדעי-טכנולוגי

נושאי משנה - דוגמה ופירוט

נושא מרכזי **1. הגדרה וייצוג שחקנים**

א. תיאור עולם הבעיה

- הדמיה של מעגל חשמלי
- המרכיבים: נורה חשמלית, חוט חשמל, מתג חשמלי, סוללה חשמלית

ב. בחירת השחקנים

נושא מרכזי **2. הגדרת מצבים**

א. הגדרה עבור כל שחקן: מצבים, פעולות, משתנים

- נורה חשמלית – שני מצבים: דולקת/לא דולקת
- הגדרת משתנה לתיאור שני המצבים
- מתג חשמלי – שני מצבים: סוגר מעגל חשמלי/ פותח מעגל חשמלי .
- הגדרת משתנה לתיאור שני המצבים

ב. כתיבה אלגוריתמית של התהליכים והקשר בין השחקנים

- שינוי מצב המתג באמצעות לחיצה עם העכבר על דמות המתג
- הצגת דמות מתג במצב פתוח – משתנה מתג = 1
- הצגת מצב הנורה החשמלית
- מתג סגור, נורה חשמלית במצב דולק – משתנה מייצג מצב נורה = 1
- מתג פתוח, נורה חשמלית במצב כבוי – משתנה מייצג נורה = 0
- עבור חוטי החשמל, מומלץ לבחור שני צבעים לייצוג שני מצבים:

  זרם חשמלי עובר / זרם חשמלי לא עובר.

א. כתיבת הקוד

עבור סביבת סקרטץ' יש לנו את המרכיבים של:

- יצירת הדמויות ולכל דמות שתי תלבושות
- הגדרת המשתנים
- הצבת הדמויות במיקום הרצוי ליצירת מעגל חשמלי
- כתיבת הקוד המדמה את השינוי במערכת לפי מצב המתג

ב. 'הרצה' חוזרת של ההדמיה, ניתוח התוצאות, חיבור בין התוצאות לעולם הבעיה המדעי-טכנולוגי

הפעלת ההדמיה שינוי מצב מתג, מעקב אחר השינוי בתצוגה, השינוי בערכים של המשתנים כאשר הנורה דולקת או כאשר הנורה אינה דולקת

כתיבת סיכום קצר התלמיד יסכם מה למדנו בממד של תכנון וכתיבת קוד. מה למדנו בממד המדעי-טכנולוגי

תיאור ההדמיה

תסריט הדמות 'נורה'

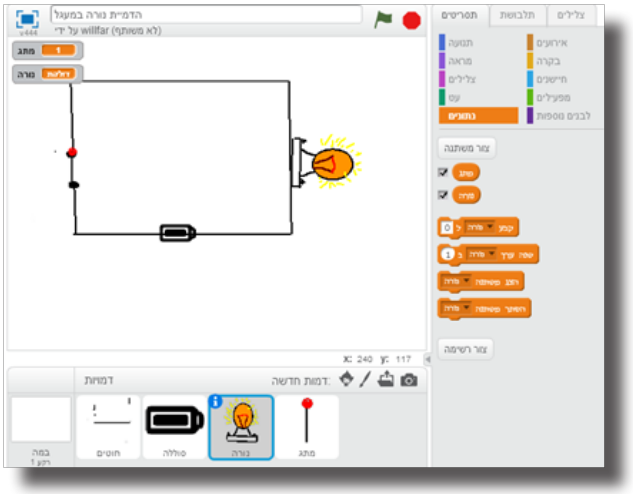


תסריט הדמות 'מתג'



תסריט הדמות 'סוללה'





נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
תיאור עולם הבעיה בחירת השחקנים	הפרק הזה בתכנית מציג כתיבת יישום מעשי עם רעיון - סיפור. יש חשיבות גדולה לעבודת של תלמידים בצוותים (2-3) חשוב לתאר את המטלה ומה מצופה מהתלמידים ולעשות זאת בדף משימה מיוחד	✓	✓	יצירת הדמויות: בכלי הגרפי בסביבת העבודה; או יבוא מרשת האינטרנט עם עיבוד גרפי בתוכנת הצייר; או הכנה וסיוע של המורה המלמד **1 שעות**
הגדרה עבור כל שחקן: מצבים, פעולות, משתנים	יש לכתוב עם התלמידים מה נדרש מכל שחקן בעולם הבעיה, אלו משתנים אנו מגדירים עבור שחקן, אלו מצבים יש לכל שחקן וכיצד הם משתנים בתהליך	✓	✓	הגדרת המצבים, יצירת המשתנים, קביעת הערכים של המשתנים בכל מצב **1/2 שעות**
כתיבת האלגוריתמים של התהליכים והקשר לשחקנים	תיאור מילולי של האלגוריתמים, אופן כתיבת האלגוריתמים בשלב זה של התכנית אינו מוגבל, יש לנסות להוביל לכתיבה מדויקת יותר ופחות 'סיפורית'	✓	✓	כתיבת האלגוריתמים על מצב המתג, מצבי הנורה ביחס למצב המתג, ציור המצבים בצורת תרשים שתבחר על ידי המורה **1 שעות**
כתיבת קוד	חשוב ליצור ולכוון את דרך היישום מתוך המסמכים שכל תלמיד/צוות תלמידים כתב. המיומנויות של שיתוף פעולה ותהליך תכנון מרמה מופשטת לקוד, הן מטרה מרכזית של תכנית הלימודים		✓	כתיבת קוד **1 שעות**
'הרצה', ניתוח תוצאות ההדמיה, חיבור בין ההדמיה לעולם הבעיה המדעי-טכנולוגי		✓	✓	הרצה וניתוח תוצאות **1/2 שעות**

פרק ז' הגדרת משתנים - הרחבה: משתנה לדמות, משתנה לכל הדמויות, משתנה ענן

תחומי תוכן: חישוביות, אלגוריתמיקה ומיומנויות המאה 21, כתיבת קוד

מושגים

- ✓ משתנה לדמות
- ✓ משתנה לכל הדמויות
- ✓ משתנה ענן

משתנה לדמות

משתנה מוכר שמוגדר עבור דמות מסוימת וניתן לשנות את ערכו רק בתסריטי הדמות לה שייך המשתנה. במדעי המחשב המושג חל על משתנה המוכר בתחום מסוים, בפעולה מסוימת – משתנה מקומי (לוקאלי).

משתנה לכל הדמויות

משתנה המוכר לכל הדמויות בפרויקט; כל הדמויות רשאיות להשתמש ולבצע פעולות על המשתנה. במדעי המחשב המושג חל על משתנה המוכר בכל התכנית – משתנה כללי (גלובלי)

משתנה ענן

משתנה שנשמר בענן. המשמעות היא שערכו נשמר גם לאחר סגירת היישום, וכאשר מפעילים את היישום מחדש ערכו של המשתנה מאוחזר מהענן . לדוגמה אפשר לשמור במשתני ענן את שמו והישגו הטוב ביותר של התלמיד במשחק מסוים, יחד עם ההישג הגבוה ביותר באותו משחק מחשב. ערך זה יעודכן בכל הפעלת המשחק.

הערה: ניתן להשתמש במשתנה ענן רק אם נכנסת לסביבת סקרטץ' עם שם משתמש וסיסמה

דרכי הערכה

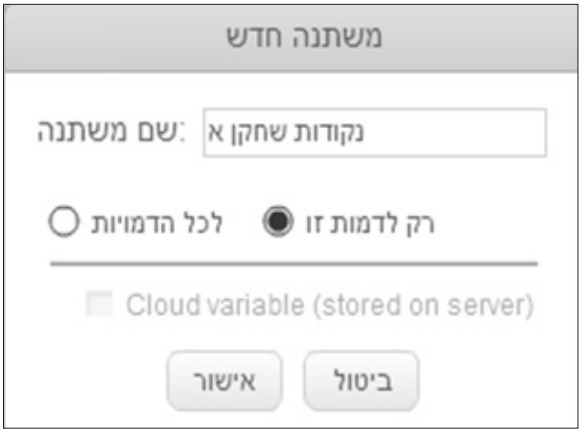
1. משימות עיצוב והגדרת משתנים: לדמויות מסוימות; לכל הדמויות.
2. מעקב אחר השתנות ערכי משתנים. מבט על משתנה לדמות ממסך התסריטים.
3. עיצוב ויישום פרויקטים העושים שימוש במשתנה לדמות; משתנה לכל הדמויות; משתנה ענן.

מטרות ביצועיות

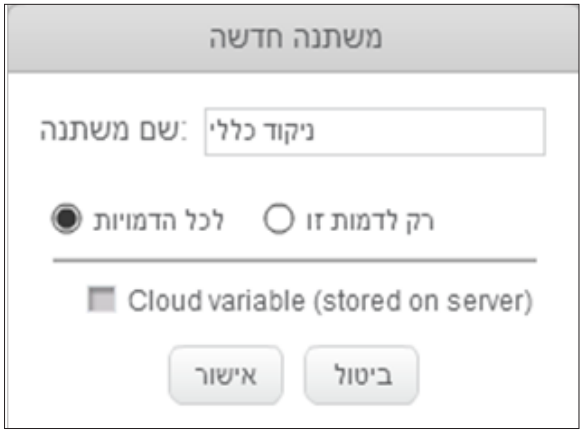
1. התלמיד ידע ליצור משתנה לדמות ומשתנה כללי – משתנה לכל הדמויות.
2. התלמיד יגדיר משתני דמות ומשתנים לכל הדמויות. השילוב בין סוגי המשתנים והוראות 'הסתר משתנה' ו'הצג משתנה' לרעיון של כימוס ושיתוף של משתנים.
3. התלמיד יציע רעיונות לשימוש במשתנה ענן; התלמיד יכתוב יישום פשוט עם משתנה ענן.

הגדרת משתנים-דוגמאות

הגדרת משתנה לדמות:



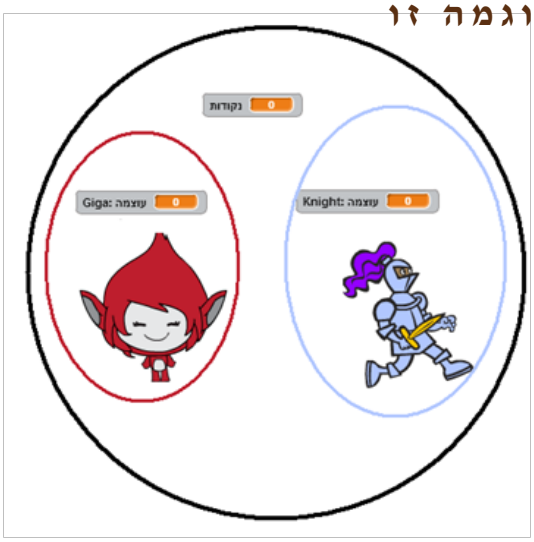
הגדרת משתנה לכל הדמויות



הסבר: משתנה לדמות; משתנה לכל הדמויות

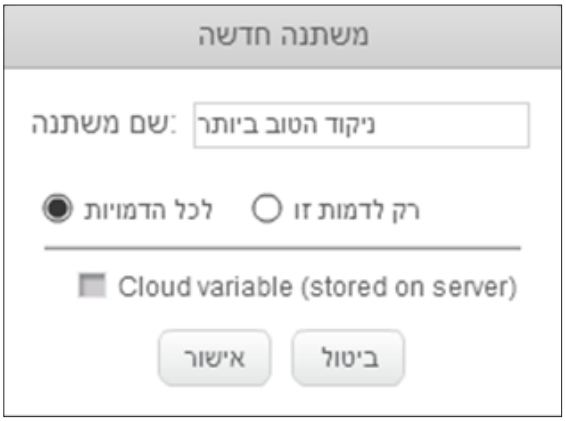
דוגמה: שתי דמויות, לכל דמות משתנה מקומי ולשתי הדמויות משתנה גלובלי: לדמויות של Giga, ו- Knight יש משתנה מקומי, כלומר לא משותף, בשם 'עוצמה'. לשתי הדמויות יש משתנה משותף 'נקודות'.

התיאור הגרפי של דוגמה זו



לכל דמות יש משתנה מקומי משלה בו היא משתמשת ומשנה את ערכו. זה אפשרי רק בתסריטים השייכים לדמות. לשתי הדמויות יש משתנה כללי ששתי הדמויות יכולות להשתמש בו בתסריט המשותף.

הגדרת משתנה ענן (דרוש רישום באתר)



הגדרת משתנים – דוגמאות

1. צור שתי דמויות; הגדר משתנה משותף; קבע ערך למשתנה; כתוב תסריט לכל דמות – בתסריט יכללו חישובים עם ערך המשתנה וההוראה **'אמור'**.

2. צור שתי דמויות; הגדר לכל דמות משתנה; הגדר משתנה אחד משותף; כתוב תסריטים המשתמשים במשתנים של הדמויות ובמשתנה המשותף.

3. צור תסריט בו נוצר משתנה ענן; קבע את ערכו; הצג את ערכו; סגור את הקובץ; פתח את הקובץ; שנה את ערכו של משתנה הענן – מה קבלת?

דוגמה א'

משתנה ענן – הרצה ראשונה של היישום



משתנה ענן הרצה שנייה של היישום



דוגמה ב' - המרדף אחר הפיל והשועל

דמות האריה ביישום זה מנסה לתפוס פיל או שועל

לדמות השועל יש משתנה – 'עוצמת שועל'

לדמות הפיל יש משתנה – 'עוצמת הפיל'

תיאור הפרויקט

הפרויקט מתחיל עם לחיצה על מקש ארוך – שני משתני העוצמה מקבלים את הערך 3 ומוצגים על הבמה. עתה האריה מנסה לתפוס את אחת הדמויות בקפיצה לכיוון ולמרחק אקראיים, בכל קפיצה יש ללחוץ על דגל ירוק, להציב את האריה במיקום התחלתי, וללחוץ על דמות האריה כדי שיבצע את הקפיצה.

אם האריה נוגע באחת מהדמויות (פיל, שועל), ערך העוצמה של הדמות בה האריה נגע קטן ב-1 .

כאשר עוצמה של אחת מהדמויות פיל או שועל שווה 0, הדמות אומרת "אני עייפה מאוד"

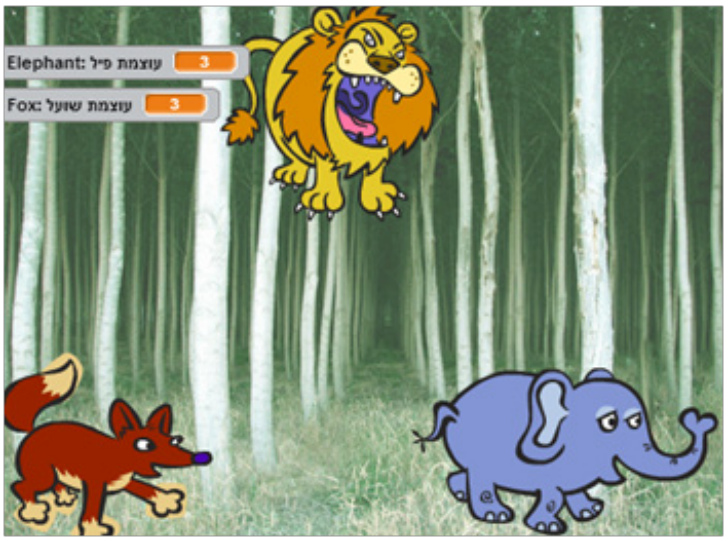
אתגרים להוספה:

1. הוספת משתנה ענן לאריה שיעקוב כמה קפיצות הוא ביצע עד שאחת מהדמויות מגיעה לעוצמה 0 וישמור את השיא (המספר הקטן ביותר) .
2. החלפת הדמות של השועל בדמות אחרת עם אפשרות להחלפת תלבושות. שנו את התסריטים בהתאם.
3. הוספה לתסריטי דמויות השועל והפיל: במצב בו העוצמה – 0 הדמות קטנה ומשמיעה קול.

טבלת דמויות ותסריטים

דמות/ תסריט	כאשר לוחצים על	כאשר נלחץ מקש רווח	כאשר לוחצים על דמות זו	כאשר מתקבל מסר תפסתי את השועל	כאשר מתקבל מסר תפסתי את הפיל
 Lion	V		V		
 Fox		V		V	
 Elephant		V			V

הבמה :



פירוט התסריטים לפי דמויות

תסריטי האריה





תסריטי השועל



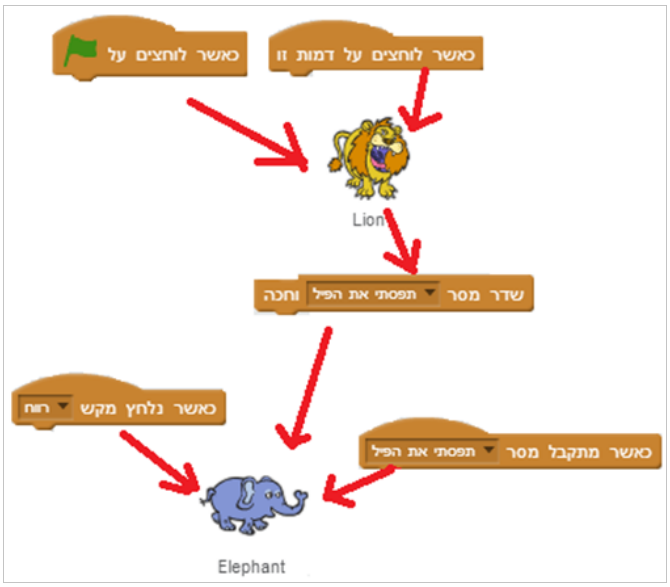


תסריטי הפיל





תרשים תזמון תסריטים – פיל נתפס על ידי האריה



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
משתנה לכל הדמויות	חשיבות בחירת שמות. אתחול ערכי משתנים, עדכון ערך משתנה, משתנה מחרוזת	✓	✓	שימוש בפעולות הגדרת משתנה. שימוש בפעולות אתחול ועדכון ערכים במשתנים. מעקב אחר השתנות הערכים, יישום אלגוריתמים חישוביים, והצגת ערכי תוצאות. **1 שעות**
משתנה לדמות	רעיון הכימוס, באמצעות המחשת הרעיון של גישה למשתנה של דמות מסוימת, ואי גישה למשתנה בתסריטים של דמות אחרת	✓		המחשה באמצעות פעילויות, הפעלת התלמידים להבנת ההפרדה בין משתנים לדמויות. משתנה לכל הדמויות יוצג על ידיי המורה לעיני כל התלמידים משתנים לדמויות יהיו נגישים לתלמידים מסוימים **1 שעות**
שילוב משתנים לדמויות, משתנים לכל הדמויות	הסתרת ערך משתנה נועד למנוע מהמשתמש לראות את הערך או שהמטרה היא לא להציג משתנה זמני	✓	✓	יצירת יישומים עם 2-3 דמויות, תרגול יצירת משתנה לדמות, משתנה לכל הדמויות. תרגול הסתר משתנה, הצג משתנה. **1 שעות**
משתנה ענן	יצירת משתנה ענן, רעיון האחסון בענן, רעיון הגישה והעדכון של משתנה ענן	✓	✓	הדגמה של יישום פשוט המעדכן משתנה ענן, סגירת היישום ופתיחתו, והצגת השינוי בערך משתנה ענן מתן מטלה בסיסית לשימוש במשתנה ענן **1 שעות**

פרק ח' אופרטורים לוגים, קינון משפט תנאי

תחומי תוכן: חישוב ביטויים לוגים, אופרטורים לוגיים, טבלאות אמת של קמטים לוגים, סדר קדימויות בביטויים לוגים, שילוב ביטויים לוגיים מורכבים בתסריטים, קינון משפטי תנאי, הקבלה בין קינון משפטי תנאי לשימוש באופרטורים לוגים.



האופרטור AND

וגם

אופרטור זה משמש לשילוב שני ביטוים לוגיים. התוצאה הסופית של חישוב הביטוי הלוגי מוגדרת באופן הבא עבור שני תנאים שנקרא להם תנאי א', תנאי ב'.

טבלת אמת AND

תנאי א'	תנאי ב'	תנאי א' AND תנאי ב'
שקר	שקר	שקר
שקר	אמת	שקר
אמת	שקר	שקר
אמת	אמת	אמת

האופרטור OR

או

אופרטור זה משמש לשילוב שני ביטוים לוגיים. התוצאה הסופית של חישוב הביטוי הלוגי מוגדרת באופן הבא עבור שני תנאים שנקרא להם תנאי א', תנאי ב'.

תנאי א'	תנאי ב'	תנאי א' OR תנאי ב'
שקר	שקר	שקר
שקר	אמת	אמת
אמת	שקר	אמת
אמת	אמת	אמת

 האופרטור לא - NOT

אופרטור השלילה. אם ערך הביטוי הלוגי **'אמת'**, התוצאה של הפעלת האופרטור היא **'שקר'**, אם ערך הביטוי הלוגי **'שקר'** התוצאה של הפעלת האופרטור היא **'אמת'**

תנאי	NOT
אמת	שקר
שקר	אמת

קינון משפטי תנאי

מבנה תכנותי בו אנו משלבים משפט ביצוע שהוא משפט תנאי בתוך משפט תנאי. כך אנו יוצרים קינון של משפטי תנאי. חשוב לבחון אלו מצבים אנו מכסים במבני קינון שונים.

נושאי משנה

א. ריענון משפט **'אם ואם לא'**

ב. האופרטור AND

ג. יישום משפט תנאי עם האופרטור AND

ד. יישום תסריט עם אופרטור AND ויישום אותו תסריט עם קינון משפטי תנאי

ה. האופרטור OR

ו. יישום תסריט עם האופרטור OR ויישום אותו תסריט ללא האופרטור OR

דרכי הערכה

1. תרגול בניסוח וכתיבה מילולית של טענות לוגיות.
2. תרגול ביישום טענות לוגיות מילוליות באמצעות קטעי תכנות בסיסיים.
3. ניסוח סיפור עם התניות ויישומו בתסריט עם אופרטורים לוגיים וללא אופרטורים לוגיים.

מטרות ביצועיות

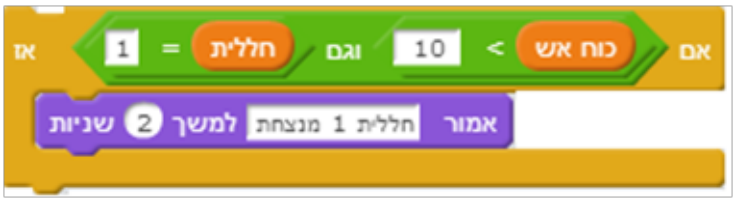
התלמיד יכיר מבנה של ביטויים לוגיים כולל האופרטורים הלוגיים: NOT , AND , OR



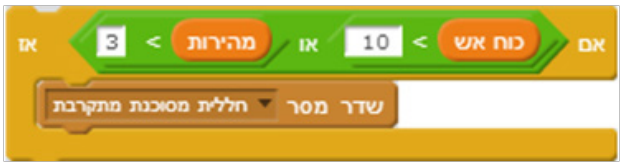
מטרות ביצועיות –דוגמאות

1. התלמיד יכיר מבנה של ביטויים לוגיים כולל האופרטורים: NOT , OR , AND

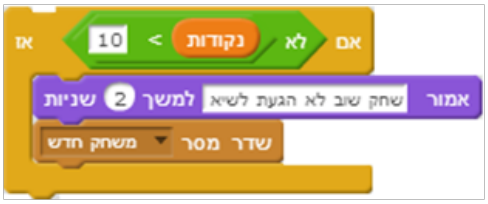
דוגמה: שימוש ב: and - וגם



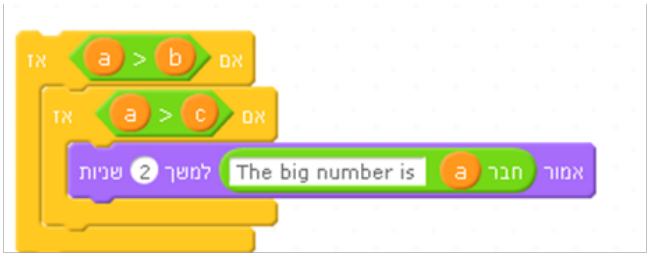
דוגמה: שימוש ב: OR - או



דוגמה: שימוש ב: NOT - לא



2. התלמיד יפתח ויישם תסריטים עם מבנה תנאי מקונן





דוגמאות – הרחבה

א. זיהוי מילים

דוגמה זו כוללת משחק פשוט של זיהוי שתי מילים באנגלית, הצלחה במשחק תתרחש רק במקרה בו המשחק זיהה נכון את שתי המילים. במשחק אנו משתמשים בשני משתנים ובמשפט '**אם ואם לא**'.

אפשרויות הרחבה ושינוי עומדות בפני התלמידים. ניתן לשנות דמויות ומילים ובכך להרחיב את המשחק למילים נוספות.

הבמה של המשחק



תיאור המשחק

על המשחק להעביר את הדמויות של הסוס ושל העץ אל הכפתורים המתאימים עם המילים הנכונות באנגלית.
אם דמות הסוס וגם דמות העץ הועברו לכפתורים הנכונים המשחק הצליח במשימה.

התסריטים של Jaime





התסריטים של Horse1





התסריטים של Tree1





ב. החיפושית הקופצת

דוגמה זו מציגה מצבים שונים. חיפושית קופצת על משטח משובץ בריבועים אדומים וכחולים. על המשטח יש חיפושית נוספת ובננה.

החיפושית הקופצת תנחת בכל קפיצה:

על משבצת אדומה

או

על משבצת כחולה

או

על בננה

או

על החיפושית השנייה

או

על חיפושית וגם על משבצת אדומה

או

על בננה וגם על משבצת כחולה...

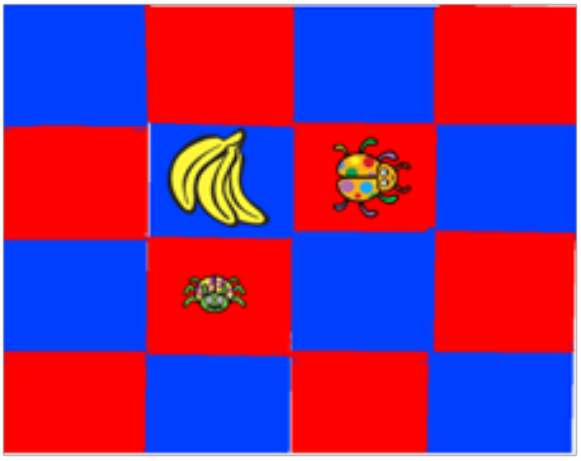
ועוד אפשרויות שמומלץ לבחון בהרצת היישום ולענות על השאלה: האם נבדקו כל המצבים של החיפושית הקופצת ביחס לרקע ולדמויות האחרות?

הרצת היישום פשוטה, כל הקשה על מקש רווח מקפיצה את החיפושית למיקום הבא.

הדוגמה מאפשרת מניפולציות שונות עם התנאים הלוגיים והאופרטור AND והאופרטור OR. אפשר ומומלץ לשנות את המשטח המשובץ: לשנות צבעים, להחליף דמויות על המשטח, להוסיף דמויות על המשטח.

הערה: המשטח נוצר כתמונה של הרקע

הבמה של המשחק



התסריטים של החיפושית הקופצת



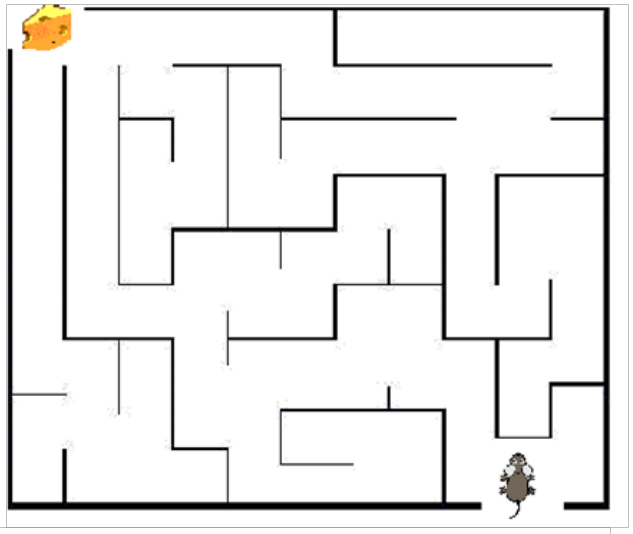




ג. העכבר במבוך

מבוכים הם משחקים אהובים על ילדים בכל הגילאים. כאן תוצג דוגמה בסיסית למבוך, עם אפשרויות הרחבה . בתסריטי היישום השתמשנו במשפט תנאי '**אם**', היישום ללא משתנים ועם תנאים לוגיים בסיסיים.

הבמה (הבמה וציור הגבינה הם רקע שיובא לצייר ונערך בצייר ונשמר כקובץ שנבחר כרקע של הבמה)



תסריטי העכבר



אפשרויות להרחבת המשחק:

המשחק הוא גרסה מצומצמת לתנועה במבוך. אפשר להשתמש בפרויקט זה כבסיס לפיתוח המשך של התלמידים. ניתן להוסיף:

- משתנה ניקוד שיעודכן בכל שלב של התקדמות.
- משתנה ענן לשמירת שיא בענן.
- הוספת 'ממתקים' שהעכבר יאכל בדרך ויגדיל את הנקודות. ניתן לעשות זאת על ידי הוספת עצמים שונים במיקומים שונים בתוך המבוך, כל עצם 'שנאכל' נעלם.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21		ציוני דרך
ריענון **'אם ואם לא'**	הסבר באמצעות דוגמאות פשוטות כמו $0<x$ ואם לא, לדבר על מה קורה עם $0=x$, לעיתים תלמידים לא מבחינים. להראות דוגמאות עם שני תנאים פשוטים, ואותה דוגמה עם משפט תנאי מורכב	✓		תלמיד יכתוב קטעי תסריטים עם המשפט **'אם ואם לא'** ואותו קטע עם אותה מטרה עם שני משפטי **'אם'** פשוטים. **1/2 שעות**
האופרטור AND	מתן דוגמאות מהחיים ללא תכנות לתנאים עם AND הצגת טבלת האמת של AND להשתמש בייצוג בסיסי ואינטואיטיבי	✓		התלמיד ינסח בעברית משפט תנאי, יציין את התנאים ואת המילה **'וגם'** בין התנאים. **1/2 שעות**
יישום משפט תנאי עם האופרטור AND	דוגמאות בסיסיות עם שני ביטויי תנאי: אפשר רק עם קבועים, אפשר עם משתנים, למתקדמים אפשר עם שילוב ביטויים מתמטיים	✓	✓	התלמיד יכתוב מספר קטעי קוד שונים עם משפט תנאי בשילוב האופרטור AND התלמיד יכתוב קוד שקול ללא שימוש באופרטור AND **1/2 שעות**
האופרטור OR	מתן דוגמאות מהחיים ללא תכנות לתנאים עם OR הצגת טבלת האמת של OR להשתמש בייצוג בסיסי ואינטואיטיבי	✓		התלמיד יכתוב קטעי קוד עם האופרטור OR ויכתוב קטעים שקולים עם שני משפטי **'אם'**
יישום משפט תנאי עם האופרטור AND		✓	✓	התלמיד יכתוב מספר קטעי קוד שונים עם משפט תנאי בשילוב האופרטור AND התלמיד יכתוב קוד שקול ללא שימוש באופרטור AND **1/2 שעות**

פרק ט' כתיבה של אלגוריתם מילולי, תכנון ויישום (מרעיון לקוד)

תחומי תוכן: חישוביות, אלגוריתמיקה, ומיומנויות המאה 21, כתיבת קוד

מושגים

- ✓ אלגוריתם
- ✓ פסיאודו קוד
- ✓ משפט קלט,
- ✓ משפט תנאי מצומצם
- ✓ משפט תנאי מורחב
- ✓ לולאה מבוקרת
- ✓ מודל תכנון מעלה-מטה
- ✓ מודל תכנון מטה-מעלה

אלגוריתם

אלגוריתם הוא דרך שיטתית (כלומר כזו שצעדיה מוגדרים היטב) לביצוע של משימה מסוימת במספר סופי של צעדים.

פסיאודו קוד – קוד מדומה

כתיבת הוראות לתכנית מחשב שאינן כתובות בשפת תכנות מסוימת אלא בניסוח כללי בעברית (תלמידים באנגליה יכתבו באנגלית, בצרפת בצרפתית). כאשר אנחנו מעבירים את התכנית – אלגוריתם למחשב, אנחנו מבצעים זאת באמצעות שפת התכנות בה אנו משתמשים.

משפט קלט

הוראת קלט שמטרתה לקבל קלט מהמשתמש בתכנית המחשב. בדרך כלל ברירת המחדל היא קלט מהמקלדת.

משפט תנאי מצומצם

משפט תנאי ללא האפשרות 'אחרת'.

משפט תנאי מורחב

משפט הכולל את ההוראות לשתי תוצאות התנאי : אם הוא מתקיים ; ואם לא מתקיים.

לולאה מבוקרת

לולאה שמתבצעת מספר מוגדר מראש של פעמים

מודל תכנון מעלה-מטה (הרחבה בהמשך)

במודל 'מעלה-מטה' מנוסחת תחילה סקירת מערכת בקווים כלליים ומבלי להיכנס לפרטים של אף אחד מחלקיה. לאחר מכן מנוסח כל חלק של המערכת לפרטיו.

מודל תכנון מטה-מעלה (הרחבה בהמשך)

במודל 'מטה-מעלה' מוגדרים תחילה החלקים הפרטיים של המערכת לפרטיהם, ואז מצורפים החלקים יחדיו לרכיבים גדולים יותר.

מבוא

נושא אלגוריתמיקה מילולית שמופיע בפרק זה כולל הכללת נושאים בלולאות שלא נלמדו. ניתן ללמד חלק מנושאי הפרק הזה גם בשלבים יותר מוקדמים, וניתן ללמד את פסיאודו קוד עם לולאות לקראת סוף תכנית הלימודים – בסוף שנת הלימודים של כיתה ד׳ או בשנת הלימודים של כיתה ה׳. חשוב לרכז את הנושא בפרק אחד ולא לפרוש את הנושא בפרקים שונים בתכנית הלימודים. הלמידה והיישום של הנושאים צריכה להיות מתואמת וצריכה להיעשות במקביל להתקדמות בלמידת הוראות ומבני בקרה בשפת סקרטץ׳.

אלגוריתמיקה מילולית / פרדיגמה פרוצדורלית בסביבת סקרטץ׳

בפרקי התכנית למדנו ויישמנו את הרעיונות והיכולות הרבות של שפת סקרטץ׳ כשפת תכנות מונחה אירועים. השתמשנו בשני פיתוחים שהוצגו בתכנית ושנועדו לסיוע בתכנון כתיבת יישום בסביבה, ובמעקב אחר ביצוע של רכיבים ביישום בסביבה זו.

כתיבת אלגוריתמיים מילוליים לפני כתיבת קוד מהווה תהליך שתומך בעבודת עיצוב וניתוח של מטלה לקראת כתיבת קוד, אולם חשוב לשייך את הכלי הזה ואת הטכניקה הזו לכלים שצמחו מתוך סביבות של שפות פרוצדורליות. כדאי לזכור שהיתרון הגדול של כתיבת אלגוריתם בפסיאודו קוד הוא באי תלות בשפת התכנות. כתיבת תכנית פסיאודו קוד אינה תלויה בשפה בה אנו מממשים את הקוד ; משמשת כרכיב חשוב בתהליכי תכנון ועיצוב האלגוריתם לקראת מעבר לקוד – זה חלק מהיעד שהוגדר לתכנית הלימודים והוא **כתיבת קוד בתהליך מובנה ופשוט**.

אלגוריתם מילולי הוא כלי בסיסי בתהליכי התכנון שמשתלב עם הכלים שהוצגו בתכנית הלימודים: טבלת דמויות ותסריטים, ותרשים תזמון אירועים. שלושת הכלים הללו תומכים בכל תהליכי התכנון וניתן להשתמש בהם באחד משני המודלים לפיתוח תוכנה **מעלה-מטה: ומטה-מעלה.**

במודל **מעלה-מטה** מנוסחת תחילה סקירת מערכת בקווים כלליים מבלי להיכנס לפרטים של שום חלק ממנה. לאחר מכן מנוסח כל חלק של המערכת לפרטיו, ואז כל פרט חדש עשוי להיות מנוסח שוב תוך כדיי שמגדירים אותו לפרטים נוספים עד שכל המפרט מפורט דיו כדי לתקף את המודל.

בניגוד לזאת, במודל **מטה-מעלה** מוגדרים תחילה החלקים הפרטיים של המערכת לפרטיהם, ואז מצורפים החלקים יחדיו לרכיבים גדולים יותר שבתורם מצורפים גם הם עד להשגתה של המערכת השלמה.

מטה-מעלה - שילוב כלים בסביבת סקרטץ׳

המודל הזה יתאים למימוש בתכנית הלימודים עבור בעיות פשוטות עם מספר מצומצם של תסריטים (1-3), ומספר קטן של דמויות (1-3). עבור מטלות מצומצמות כגון חישוב מתמטי, ציור על המסך ואנימציה פשוטה – בנייה של היישום עבור כל תסריט בנפרד תתאים יותר. במודל הזה נשתמש יותר בכלי של אלגוריתם מילולי כשהשילוב של אלגוריתם מילולי יחד עם גישת מטה-מעלה, יאפשר את פישוט התהליך ואת המעבר לקוד בצורה מובנת וקצרה.

מעלה-מטה - שילוב כלים בסביבת סקרטץ׳

המודל הזה יתאים למימוש בתכנית הלימודים עבור בעיות יותר מורכבות ועבור פרויקטים מסוגים שונים הכוללים יותר מ-3 דמויות ויותר מ-2 תסריטים לדמות. במודל זה נעשה שימוש בשילוב הכלים הבאים שהוצגו בתכנית: טבלת אירועים ; תרשים תזמון אירועים ; אלגוריתם מילולי. שילוב שלושת הכלים נעשה כך שבשלב הראשון של הפשטת ועיצוב היישום אנחנו משתמשים בכלים : טבלת אירועים ותרשים תזמון אירועים, ובשלב השני אנו משתמשים באלגוריתם המילולי כדי לפרט את המרכיבים הבסיסיים של כל מרכיב בפתרון (תסריט).

הגישה המשולבת

מומלץ לאמץ גישה משולבת – מתחילים את שלב התכנון במודל מעלה-מטה; מגדירים את המשימה הכללית; מפרקים את המשימה הכללית לתתי משימות. בשלב התכנוני בו כמעט כל המרכיבים הנדרשים ידועים, נוקטים בגישת מטה-מעלה: כותבים כל מרכיב בצורה מפורטת, ויוצרים את היישום בצורה מפורטת.

מיומנויות המאה 21 ושילוב מודלים של פיתוח

ככל שמורכבות היישום גדלה (ברמת התלמידים) ואנו נדרשים למודלי תכנון, אנחנו נדרשים יותר גם לתהליכי יישום מיומנויות המאה 21. בעיה/מטלה מורכבות יותר, יותר דמויות, יותר תסריטים, יותר משתנים וכדומה – משמעותם יותר תלמידים המשתתפים בכתיבת היישום. לכן אנחנו נדרשים להתמודד עם סביבה בה התלמידים צריכים לרכוש וליישם את מיומנויות המאה 21, כולל:

- יצירתיות וחדשנות
- חשיבה ביקורתית ופתרון בעיות
- תקשורת ושיתוף פעולה
- אוריינות מידע, אוריינות מדיה (שימוש במדיות שונות) –ICT ועוד

נושאי משנה:

- כתיבת אלגוריתם מילולי – הרעיון המרכזי
- כללי כתיבה מוסכמים
- מספור הוראות, מספור משנה, הזחה
- מעבר מאלגוריתם לקוד באופן מובנה

פסיאודו-קוד – מבוא

פסיאודו-קוד (פסיאודו-אלגוריתם) – תיאור מצומצם ולא רשמי לאלגוריתם של תכנית מחשב. התיאור מיועד לקריאה על ידי בני אדם ולא לקריאה על ידי מחשב.

לאחר כתיבת פסיאודו-קוד, ניתן לכתוב את הקוד האמיתי בשפות תכנות שונות (עם שינויים המתאימים לשפת התכנות). אין דרך פורמלית יחידה לכתוב פסיאודו-קוד וחשוב שהקורא יבין את הקוד המילולי ויוכל להעבירו לקוד בסביבת העבודה בה הוא עובד. חשוב לשמור על עקביות במהלך הכתיבה.

לכל הוראה יש לתת מספר, המספור מתחיל מ-1. עבור קבוצת פקודות (בתוך מבנה לולאה, מבנה תנאי), יש להשתמש במספור משני 1.1, 1.2 וכך הלאה, עד סיום קבוצת הפקודות השייכת למבנה הבקרה.

ניסוח מוסכם של פקודות באלגוריתם מילולי בתכנית הלימודים זו:

משפט השמה:

ביטוי ==> משתנה

משפט קלט:

קלוט ==> משתנה

משפט תנאי מצומצם:

'אם' (תנאי) בצע

משפט ביצוע

משפט תנאי מורחב

'אם' (תנאי) בצע

משפט ביצוע א'

אחרת

משפט ביצוע ב'

לולאה מבוקרת

'חזור (ביטוי) **פעמים'**

משפט ביצוע

לולאה מותנת

'חזור עד ש' (תנאי)

משפט ביצוע

לולאה לעולמים (לולאה אין-סופית)

בצע **'לעולמים'**

משפט ביצוע

משפט ביצוע

משפט ביצוע פשוט – רק הוראה אחת

משפט ביצוע מורכב – מספר הוראות עם מספור פנימי

דוגמה לתנאי עם משפט ביצוע פשוט ועם משפט ביצוע מורכב:

משפט ביצוע פשוט

1. קלוט ==> ציון
2. **'אם'** (ציון>90)

   2.1 **'אמור'** "כל הכבוד"

משפט ביצוע מורכב

1. קלוט ==> ציון
2. **'אם'** (ציון < 56)

   2.1 ציון +10 ==> ציון

   2.2 **'אמור'** "קבלת תוספת לציון"
3. המשך האלגוריתם.

דוגמה ללולאה עם משפט ביצוע פשוט ועם משפט ביצוע מורכב:

משפט ביצוע פשוט

1. קלט ==> צלילים
2. **'חזור** (צלילים) **פעמים'**
   2.1 **'השמע צליל'**

משפט ביצוע מורכב

1. 0 ==> סכום
2. קלט ==> שחקנים
3. **'חזור** (שחקנים) **פעמים'**
   3.1 קלוט ==> נקודות_שחקן
   3.2 סכום + נקודות_שחקן ==> סכום
4. 'הצג' (סכום)

דרכי הערכה

1. כתיבת אלגוריתם מילולי לבעיה (קטנה)
2. מעבר מאלגוריתם מילולי לקוד

מטרות ביצועיות

1. התלמיד ידע לרשום הוראות ביצוע פשוטות באלגוריתם מילולי.
2. התלמיד ידע לכתוב אלגוריתם מילולי הכולל תנאי פשוט עם משפט ביצוע פשוט.
3. התלמיד ידע לכתוב אלגוריתם מילולי עם **'אם כן ואם לא'**.
4. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים פשוטים שכתב (לכתוב קוד לאלגוריתמים מילוליים בסיסיים).
5. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה ולכתוב הוראה אחת שכלולה בלולאה (משפט ביצוע פשוט).
6. התלמיד ידע לכתוב אלגוריתם מילולי עם הוראה ומספר של הוראות הכלולות בהוראה זו.
7. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים שכוללים לולאות ומשפטי ביצוע מורכבים.
8. בהינתן בעיה, התלמיד ידע לכתוב אלגוריתם מילולי עם: הוראות תנאי, הוראות לולאה ומשפטי ביצוע במבנים אלו.
9. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים.

מטרות ביצועיות - דוגמאות

א. התלמיד ידע לכתוב אלגוריתם מילולי עם **'אם כן ואם לא'**

1. קלט ==> מספר _ א
2. קלט ==> מספר _ ב
3. **'אם'** (מספר _ א) > (מספר _ ב)
   - 3.1 **'אמור'** (מספר _ א)
   - 3.2 אחרת
   - 3.3 **'אמור'** (מספר _ ב)

ב. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים פשוטים שהוא כתב (לכתוב קוד לאלגוריתמים מילוליים בסיסיים)

האלגוריתם:

1. קלט ==> מספר _ א
2. קלט ==> מספר _ ב
3. **'אם'** (מספר _ א) > (מספר _ ב)
   - 3.1 **'אמור'** (מספר _ א)
   - 3.2 אחרת
   - 3.3 **'אמור'** (מספר _ ב)

העברה לקוד בסביבת העבודה – SCRATCH



ג. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה, והוראה אחת הכלולה בלולאה (משפט ביצוע פשוט)

1. מונה <== 0

2. חזור **'לעולמים'**

2.1 **'אמור'** ("אני בלולאה אינסופית")

ד. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה, ומספר הוראות שכלולות בהוראה זו.

1. מונה <== 0

2. **'חזור** (4) **פעמים'**

2.1 **'אמור'** ("אני בלולאה")

2.2 מונה <== מונה +1

3. **'אמור'** (מונה)

ה. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים הכוללים לולאות ומשפטי ביצוע מורכבים.

אלגוריתם מילולי:

1. עכברים <== 0

2. **'חזור עד ש'** (עכברים=10)

2.1 אם (חתול נוגע ב עכבר)

2.1.1 עכברים <== עכברים+1

2.1.2 **'אמור'** ("תפסתי את העכבר")

3. **'אמור'** ("תפסתי עשרה עכברים")

תכנית בסביבת העבודה SCRATCH



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
כתיבת אלגוריתם מילולי הרעיון המרכזי	חשוב להדגיש כי מדובר בתהליך. אנחנו לא עוברים ישירות מתיאור בעיה לכתיבת תכנית מחשב, יש לנתח את הדרישות ולנסח מילולית את הפתרון	✓		התלמיד ינסח פתרון מילולי עבור בעיה מסוימת בשפה העברית ללא כללים ברורים **1 שעות**
כללי כתיבה מוסכמים	הצגת כללי הכתיבה ומתן דגש על ניסוח ההוראות המרכזיות כפי שמופיע במבוא לפרק	✓	✓	התלמיד ידע לכתוב הוראות בשפת האלגוריתמיקה הכללית לפי ההנחיות שנתנו **1 שעות**
מספור הוראות, מספור משנה, הזחה	הצגת רעיון המספור והמספור המשני			התלמיד יקבל תיאור מילולי כללי של אלגוריתמים, יפרט את האלגוריתמים הכללים, ויעביר את האלגוריתמים לקוד **2 שעות**
מעבר מאלגוריתם לקוד באופן מובנה	שילוב תנאים מגוונים, יחסים בין ערכים, יחסים בין דמויות הדגשת השימוש בשפה ללא קשר לסביבת העבודה	✓	✓	התלמיד יבצע את חמשת השלבים בפתרון בעיה עבור מטלות ובעיות שונות א. הבנת הבעיה ב. ניתוח וחשיבה על דרכי פתרון ג. כתיבת אלגוריתם מילולי מפורט ד. העברה לקוד ה. הרצה, בדיקות ושיפורים **2 שעות**

פרק י' מבוא למבנה בקרה לולאה: לולאת 'לעולמים'

תחומי תוכן: חישוביות אלגוריתמיקה ומיומנויות המאה 21, כתיבת קוד

מושגים

✓ לולאה אינסופית - בסביבת סקרטץ' - לולאת **'לעולמים'**, מבנה בקרה המאפשר ביצוע חוזר של קבוצת פקודות ללא סיום

נושאי משנה:

א. לולאת **'לעולמים'**

ב. תכנון וניתוח מערכת – שלב לפני כתיבת אלגוריתמים וקידוד

ג. תהליכי הרצה ובדיקת נכונות

דרכי הערכה

1. כתיבת תסריטים עם לולאת **'לעולמים'**

2. שילוב לולאת **'לעולמים'** עם מבנה תנאי. שימוש בהוראות מקבוצת הפקודות –'תנועה', חיישנים, 'עט', 'מראה' – כדי ליצור הדמיות ומשחקים בסיסיים.

מטרות ביצועיות (בפרק זה דוגמאות יוצגו בסוף הפרק)

1. היכרות ויישום של לולאות נעשים באמצעות מבנה של לולאת **'לעולמים'**. מבנה זה יאפשר לנו ליצור יישומים מעניינים יותר ללא כניסה מעמיקה ומורכבת לכל נושא לולאה מבוקרת ולולאה מותנת.

לולאות, הרעיון ויישומו – הרחבה

לולאה היא מבנה בקרה, כלומר מבנה המארגן את זרימת ביצוע פקודות בתכנית מחשב/תסריט. הלולאה מאפשרת חזרה על קבוצת פקודות מספר פעמים. יש אפשרות לבצע 3 סוגי לולאה:

א. לולאה מבוקרת - אנו קובעים את מספר הפעמים שהלולאה תתבצע ואת המבנה של הלולאה:

1. **'חזור _ פעמים'**

   1.1 .....

   1.2 .....

   1.3 .....

2. .....

אנחנו רואים מבנה של חזרה המתבצע במספר הפעמים שנדרש (יש להוסיף ערך קבוע או ביטוי) בתום מספר הפעמים שהתבצע הלולאה מסתיימת.

ב. לולאה מותנת – הלולאה מתבצעת עד שמתקיים תנאי שאנחנו קובעים. הלולאה תתבצע מספר פעמים שונה בכל ביצוע וזה תלוי בקיום התנאי: יתכן שהלולאה לא תתבצע אפילו פעם אחת; שהיא תתבצע אינסוף פעמים (התנאי לעצירה לעולם לא מתבצע); או שהיא תתבצע מספר פעמים כלשהו.

1. **'חזור עד ש'**_____
   - 1.1 .....
   - 1.2 .....
   - 1.3 .....
2. .....

ג. לולאה אינסופית – לולאת 'לעולמים': הלולאה תתבצע אינסוף פעמים. בסביבת סקרטץ' ניתן לעצור את הלולאה בלחיצה על 'עצור' שמופיע בחלק העליון של הזירה. בשפות תכנות אין מבנה לולאה הנקרא **'לעולמים'**, מצב כזה עלול להיווצר בכתיבת קוד ללולאה מבוקרת – באמצעות שינוי מוטעה שמבצע המתכנת כשישנו תנאי לוגי שמתקיים תמיד בלולאה מותנת כשכותב הקוד לא הבחין שתנאי הלולאה מתקיים תמיד ויש ביצוע אינסופי. בשפת סקרטץ' ובמיוחד בתכנות משחקים ואנימציות יש שימוש יותר תדיר בלולאה אינסופית, ולכן יוצרי השפה שילבו בלוק מיוחד למצב זה – **לעולמים**. לדוגמה: כאשר אנחנו כותבים תסריטים לכמה דמויות, לעיתים קרובות דמות אחת או יותר מבצעת תהליך אינסופי מתוכנן. למשל משחק פשוט של חתול הרודף אחרי העכבר: ניתן לבנות תסריט בו העכבר 'מתרוצץ' ברחבי הזירה בלולאה אינסופית, והחתול נשלט בתנועתו באמצעות החצים. המטרה שלנו היא לתפוס את העכבר וכל נגיעה בעכבר מוסיפה נקודות למשחק כך שהמרדף לא מסתיים לעולם כמו אצל טום וג'רי.

דוגמאות לולאה אינסופית – 'לעולמים'

חתול מנסה לתפוס עכבר. העכבר מתרוצץ על הבמה. והחתול, בעזרתנו ובאמצעות ארבעת מקשי החצים, מנסה לגעת ולתפוס את העכבר. שים לב, מהירות תנועה העכבר תלויה בזמן ההמתנה בין התנועות. אלו תסריטים בסיסיים של הדמיית משחק 'חתול ועכבר'. ניתן לשפר את המשחק והמורים מוזמנים לאתגר את התלמידים.

התסריטים של החתול



התסריט של העכבר



אנחנו רואים המחשה לאופן בו מבנה הבקרה **'לעולמים'** בשילוב תכנות המונחה על ידי אירועים – EVENTS DRIVEN PROGRAMMING – מאפשרים יצירת 'משחק' מעניין ללא מאמץ. ברור שנדרש תהליך תכנות מסודר, משתנים נוספים ומורכבות גדולה יותר בכדי ליצור יישום מרשים ויותר מובנה.

דוגמאות – הרחבה

1. משחק בסיסי – Pong ליחיד

משחק זה הוא משחק גנרי. ישנן גרסאות רבות למשחק ופה תוצג גרסה בסיסית.

הבמה



דמויות ותפקידים

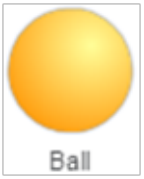
– Bar



דמות של פס/חסם נבחרה מקבוצת הדמויות בסביבת סקרטץ' ושמה המקורי PADDLE.

תפקיד הדמות לנוע אופקית בתחתית המסך משמאל לימין ומימין לשמאל. השליטה על התנועה נעשית עם מקשי חצים: חץ ימינה; חץ שמאלה

– Ball



דמות כדור נבחרה מאוסף הדמויות של סביבת סקרטץ'. תפקיד הדמות לנוע על המסך. יכולים להתקיים שני אירועים: הדמות נוגעת בקו הגבול של הבמה (סומן עם פס כחול); הדמות נוגעת בדמות – BAR

אלגוריתם מילולי לדמויות Ball



1. **'לעולמים'**
   - 1.1 **'אם נוגע ב'** מסגרת
     - 1.1.1 **'הפוך כיוון'**
   - 1.2 **'אם נוגע ב'** צבע כחול
     - 1.2.1 **'נגן צליל'**
   - 1.3 **'אם נוגע ב'** פס (Bar)
     - 1.3.1 **'הפוך כיוון'**
     - 1.3.2 **'השמע צליל'**

מימוש תסריטי הדמות Bar

1. **'אם נלחץ** חץ ימני'
   - 1.1 **'פנה לכיוון'** 90
   - 1.2 **'זוז** 5 **צעדים'**
1. **'אם נלחץ** חץ שמאלי'
   - 1.1 **'פנה לכיוון'** 90-
   - 1.2 **'זוז** 5 **צעדים'**

מימוש תסריטי הדמות



משחקי Ball, Pong בסקרטץ': קיימות גרסאות שונות ומגוונות למשחק ה- PONG בסביבת סקרטץ'. על בסיס הגרסה שהוצגה בתכנית זו אפשר להרחיב ולשפר את המשחק. חשוב כמובן לתכנן את ההרחבה: אלו דמויות נוספות נדרשות, משתנים, ועוד.

הרחבות, שיפורים ושינויים:

1. משחק של שני שחקנים: המשחק יכלול שני BAR, כאשר כל שחקן מזיז את ה- BAR שלו.
2. משתנים: משתנה שימנה את מספר הפגיעות עם ה- BAR בכדור, או משתנה שימנה את מספר ההחמצות (פגיעה במסגרת). עבור שני שחקנים יתכנו יותר משתנים.
3. רמת המשחק משתנה למשל ע"י הגדלת מהירות הכדור ובכך עולה רמת הקושי.
4. רמות משחק עם החלפת רקע, ככל שרמת המשחק עולה. מחליפים רקע
5. מספר גדול של BAR הנמצאים בקצה העליון של הבמה ללא תנועה, שנעלמים עם כל פגיעה בכדור. משחק המיועד לשחקן בודד.

חשוב! לתכנן בגישת 'מעלה-מטה' וגישת 'מטה-מעלה'. חשוב לשלב אלגוריתמים פסיאודו-קוד, במיוחד עבור תסריט מורכב. חיפוש בגוגל או בסביבת הפיתוח של סקרטץ' עם המילים SCRATCH, BALL, PONG, GAME, יניב קישורים רבים למשחקים מסוג זה.

2. משחק בסיסי – החתול תופס בלונים

משחק שעקרונותיו פשוטים והוא מהווה הרחבה למשחקי תנועה של דמות ולמרדף אחר דמויות אחרות. במשחק זה דמות החתול נקלעה למסיבת בלונים ש'משתחררים' ונעים בחלל החדר. החתול מנסה לתפוס בלונים ככל יכולתו.

הבמה



דמויות ותפקידים

הבלונים כל הבלונים כוללים את אותו קוד, תכונה שמאפשרת פשטות וחסכון בכתיבת הקוד. למרות הקוד הזהה כמעט בכל המרכיבים, אנחנו מקבלים הבדלים בהתנהגות הבלונים באמצעות שימוש בערכים אקראיים. כך כל בלון מופיע במיקום אחראי אחר ובתזמון אקראי.

תסריטי דמות הבלון



אירוע מפעיל – 

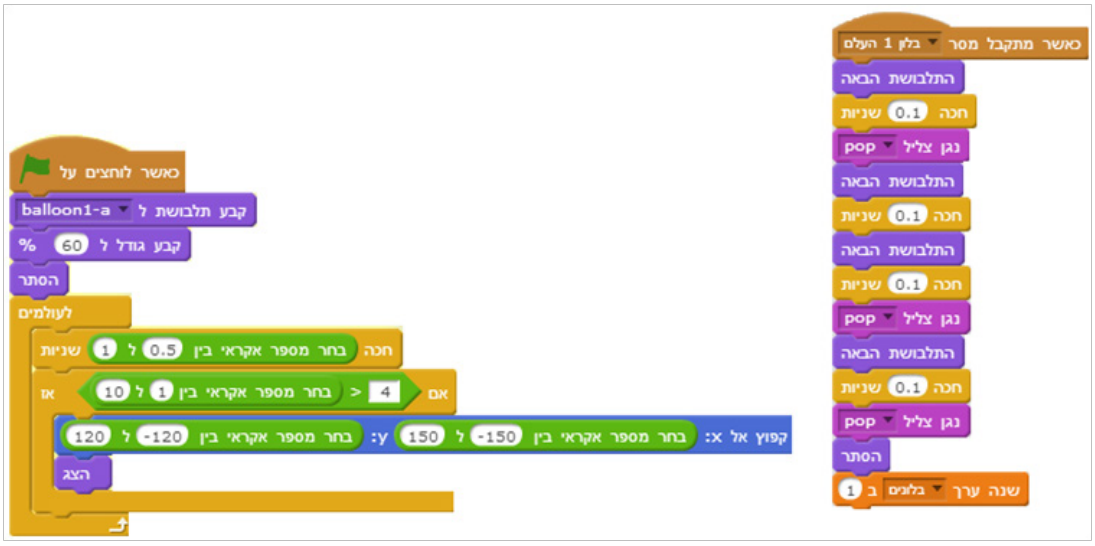
1. קביעת תלבושת התחלתית
2. קביעת גודל דמות
3. **'הסתר'** דמות
4. **'לעולמים'**
   - 4.1 **'המתן'** מספר אקראי של שניות (0.5-1)
   - 4.2 **'אם'** (ערך מספר אקראי בתחום 1-10) < 4
     - 4.2.1 **'קפוץ'** למיקום אקראי על הבמה

אירוע מפעיל – 

1. **'החלף'** לתלבושת הבאה
2. **'נגן צליל'**
3. **'המתן'** 0.1 **שניות'**
4. **'העלם'**
5. בלונים +1 (הוספת 1 למשתנה מספר הבלונים שתפס החתול) ==> בלונים

קטע ההוראות 1-3 יחזור מספר פעמים על ידי כתיבה חוזרת (בהמשך תכנית הלימודים נלמד ונשתמש בלולאות נוספות, מלבד הלולאה **'לעולמים'**, שיתמכו בפתרון יעיל יותר של המטלה)

תסריטי דמות הבלון



התסריטים של שתי דמויות הבלונים הנוספות מלבד התסריט המופעל 'כאשר מתקבל מסר' - עבור כל בלון מתקבל מסר 'כאשר החתול נוגע בו', ולכן הוראת האירוע בכל אחד משלושת הבלונים תהיה:





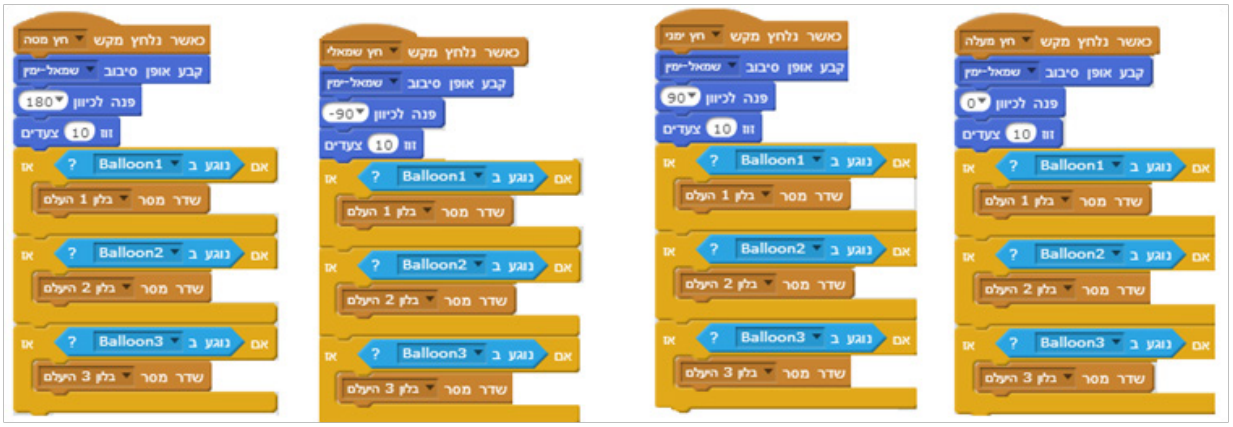
תסריטי דמות החתול

התסריטים המרכזיים של החתול מטפלים בתנועה של החתול על הבמה באמצעות חיצים ועל ידי בדיקה האם החתול נוגע באחד מהבלונים. לכן אציג אלגוריתם מילולי רק של תסריט אחד.

אירוע מפעיל – 

1. קבע את אופן סיבוב הדמות –מניעת היפוך הדמות בסיבוב
2. **'פנה לכיוון'** 90
3. **'זוז** 10 **צעדים'**
4. **'אם נוגע ב'** בלון 1
   4.1 **'שדר מסר'**: בלון 1 היעלם
5. **'אם נוגע ב'** בלון 2
   5.1 **'שדר מסר'**: בלון 2 היעלם
6. **'אם נוגע ב'** בלון 3
   6.1 **'שדר מסר'**: בלון 3 היעלם

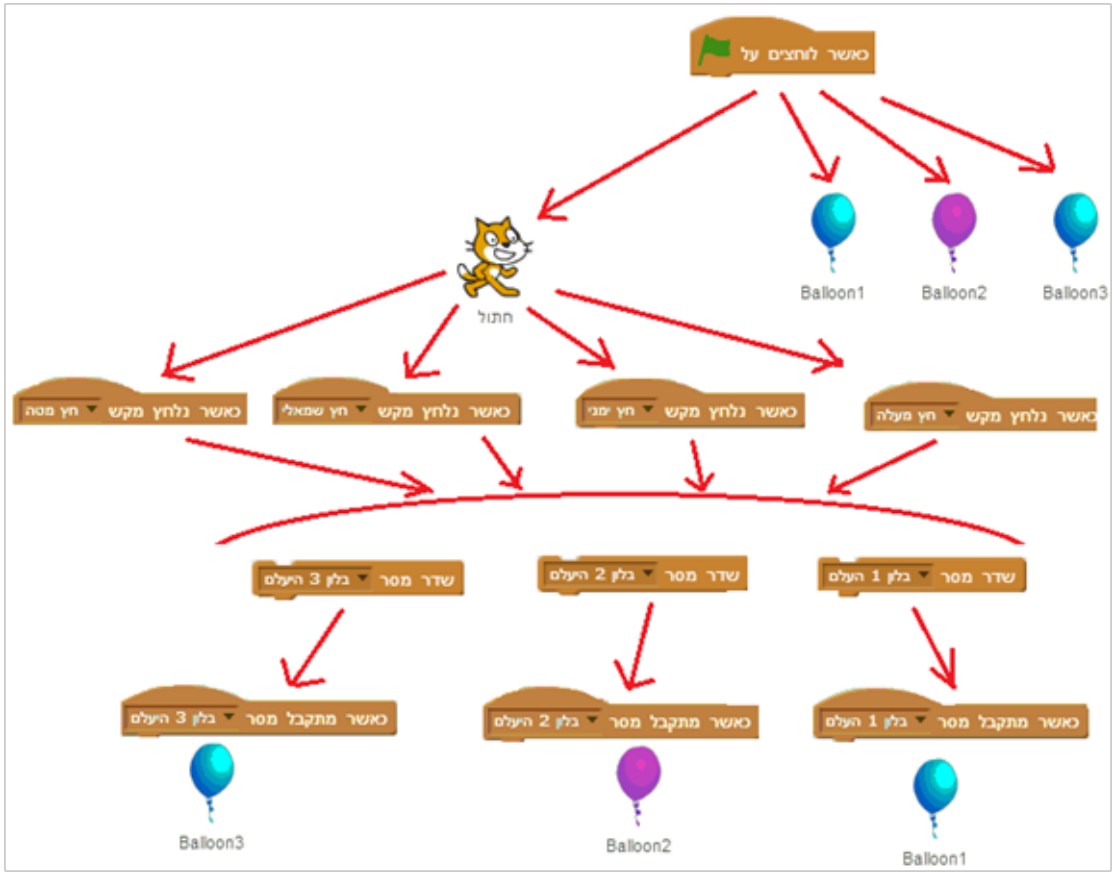
התסריטים (על פי ארבע החצים)



תסריט אתחול משתנה מספר בלונים.



תרשים תזמון תסריטים



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
לולאת **'לעולמים'** – לולאה אינסופית בסקרטץ'	אפשרי לכתוב תסריטים ללא הוראת פתיחת אירוע, לתת לתלמידים להתנסות על ידי שימוש בהוראה **'לעולמים'** בשילוב פקודות שונות	✓	✓	תלמיד ישתמש בלולאת **'לעולמים'** ליצירת אנימציות, משחקים והדמיות, תוך שימוש בתסריטים של דמויות עם תלבושות והוראות הקשורות ומטפלות בשינוי תלבושת של דמות. מומלץ יישומים עם 2-4 דמויות לכל היותר **3 שעות**
לולאת **'לעולמים'** שילוב ויישום במשחקים	יצירת משחקים; יישומים מדעיים עם 2-3 דמויות ותלבושות. מומלץ להשתמש בכלים שהוצגו.	✓	✓	תלמידים בצוותים יציגו רעיון ויממשו את הרעיון באמצעות מבנה בקרה כולל לולאות, שילוב הוראות גרפיקה, תלבושות וקולות. **3 שעות**

מודול מדעי המחשב - פרויקטי העשרה לפי רמות

תחומים: חישוביות, אלגוריתמיקה, מיומנויות המאה 21, מרעיון לקוד

תפקידו של מודול זה לתמוך ולהעשיר את ההוראה ואת הלמידה של מודול מדעי המחשב. ההוראה מותאמת לקבוצות לימוד שונות ומגוונות. המודול מחולק ל-3 רמות של יישום בהתאם למה שהוצג במודול מדעי המחשב.

תכנית רמה א':

בתכנית רמה א' שחלקה העיקרי מהתכנית המלאה הוא כ- 45 שעות, תתבצע ההרחבה ל-60 שעות בהגדלת מספר הדוגמאות, במתן אפשרויות והזדמנויות לכתיבת יישומים, בהקצאת זמן רב יותר לתרגול, בגיוון בדוגמאות 'משחקיות'. בחוברת זו יש 15 דוגמאות העשרה המותאמות לרמה לפי הסעיפים שצויינו.

פרויקטים העשרה רמה א'

פרויקט א' - מי הגיע לשיעור

פרויקט DEBUG IT 3.3 REMIX

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/111316546/

חיפוש פרויקטים דומים עם התגיות: SPEED DRAW

הרעיון המרכזי

החתול נמצא במרכז הבמה ובודק ושואל "מי נמצא", 4 דמויות שכל אחת עונה בתורה "אני כאן"

מורכבות תכנותית

קושי בסיסי, כולל ההוראות: **'שדר מסר וחכה'**; **'כאשר התקבל מסר'**; **'הצג'**; **'הסתר'**; **'קבע גודל'**.

מורכבות גרפית אין מורכבות גרפית

מה רצוי ללמוד ולהפיק מהמטלה הזו? תרגול מעניין בדיאלוג בין דמויות, שליחת מסר, קבלת מסר.

מצורף: התסריט של החתול שבודק **'נוכחות'**, התסריט של TERA , שאר התסריטים של שלוש הדמויות באותה תבנית כמו התסריט של TERA עם התאמה ושינוי לפי הדמות.



פרויקט ב' - לחבק עצים

פרויקט SAVE TREES

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/107723412/

חיפוש פרויקטים דומים עם התגיות: SCIENCE

הרעיון המרכזי

פרויקט בסיסי, מעין תשדיר שירות לעידוד שמירה על עצים.
שתי דמויות, תסריט יחיד.

מורכבות תכנותית

קושי בסיסי, כולל הוראות: **'שדר מסר וחכה'**; **'כאשר התקבל מסר'**; **'הצג'**; **'הסתר'**; **'קבע גודל'**.

מורכבות גרפית רק החלפת תלבושות.

לאן אפשר לקחת את הפרויקט?

לכיוונים של משימת 'יצירת מסע 'פרסום' בנושאים של איסור לנסוע באופניים ללא קסדה, חצייה נכונה של מעבר חצייה, לא לשחק באש, לזרוק בקבוקי פלסטיק למיחזורית וכדומה. זה פרויקט שניתן לבצע בשיעור אחד כולל כל השלבים, אפשר לשלב אנימציה בסיסית עם חילוף תלבושות והמתנה, וכתיבה סדרתית ללא לולאות.



פרויקט ג' - מה זה אטום

פרויקט WHAT IS AN ATOM?

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/107172152/

חיפוש פרויקטים דומים עם התגיות: SCIENCE

הרעיון המרכזי

פרויקט בסיסי, שתי דמויות, תסריט יחיד, מעין מצגת עם 'מורה' – דמות החתול שמצביעה על מרכיבים במודל האטום ומסבירה. ברור שהנושא המדעי אינו קשור לתכנית הלימודים של תלמידים צעירים, אולם היישום מציג דרך לקחת נושא עם מודל ועם איור של הנושא המדעי, ובאמצעות דמות להציג כל מרכיב ולהסביר את המרכיבים.

מורכבות תכנותית

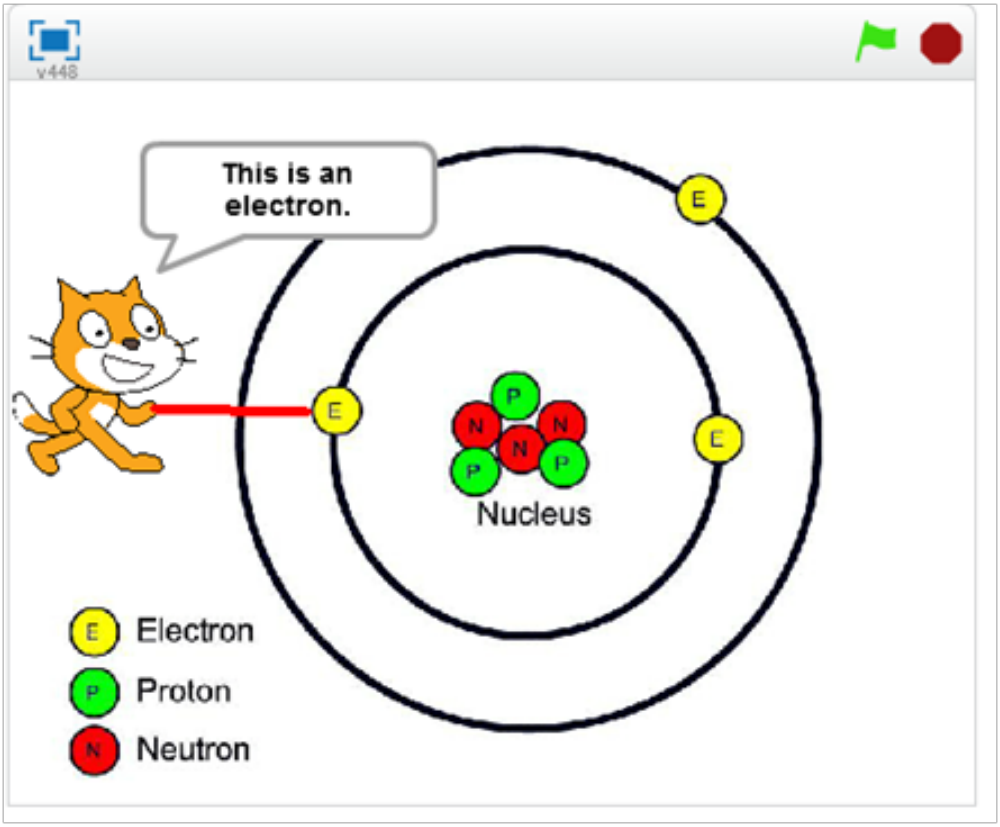
קושי בסיסי, כולל הוראת: **'אמור'**, **'קפוץ אל'**

מורכבות גרפית

המטלה לציור הגרפי של המודל תהיה הכנה ובנייה גרפית של דמות 'מבנה האטום' עבור כל פרויקט אחר. אפשר לייבא את תמונת המודל ממקור חיצוני.

לאן אפשר לקחת את הפרויקט?

לכל נושא בו יש מודל גרפי. ניתן להשתמש ברעיון הבסיסי של יישום זה ולבנות הרצאה מדעית. הפרויקט ניתן ליישום, כולל כל השלבים, בשעה אקדמית אחת.



פרויקט ד' - תרגול בתרגילי כפל

פרויקט MULTIPLY WITH TERA!

2 דמויות, 2 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/108241886/

חיפוש פרויקטים דומים עם התגיות: TERA, MATH, MULTIPLY

הרעיון המרכזי

תהליך המתבצע לעולמים. מוצג תרגיל כפל (מספרים אקראיים), המשחק נדרש להקליד תשובה ותשובתו נבדקת.

בתכנית הלימודים ישנו יישום דומה וקיימים באתרי השיתוף פרויקטים רבים העוסקים בתרגול בחשבון. הפרויקט לא מורכב ליישום, אולם ישנה חריגה מהמסגרת שהוגדרה לרמת הבסיס – שימוש בשני משתנים – ולכן השימוש בדוגמה מותנה בהבנת היישום של המשתנים. הפרויקט מומלץ כמטלת סיום שנה. כולל: לולאה; שני משתנים; מבנה תנאי; שילוב צלילים.

מורכבות תכנותית

רמת קושי סבירה. יכול לשמש, עם הרחבות ושדרוג, כפרויקט סיום שנה.

מורכבות גרפית

הפרויקט פשוט בהיבט הגרפי. שימוש בדמות מהספרייה של סביבת העבודה, עם 3 תלבושות.

לאן אפשר לקחת את הפרויקט?

הרחבה לתרגילים עם מספר פעולות חשבון, בחירת דרגת קושי, יותר אנימציה וצלילים.

פרויקט ה' - אנימציה ומוסיקה

פרויקט COLORFUL DESIGNER / DRAWER

1 דמויות, 1 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/78357976/

חיפוש פרויקטים דומים עם התגיות: ANIMATIONS , MUSIC

הרעיון המרכזי

הרעיון פשוט, הביצוע קצת יותר מורכב.
הפרויקט כולל שיר-צליל המושמע מתחילתו עד סופו, ואנימציה מלווה.

מורכבות תכנותית

רמת קושי בסיסית, השמעת צליל-שיר בלולאה אינסופית, ובמקביל אנימציה בלולאה אינסופית. האנימציה של דמות עם 20 תלבושות. האתגר בפרויקט זה הוא הדמויות. תלמיד בבית ספר היסודי, גם אם הוא צייר ואומן מוכשר, יידרש לעשרות שעות בכדיי ליצור אנימציה ברמה הזו, אבל נפרט ב 'לאן אפשר לקחת את הפרויקט' איך ניתן ליישם פרויקטים דומים עם פחות מאמץ וזמן.

מורכבות גרפית

הפרויקט מורכב ביותר בהיבט הגרפי. תלמיד שצריך לייצר אנימציה עם תלבושות ברמה המוצגת בפרויקט זה יידרש לעשרות שעות.

לאן אפשר לקחת את הפרויקט?

תלמידים ברמת בסיס יכולים לבחור צליל-שיר; לצרף את הצליל שאחזרו (מרשת האינטרנט) ולצרפו לפרויקט; לבחור דמות רוקדת עם מספר תלבושות (יש כמה דמויות כאלו בספרייה של סביבת סקרטץ'); ליצור את שני התסריטים:

תסריט לולאת אינסוף להצגת האנימציה.

תסריט לולאת אינסוף להשמעת הצליל-שיר.

ברמות יותר גבוהות ועם תלמידים שהם מאיירים מוכשרים, ניתן ליצור דמות ותלבושות מקוריות. רק חשוב לתת לפרויקט מספר שעות ליצירה. לפרויקט הזה יש המון שיתופים, ומאוד חשוב, המון רימקסים (שינוי ושיפור) – מתכנתי סקרטץ' (ילדים ונוער) שעשו שינויים בתלבושות של החתול והוסיפו נגיעות משלהם. זוהי דרך לקחת את הפרויקט הזה לכיוונים יצירתיים ואישיים של כל תלמיד.

פרויקט ו' - ציור

פרויקט COLOURFUL DESIGNER / DRAWER

1 דמויות, 1 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/108323449/

חיפוש פרויקטים דומים עם התגיות: LOOKS 3D, ART

הרעיון המרכזי

דמות BALL נעה לכיוון סמן העכבר ו'משאירה' שובל צבעוני של מסלול התנועה.

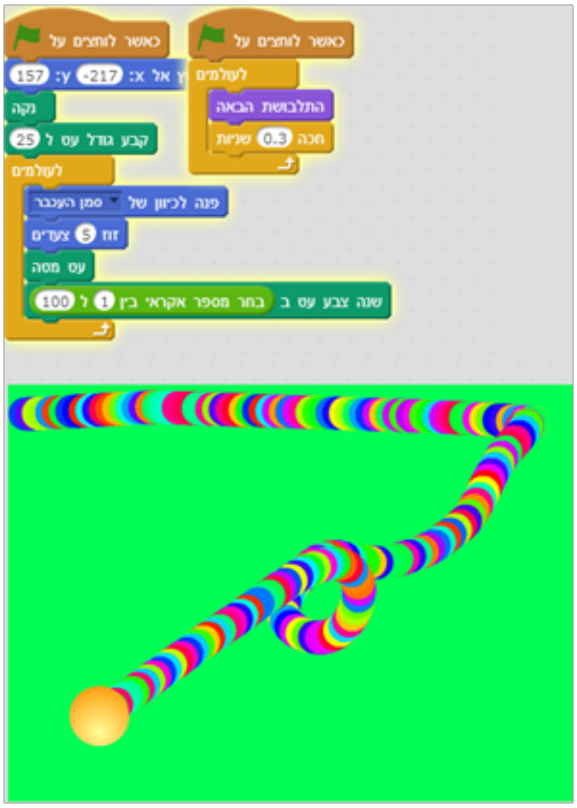
מורכבות תכנותית

המרכיב של התנועה ויצירת 'שובל' צבעוני באמצעות שינוי הצבע עם מספר אקראי של צבע הוא פשוט ויוצג כאן. יש תסריט נוסף שמאותחל עם אירוע **'דגל ירוק'**, שתכליתו בחירת צבע הרקע של הבמה.

הבחירה מבוססת על יצירת תלבושות של הרקע בצבעים שונים ושימוש בהוראה **'שאל'** בה המשתמש צריך להקיש את שם הצבע באנגלית, כשהאות הראשונה היא אות גדולה (לדוגמה Blue עבור כחול) ובהתאם לבחירה משתנה הצבע. אם בוחרים ALL הרקע משתנה בלולאה אינסופית, וגם אפשרות זו מוסיפה לרמת המורכבות של הפרויקט. ייתכן שהפרויקט יעניין בהקשר של תרגול השפה האנגלית לתלמידים דוברי עברית/ערבית כאן בישראל. זה יוצר שילוב בין למידה של מדעי המחשב ויישום בתחום ידע מסוים. כאן בפרויקט זו אנגלית ונושא הלמידה הוא שמות של צבעים.

מורכבות גרפית

אין. תנועה של דמות עם מצב **'עט מטה'** היא מרכיב בסיסי.



לאן אפשר לקחת את הפרויקט?

יש פרויקטים רבים עם שימוש בתנועה עם מצב **'עט מטה'**; בלולאה אינסופית; עם שינוי בגודל העט; בצבע העט; שילוב הפעולה **'החתם'**, ועוד. פרויקטים אלו מתאימים לתרגול של הקבוצות **'עט'** ו **'תנועה'** ושל מפעילים כמו יצירת מספר אקראי. היתרון של פרויקטים מסוג זה הוא יצירת מסך מעניין, מאוד דינמי, באמצעות פקודות פשוטות. כמו כן ישנו שימוש במשתני מערכת כמו: צבע עט, גודל עט והפקודות: **'שנה צבע עט ב'**, **'שנה גודל עט ב'**, **'קבע גודל עט'**, **'קבע צבע עט'**. הוראות אלו והייצוג של התוצאה שלהם בצורה גרפית מסייע לקידום הבנת הרעיון של משתנים והשימוש שלהם ביישומים שלנו, בדרך שעוקפת את הקשיים בהבנת וביישום של משתנים.

הערה: לא הוצג התסריט לבחירת צבע הרקע באמצעות שינוי תלבושת רקע. זה מרכיב מסובך יחסית ולא נדרש. ניתן לבחור רקע אחד מעניין. השאלות והדיון בנושא הם נושאים מעניינים לדיון עם תלמידים.

פרויקט ז' - אנימציה וצלילים

פרוייקט DANCE!

2 דמות, 5 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/30366118/

חיפוש פרויקטים דומים עם התגיות: ANIMATION

הרעיון המרכזי

אנימציה וצלילים של שתי דמויות ורקע. המעניין בפרויקט זה שמחברו בחר ליצור אנימציה נוספת עם תמונת הרקע שכוללת שלוש תלבושות שמתחלפות.

מורכבות תכנותית

לולאת **'לעולמים'**; שינוי תלבושות והעברת מסרים; **'נגן צליל'**; תסריטים בסיסיים; תרגיל מעניין.

מורכבות גרפית מורכבות בסיסית, דמויות מסביבת סקרטץ' ועיצוב רקע מינימלי.

לאן אפשר לקחת את הפרויקט?

כל יישום להצגת אנימציה מלווה בקובץ קול מעניין ומהנה. תרגיל מתאים להיקף שיעור בודד של 45 דקות לכל השלבים, מרעיון ועד קוד.

תסריטי הדמויות



פרויקט ח' - שילוב דמויות ותלבושות

פרויקט JUST FOR FUN. YOU SHOULD WATCH THIS.

מקור: אוסף הפרויקטים ליום הסקרטץ' הבינלאומי 2016

1 דמות, 2 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/111689360/

חיפוש פרויקטים דומים עם התגיות: FUN , SPRITE

הרעיון המרכזי

הרעיון המעניין בפרויקט זה הוא שימוש במספר תלבושות כשכל תלבושת היא עותק של אחת מדמויות סקרטץ' שכלולות בסביבת העבודה. היישום יוצר עניין. עם בחירת תלבושת בצורה אקראית (מספר 1-6) ויצירת שינוי באפקט הצבע, אנחנו מקבלים יישום פשוט ומעניין שמתאים לרמת הבסיס.

מורכבות תכנותית

לולאת '**לעולמים**' ; שינוי תלבושות ; שינוי אפקט צבע.

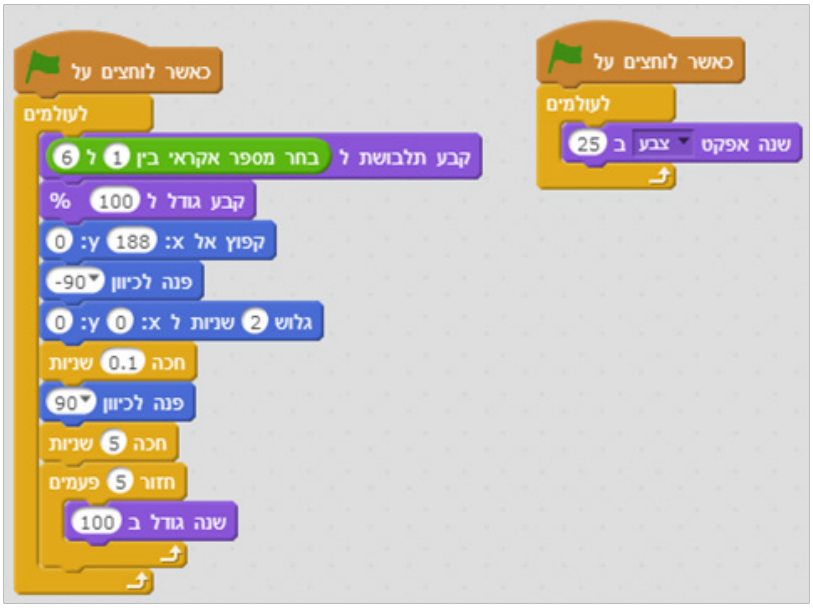
מורכבות גרפית

מורכבות בסיסית, שימוש בדמויות של סקרטץ'.

לאן אפשר לקחת את הפרויקט?

רעיון החלפת תלבושת עם מספר אקראי ושילוב תלבושות מדמויות שונות יכול להיות מאוד יצירתי ומעניין, מומלץ לשלב צלילים.

תסריטי הדמות



פרויקט ט' - גם תלמידות יכולות

פרוייקט GIRLS CAN'T WHAT?

1 דמות, 4 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/109942458/

חיפוש פרויקטים דומים עם התגיות: STOP THE STEREO TYPES, STAY STRON GIRLS, GIRLS CAN TOO

הרעיון המרכזי

פרויקט בסיסי אבל חשוב. הפרויקט מעביר את המסר 'גם תלמידות יכולות' – כל מה שתלמידים עושים גם תלמידות יכולות. מרב התנועה והדינמיות של הפרויקט מושגות על ידי החלפת רקעים. בכל רקע אנחנו רואים תחום נוסף שתלמידות יכולות.

מורכבות תכנותית

חשוב לשים לב: רוב 'העבודה' נעשית על ידי תסריט של הרקע – היישום מציג לנו סדרה של שקופיות עם מוסיקה מלווה, המסר מעניין והתכנון בנוי היטב. המעבר בין הרקעים נעשה באמצעות חיצים ימינה ושמאלה.

מורכבות גרפית

כמובן שלא ברור מקור השקפים – רקעים. כל הרקעים כוללים דמות נערה בסגנון קומיקס שמבצעת מטלה/ משימה ש'רק בנים יכולים'. הכנת מערכת רקעים כזו הינה משימה לא פשוטה.

לאן אפשר לקחת את הפרויקט?

א. העברת רעיון חשוב ומרכזי לגבי יכולות של תלמידות בהשוואה לתלמידים. ניתן להעביר מסר חברתי חשוב על שיוויון. (הפרויקט נוצר ושותף ע"י תלמידות)

ב. יישום זה מציג איך ניתן לעשות מצגת עם סקרטץ'. ישנה אפשרות לנווט בין השקפים וגם לשלב אנימציה וקולות. זו גם דרך להכנת מצגת לשיעור מדעים, היסטוריה וכדומה.

פרויקט י' - אופנה

פרויקט ♡DRESS UP SKY ♡ REMIX

חשוב! כל הדמויות כולל דמות הילדה הן דמויות קומיקס ואין בעיה של חשיפה או עניין ערכי מוסרי. אולם יתכן שמשחק זה אינו מתאים לתלמידי בית ספר דתי – לשיקול דעת המורה האם להשתמש בדוגמה, בהתאם לערכי בית הספר. בכל מקרה מומלץ שהמורה תעיין/יעיין בדוגמה וייקח/ותיקח אותה לכיוונים שקשורים לנושאים אחרים, שיוצגו בהמשך.

21 דמיות, 69 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/111715999/

חיפוש פרויקטים דומים עם התגיות: WORK IN PROGRESS

הרעיון המרכזי

פרויקט נוסף שיעודד תלמידות ללמוד מדעי המחשב. היישום, על פניו, נראה מורכב: 21 דמויות, 69 תסריטים, זה המון. אבל כאשר נכנסים פנימה ליישום רואים שהשד לא כל כך נורא והפרויקט מתאים לכל הרמות. הרעיון המרכזי הוא ביצוע משימת הרכבה בשלבים – את הסדר שלהם ניתן לשנות בכל הרצה וזה הדבר המעניין. כאן יש על המסך דמויות המייצגות חלקי לבוש של נערה, כל הקלקה על חלק מחלקי הלבוש מעבירה את הפריט אל הנערה ולמיקום הנכון. חשוב: הנערה בסוף תראה שונה בכל הרצה, תלוי בסדר ההלבשה שנחבר.

מורכבות תכנותית

התכנות פשוט. בהמשך נציג תסריטים של אחת הדמויות שנותרת זהה מלבד המיקומים המצוינים בהוראות לכל הדמויות. כל דמות, בלחיצה על דגל ירוק, מתחילה בתלבושת בסיסית (פריט הלבוש בצבע בהיר), ונשלחת למיקום על הבמה. לחיצה על הדמות מעבירה את פריט הלבוש למיקום הנכון על הנערה ומשנה את התלבושת. עד שמסיימים לסייע לנערה להתלבש.

מורכבות גרפית

הפרויקט מורכב בהיבט הגרפי. המון שעות עבודה לחיפוש ועיצוב המרכיבים הגרפים. צריך לקחת את הפרויקט לתחומים שונים ולנסות למצוא אתר אחד/מקור אחד לכל קבוצות הדמויות – לעניין זה נתייחס בהמשך. ובמקביל ישנה חשיבה על רימיקס עם מספר דמויות קטן יותר.

לאן אפשר לקחת את הפרויקט?

זה אחד הפרויקטים החשובים, ולכן למרות בעיית אי ההתאמה לבתי ספר דתיים (אולי, זה לא מובן מאליו אבל מטעמי זהירות נציין זאת), ניתן לקחת את הפרויקט לכל נושא. בהקשר לבית ספר דתי (ולא רק), הנה רעיון גנרי לפרויקט שיציג את הפוטנציאל של הפרויקט גם לחיבורים רב-תחומיים. נניח שהתמונה המרכזית היא שולחן שבת. זוהי הדמות המרכזית. על הבמה יהיו כל המרכיבים של מה שנדרש שיהיה על שולחן שבת: מפת שולחן; שתי חלות; זוג פמוטים עם נרות (כאשר נניח אותם על השולחן תתחלף התלבושת לנרות דולקים); גביע קידוש (במעבר לשולחן יתמלא ביין); צלחות וכדומה. המשימה לסדר את שולחן השבת בסדר הנכון.

הפוטנציאל של היישום ברור: ניתן לחבר אותו לשיעורים בנושא השבת, הלכות שבת וכדומה. כך להרוויח חיבור בין-תחומי. יתכן שיתווספו שעות לעבודת ההכנה כי המורה המלמדת תרבות ישראל/הלכות וכדומה תהיה שותפה ו'תשקיע' חלק מהשעות הנדרשות. זה יכול להיות פרויקט סיכום עבור התחום שלה והתחום במדעי המחשב בקיצור WIN-WIN.

פרויקט י"א - מסיבת ריקודים

פרויקט DANCE PARTY

2 דמיות, 2 תסריטים

קישור לפרויקט - **HTTPS://SCRATCH.MIT.EDU/PROJECTS/10128067/**

שתי דמויות רוקדות על במה

מורכבות תכנותית

החלק המעניין בפרויקט הזה הוא הרקע. הרקע מתחלף (6 רקעים שונים) ויוצר במה מחשמלת והוא המרכיב המקורי והחשוב בפרויקט. יבחרו שתי דמויות מתוך ספריית סקרטץ' עם תלבושות הכלולות בדמות. המוסיקה המלווה יוצרת באמת מסיבת ריקודים .

מורכבות גרפית

יצירתן של 6 תמונות הרקע זו העבודה הגרפית החשובה והמרכזית בפרויקט.

לאן אפשר לקחת את הפרויקט?

5422 אהבו את הפרויקט הבסיסי ולמעלה מ-5500 רימקסים נעשו לפרויקט. הכיוון הוא להוסיף דמויות רוקדות נוספות, להוסיף רקעים על בסיס אותו רעיון של שינוי בצבעים, וכן ניתן לשנות את הצלילים המלווים.

פרויקט י"ב - לצייר עם הדמות

פרויקט PAINT WITH GOBO

3 דמויות, 4 תסריטים

קישור לפרויקט - **HTTPS://SCRATCH.MIT.EDU/PROJECTS/10015857/**

דמות המציירת על הבמה

הרעיון הייחודי ביישום הפשוט הזה הוא השימוש בהוראה **'החתם'**. כשאנחנו משתמשים בהוראה זו הדמות מוחתמת על הבמה, כך שאם ממשיכים לנוע והדמות מוחתמת עם שינוי של אפקט הצבע, מקבלים תוצאה גרפית מהנה. אבל יוצרי התסריט לא הסתפקו בכך ושילבו שני מנגנונים לכאורה פשוטים אבל כאלו שתרמו לשיפור בתוצר ולהנאה מיופיו. התוספות כוללות: סיבוב אקראי ימינה או שמאלה של הדמות (שימו לב יש כאן הגרלה של מספרים בתחום: 30- עד 30, יש להסביר זאת בשינוי כיוון הסיבוב ללא כניסה להסבר על מספרים מכוונים); החלפת תלבושת, כאשר התלבושת הבאה אינה מבוססת על שינוי בתלבושת הקודמת אלא על דמות אחרת מהספרייה של סקרטץ'. כך מקבלים בסופו של עניין יישום ציור מהנה.

מורכבות תכנותית

יש לנו לולאת **'לעולמים'** עם תנועה, עם החלפת תלבושת, עם שינוי אפקט צבע, ועם שילוב ההוראה של שינוי כיוון אם נוגעים במסגרת. מורכבת תכנותית בסיסית.

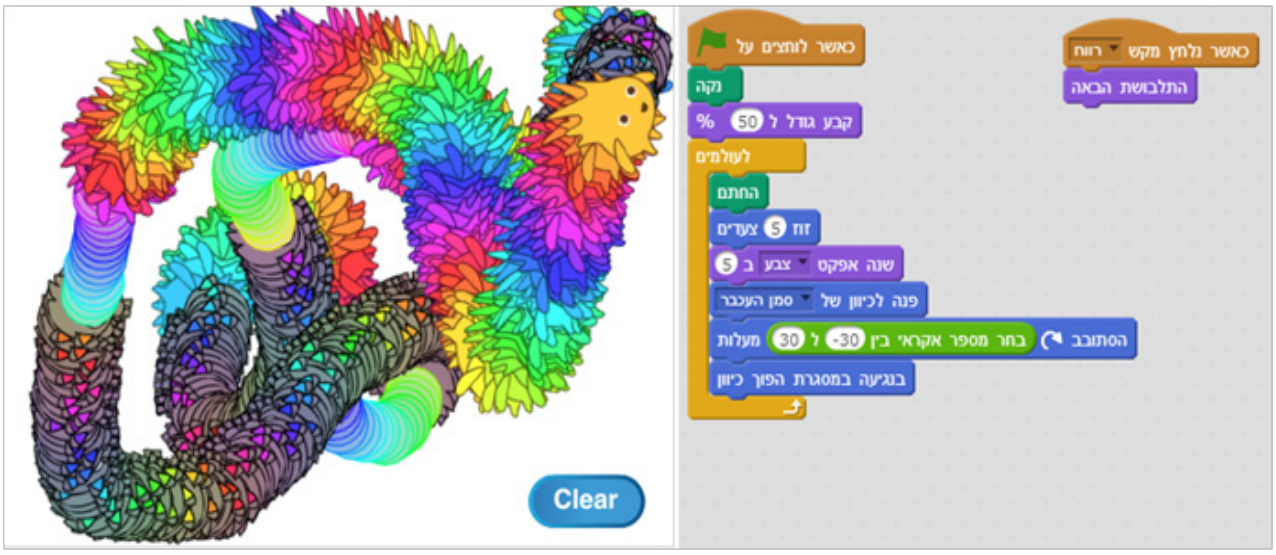
הדמות CLEAR BUTTON מנקה את המסך.

מורכבות גרפית

מורכבות בסיסית. שימוש בדמויות ותלבושות מהספרייה של סקרטץ׳.

לאן אפשר לקחת את הפרויקט?

כמעט 6000 אהבו את הפרויקט וישנם מעל 4400 רימקסים, זה אומר שהיישום אמנם פשוט אבל מלהיב ומזמין שינויים ושיפורים, כולל הוספת תלבושות של דמויות שונות מספריית סקרטץ׳, הוספת מוסיקה, שינוי רקע ועוד.



פרויקט י״ג - לכידת עטלפים

פרויקט BAT SPLATTER

2 דמיות, 2 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/22338140/

משחק בסיסי עם שתי דמויות: מטרה, ועטלף שיש לתפוס

מורכבות תכנותית

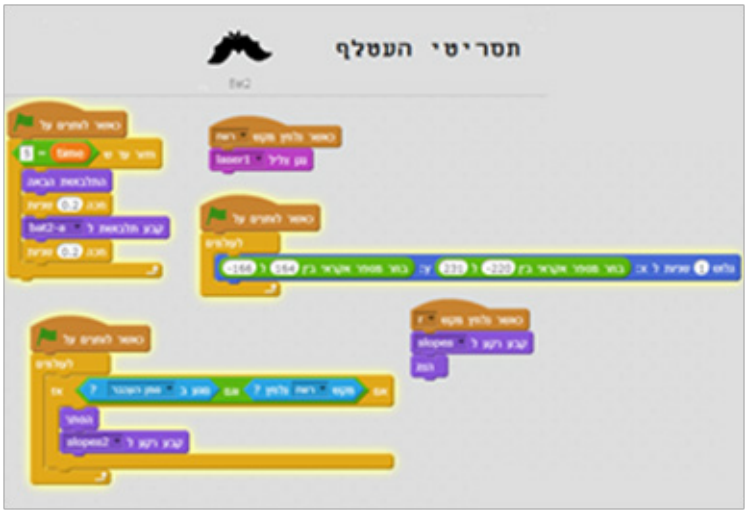
התכנות בסיסי. מעקב עם מטרה עם תסריט פשוט ביותר: ׳קפוץ אל העכבר׳. המטרה נעה עם תנועת העכבר. עטלף שנע אקראי על המסך. ברגע שהמטרה נחה על העטלף ולוחצים על מקש רווח (ירי), פגענו בעטלף. יש חריגה מהכללים שקבענו לרמת הבסיס והיא באה לידי ביטוי בהוראה **׳חזור עד ש׳**. אפשר להחליפה בהוראת לולאת **׳לעולמים׳**; אפשר להציגה רק לחלק מהתלמידים שיתכן שהם בשלים לקפוץ ׳מדרגה׳ ברעיון של לולאות. כל המרכיבים האחרים של הפרויקט פשוטים ובסיסיים.

מורכבות גרפית

מורכבות בסיסית. דמות העטלף עם שתי התלבושות כלולה בספרייה של סקרטץ׳, דמות המטרה מורכבת מעיגול פשוט ושני קווים ניצבים.

לאן אפשר לקחת את הפרויקט?

זהו משחק בסיסי אבל הוא תשתית למשחקים רבים. אפשר לעשות רימקסים לכיוון של עדכון מספר נקודות על כל פגיעה בעטלף ומעבר למספר נקודות מסוים לעבור שלב (למשל עם שני עטלפים), אפשר לקבוע עם משתנה את מהירות תנועת העטלף ועוד.



פרויקט י"ד- להציל את החתול

פרויקט PICK UP THE SCRATCH CAT!

3 דמויות, 2 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/11274982/

זהו פרויקט מאוד בסיסי ומתאים לשיעורים הראשונים של הלמידה ברמת הבסיס. ולמרות שאין לולאות, אין מבנה תנאי ואין מסרים, יש עניין בפרויקט כמטלת סיכום בשיעור של 45 דקות, והחשוב ביותר: הצגת המטלה הבסיסית ואתגרים להרחבת היישום.

מורכבות תכנותית

הוראות תנועה בסיסיות, הוראת **'אמור'** ו**'נגן צליל'**.

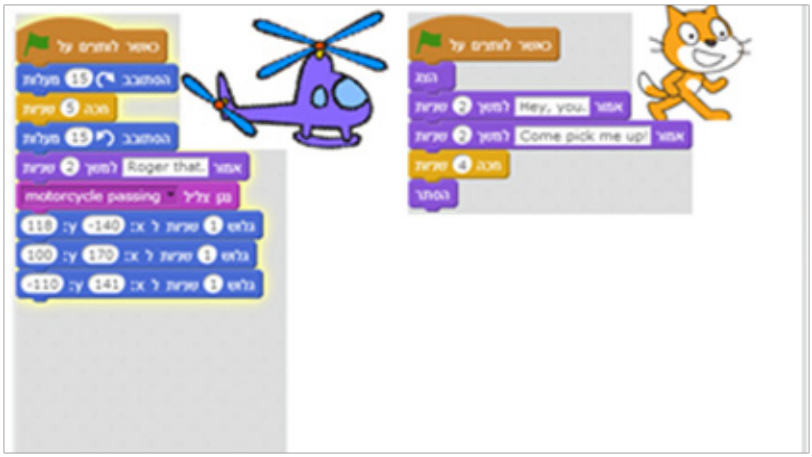
מורכבות גרפית

דמות ההליקופטר ודמות החתול מהספרייה של סקרטץ', נדרש ליצור את הרקע (או לייבא)

לאן אפשר לקחת את הפרויקט?

הפרויקט בסיסי ויתרונו בכך שהוא מהווה מסד להרחבה, לדוגמה: ניתן היה לשלב בתסריט של ההליקופטר החלפת תלבושות וכך היינו מקבלים תנועה של הרוטור הראשי של ההליקופטר; ניתן לתת למשחק להניע את ההליקופטר באמצעות חצים לעבר היעד; אפשר ללכת לתסריט שונה לגמרי בהצלת החתול ובמקום "לאסוף" את החתול ואז דמותו נעלמת ליצור אנימציה על בסיס דמות החתול בה החתול נתלה על ההליקופטר כדי להינצל. גם יצירת אנימציה של להבות מהבניין הייתה משפרת את הדרמה.

לסיכום, פרויקט זה הוא הדגמה כיצד להציג רעיון לתלמידים, לבקש מהם לחשוב על דרך לשפר את הסיפור ובתלות בנקודת הזמן בה מוצגת המטלה (האם למדו לולאת **'לעולמים'**, תנאים וכדומה), התלמידים יוכלו לשפר ולהרחיב את הפרויקט.



פרויקט ט"ו - Pong 2

פרויקט PING PONG GAME

4 דמויות, 7 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/2457848/

מילות מפתח לחיפוש: PING PONG, GAME, EASY

זהו משחק פונג שני שמוצג כאן בפרויקטים כדוגמה ברמת ביניים. זה מעניין וחשוב להציג שני משחקים שכמעט עושים את אותן פעילויות אבל הם שונים לגמרי. כך מוצג לתלמידים הרעיון שיש פתרונות רבים לאותה בעיה/מטלה ומתקיים בסיס לחשיבה מה יותר טוב, ואלו קריטריונים משמשים אותנו להשוואה בין יישומים שיועדו לפתור את אותה מטלה.

מורכבות תכנותית

הרעיון הראשון כאן במשחק – אין רקע. כל המרכיבים של המשחק הן דמויות. הכדור והפס החוסם שזו הבחירה הטבעית, אבל גם ה'מחסום' בתחתית הבמה שפוגש את הכדור שאנו מחמיצים הוא דמות, וגם ההודעה על המשך במקרה של החמצה היא דמות. זו אסטרטגיה שונה ולא תמיד נכונה. גם תנועת הכדור ימינה ושמאלה מושגת בצורה מעניינת ושונה, לא בבחירה הטבעית של חץ ימינה וחץ שמאלה אלא לפי מיקום ה- x של העכבר. ביישום נעשה שימוש במשתנה המערכת **'כיוון'** כדי לדעת מה הכיוון של הכדור ולהפוך את הכיוון. חשוב לתת את הדעת לעניין משום שנעשה גם שימוש במשתנה שאמנם לא הוגדר על ידי המשתמש, אבל מקיים את ההגדרה של משתנה וכולל את אותו הקושי בהבנה וביישום.

מורכבות גרפית

גרפיקה פשוטה ביותר, כל הדמויות הן פסים או עיגול .

לאן אפשר לקחת את הפרויקט?

ההמלצה בפרויקט זה היא להשוות אותו עם הפרויקט שכלול ברמת הביניים. בפרויקט זה יתכן שיש יותר פשטות בפתרון, אך יחד עם זאת ישנו שימוש בכלים שונים וניתן לומר שיתכן שהגישה יותר יצירתית – ההשוואה והדיון חשובים ביותר.

תכנית רמה ב'

בתכנית רמה ב' שחלקה העיקרי בתכנית המלאה הוא כ- 50 שעות, ההרחבה ל-60 תתבצע בהגדלת מספר הדוגמאות, במתן אפשרויות והזדמנויות לכתיבת יישומים, בנתינת זמן רב יותר לתרגול, בגיוון הדוגמאות המשחקיות, בהוספת דמויות, בהוספת תסריטים, בעבודה על גרפיקה אישית (המעוצבת על ידי התלמיד ולא מיובאת) ובשילוב צלילים.

פרויקטי העשרה רמה ב'

פרויקט א' - ציור מהיר

פרויקט CUBE SPEED PAINT

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/106886920/

חיפוש פרויקטים דומים עם התגיות - SPEED DRAW

הרעיון המרכזי

ציור המוצג באופן מהיר על ידי החלפת תלבושות (162 תלבושות). מעבר מהיר בין התלבושות יוצר אשליה של ציור המתבצע בצורה מהירה.

מורכבות תכנותית

קושי בינוני, כולל הוראת '**חזור_פעמים**', שינוי תלבושת, ושינוי גודל של דמות.

מורכבות גרפית

הפרויקט דורש המון שעות לציור התלבושות. צריך לתכנן ציור מלא ולפרק אותו לסדרה ארוכה של ציורים חלקיים – עם כל הוספת קו, לעיתים אפילו הוספה שולית – צריך להכין תלבושת נוספת.

לאן אפשר לקחת את הפרויקט?

לתת לתלמידים לבצע משהו דומה אבל עם ציור פשוט, נניח סמיילי. הפרויקט לא מומלץ ליישום עם ציורים מורכבים. ציור מורכב ידרוש הכנת עשרות תלבושות, מטלה שתיקח זמן רב.

פרויקט ב'-נגינה עם 'פסנתר'

פרויקט MUSIC!

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/106640503/

חיפוש פרויקטים דומים עם התגיות: MUSIC

קישור לפרויקט - /https://scratch.mit.edu/projects/106886920

חיפוש פרויקטים דומים עם התגיות – SPEED DRAW

הרעיון המרכזי

מוסיקת רקע המופעלת עם תסריט של הרקע.

8 דמויות של 'קלידי פסנתר' על הבמה, כל הקלקה על דמות משמיעה צליל שונה.

מורכבות תכנותית

קושי בינוני. אירועים מפעילים: 'כאשר לוחצים על דמות זו'; 'כאשר לוחצים על דגל ירוק'; יש בדיקה אם העכבר נוגע 'בקלידי'; יש החלפת תלבושת שמייצגת את בחירה ה'קלידי'; לחיצה על ה'קלידי' משמיעה צליל נגינה.

מורכבות גרפית

גרפיקה בסיסית, יצירת דמויות המקלדת פשוטה ביותר.

לאן אפשר לקחת את הפרויקט?

יש מגוון אפשרויות רחב של יישומי מוסיקה. ככל ש'האוזן' של התלמיד משוכללת יותר וככל שהיכולת המוסיקלית שלו גבוהה יותר, ההנאה מיצירת יישומי המוסיקה גוברת.

פרויקט ג'-אנימציה

פרויקט WHIRLING BALL

1 דמות, 2 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/105929297/

חיפוש פרויקטים דומים עם התגיות: ART, ANIMATIONS

הרעיון המרכזי

שימוש בלולאות כולל לולאות מקוננות ושינוי באפקטים.

מורכבות תכנותית

הפרויקט הזה מוצג כאן לא בגלל רעיון תכנותי מבריק או התאמה והלימה לתכנית הלימודים. מצד אחד, מרכיבים בתסריט המרכזי בפרויקט כמו לולאה מקוננת אינם חלק מתכנית הלימודים, ומצד שני התסריט אינו מורכב או מסובך למרות שנעשה שימוש בלולאות מקוננות עם שינוי באפקט הגרפי.

הפרויקט חשוב ומוצג משתי סיבות:

1. לעיתים ניתן להציג נושא שחורג מתכנית הלימודים הרשמית, להציג תסריט מעניין ולאפשר לתלמידים לשנות (לדוגמה בתסריט זה ניתן לשנות את האפקט הגרפי על הדמות). יש שבעה סוגי אפקטים כולל אפקט 'רוח רפאים' שמשתמשים בו במשחקים רבים כשרוצים להעלים דמות בשלבים. כל שינוי מציג תוצאה שונה ומעניינת. השימוש בלולאה מקוננת, אינו 'מאיים', אין צורך להבין לעומק תהליכים שלעיתים קרובות קיימים באלגוריתמים עם לולאות מקוננות, ויש בכך סוג של למידה מטרימה לקראת המשך – ההצצה לנושאים שאינם קשורים לתכנית הלימודים שמעידים על מה שעוד לא למדנו.

2. יש בתסריט השני שמוצג בפרויקט שימוש בכלי התיעודי 'הערה'. היכולת להוסיף הערה בתכנית לא שולב בתכנית הלימודים, ויתכן שזהו חסרון של תכנית הלימודים כי הערות יוצרות תיעוד של התכנית והן כלי מרכזי וחשוב בתהליכי פיתוח תוכנה. החשש מהשילוב נובע מהבלבול שהשילוב יכול ליצור. אולם פרויקט זה מציג שימוש מעניין. ניתן להוסיף הערה המזמינה את המשתמש להכניס 'למשחק' מרכיב נוסף באנימציה המוצגת. רק הזזת התסריט לעבר חיבור עם הוראה מפעילת אירוע (כאן זה אירוע דגל ירוק) תפעיל את התסריט, ורעיון זה מעניין וראוי לחיקוי בהקשר שהוא נכון פדגוגית ודידקטית.

מורכבות גרפית

קושי בסיסי, יישום של שינוי אפקט.

לאן אפשר לקחת את הפרויקט?

ללמידה מטרימה של לולאה מקוננת עם תלמידים מתקדמים; לשילוב יותר תדיר של ההוראה '**שנה אפקט**'; לשילוב יותר תדיר של הכלי 'הערה' בתסריטים.

פרויקט ד'- טיפול בציפורניים

פרויקט NAIL ART DESIGNER

14 דמויות, 42 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/2698525/

חיפוש פרויקטים דומים עם התגיות: PAINT ,ART

הרעיון המרכזי

סוג של תוכנת צייר, אבל ייחודית. ביישום 'צייר' קלסי שהוא יישום נפוץ בהוראת סקרטץ' ובאתרי סקרטץ' הרעיון המרכזי הוא לבחור סוג של מברשת, לבחור צבע, ולצייר על המסך. כאן הרעיון נלקח לסביבה אחרת – מעין סימולטור עבור קורס טיפוח ציפורניים. יש בכך יתרון חשוב: אולי זה יעודד תלמידות ללמוד מדעי המחשב.

מורכבות תכנותית

ברמת שימוש בפקודות יש לנו לולאת '**לעולמים**', מסרים בין דמויות והוראות קבוצת '**עט**'. הקושי הוא בהיקף מספר הדמויות והיקף מספר התסריטים, בשימוש רב ברעיון שיתוף בין דמויות, ובהעברת מסרים וקבלתם.

מורכבות גרפית

גבוהה. עיצוב מספר דמויות רב; שליטה בפקודות קבוצת 'עט'; עיצוב רקע מקצועי ועיצוב דמויות מיקצועי. חשוב לראות את הרעיון שאיפשר את המצב בו כשצובעים ציפורן לא חורגים מגבולות הציפורן גם אם מנסים. היישום הטכני טמון באופן ציור דמות היד: הציפורניים אינן כלולות בדמות היד ולכן הן שייכות לרקע שאותו ניתן לצבוע עם פעולות '**הורד עט**' ו'**זוז**' – הוראות שמשפיעות רק על הרקע ולא על דמויות.

לאן אפשר לקחת את הפרויקט?

לפרויקט יש 11 רימקסים ו 138 שעיינו והגיבו על הפרויקט, הוא מסקרן ומחבר תלמידות לתחום מדעי המחשב, הוא ניתן לשינוי לתוכנת ציור רגילה, ובאופן גנרי הוא ניתן לשינוי להרבה יישומים כולל עיצוב ציור לצביעה אישית שזה אופנה עכשווית.

פרויקט ה'-ברכות ליום הסקרטץ' הבינלאומי

פרויקט HAPPY SCRATCH DAY!

מקור: אוסף הפרויקטים ליום הסקרטץ' הבינלאומי 2016

9 דמויות, 26 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/108453927/

חיפוש פרויקטים דומים עם התגיות: SCRATCH

היישום של פרויקט זה הוא 'ברכות ליום סקרטץ' 2016'. זהו יישום קלאסי כי הוא משמש מעין ברכה לאירוע מסוים עם אנימציה וצלילים.

מורכבות תכנותית

המורכבות נובעת מהמספר הגדול של הדמויות, מסדר הופעתן ומזימונן המורכב של תסריטי הדמויות. הוא מקושר לרעיון של סדר הברכות של יום הסקרטץ' הבינלאומי 2016. כל דמות מופיעה בתזמון שונה עם אנימציה וכל ניהול ה'זרימה' של הברכה מורכב ודורש תכנון מוקדם. האנימציה היחידה כלולה בדמויות של הבלונים, ודמויות אלו והתלבושות הן חלק מהספרייה של סביבת סקרטץ'.

מורכבות גרפית

רמת קושי בינונית. יש אמנם 9 דמויות, אבל ל-7 דמויות יש רק תלבושת אחת. הדינמיקה מושגת ביישום עם שינוי בדמויות ועם אפקטים של צבע. האנימציה היחידה כלולה בדמויות של הבלונים, ודמויות אלו והתלבושות הן חלק מהספרייה של סביבת סקרטץ'.

לאן אפשר לקחת את הפרויקט?

לכל יישום שקשור לאירועים של התלמידים. פרויקט זה הוא הזדמנות עבור התלמידים ללמידה רלוונטית ואותנטית. התלמידים יכולים ליצור ברכה ליום ההולדת של ההורים או של חברים, ברכה לרגל אירוע משמח אחר ברמה המשפחתית, הבית-ספרית, הקהילתית או הארצית.

פרויקטים מעניינים נוספים ליום הסקרטץ' הבינלאומי:

יישום מורכב - HTTPS://SCRATCH.MIT.EDU/PROJECTS/107611116/

יישום פשוט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/109688140/

פרויקט ו׳- תזמורת

פרויקט MOJA MUZA NR 1 (LOSOWE D׳WI׳KI) (כך זה נקרא באתר)

מקור: אוסף הפרויקטים ליום הסקרטץ׳ הבינלאומי 2016

5 דמויות, 5 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/109013269/

חיפוש פרויקטים דומים עם התגיות: MUSIC

הרעיון המרכזי

הרעיון המרכזי מעורר התפעלות בפשטותו וביצירתיות שלו. קיימים פרויקטים רבים של נגינה ומוסיקה בסקרטץ׳ וזה פרויקט פשוט וייחודי.

ישנם 5 כלים שונים על הבמה. עם הפעלת היישום אף כלי לא מנגן. בכל לחיצה של מקש (יוצר היישום בחר במקשי חצים ומקש רווח, אפשר לבחור מקשים אחרים) יזומן אירוע כשלחיצת המקש מכניסה כלי נוסף לתזמורת. מאוד מעניין ונעים לשמוע.

מורכבות תכנותית

פשוטה. זימון אירועים עם **׳כאשר נלחץ מקש׳**, כך מזומנים 5 התסריטים – תסריט לכל כלי מוסיקלי. מרגע שהכלי ׳זומן לנגן׳ הוא מנגן בלולאה אינסופית וכך מקבלים תזמורת מושלמת.

מורכבות גרפית

כל הדמויות מספריית סקרטץ׳. אנימציה בסיסית רק עם שתי דמויות התופים (מתוך סביבת סקרטץ׳)

לאן אפשר לקחת את הפרויקט?

ניתן להרחיב את כלי התזמורת, לקבוע סדר שונה, לשנות צלילים וכדומה. זהו יישום מעולה לכל הרמות בתכנית הלימודים. אפשר כמובן לראות את כל התסריטים בקישור, אולם מאחר ומדובר בקבוצה קטנה וקצרה של תסריטים שנציג כאן, רצוי שהתלמידים יבצעו ויצרו הכול מהתחלה וכך יוכלו לקחת את היישום למחוזות חדשים. עבור מספר כלים, עם צלילים בעלי משמעות ׳מוסיקלית׳, היישום הזה יכול להיות פרויקט סיכום שנה מעולה לתלמידים עם ידע, הבנה ושמיעה מוסיקלית.

פרויקט ז׳- ׳הקיפוד והפטריות׳

פרויקט HEDGEHOG

4 דמויות, 9 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/10110854/

משחק מעניין עם כמה מרכיבים ייחודיים. הפרויקט נכלל ברמת ביניים כי יש בו שימוש במשתנה רגיל וגם במשתנה ענן. המטרה של הקיפוד לאכול 5 פטריות, הזמן שנמשך עד להשגת המטרה נשמר במשתנה, ובכל סיום משחק הזמן מושווה לזמן שבמשתנה הענן. כך משתנה הענן ׳זוכר׳ את הזמן הטוב ביותר אותו השגנו.

מורכבות תכנותית

יש מספר מאפיינים שעושים את המשחק למעט מורכב: השמירה על הערך הקטן ביותר של זמן הביצוע. זהו רעיון לא פשוט לתלמידים בכיתות ד'-ה'. במידת הנדרש המורה יכול לוותר על מרכיב זה. עובדה מעניינת נוספת: כשהקיפוד אוכל פטרייה אנחנו רואים בחלק הימני העליון של הבמה שמתווספת דמות מוקטנת של מטרייה. כך זה נמשך עד שהקיפוד אוכל 5 פטריות ועל הבמה יופיעו 5 פטריות קטנות המעידות שאכן הקיפוד אכל 5 פטריות. זהו שימוש מעניין במונה אירועים, כשהאירוע הוא כמה פטריות אכל הקיפוד בכל נקודת זמן ללא שימוש במשתנה. זוהי למידה מטרימה לנושא משתנה ולתבנית מנייה. מעניין שהתוצר הזה מושג על ידי שימוש בתלבושות של דמות.

הדמות כוללת 5 תלבושות:

תלבושת א' - עם פטרייה אחת קטנה.

תלבושת ב' - עם שתי דמויות של פטריות קטנות.

וכך הלאה עד להגעה ל-5 פטריות.

מורכבות גרפית

קיימת. יש צורך בעיצוב הפטרייה הגדולה, הפטריות הקטנות והקיפוד .

לאן אפשר לקחת את הפרויקט?

הכי חשוב זה הרעיון של מונה פטריות שנאכלו. זו אסטרטגיה מעניינת למנות מספר 'ניצחונות' של דמות ללא שימוש במשתנה – ורעיון מקורי.

פרויקט ח'-להתחמק מהמכשפה

פרויקט SIMPLE GAME WITH WITCH AND CRAB

3 דמויות, 9 תסריטים

קישור לפרויקט - **HTTPS://SCRATCH.MIT.EDU/PROJECTS/10110854/**

זהו משחק קלסי לרמיקס. הוא פשוט, קל להבנה ועם רעיון מעניין, אבל המשחק כמעט לא אפשרי. מטרת המשחק שהסרטן יגיע למפתח הפלא, אבל המכשפה חוסמת את המעבר בתנועה אנכית ואין סיכוי שניתן לעבור אותה מבלי שהיא תרגיש. התלמידים צריכים לפתור את הבעיה – מתפתח דיון סביב הרעיון שמשחק צריך להיות מעט קשה אבל לא בלתי אפשרי.

מורכבות תכנותית

התסריטים פשוטים: המכשפה נעה בתנועה אינסופית מלמעלה למטה, אם הסרטן נוגע במכשפה המשחק מסתיים ברעש גדול. הסרטן נע באמצעות חיצים ימינה ושמאלה. הוא צריך לעבור את המכשול, אם הוא מצליח המפתח מקבל מסר מהסרטן שמשמיע צליל והמשחק מסתיים.

מורכבות גרפית

אין דמויות בסיסיות, אין רקע.

לאן אפשר לקחת את הפרויקט?

זו דוגמה למשחק בסיסי פשוט שניתן לשפרו. זהו אתגר לתלמידים לחשוב מה ניתן לשפר: ברקע, במהירות התנועה של המכשפה – אולי עם משתנה ניתן לשנות את מהירות התנועה שלה, ובכך לשפר את רמת המשחק.

פרויקט ט'-מסע הפרפר

פרוייקט SIMPLE FLAPPY BIRD KS2 KS3

2 דמויות, 4 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/24266375/

המשחק הזה מיוחד, פשוט, אבל יש בו רעיון גאוני – טריק תכנותי וגרפי ליצירת תנועה. בתחילת עידן משחקי המחשב פותחו משחקים בהם דמות נעה לאורך המסך ומכשולים הגיעו לדמות, בצורת כל מיני "אויבים שיש לחסל", כשהמסך נע מימין לשמאל בתנועה זורמת. המשחק הזה משיג את רעיון התנועה של הרקע מימין לשמאל בתכנות מקורי ועיצוב גרפי תומך, וזה ייחודו. המשחק מלמד גישה מעניינת למימוש רעיון התנועה של הרקע.

מורכבות תכנותית

4 תסריטים, לא מורכבים, בדרגת רמה בינונית, אבל צריך לשים לב לחילוף התלבושות של דמות המכשולים ולכך שהתנועה של המכשולים מתבצעת מימין לשמאל לעומת התנועה של הפרפר וכך משיגים אשליית תנועה.

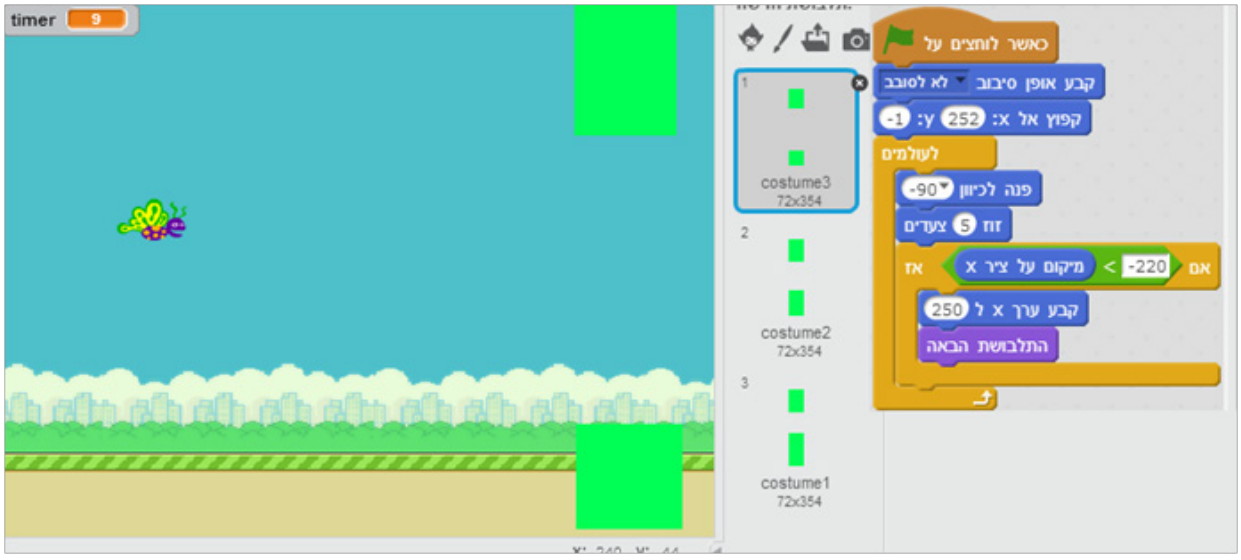
מורכבות גרפית

דמויות בסיסיות, רקע בסיסי.

לאן אפשר לקחת את הפרויקט?

אפשרות לשנות את מהירות תנועת המכשולים, לגוון את התלבושות של המכשול, להעביר את הרעיון של תנועת המסך ליישום במשחקים אחרים; המכשול מהווה תלבושת והתנועה מימין לשמאל – התלבושת מתחלפת ולכן המכשול מול הפרפר מפתיע ומשתנה.

הסבר על המימוש של הרעיון לתנועת הבמה מימין לשמאל:



פרויקט י'- משולש עיגול או מלבן

פרויקט LM1

6 דמויות, 15 תסריטים

קישור לפרויקט - **HTTPS://SCRATCH.MIT.EDU/PROJECTS/102199722/**

זהו פרויקט סיכום לרמת ביניים. המורכבות של הפרויקט מאופיינת בשימוש ב-3 משתנים, בתנאי עם שימוש באופרטור **'וגם'**, ובמספר גדול יחסית של דמויות ותסריטים.

הרעיון מאחורי היישום הזה הינו ללמד את המחשב מה אני אוהב. תלמיד שמשחק במשחק מקבל הנחייה ללחוץ על דמות הכדור עם המילה PLAY, על המסך תופיע אנימציה קצרה שבה מוצגות צורות (משולש, ריבוע, עיגול), שישנו את צבען ומיקומן על המסך. בשלב שהאנימציה מסתיימת התלמיד בוחר איזו צורה הוא מעדיף. כאשר לוחצים על הדמות עם המילה END, המחשב קובע איזו צורה הנדסית המשחק מעדיף. הבחירה של המחשב בצורה המועדפת תלויה כמובן במנייה שהתכנית מבצעת בכל פעם שנבחר משולש, ריבוע, או מלבן. זה יישום פשוט עם שלושה משתנים ורעיון מעניין.

מורכבות תכנותית

המורכבות היא בשימוש בשלושה משתנים כמונים ובשימוש באופרטור **'וגם'**. כל היישום מתבסס על תכנות מונחה אירועים, ומנצל את הסביבה כדי ליצור את הסדר של האירועים.

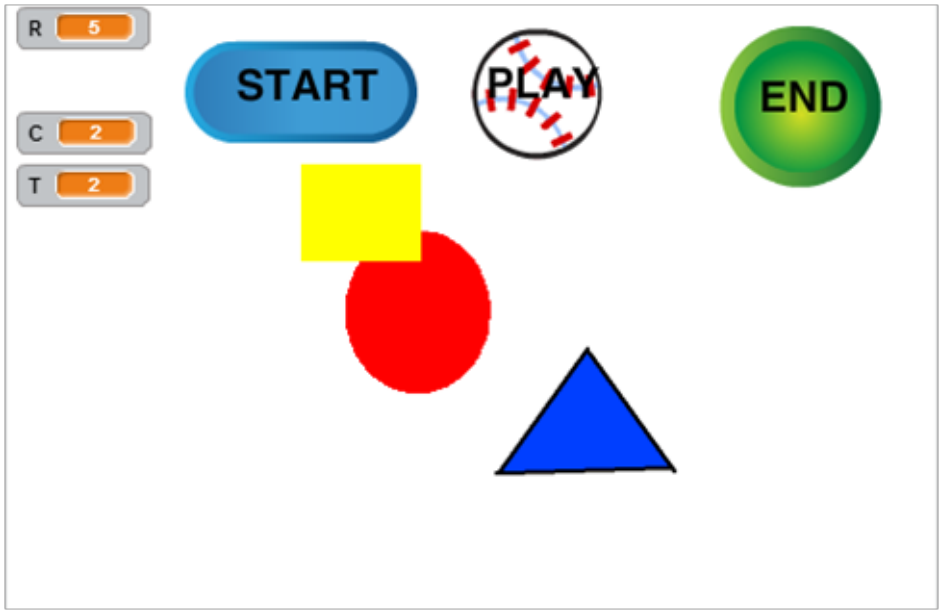
מורכבות גרפית

הגרפיקה אינה מורכבת. כל הדמויות נבחרו מהספרייה של סקרטץ' וללא רקע.

לאן אפשר לקחת את הפרויקט?

אפשר לשנות: 3 צורות גיאומטריות להחליף ל-3 סוגי פירות, או ל-3 סוגי מכוניות וכדומה. הפעילות יכולה להיות חברתית ובאווירה של שיתוף בכיתה בהתאם להעדפות התלמידים. מספר הדמויות והתסריטים גדול יחסית.

כל הפרטים על היישום באמצעות הלינק המצורף.



תכנית רמה ג':

את תכנית רמת ג' שחלקה העיקרי מהתכנית המלאה הוא כ- 60 שעות, מומלץ להרחיב ולהמשיך לפרקים י"א-י"ב; לכלול לולאת **'חזור ___ פעמים'**; ולכלול שימוש מוגבל בלולאת **'חזור עד ש'**; להשתמש בגרפיקה אישית לדמויות ולרקעים; להרחיב בשימוש בדמויות ותסריטים; להעביר מסרים ופרויקטים המשלבים ידע מדעי וטכנולוגי ואף לשלב נושאים מתחומים רבי מלל.

פרויקטי העשרה רמה ג'

פרויקט א'-מערכת השמש

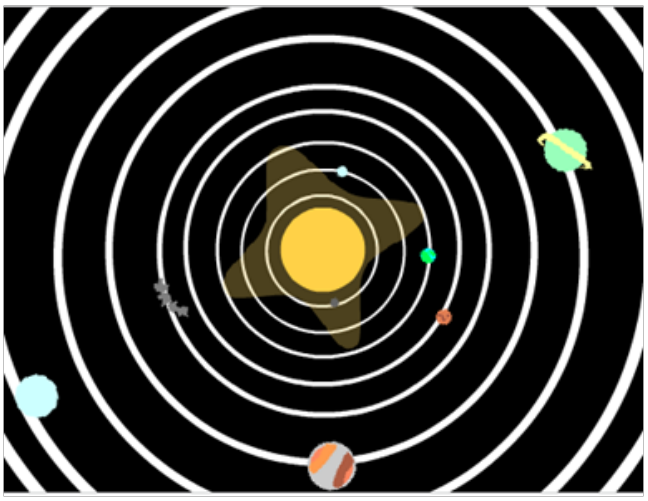
פרוייקט SOLAR SYSTEM

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/95244140/

חיפוש פרויקטים דומים עם התגיות: SCRATCH SCIENCE GROUP

הרעיון המרכזי

הרקע הוא מערכת השמש, עם מסלולי כוכבי הלכת. כל כוכבי הלכת נעים במסלולם.



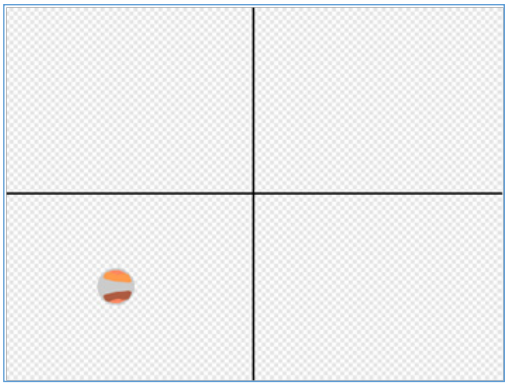
המיקום של כוכבי הלכת נקבע באמצעות מיקומה של השמש ב- 'תלבושת'. 'רוח השמש' ממורכזת באמצע וכל כוכב לכת ממוקם לפי המסלול שלו ביחס לשמש, כך קיבלנו את כוכבי הלכת במיקום היחסי לשמש. במגע בין כוכב לכת לעכבר – כוכב הלכת אומר את שמו.

מורכבות תכנותית

קושי בינוני. לולאות **'לעולמים'**; לולאת **'חזור'**; פקודות גרפיות פשוטות.

מורכבות גרפית

מורכבות גבוהה יחסית. יש למקם את הדמויות על המסך באמצעות לשונית 'תלבושת' ובהתאם למיקום היחסי מהשמש, זו לא פעולה פשוטה וצריך לדייק.



חשוב! זוהי אסטרטגיה מאוד יצירתית לקביעת המיקום ביחס לשמש. אין כל התייחסות למיקום מסוים – כל הדמויות ממוקמות במיקום 0,0, אבל בעיצוב התלבושת המיקום של הדמות ביחס למרכז הוא לפי המיקום שלה ביחס לשמש – בצורה פשוטה מתקבלת כאן האשליה של התנועה. בציור הרקע מקפידים שקווי מסלולי התנועה של כל כוכב לכת יתאימו למיקום הגרפי שלו.

לאן אפשר לקחת את הפרויקט?

ליצור הדמיות של תנועת הירח סביב כדור הארץ; להמחיש את תנועת כדור הארץ סביב השמש וסביב עצמו עם שימוש בכמה דמויות של כדור הארץ. אפשר לבחון את היתכנות היישום של הרעיון היצירתי למיקום דמויות, ולעשות זאת באמצעות שינוי נקודת המרכז שלהן במסך העיצוב הגרפי של תלבושות הדמות.

פרויקט ב'-'מבוך'

פרויקט THE CRABBIE MAZE HARD MODE

4 דמויות, 4 תסריטים.

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/97952791/

חיפוש פרויקטים דומים עם התגיות: GAMES

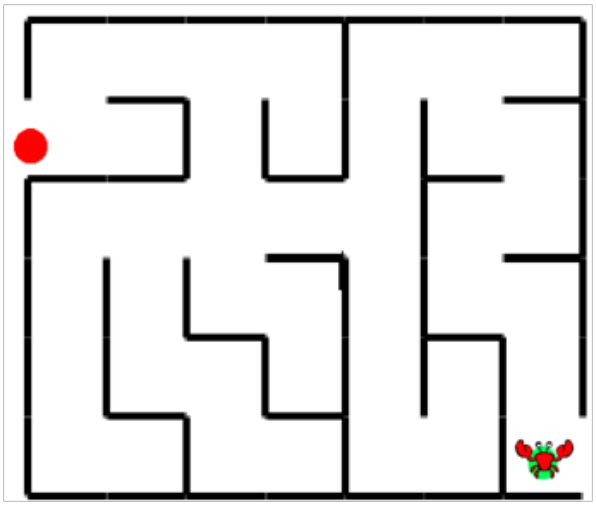
הרעיון המרכזי

זהו פרויקט 'ממשפחת 'המבוכים'. בתכנית הלימודים כלולה גרסה של משחק מבוך, זו גרסה נוספת ומעניינת. הדמות המרכזית היא דמות סרטן שנעה במבוך באמצעות חיצים. הדמות אינה משנה כיוון ולא תמיד ברור האם הדמות עוברת במעבר או לא, עובדה שמקשה על המשחק. התסריט המרכזי הוא של הסרטן שעיקרו:

לולאת '**לעולמים**' – תנועה לפי ההוראה אם מקש נלחץ.

'**אם נוגע ב**' צבע שחור המשמעות היא פסילה וחזרה לנקודת ההתחלה.

הציור של המבוך אינו חלק מהרקע אלא דמות, וזו הבחירה של יוצר המשחק.



מורכבות תכנותית

קושי בינוני+, לולאת **'לעולמים'**, תנאים מקוננים, שינוי במיקומי הדמות לפי X ולפי Y.

מורכבות גרפית

מורכבות בסיסית, כל הדמויות הן חלק מאוסף השפה. אין יצירת גרפיקה.

לאן אפשר לקחת את הפרויקט?

קיימים כמעט אינסוף משחקי מבוך, עולם ומלואו, מומלץ להתחיל בבניית משחק פשוט.

פרויקט ג-'טמגוצ'י גרסת סקרטץ'

פרויקט YOUR MONKEY

6 דמויות, 26 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/101565273/

חיפוש פרויקטים דומים עם התגיות: WORK-IN-PROGRESS, ANIMATIONS

הרעיון המרכזי

זהו משחק מעניין שהוא מחווה למשחק פופולארי של שנות ה-90 – 'טמגוצ'י

טָמָגוֹצִ'י (ביפנית: たまごっち - "ביצה בתוך שעון"), הוא צעצוע דיגיטלי קטן שפותח ביפן ב-1996. למכשיר שלושה כפתורים בדרך כלל, ניתן "לגדל" (על הצג) יצור דיגיטלי שזקוק למזון, לחינוך, לניקיון, לחיבה ולבריאות (הצרכים משתנים בהתאם למכשיר). בטמגוצ'י עשויות להיות דמויות שונות (מתוך ויקיפדיה)

במשחק זה אנחנו 'דואגים' לקופיף חמוד: שיאכל, שישתה, שישחק, ושילך לישון. תפעול המשחק פשוט: אם רוצים שהקופיף ילך לישון מקליקים על מיטה, אם רוצים שהקופיף יאכל מקליקים על בננה וכדומה.

מורכבות תכנותית

זהו פרויקט לא פשוט תכנותית, במיוחד ברמה ובהיקף שיושם. הכללת משתנה לכל מאפיין 'חיים' של הדמות, עדכון משתנים: אם הקופיף לא ישן מדד העייפות שלו גדל וכך לגבי שתייה, שעמום ועוד.

מורכבות גרפית

מורכבות בסיסית, כמעט כל הדמויות הן מתוך הספרייה של סביבת העבודה והדמיות הנוספות הן ברמה של העתק והדבק ממקור חיצוני.

לאן אפשר לקחת את הפרויקט?

הפרויקט ניתן ליישום עם פחות "צרכים", נניח לדאוג לדמות רק לאוכל ושתייה. ניתן לקחת את הפרויקט למערכות מכניות כמו מנוע מכונית שאם הוא מתחמם צריך להוסיף לו מים ואם נגמר הדלק צריך לתדלק ועוד. זה משחק שמציג היבט של הדמיית תהליך עם משתנים המציגים את מצב התהליך, וזו נקודה מעניינת.

בעמוד הבא מוצגים כל התסריטים וניתן כמובן לגשת לקישור ולעשות רמיקס ליישום.

פרויקט ד' 'כדור נופל'

פרויקט RAINING BALLS

1 דמויות, 5 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/36707652/

מילות חיפוש: GAME, RAINING, WRECKING

משחק פשוט הכולל כמה מהרעיונות החשובים בבניית משחקים על ידי תלמידים. ניקוד בסיסי על הישג – משתנה יחיד. תנועה אנכית מלמעלה למטה של דמות. החמצת הדמות מסיימת את המשחק.

מורכבות תכנותית

המשחק מוכנס לקטגורית הפרויקטים ברמה הגבוהה בגלל השימוש בשלושה מרכיבים המאפיינים רמה זו:

שימוש בשני משתנים, משתנה אחד תפקידו לוגי (1 או 0);

תנאי עם אופרטור לוגי AND;

לולאת '**חזור עד ש**'.

מורכבות גרפית

גרפיקה פשוטה. עיצוב רקע עם 'בניינים' שמבוסס על צורות גיאומטריות פשוטות, הדמות המרכזית היא עיגול שחור וזה בוודאי פשוט ליצירה.

לאן אפשר לקחת את הפרויקט?

ליצירה של משחקים שכוללים דמויות ש'נופלות' מלמעלה למטה, שינוי במיקום ממנו מתחילה 'הנפילה' נעשה באמצעות קביעת מיקום X בצורה אקראית, שליטה על מיקום בציר ה- Y ועל הקצב של השינוי, אפשר להוסיף תלבושות לדמות, ולהוסיף צלילים.

פרויקט ה'-מסע הרפתקאות

פרויקט TEENS AT THE CASTLE

4 דמויות, 12 תסריטים

קישור לפרויקט - HTTPS://SCRATCH.MIT.EDU/PROJECTS/10128197/

משחק זה מייצג ז'אנר (סוגה) מאוד רחב של פרויקטים. משחקים מסוג זה הם מסע הרפתקאות שבו עוברים מסצנה/אירוע אחד לאירוע הבא בסיפור, בדרך להצלת הנסיכה מהדרקון או לכל מטרה חשובה אחרת.

המאפיינים של המשחקים הללו: רקעים מיוחדים שמייצגים תחנה ברצף של האירועים. מעבר בין האירועים נעשה בדרכים שונות, כשהדרך הבסיסית היא באמצעות שתי דמויות של חיצים בתחתית הרקע, חץ ימינה לדפדוף ימינה וחץ שמאלה לדפדוף שמאלה. לעיתים המעבר מתחנה אחת בהרפתקה לתחנה הבאה תלוי בהשלמת משימה כמו להילחם במפלצת בדרך, או לענות לחידה של קוסם שפגשנו. המשחק כאן הוא רק מעטפת ומסגרת התחלתית, ויוצר המשחק מזמין אותנו לעשות רמיקס למשחק.

מורכבות תכנותית

במשחק המסגרת המרכיב התכנותי פשוט: דמות חץ המקדמת אותנו לשלב הבא, ושתי דמויות שרק מנהלות דיאלוג.

מורכבות גרפית

בכל משחקי הרפתקה יש כמה אתגרים בהיבט של הגרפיקה כשהאתגר המרכזי הוא עיצוב הרקעים והדמויות. בחלק מהמקרים הרקעים והדמויות מיובאים וצריך רק להתאימם לגודל הבמה ולסיפור של ההרפתקה. כשיש ליצור רקעים ודמויות העניין כרוך בהמון שעות עבודה וכמובן במחשבה יצירתית.

לאן אפשר לקחת את הפרויקט?

זהו פרויקט המייצג את הז׳אנר של משחקי קווסט – הרפתקה. הפוטנציאל של יצירת משחק כזה הוא אינסופי. אפשר להתחבר לתחומי לימוד שונים: לספרות – באמצעות יצירת משחק עבור דמות שמופיעה בסיפור שנלמד בכיתה; לנושא בהיסטוריה – לבנות משחק קווסט המתאר אירועים היסטוריים ופעולות של דמויות באירועים הללו. ילדים יכולים לקחת למשל אגדת ילדים, נניח לדוגמה את סיפור ׳החתול במגפיים׳, לייבא תמונות רקע של הסיפור ממקורות שונים ברשת, ליצור או לייבא את הדמויות בסיפור והמשחק יעביר את החתול ממעמד של חתול רחוב עד לעוזר האישי של הנסיך. ואפשר תמיד לנהל את הדיאלוגים במשחק באנגלית והרווח כמובן ברור.

חלק שני של מודול מדעי המחשב

היקף 30 שעות. פרקי חלק שני: פרק י"א – פרק י"ג.

מודול זה מהווה חלק ממודול מדעי המחשב.

המודול מיועד לתת מענה לאחד מהמצבים הבאים:

1. לבית ספר שאפשר לתלמידיו תכניות ללימוד עקרונות מדעי המחשב במסגרת פרויקטים שונים (חוגים, שעורי תגבור וכדומה) וכתוצאה מכך קבוצת התלמידים סיימה את תוכנית 60 השעות הבסיסית לפני סיום שנת הלימודים – המודול הזה יכול להוות תוכנית המשך.
2. לבית ספר שיחליט ללמד 3 שעות שבועיות – המודול משלים את התוכנית ל- 90 שעות.
3. לבתי ספר שמסיבות שונות לא ילמדו בכיתה ה' רובוטיקה – תוכנית זו תעמוד לרשותם כהמשך למודול 60 השעות בשילוב עם מודול פרויקטי העשרה.
4. לבתי ספר שמקיימים קורסי המשך ומעוניינים להשתתף בימי שיא ותחרויות כמו יום הסקרטץ' הבינלאומי או הארצי, תחרות קוד גורו וכדומה – מודול זה יכול לסייע ולשמש כמקור להשלמת ידע על הסביבה וכמקור לרעיונות לפרויקטים.

המודול כולל פרויקטים שונים שיכולים לשמש את המורים במתן מטלות סיכום לסיום מחצית שנה ולסיום השנה, הדוגמאות מפורטות עם הסברים ויסייעו למורה להנחות את התלמידים.

פרק יא' הכל מתחבר

מטרות הפרק

1. תלמידים יציגו את תהליך ניתוח דרישות היישום ואת מבנה הנתונים של היישום (משתנים ותפקידם).
2. תלמידים יציגו טבלת תסריטים ודמויות עבור פרויקטים מתוכננים, לפני ביצוע וכתיבת קוד.
3. תלמידים יכירו את תרשים תזמון תסריטים המוצג בתכנית לימודים זו, ויישמו שימוש בתרשים זה בפרויקטים אותם הם כותבים.
4. תלמידים יכירו את רעיון תכנון פרויקט ויתוודעו אל תהליך פירוק הפרויקט למרכיבים (לקראת שלב הגדרת לבנה אישית).
5. מעקב אחר תהליכי העבודה; מתן הערכה לתפקוד קבוצה; עבודה בשיתוף פעולה ובשיתוף במידע; התמודדות עם פתרון בעיות.
6. קביעת סולם הערכה ברור לדרישות מהיישום כולל: שימוש במשתנים; מבני מקרה מורכבים; תכנות מקבילי; העברת מסרים; שימוש בגרפיקה כולל עיצוב דמויות, תלבושות ורקע; שימוש בצלילים.

פרק זה מסכם שלב חשוב בלמידה ויוצר אינטגרציה של רעיונות וכלים לצורך כתיבת יישומים מרובי דמויות ומרובי תסריטים. הפרק מתאפיין בקווים מנחים לשימוש לא מורכב במבני בקרה כולל תנאים ולולאות. הוא כולל שימוש בסיסי במשפטי תנאי (לא תנאים מורכבים ולא משפטי תנאי מקוננים), הלולאה היחידה בה נעשה שימוש היא לולאת '**לעולמים**'.

לצד הפשטות של השימוש במרכיבים ומאפיינים של תכנות בפרדיגמה פרוצדורלית, יוצג שימוש יותר נרחב באירועים ככלי לניהול ותזמון התסריטים בסביבה של פרדיגמת EVENT DRIVEN PROGRAMMING. גישה זו מאפשרת יישום מושכל של סביבת הלמידה, ושימוש מגוון בהוראות מקבוצת 'אירועים' ו 'ניהול תזמון האירועים' כדי ליצור משחקים ויישומים מהנים אחרים ב 'אמצעים פשוטים'.

הפרק 'הכול מתחבר' מציג דוגמאות מגוונות. חלק מהיישומים מוצגים במלואם עם כל התסריטים וחלק אחר מציג את המטלה ומאפייניה ובדוגמאות אלו יש להשתמש לעבודה במעבדה. ברור שניתן ואף רצוי לכתוב דוגמאות לפרויקטים נוספים ו/או לחפש רעיונות ויישומים נוספים בשילוב תחומי ידע ונושאים מגוונים.

הכול מתחבר-משימה מסכמת

משחק אוספים תפוחים

פקודות משתתפות בתסריטים

אירועים:

כאשר לוחצים על



תנועה:

קבע אופן סיבוב



פנה לכיוון



פנה לכיוון של



זוז



מראה:

התלבושת הבאה



בקרה:

לעולמים



חכה



מטרת המשימה

ילד הולך במדבר באנימציה של תנועה והליכה, בדרכו במדבר הילד מחפש תפוחים וכל תפוח שהוא מוצא הוא אוכל וכך הוא ושורד.

הסבר על מרכיבים בפרויקט

רקע - רקע מספריית הרקעים של סביבת העבודה – שם הרקע DESERT

דמות - BOY3 WALKING

לדמות BOY3 WALKING יש 5 תלבושות מוכנות אשר מציגות את הילד במצבי הליכה שונים. ל-5 התלבושות הללו הוספנו עוד תלבושות שהם העתקה של תלבושות קיימות.

שינינו את שם הדמות וקראנו לה 'ילד הולך'



שינוי סדר התלבושות

באמצעות בחירת 'גרירה ושחרור תלבושות' בלשונית תלבושת, שינינו את הסדר של התלבושות כך שהשימוש בהוראה '**התלבושת הבאה**' תיצור הדמיה נאה ורציפה של תנועת ההליכה של הילד.

דמות APPLE

יצרנו 4 דמויות של התפוח שנקראות APPLE, APPLE2, APPLE3, APPLE4

חשוב! לדמות התפוח יש צבע אדום כמעט בכל שטח הצורה, תכונה זו תאפשר לזהות את המצב בו הילד נוגע בתפוח באמצעות החיישן . 

הסבר על התסריט הבסיסי המצורף למשימה

תסריט דמות הילד

דמות הילד מכוונת ל-90 מצב התחלתי. קביעת אופן הסיבוב '**שמאל-ימין**' נדרשת כדי שחזית דמות הילד המחליפה כיווני הליכה תהיה תמיד לכיוון ההליכה.

הבלוק '**לעולמים**' יוצר תנועה אינסופית של הדמות שעוקבת אחר סמן העכבר, כך אנו מקבלים את ההדמיה שרצינו - ילד שמטייל במדבר ואוסף תפוחים.

תסריט דמות התפוח

אופן הסיבוב של התפוח נקבע למצב '**לא לסובב**', אין לתפוח /פנים/חזית ולכן אנחנו נקבע שהדמות שלו לא תשנה את כיוונה בכל פעם שישנה פגיעה במסגרת.

בבלוק '**לעולמים**' אנחנו נותנים לתפוח לנוע בכיוון ובמרחק אקראיים, ובודקים שהתפוח לא 'עף לשמים' כלומר שהבדיקה היא ש -Y>25 - במצב זה אנו משנים את כיוון התפוח ומחזירים אותו 'לאדמה'.

בתסריט של התפוח הדמות APPLE מועתקת אל שאר דמויות התפוחים על ידי 'גרור וזרוק' – 'DRAG AND DROP'

כך אותו תסריט פועל על כל התפוחים וכולם מתחילים להתרוצץ במדבר. צריך ללמד את התלמידים גם את האופציה השנייה להעתקת תסריטים – שכפול דמות. בטכניקה זאת יש להכין את כל התסריטים הרלוונטיים לדמות אחת, ואחרי סיום הכנה להשתמש בפקודה '**שכפל**'. בשני התסריטים הללו הנחנו בסיס למשחק קצר.

מטלות ושיפורים:

1. תנועה של הילד באמצעות חיצים, ולא מעקב אחר העכבר

2. הוספת תסריט לדמות הילד או התפוחים; לעקוב: מה מתרחש כאשר הילד נוגע בתפוח; אפשר להעלים את התפוח; לעדכן משתנה נקודות של דמות הילד וכדומה.

3. שינוי בתלבושת התפוח כאשר הילד נוגע בו; שינוי בגודל של דמות הילד כאשר הוא נוגע בתפוח (לאחר אכילתו).

4. כל רעיון אחר שיעלה על דעת התלמידים.

רקע המשחק



התסריטים של דמות הילד ודמות התפוח:

תסריט התפוחים



תסריט הילד



משימת סיפור החתול במגפיים

משימה זו מהווה בסיס רעיוני למשימות רבות שביצועם דורש יצירתיות רבה, יכולת ושליטה בעורך הגרפי, הפעלה של עורך הצלילים, יצירת אנימציות וכתיבה ויכולת ארגון של מרכיבי הפרויקט לכלל יצירה אחת.

הרעיון המרכזי

יצירת סיפור דמיוני, או סיפור המבוסס על אגדה, או סיפור על טיול מהקיץ האחרון. מחלקים את הסיפור ל'דפים' כשכל דף יכיל תמונה שתשמש רקע שהוא חלק מרצף הסיפור. הסיפור מנוט על ידי התסריטים. צריך להציג על הבמה את הסיפור ואת התפתחות העלילה. בסיפור ישולבו דמויות שיופיעו בדפי הסיפור. הדמויות ינהלו דיאלוגים, יבצעו אנימציות וישולבו צלילים מתאימים שילוו את הסיפור.

לצורך הביצוע יכלול תיאור המשימה הוראות העוסקות בהחלפת רקעים, בהצגת דמויות והסתרתם והוראות צלילים ותנועה רבות ומגוונות. כך ניתן יהיה לתת לסיפור חיים. למחפשים אתגר ניתן להפוך את הסיפור למסע הרפתקאות בו בכל דף בסיפור הגיבור מבצע מטלות שביצוען מאפשר מעבר לדף הבא. המטלות תהיינה מגוונות ותכלולנה פתרון חידות, יציאה ממבוך וקרבות עם מפלצות.

להמחשת הרעיון, היישום 'סיפור בתמונות' יהיה סביב הסיפור 'החתול במגפיים', ניתן להשתמש בדמות החתול של סקרטץ', ואפשר לחפש ברשת דמות של חתול חלופית.

תלבושות שונות

יש לעבוד עם העורך הגרפי וליצור תלבושות שונות של החתול שיאפשרו יצירת אנימציית תנועה, תלבושות עם מגפיים וללא מגפים ועוד. יותר תלבושות ובאיכות ובדיוק גבוהים יותר יהפכו את 'סיפור בתמונות' למעניין ומהנה.

צלילים שונים

יש לעבוד עם עורך הצלילים על צלילי רקע, צליל החתול בקבוצת הצלילים הוא בסיסי, וחשוב לחפש ולמצוא צלילים נוספים.

רקעים שונים

תכנן את הסיפור לפי תמונות: 5-7 תמונות של רקעים ושלבים בסיפור.

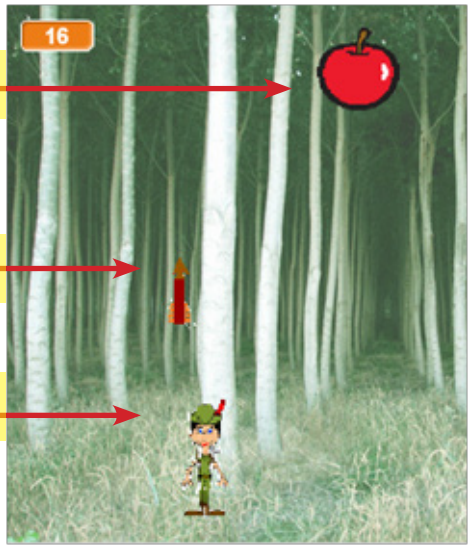
תכנת כל דף בסיפור על כל המרכיבים: צלילים, אנימציות, דיאלוגים ומטלות ביצוע.

בספריית המשאבים של ספר זה יש כמה תמונות בסיס לדמות החתול שניתן לעבד אותן בעורך הגרפי בכדי ליצור את דמות החתול והתלבושות שלו.

מטלות לחתול

בכל תמונה בסיפור החתול מבצע מעשה גבורה: מציל את המלך מטביעה וכדומה. חשוב לשלב אתגרי משחק ולהפוך את הסיפור למסע הרפתקאות שגם יעודד לקרוא וגם יהיה מקור להנאה.

משחק רובין הוד יורה חיצים בתפוחים



- ביכולתכם להניע אותו ימינה ושמאלה ע״י לחיצה על מקשי החיצים
- שיגור חץ באמצעות מקש ארוך (רווח)

אם מצליחים לפגוע בתפוח הניקוד גדל ב-1 ומיד מופיע תפוח 'חדש' שנע הלוך ושוב.

דמויות ודרישות

- תחילת המשחק ע״י לחיצה על דגל ירוק
- המשחק כולל משתנה אחד : 'פגיעות'

דמות 1: תפוח

- מופיעה במקום אקראי בראש המסך
- נעה ימינה ושמאלה והופכת כיוון לאחר נגיעה במסגרת

דמות 2: רובין הוד

- ניתן להניע אותה ימינה ושמאלה ע״י לחיצה על מקשי החיצים (חץ ימני וחץ שמאלי)
- הופכת כיוון לאחר נגיעה במסגרת

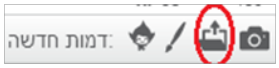
דמות 3: חץ

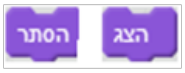
- עד שלא שוגרה היא מוסתרת ואינה מוצגת
- היא משוגרת מתוך רובין הוד ע״י לחיצה על מקש רווח
- נעה בתנועה ישרה כלפי מעלה

פיצוץ: כאשר חץ נוגע בתפוח

- התפוח מתרסק
- הגדלת מונה פגיעות ב-1
- הופעת תפוח חדש (החזרת תלבושת תפוח שלם) במקום אקראי בראש המסך

יצירת דמויות:

- אנחנו מחפשים תמונה באינטרנט ושומרים אותה על המחשב (למשל 'בשולחן העבודה')
- מתוך סקרטץ': יבוא הדמות החדשה ע"י:  ובחירת הקובץ ששמרנו

- הצגה או הסתרה של דמות. ע"י פקודות 

- שינוי לתלבושת מבוקשת: ע"י פקודת 

- טיפול בשיגור חץ:
  - החץ מוסתר כל עוד לא נלחץ מקש רווח
  - כאשר נלחץ מקש רווח דמות רובין הוד משדרת מסר מתאים לדמות החץ ע"י 

  - טיפול החץ במסר ע"י 

    - הצגת החץ במיקום הנוכחי של רובין הוד: ע"י פקודת 

    - החץ נע בתנועה ישרה כלפי מעלה
    - זיהוי נגיעת החץ בתפוח: ע"י חיישן 

    - אם אירע פיצוץ, דמות החץ משדרת מסר מתאים לתפוח

במת המשחק



דמויות המשחק



תסריטי המשחק



דמות

תסריט מרכזי אחד מניע את הדמות בחלק העליון של הבמה בצורה אופקית, ימינה ושמאלה בתנועה אינסופית. אם הדמות נפגעת מהירי (פגיעה בדמות ARROW), מתבצעת החלפת תלבושות למשך זמן קצר למראה של 'התפרקות התפוח'. משתנה פגיעות גדל ב-1. לאחר הפיצוץ הדמות 'חוזרת לחיים' וממשיכה בתנועתה האינסופית וחוזר חלילה.

אלגוריתם התסריט המרכזי של דמות apple:

אלגוריתם התסריט המרכזי של דמות APPLE :

1. פגיעות <== 0
2. קבע תלבושת תפוח שלמה
3. מקם חללית במיקום אקראי (מיקום X אקראי , Y =150)
4. פנה לכיוון 90
5. **'לעולמים'**

5.1 **'זוז** 8 **צעדים'**

5.2 בנגיעה במסגרת הפוך כיוון

5.3 **אם** נפגעתי מחץ

5.3.1 פגיעות + 1 <== פגיעות

5.3.2 החלף תלבושות לתפוח מתפרק (כמה תלבושות)

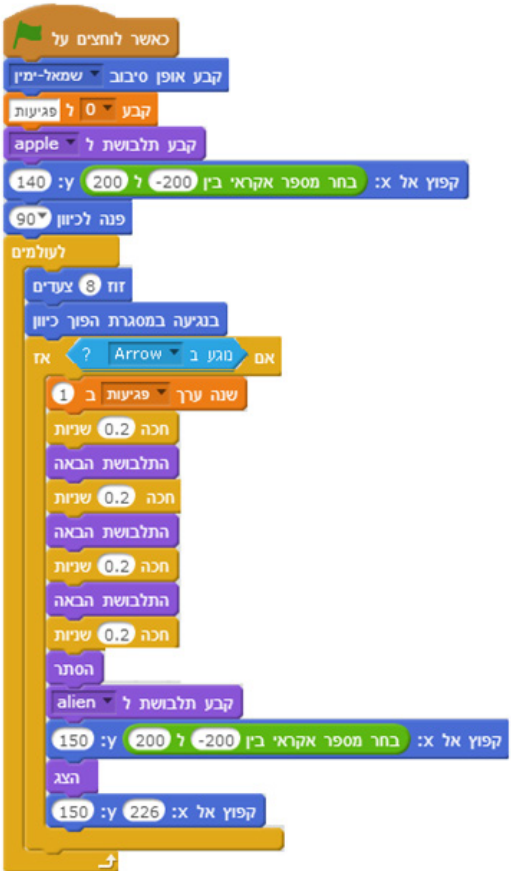
5.3.3 המתן זמן קצר מאוד

5.3.4 העלם את דמות התפוח

5.3.5 מקם תפוח 'חדש' במיקום אקראי

5.3.6 הצג את תפוח

מימוש האלגוריתם:





דמות

תסריט מרכזי אחד, דמות ROBIN נעה בחלק התחתון של המסך בצורה אופקית בלבד, ימינה ושמאלה עם מקשי החיצים. ברגע שלוחצים על מקש רווח נשלח מסר 'FIRE' לדמות ה- ARROW.

אלגוריתם התסריט המרכזי של דמות robin: מימוש האלגוריתם:



1. **'לעולמים'**
   - 1.1 **אם** נלחץ מקש חץ ימני
     - 1.1.1 **'פנה לכיוון'** 90
     - 1.1.2 **'זוז** 10 **צעדים'**
   - 1.2 **אם** נלחץ מקש חץ שמאלי
     - 1.2.1 **'פנה לכיוון'** -90
     - 1.2.2 **'זוז** 10 **צעדים'**
   - 1.3 **אם** נוגע במסגרת
     - 1.3.1 **'הפוך כיוון'**



דמות

בלחיצה על דגל ירוק הדמות במצב מוסתר. הדמות נכנסת לפעולה עם ההוראה **'כאשר התקבל מסר'** - FIRE. השימוש בהוראה **'קפוץ אל'** ROBIN נעשה כך שדמות של 'חץ' נמצאת בדיוק מעל ROBIN. במצב ירי הדמות עוברת למצב הצג ולולאה בסיסית של **'חזור'** מקדמת את החץ אל החלק העליון של הבמה. בדיקה האם החץ פגע מתבצעת על ידי דמות APPLE.

חשוב! הדוגמה הזו בתסריט מקדימה מעט את הסדר של הלמידה כי רק בפרק הבא תוצג לולאת **'חזור עד ש'**, השימוש כאן מוגבל וקל להבנה, מהווה למידה מטרימה לנושא לולאת **'חזור עד ש'** ומשמש דוגמה טובה ליישום של לולאה זו.

אלגוריתם התסריט המרכזי של דמות חץ:

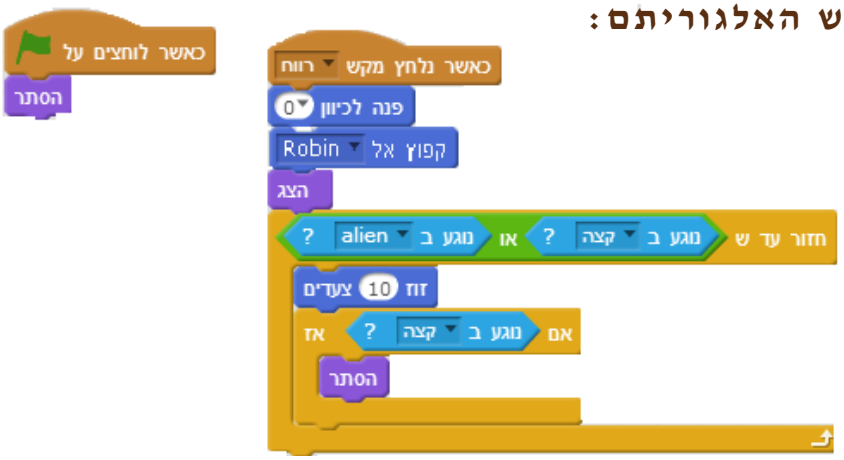
1. **'קפוץ אל'** דמות ROBIN – הוראה זו מייצרת את המצב הגרפי בו דמות החץ נמצאת בדיוק 'בתוך' דמות ROBIN והירי יראה כאילו נורה מתוך דמות ROBIN

2. **'הצג'** את דמות חץ

3. **'חזור עד ש'** (**'נוגע ב'** קצה הבמה או **'נוגע ב'** APPLE)

4. **'הזז'** את החץ 10 צעדים למעלה – הזזת החץ תתבצע על ידי שינוי מיקום ערך החץ של הפגז

מימוש האלגוריתם:



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
פרויקטים ומטלות סיכום. הבנה וכתיבת שיפורים.	בפרק זה צריך לאפשר עבודה יותר נרחבת בצוותים, לתת מרחב לבחירת רעיונות של תלמידים, לאפשר להם לחפש ברשת ובמיוחד באתרי שיתוף בנושא סקרטץ ובמיוחד באתר הבית של סביבת הלמידה.	✓	✓	הבנה וכתיבה של פרויקטים שהוצגו חלקית בכיתה כבסיס להמשך פיתוח. הרחבות של הפרויקטים ; הוספת משתנים ; שיפור אנימציות ; שילוב מוסיקה ; יצירת פרויקטים דומים עם דמויות חדשות וסיפורים שונים ; שילוב נושאים מהיסטוריה, גיאוגרפיה או ספרות. **6 שעות**
פרויקטים ומטלות סיכום.	היעד כאן הוא כתיבה של יישום מורכב (מספר דמויות ותסריטים), בצוותים ובאופן עצמאי.	✓	✓	כתיבת פרויקט בצוותים. **6 שעות**

פרק יב' לולאות - הרחבה

תחומי תוכן: מבנה בקרה לולאה, לולאה מבוקרת, לולאה לעולמים, מחזור בלולאה

מושגים

- ✓ מבנה בקרה לולאה
- ✓ לולאה 'לעולמים'
- ✓ לולאה מבוקרת
- ✓ מחזור בלולאה
- ✓ תבנית מנייה
- ✓ לולאות מקוננות

מבנה בקרה לולאה

מבנה תכנות המכיל הוראה אחת או יותר ומתבצע מספר פעמים. בלולאות מסוג מסוים מספר הפעמים מוגדר מראש, ובלולאות אחרות מספר הפעמים תלוי בהתקיימות של תנאים שונים.

לולאה 'לעולמים' (השתמשנו בה בפרקים קודמים)

לולאה שמתבצעת אינסוף פעמים ללא סיום, לולאה זו ממומשת בבלוק:



לולאה מבוקרת

לולאה שמתבצעת מספר פעמים (מחזורים) מוגדר מראש על ידי ביטוי מתמטי, לולאה זו מיושמת בבלוק:



מחזור בלולאה

לולאה שמתבצעת מספר פעמים, כל פעם שהלולאה מתבצעת נקראת מחזור.

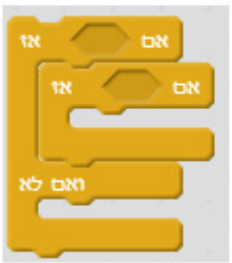
תבנית מנייה

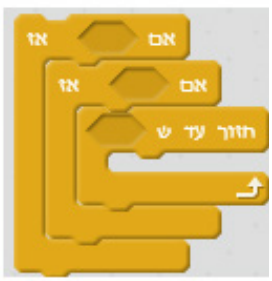
ארגון הוראות בתסריט שכולל לולאה, שמאפשר למנות 'אירוע' שהתרחש ואנחנו מעוניינים למנות את מספר הפעמים שהוא התרחש.

לולאות מקוננות

שילוב לולאה בתוך לולאה. שילוב לולאות בצורה מקוננת יוצר הזדמנויות תכנותיות למימוש רעיונות שונים, אולם קשה יותר לעקוב אחר ביצוע של תסריט שכולל לולאות מקוננות. קינון מבנה בקרה אינו מצומצם רק למבנה לולאה אלא גם למבנה תנאי, וכן לכל שילוב בין המבנים.







נושאי משנה:

- רעיון הלולאה, דיון, פעילות ללא מחשב
- כתיבת אלגוריתמים ומימושם עם לולאות '**חזור** ___ **פעמים**'
- מימוש אלגוריתמים עם לולאות
- מנייה בלולאה
- מטלות עם לולאות

דרכי הערכה

1. תרגול בניסוח וכתיבה מילולית של טענות לוגיות.
2. תרגול ביישום טענות לוגיות מילוליות באמצעות קטעי תכנות בסיסיים.
3. ניסוח סיפור עם התניות, ויישומו בתסריט עם אופרטורים לוגיים וללא אופרטורים לוגיים.

מטרות ביצועיות

1. התלמיד ידע להבחין מתי נדרשת לולאה.
2. התלמיד יבחר את הוראה '**חזור** __ **פעמים**' או '**חזור לעולמים**'.
3. התלמיד ידע לבחור את מספר הפעמים שהלולאה צריכה להתבצע.
4. התלמיד ידע לשלב בלולאות משתנים נוספים, וידע להכיר ולהבין את משמעות השתנות הערכים בכל מחזור של הלולאה.
5. התלמיד ידע לשלב בלולאה משתנים שתפקידם "לזכור" האם מתקיים 'אירוע'.
6. התלמיד ישלב הוראות מקבוצות הוראות שונות כדי ליצור יישומי משחקים, אנימציות, סיפורים ועוד.

מטרות ביצועיות - דוגמאות

1. התלמיד ידע להבחין מתי נדרשת לולאה. להשגת מטרה זו מוצגות כמה דוגמאות, מטרת הדוגמאות היא להטמיע בתלמידים את המושג 'לולאה': כאשר אנחנו כותבים קבוצת הוראות, וקבוצה זו חוזרת מספר פעמים, עולה הצורך לארגן את הקבוצה בתוך מבנה בקרה של לולאה. ניתן לקיים מספר פעילויות ללא מחשב וגם לתת הוראות לתלמידים לבצע פעולה מסוימת מספר פעמים – לתת את אותה הוראה מספר פעמים. לדוגמה: תלמיד מקבל שתי הוראות: 'עמוד' ו'שב על הכיסא'. ההוראות נאמרות 5 פעמים. הדיון בכיתה: אלו מההוראות שנאמרו חזרו 5 פעמים.

   בשלב הבא ניתנת ההוראה: תלמיד 'עמוד ושב 5 פעמים', או ניסוח יותר קרוב להוראה 'חזור':

   'בצע 5 פעמים':

   'עמוד'

   'שב'

   ביצוע מטלות מסוג זה ללא מחשב יעזור להבין מהי לולאה, וכיצד המבנה הזה מייעל כתיבת אלגוריתם.

המטלות הבאות הן מטלות בסיסיות שממחשות מבנה לולאה בסביבת סקרטץ', מומלץ לשלב בדוגמאות תסריטים הכוללים תנועה והחלפת תלבושות.

מטלות הדגמה בשיעור

א. תסריט רקדן היפ-הופ

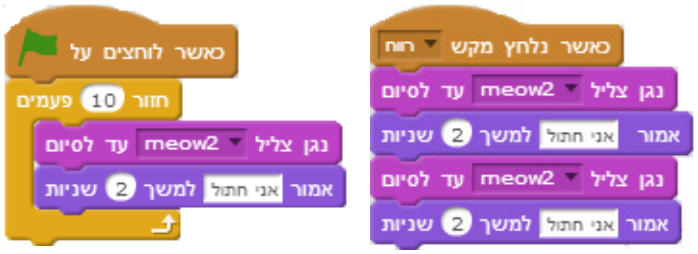
לדמות HIP-HOP יש 13 תלבושות, תכונה זו בשילוב לולאה מאפשרת להדגים אנימציה שממומשת באמצעות לולאת **'חזור'**



ב. כתוב תסריט בו החתול משמיע פעמיים "מיאו" ואומר "אני חתול" . כתוב תסריט בו החתול משמיע 10 פעמים "מיאו" ואומר "אני חתול".

אלו מטלות שמאפשרות לנו להעביר לתלמידים את הנחיצות של מבנה הבקרה לולאה.

הפתרון למטלה ראשונה



מטלות מסוג זה מאפשרות לתלמיד להבין את המשמעות של לולאה ואת היתרון בשימוש בלולאה במקום במבנה הסדרתי. התלמיד יוכל להבין את העוצמה של מבנה הלולאה ולרתום עוצמה זו לצרכיו.

2. התלמיד יבחר את הוראה **'חזור __ פעמים'** או **'חזור לעולמים'**.

המטלה צייר ריבוע באורך 100



תסריטים אלו עם קולות וציור על המסך מאפשרים להבחין בהבדל בין הלולאות באמצעות שמיעה וציור, ועוזרים להחליט מתי לבחור סוג מסוים של לולאה.

3. התלמיד ידע לבחור את מספר הפעמים שהלולאה צריכה להתבצע.

מטלה: כתוב תסריט לציור של מצולע משוכלל: ריבוע, מחומש, משושה (מצולע משוכלל: כל המצולעים בעלי אורך צלע שווה).



חשוב לציין את השוני בין התסריטים שמתבטא בשני הבדלים חשובים:

א. מספר הפעמים שהלולאה מתבצעת עבור כל מצולע.

ב. שוני בזווית הסיבוב של הדמות – לחישוב זווית הסיבוב של הדמות אנחנו משתמשים בביטוי חשבוני שיש לו משמעות גיאומטרית ויש לקיים דיון גם בהיבט המתמטי וגם בהיבט הגיאומטרי. אין להגיע לתסריטים אלו מיד אלא לאפשר לתלמידים לנסות, לטעות ויחד איתם להגיע לתוצאה הסופית.

4. התלמיד ידע לשלב בלולאות משתנים נוספים וידע להכיר ולהבין את משמעות השתנות הערכים בכל מחזור של הלולאה.

השאלה שעולה ממטלה ביצועית (3): האם אנו יכולים ליצור תסריט אחד עבור כל המצולעים. כאן אנחנו מתחילים לשלב בתוך לולאה משתנים נוספים לביצוע המטלות. שימוש במשתנים אלו מאפשר ליצור הפשטה של הבעיה, סוג של פונקציה המקבלת קלט, והתאמה לדרישות המשתמש. כל הרעיונות הללו טמונים בשימוש במשתנה אחד.

המשתנה בו נשתמש יהיה מספר הצלעות הנדרש, נקרא למשתנה 'מספר צלעות', הערך יתקבל מהמשתמש וישמש כערך גם עבור מספר הפעמים שהלולאה תתבצע, וגם כערך בביטוי של חישוב הזווית של הסיבוב.

פתרון המטלה הביצועית עם הכללת חלק מהנושאים החשובים שנלמדו ביחידת לימוד זו:

אלגוריתם מילולי

1. '**נקה**' מסך
2. '**הורד**' עט
3. קלט ==> מספר צלעות
4. '**חזור** (מספר צלעות) **פעמים**'
   - 4.1 '**זוז**' 100
   - 4.2 '**הסתובב**' ימינה (360/מספר צלעות)
   - 4.3 '**המתן**' שנייה

נוכל ליצור הכללה נוספת של המטלה ולבקש שיהיה גם קלט של אורך הצלע.

התסריט הבא כולל את השיפור:



סיכום המטלה הביצועית 4

הגענו לפתרון שמשמעותו שמרעיון מופשט אנחנו מקבלים בסוף התהליך תסריט סופי. אם נרצה כעת לנסח את הדרישה הכללית שתוצאתה התסריט ניסוח הבעיה יהיה:

כתוב תסריט אשר ישרטט על המסך כל מצולע משוכלל שנבקש ושנוכל לבחור את מספר הצלעות ואת אורך הצלע.

התמודדות עם דרישה כזו לתלמידים בכיתות ד'-ה' היא מורכבת ולא פשוטה. כפי שראינו בשלבי הפתרון יש לפתור בעיה ואתגר גיאומטרי; לפתור בעיה מתמטית לצורך ניסוח הביטוי המחשב את זווית הפנייה; ליצור את המשתנים המייצגים את מספר הצלעות ואת אורך הצלע.

בנקודה זו של ניתוח הבעיה יש לחשוב על מבנה הבקרה בו נשתמש; לאלו הוראות מסביבת העבודה הייחודית נדרש. כאשר יש לנו את כל התשובות עלינו לנסח את האלגוריתם המילולי ובסוף התהליך – כתיבת קוד בסביבת העבודה סקרטץ', הפעלת התסריט ותיקון שגיאות במידת הצורך.

מכלול התהליכים הללו מביא לידי ביטוי את אחת ממטרות העל של תכנית הלימודים **ניסוח אלגוריתם וכתיבת קוד בתהליך מובנה.**



5. התלמיד ידע לשלב בלולאה משתנים שתפקידם למנות 'אירוע' שמתקיים

תבנית מנייה - הסבר כללי

תבנית מנייה היא הבסיס לכל מטלה שכוללת מנגנון הסופר כמה פעמים התרחש 'אירוע'. האירוע תלוי בתיאור הבעיה והדרישות. דוגמה למקרים בהם אנו משתמשים בתבנית מנייה:

א. כמה פעמים דמות נגעה בדמות

ב. כמה ציונים שנקלטו היו ציונים שליליים

ג. בכמה תרגילים המשחק ענה נכון

תבנית מנייה כוללת משתנה שמונה את מספר הפעמים ש'האירוע' התרחש, ומנגנון עדכון. האלגוריתם הבא מתאר תבנית מנייה כללית:

1. מונה <==0

1.1 **'חזור ____ פעמים'**

1.2 **'אם'** אירוע התרחש

1.2.1 מונה <== מונה + 1

משתנה מונה יעודכן גם בהוראה מונה <== מונה + 1 או בהוראה מונה <==מונה -1

מטלת דוגמה א':

בתחנה המטאורולוגית בהר החרמון יש מד טמפרטורה שמודד את הטמפרטורה ברכבל העליון, ידוע שכדי שיהיה אירוע שלג הטמפרטורה חייבת להיות נמוכה מ-0. מנהל האתר פנה אליך כדי שתכתוב תסריט שיקלוט את הטמפרטורה שנמדדה בהר בכל יום. על התסריט להשמיע צליל בכל פעם שנקלטת טמפרטורה שלילית, וכן להציג בסוף התסריט הודעה שתציג כמה ימים היה סיכוי לשלג (כלומר הטמפרטורה הייתה מתחת ל-0).

פתרון המטלה

זוהי תבנית מנייה, אנו רוצים למנות כמה פעמים מתקיים ערך מסוים: < 0, כשאנחנו רוצים למנות אירוע מסוים עלינו להגדיר משתנה למטרה זו, המשתנה יתעדכן בכל פעם שהתסריט קולט מספר שלילי.

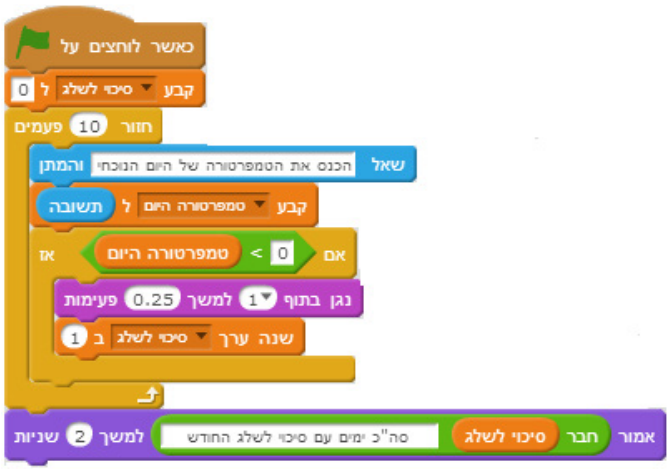
שאלה: איזה ערך מקבל המשתנה של המונה בתחילת התסריט?

תשובה: המשתנה מקבל את הערך 0, כי לא התחלנו לקלוט ולבדוק ולכן ערכו מושם להיות 0, אנו קוראים לזה 'אתחול' המשתנה.

שאלה: בכמה משנים את ערך המשתנה בכל פעם שמקבלים את הערך שקטן מ-0?

תשובה: משנים את הערך של המשתנה ב-1, כי האירוע התקיים ולכן צריך להוסיף 1 למונה של התופעה.

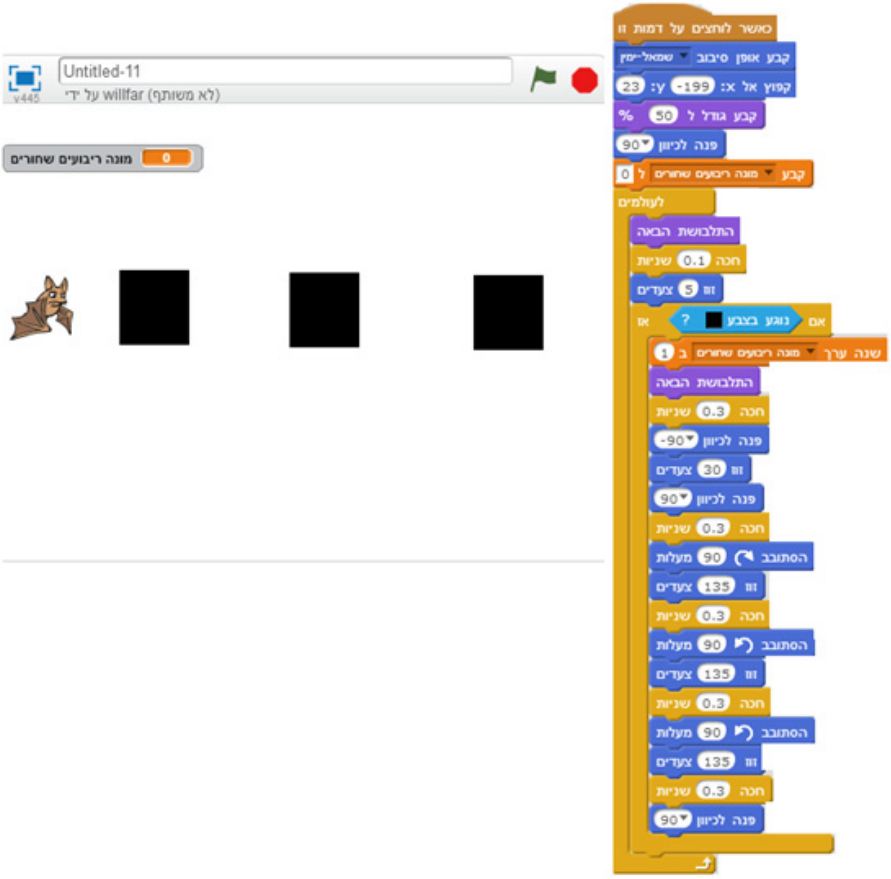
התסריט



מטלת דוגמה ב':

דוגמה זו מעט מורכבת ומטרתה להמחיש שימוש בתבנית מנייה. בהצגת הדוגמה ניתן לשנות הוראות שונות ולצמצם את התסריט על פי התאמתו לשיעור ולקבוצת הלימוד.

בדוגמה שלנו עטלף מוצב בקצה השמאלי של המסך. לפניו מכשולים בצורת ריבועים שחורים (הרקע של המסך). העטלף עף ימינה. כאשר הוא נתקל בריבוע שחור הוא מעדכן מונה ריבועים, כלומר מוסיף 1, עוקף את הריבוע השחור וממשיך במעופו. כאשר העטלף מגיע לקצה המסך אנחנו רואים שמספר המכשולים שהוא עבר (ריבועים שחורים) הוא 3.



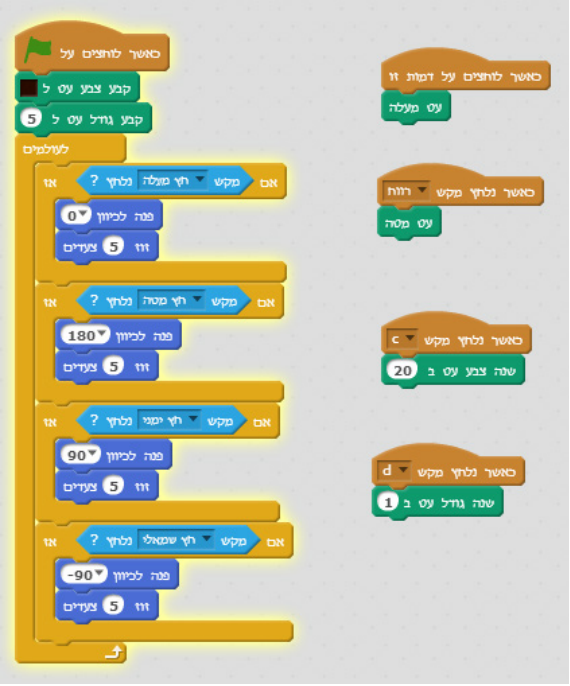
מטלות נוספות - מנייה:

מומלץ לתרגל עם התלמידים כתיבת תסריטים בסיסיים עם תבנית מנייה. הגדירו במטלות אלו את הקלט, הגדירו את מה שנדרש למנות ותרגלו עם התלמידים כתיבה של תסריטים אלו.

6. התלמיד ישלב הוראות מקבוצת הוראות שונות הכוללות שימוש בבלוקים: '**חזור**' '**לעולמים**', '**חזור_פעמים**', ושימוש בלולאות מקוננות, בתנאים בסיסיים ובתנאים מורכבים כדי ליצור יישומי משחקים, באנימציות, בסיפורים ועוד.

יישום תוכנת צייר

הפרויקט בדוגמה זו כולל תסריט מרכזי שיעדו הנעת דמות על המסך לכל הכיוונים, וקבוצת תסריטים קצרים המזומנים עם ההוראה של האירוע '**כאשר נלחץ מקש**' ועם ההוראה '**כאשר לוחצים על דמות זו**'.



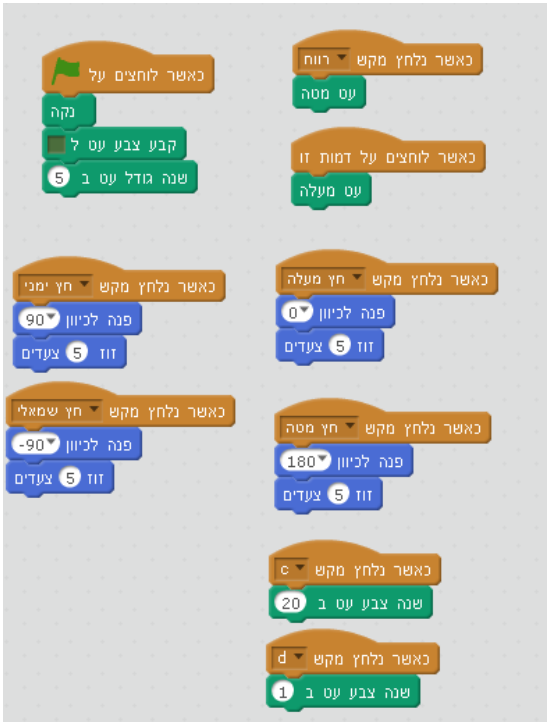
חשוב להציג דוגמה זו ולהמחיש שוב איך הפרדיגמה התכנותית EVENT DRIVEN PROGRAMMING של סקרטץ' מתבטאת ביישום.

התסריט המרכזי המופעל על ידי האירוע **'כאשר לוחצים על דגל ירוק'** מאפשר הנעת הדמות על המסך, זיהוי הכיוון וביצוע פנייה ותנועה לכיוון. התנועה מבוססת על משפט תנאי עם החיישן. 

(עם מקשי החצים).

היישום משנה את מאפייני העט עם הוראות האירועים : **'כאשר נלחץ מקש'** , **'כאשר לוחצים על דמות זו'.** הבהרת ההבדלים בין משפט תנאי **'אם'** בשילוב החיישן **'נלחץ מקש?'** לבין האירוע **'כאשר נלחץ מקש'** מצביעה על שילוב מעניין בתכנות בסביבת סקרטץ', שיש בו ממד של תכנות פרוצדורלי שמשולב עם תכנות מונחה אירועים. במכלול כזה ישנם תהליכים המתבצעים במקביל ונוצרת מורכבות. התרשימים והטבלאות שהצגתי מסייעים בעיצוב ובבנייה של היישום וביישומו בסביבת העבודה.

הדוגמה הבאה מציגה תסריט לביצוע אותה המטלה ללא שימוש בלולאה וללא שימוש בהתניה.



יישום – לומדים לספור

היישום משלב קבוצות פקודות מגוונת, נעשה בו שימוש בלולאת '**חזור_פעמים**' ונעשה בו שימוש נרחב בזימון תסריטים באמצעות מסרים.

תיאור כללי

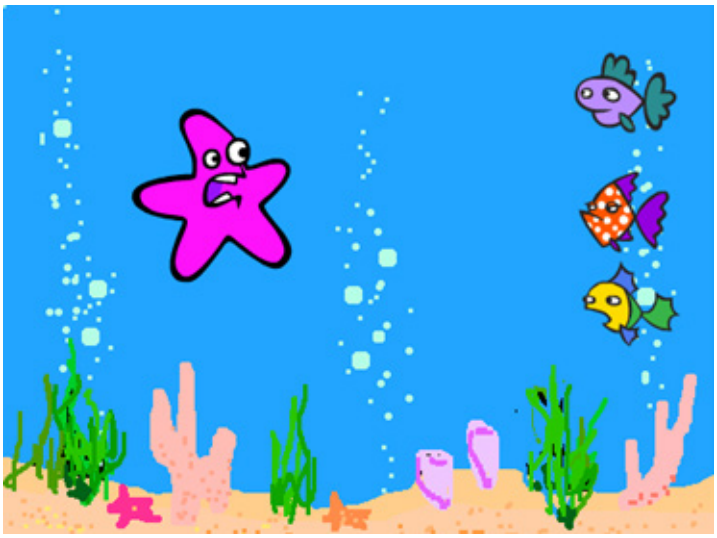
בלחיצה על הדגל הירוק מתחיל משחקון. כל משחקון מציג 3 דמויות על המסך. בהקלקה על הדמות עם העכבר המשחק בוחר דמות. לאחר שנבחרה דמות היא תוצג מספר פעמים לפי ערך אקראי בתחתית הבמה.

בשלב זה המשחק נדרש לענות על השאלה: כמה עותקים של דמות מופיעים בתחתית המסך?

תשובה נכונה מזכה במשוב חיובי, תשובה לא נכונה במשוב שלילי.

לחיצה על הדגל מתחילה משחקון חדש.

הפקודות הכלולות בתסריטים:



תיאור המשחק

על הבמה – כוכב ים שמנהל את ההצגה.

בצד הימני של הבמה – שלושה דגים.

המשחק בוחר את אחת מדמויות הדגים. הדמות מופיעה למטה למספר אקראי של פעמים. המשחק צריך לענות על השאלה כמה דגים הוא רואה בתחתית הבמה.

אם הילד המשחק מצליח הוא יזכה למשוב חיובי ואם לא הוא יוזמן ללחוץ על הדגל הירוק למשחקון נוסף.

בפרק לעיל דנו ברעיון תכנון 'מעלה-מטה' או תכנון 'מטה-מעלה'.
נשתמש באסטרטגית ניתוח המשימה לפי מודל 'מטה-מעלה'.

תכנון מטה-מעלה : משחק לומדים לספור

כוכב הים שמנהל את המשחק
מבצע בדיקה האם המשחק
הקליד את מספר הדגים הנכון

אם לוחצים על דמות דג
הדמות של הדג שנבחר מוחתמת
בתחתית הבמה, מספר פעמים
בתחום 1-10



תסריטי כוכב הים  שמנהל את המשחק:



תסריט האתחול כולל ניקוי מסך. נבחר את מספר הדגים שיוצגו על הבמה באופן אקראי, ולאחר סיום ההכנות ערך המשתנה 'אישור החתמה' נקבע ל-1, ומשמעו, מתן אישור לאחד משלושת הדגים שנבחר להחתים את דמותו. התיאום הזה בין התסריטים מושג על ידי ההוראה **'חכה עד ש'** שכלולה בתסריט של כל אחת מדמויות הדגים באופן הבא:



בסיום התהליך של ההחתמה הדג שמחתים שולח מסר 'דגים' לכוכב הים. זהו תיאום נוסף בין הדמויות לתסריטים של הדמויות. עתה כוכב הים יכול לשאול את השאלה: כמה דגים השחקן רואה?



תסריטי הדגים



לשלוש דמויות הדגים יש אותם תסריטים (יש שינוי בערכי המיקום על הבמה של כל דג), יוצגו התסריטים של דמות 'הדג הצהוב'.

תסריט האתחול (הכנות לפני משחק)



כל דג נשלח למיקום שלו על הבמה, בהפניית 'המבט' אל כוכב הים 'המנהל' את המשחק הגודל מוקטן ל-50%.

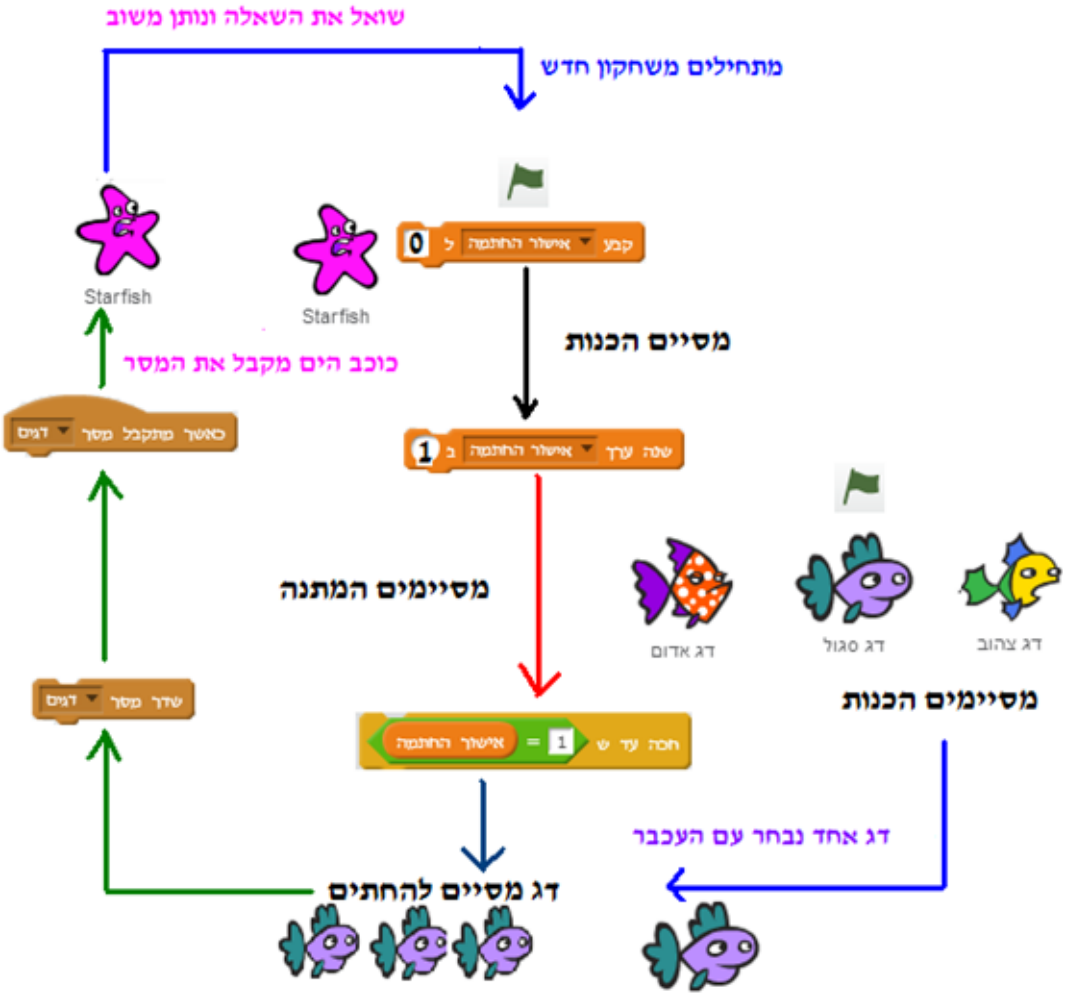
תסריט החתמה

רק הדג שנבחר באמצעות העכבר בוחר בצורה אקראית את מספר הדגים להצגה, קופץ אל נקודת התחלת ההחתמה על הבמה, מוריד את העט, ובאמצעות לולאת '**חזור**' מחתים את הדמות שלו לפי מספר הפעמים. דמות הדג שנבחר מסיימת ושולחת מסר 'דגים' לכולם. דמות כוכב הים היא המאזינה למסר, והתסריט של קבלת תשובה מהשחקן ובדיקתה מופעל.



סיכום

חשוב על ניהול ביצוע התסריטים, על התזמון של הפעולות באמצעות מסרים, על ההוראה '**חכה עד ש**', ועל המשתנה 'אישור החתמה'.



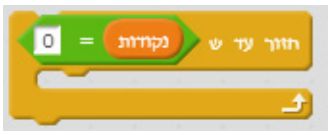
איור: תרשים תזמון אירועים

לולאת 'חזור עד ש'



ההוראה -

הוראה ליישום מבנה של לולאה (חזרה על תסריט מספר פעמים). בניגוד להוראה **'חזור'** שבה נקבע מספר הפעמים, הרי שבהוראה **'חזור עד ש'** הלולאה תתבצע עד שיתקיים תנאי כלשהו.



דוגמה

PICO שלנו משוטט ומחפש את הדמות GIGA שאינה נעה ועומדת במקום כלשהו על הבמה. PICO לא יפסיק לחפש את GIGA עד שימצא אותה.

אם PICO מתרוצץ על הבמה, אין לדעת בכמה זמן הוא ימצא את GIGA, לא נוכל לקבוע מספר ניסיונות (מחזורים) שפיקו יבצע – יתכן ש PICO ימצא את GIGA מיד ויתכן שימצא אותה לאחר צעידה ארוכה.

זה המקרה שיש להשתמש בלולאת **'חזור עד ש'**. נגיעה של PICO ב GIGA הוא האירוע שבו הלולאה תיעצר.



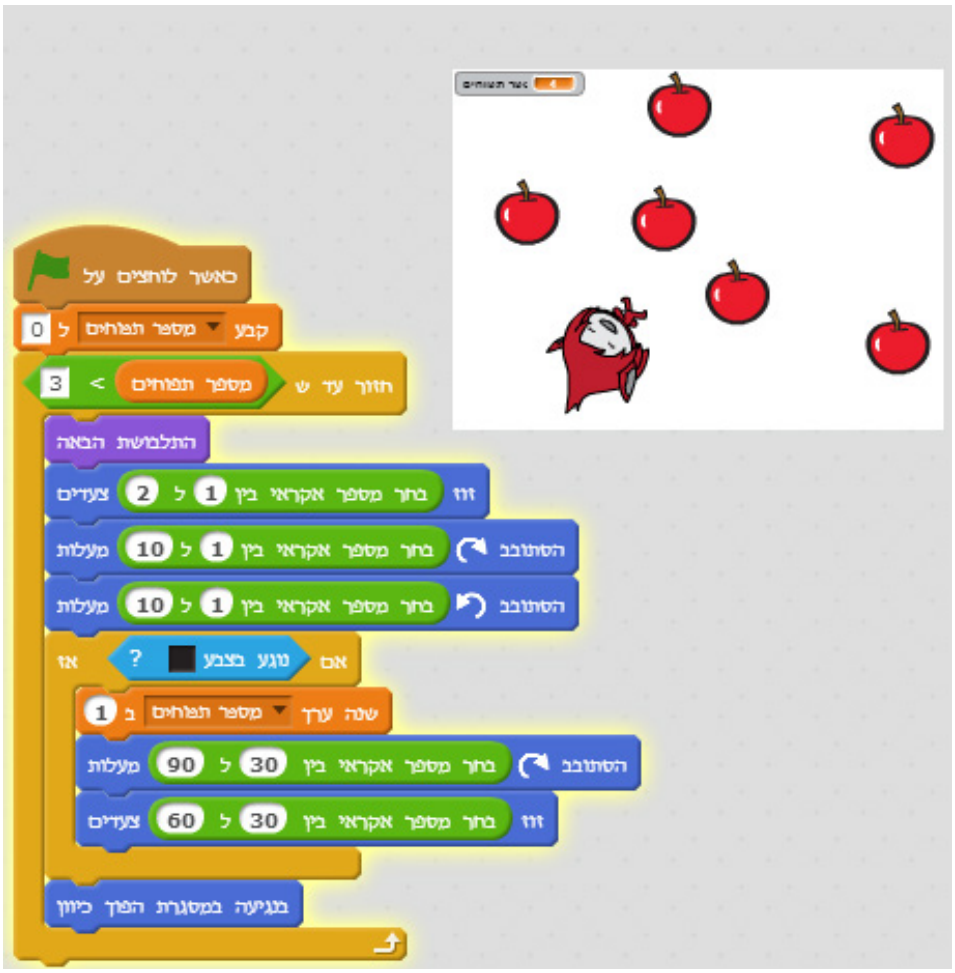
מטלות

א. לתסריט חדש בחר בדמויות: GIGA WALKING, PICO WALKING, צור לולאת '**חזור עד ש**' ותן לשתי הדמויות להתרוצץ על המסך עד שהן תפגשנה. הוסף את ההוראה '**בנגיעה במסגרת הפוך כיוון**'.

ב. שלב תבנית סכום בפתרון, שתחשב עבור כל דמות כמה צעדים הדמות ביצעה עד למפגש בין הדמויות.

דוגמה

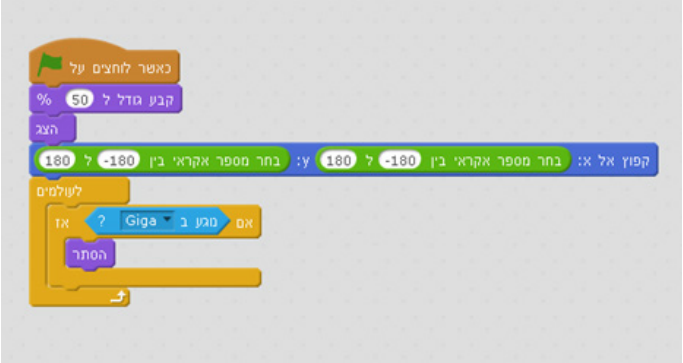
GIGA מתרוצץ על הבמה ומחפש תפוחים לאכול, כאשר GIGA יסיים לאכול לפחות 3 תפוחים הוא ייעצר. התסריט הבא מבצע את המשימה באופן לא מושלם (המטלה תהיה לשפר את הביצוע), התסריט משתמש בלולאה '**חזור עד ש**'.



שולבה בתסריט גם תבנית מנייה שמנתה כמה תפוחים אוכל גיגה, ותנאי העצירה הוא: 

מטלה

התנאי שבודק האם GIGA נוגע בתפוח הוא **'נוגע בצבע'**,  והסיבה היא שבמסגרת סביב כל דמות של תפוח יש פס שחור. כאשר GIGA נוגע בתפוח ונע מספר צעדים, לעיתים הוא נוגע באותו תפוח מספר פעמים ו'אוכל' אותו שוב. על מנת להסתיר תפוח ניתן להוסיף את התסריט הבא (ולשכפל דמות של תפוח מספר פעמים):



מטלה מסכמת לולאת 'חזור עד ש'

בנושא זה חשוב לתת מטלות מסכמות שכוללות שימוש בבלוקים מקבוצות שונות. מומלץ לשלב אנימציה, קולות וכדומה.

תיאור המטלה

שתי דמויות ניצבות על מסלול הריצה, שופט מזניק את הדמויות והדמויות שועטות קדימה, כאשר אחת הדמויות הגיעה לקו המטרה הדמות מודיעה לשופט שמכריז על הניצחון והדמות שהפסידה מודה בכישלונה.

הדמויות


Referee
השופט REFEREE

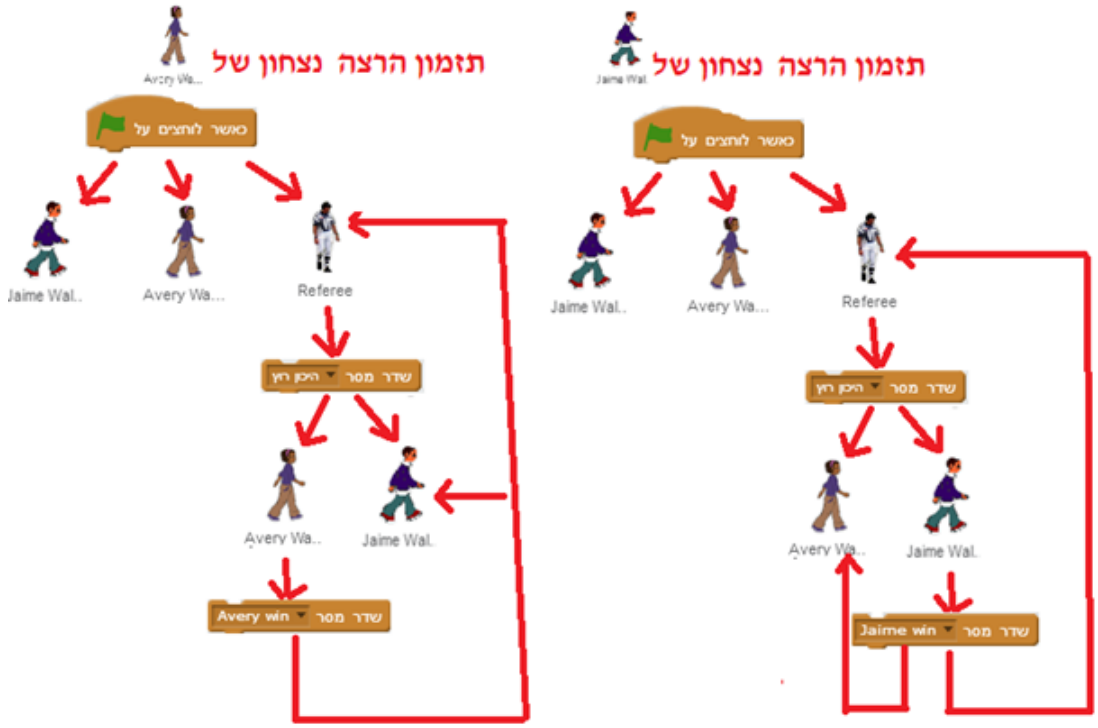

Avery Wa...
הרצה AVERY


Jaime Wal..
הרץ JAIME

טבלת תסריטים ודמויות

דמות / אירוע	כאשר לוחצים על	כאשר Avery win מתקבל מסר	כאשר Jaime win מתקבל מסר	כאשר היכן רץ מתקבל מסר
Jaime Wal..	כאשר לוחצים על קבע גודל ל 70 % קפוץ אל x: -201 y: 105	כאשר Avery win מתקבל מסר אמור אני אנצח במרוץ הבא למשך 0.5 שניות עצור הכל		כאשר היכן רץ מתקבל מסר חזור עד ש נוגע בצבע ? זז בחר מספר אקראי בין 3 ל 8 צעדים התלבושת הבאה חכה 0.2 שניות שדר מסר Jaime win
Avery Wa...	כאשר לוחצים על קבע גודל ל 70 % קפוץ אל x: -208 y: -18		כאשר Jaime win מתקבל מסר אמור אני אנצח במרוץ הבא למשך 0.5 שניות עצור הכל	כאשר היכן רץ מתקבל מסר חזור עד ש נוגע בצבע ? זז בחר מספר אקראי בין 3 ל 8 צעדים התלבושת הבאה חכה 0.2 שניות שדר מסר Avery win
Referee	כאשר לוחצים על קבע גודל ל 40 % קפוץ אל x: 167 y: 129 אמור צאו לדרך למשך 2 שניות שדר מסר היכן רץ וחכה	כאשר Avery win מתקבל מסר אמור Avery won the race למשך 4 שניות	כאשר Jaime win מתקבל מסר אמור WOO..... Jaime won the race למשך 4 שניות	

תרשים תזמון אירועים



הצגת תרשים תזמון תסריטים חשובה ביותר. כאן אנו מביאים לידי ביטוי את המאפיינים החשובים והמרכזיים של שפת סקרטץ' – שילוב של טבלת התסריטים, של תרשים תזמון תסריטים ושל תרשמי תכנון מעלה-מטה /מטה-מעלה – כל אלו הן מיומנויות תכנון וארגון חשובות ביותר לא רק בהיבט של כתיבת קוד בסביבת EVENT DRIVEN PROGRAMMING, אלא בכל פרדיגמת תכנות ובכל תהליכי תכנון מדעיים הנדסיים.

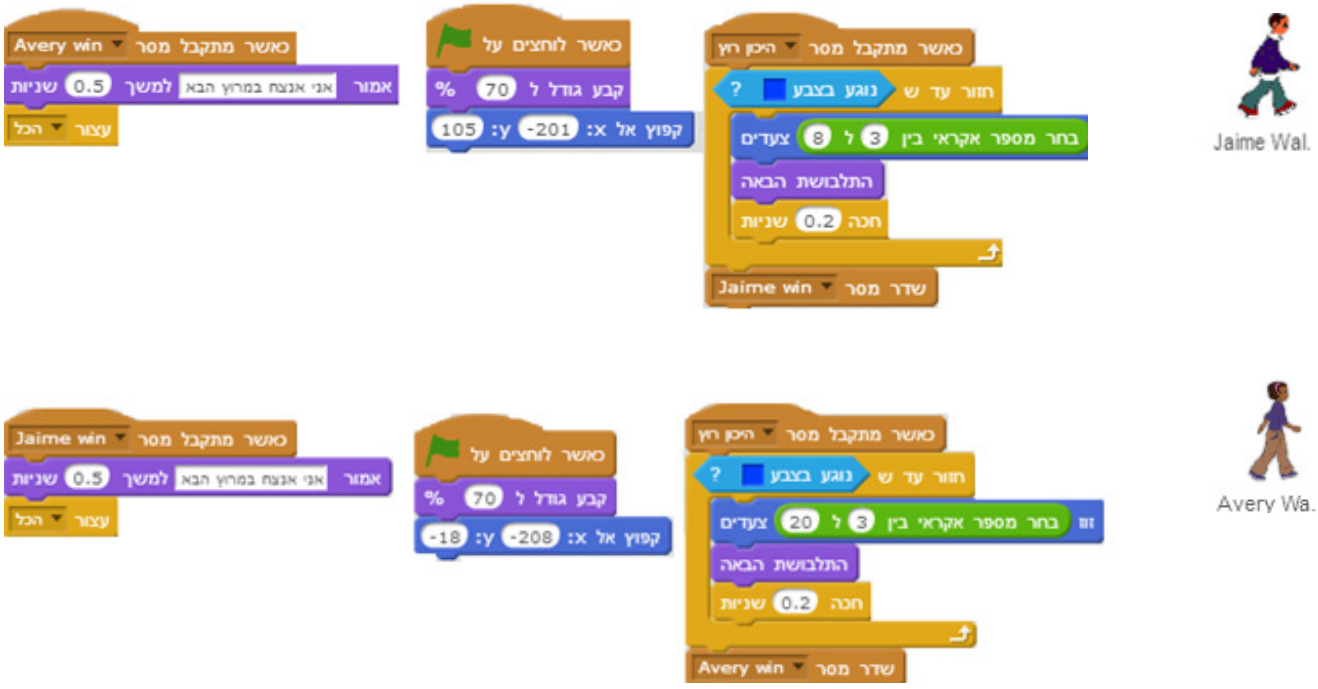
כאן הוצגו שני תרשימי תזמון תהליכים – תרשים לכל מהלך תזמון של ניצחון של כל אחת מהדמויות. בדוגמה זו אפשר לבקש מהתלמידים לשנות את מהירות הרצים כך שרץ מסוים יגיע ראשון, ולראות איך תזמון התסריטים משתנה.

הבמה



תסריטי הדמויות





לולאות מקוננות

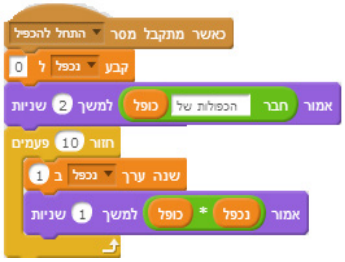
במבנה הבקרה לולאה (כל סוגי הלולאות) אנחנו מכניסים בלוקים – הוראות. ניתן לשלב בלוק של הוראת לולאה בדומה לשילובו של כל בלוק אחר. כשמשלבים לולאה בתוך לולאה מכנים זאת **'לולאה מקוננת'**. חשוב מאוד שתלמידים יבינו מה המשמעות של שילוב הוראות לולאה (מכל הסוגים).

דוגמה:

דמות א': ניצבת במרכז המסך ואומרת את המספרים מ-1 עד 10

דמות ב': ניצבת בקצה השני של המסך ואומרת – לאחר כל אמירה של דמות א' – את כל הכפולות של המספר שנאמר על ידי דמות א' 1-10.

טבלת תסריטים ודמויות

דמות / אירוע	כאשר לוחצים על	כאשר מתקבל מסר התחל להכפיל
 Giga		
 Giga walk...		

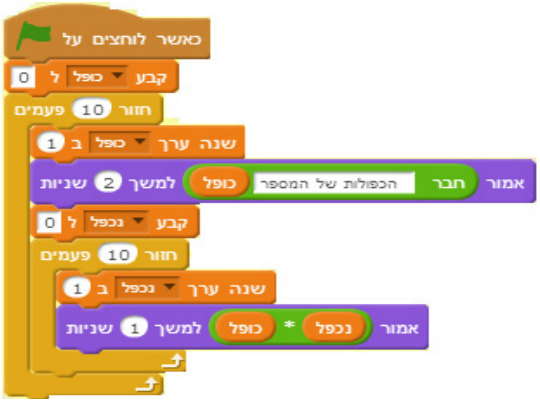
תרשים תזמון אירועים



מתבצע 10 פעמים

אנחנו רואים לולאה מרכזית שמבצעת הדמות GIGA, שבכל מחזור 'מפעילה' את הדמות GIGA WALK , שמבצעת לולאת חישוב מכפלות 1-10 בערך הכופל. שים לב: הפקודה **'שדר מסר וחכה'** מאפשרת את סיום התסריט של הדמות שמופעלת עם המסר. אין כאן לולאה מקוננת!

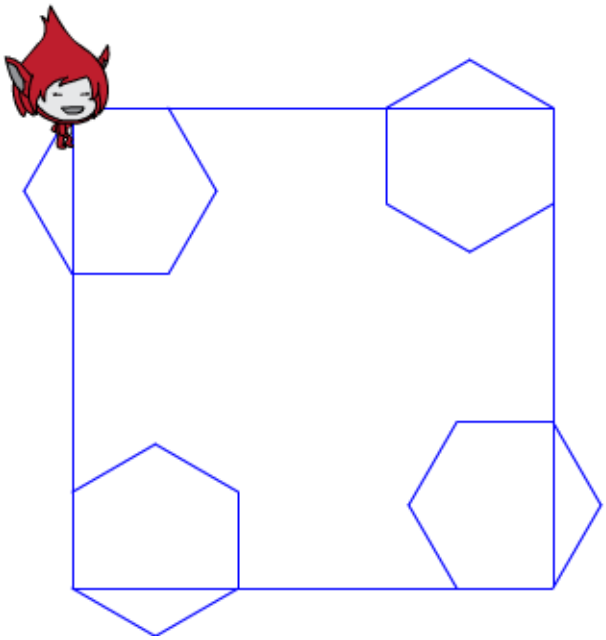
התסריט הבא מציג מהי לולאה מקוננת: הדמות GIGA החליטה להיות "עצמאית" ולבצע את חישוב לוח הכפל ללא סיוע של הדמות GIGA WALK. היא תעשה זאת באמצעות קינון לולאות, והתסריט עתה יהיה:



התסריט הזה מבצע את אותה מטלה של חישוב לוח הכפל באמצעות לולאה מקוננת.

מטלה דוגמה – לולאות מקוננות

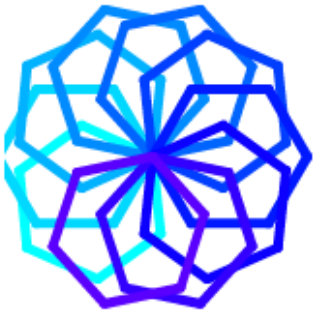
הדמות GIGA WALK מבצעת מסלול סיבוב סביב ארבע צלעות מסגרת הבמה. בכל תנועה שהדמות מבצעת לאורך או לרוחב הבמה הדמות מבצעת סיבוב של 6 צלעות, וכך אנו מקבלים את הציור הסופי הבא על המסך:





השתמשנו בלולאה מקוננת לביצוע הציור הזה שנוצר תוך כדי תנועת הליכה של GIGA WALK.

לולאות מקוננות יודגמו טוב יותר באופן גרפי מאשר באמצעות אלגוריתמים לחישוב כל שהוא. בתסריט הבא יודגם ויוצג ציור שנוצר באמצעות לולאה מקוננת. אין חשיבות לדמות המבצעת את הציור, ולכן דמות זו עוברת למצב הסתר בסיום הציור.





המלצות! הציגו בפני התלמידים תסריטים בהם הדמויות במצב **'עט מטה'**. כשיוצרים לולאות מקוננות מקבלים פלט גרפי נאה שגם ממחיש שאכן התבצעה לולאה מקוננת. הכנסת השהייה בשניות מציגה את התהליך בצורה מיטבית. התלמידים יכולים לקחת תסריטי בסיס אלו ולשלב קלט ומשתנים.

משחק – הדוב יוגי מלקט בננות

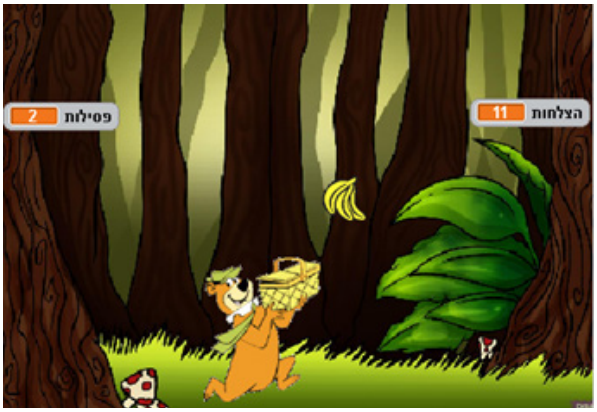
מטרה

עליך לכתוב את המשחק **'יוגי מלקט בננות'**

דמויות ודרישות

כללי

- תחילת המשחק: לחיצה על מקש רווח
- המשחק כולל שני משתנים: 'הצלחות' ו'פסילות'



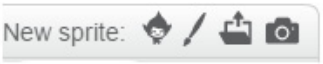
דמות 1: יוגי הדוב

- הזזת יוגי ימינה ושמאלה: לחיצה על חץ ימני או שמאלי
- כאשר יוגי נוגע במסגרת הוא הופך כיוון

דמות 2: בננות נופלות

- כל פעם נופל אשכול בננות ממקום אקראי בקצה העליון של המסך. הנפילה היא בתנועה ישרה כלפי מטה
- **הצלחה:** אם יוגי מצליח להכניס את האשכול לתוך הסל – מגדילים את מונה ההצלחות ב-1
- **פסילה:** אם יוגי פספס והאשכול נפל על הרצפה (כלומר: נגע במסגרת) – מגדילים את מונה הפסילות ב-1

הדרכה

- יצירת דמויות:
  - חיפוש תמונה באינטרנט ושמירתה על המחשב
  - מתוך סקרטץ': יבוא דמות חדשה ע"י: 

ובחירת הקובץ ששמרתם
- נפילת אשכול בננות ממקום אקראי למעלה:
  - ערך ה-Y קבוע (תמיד מלמעלה!)
  - ערך ה-X אקראי: שימוש במפעיל 

- זיהוי האם אשכול הבננות נוגע בסל: למשל ע"י חיישן 

ובדיקה האם נוגע בצבע הסל של יוגי
- זיהוי האם אשכול הבננות נוגע במסגרת: ע"י חיישן 

במת המשחק



דמויות במשחק


יוגי


בננה

תסריטי הדמויות


יוגי

תסריט מרכזי שמתחיל עם האירוע 'כאשר נלחץ מקש רווח'. לולאה אינסופית ובדיקה האם נלחצו מקש חץ ימין או מקש חץ שמאל. כל לחיצה על מקש חץ מזיזה את יוגי אופקית בתחתית הבמה לכיוון המקש שנלחץ. בנגיעה במסגרת יוגי הופך כיוון.





בתסריט המרכזי של הדמות: שתי לולאות יוצרות את המרכיב המרכזי של פעילות הדמות -

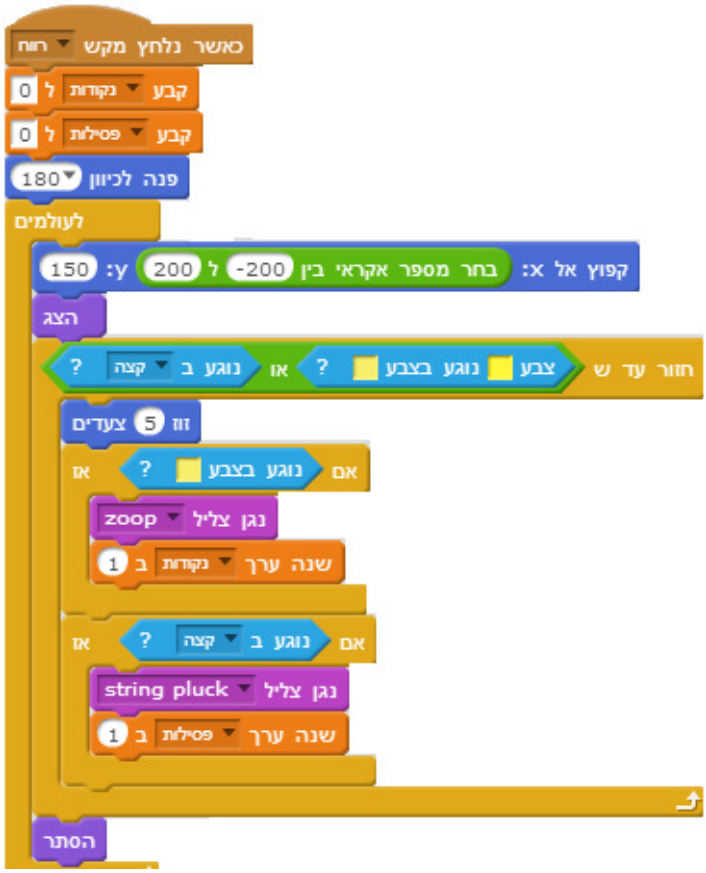
(1) לולאה אינסופית שמציבה בכל פעם מחדש את דמות הבננה בחלק העליון של הבמה במיקום אופקי (x) אקראי, כך שהבננה מופיעה באופן 'מפתיע' במקומות שונים.

(2) לולאה פנימית, לולאת '**חזור עד ש**'. לולאה זו מותנת על ידי תנאי: או בדיקה האם הבננה נגעה בסל של יוגי (צבע צהוב) או בדיקה האם הבננה נגעה בקצה (בתחתית הבמה). אם הבננה נגעה בסל (צבע צהוב), יש צליל של עידוד והערך 1 מתווסף למשתנה 'נקודות'. אם הבננה נגעה בקצה (בתחתית הבמה) יש צליל אכזבה על כך שיוגי לא תפס את הבננה. סיום הלולאה הפנימית, חזרה ללולאה החיצונית, והבננה מתחילה מחדש את המסע מלמעלה למטה.

מטלות

א. קבע את סיום המשחק – כאשר יש למעלה ממספר פסילות מסוים.

ב. חשוב על הוספת רמה: עם שינוי ברקע, בהוספת פירות שנופלים, עם שיבוטים של הבננה ועוד.



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
רעיון הלולאה, דיון, פעילות ללא מחשב	שעות אלו יוקדשו לפעילות של חשיבה ללא מחשב, יש לתת לתלמידים מטלות שחוזרות על עצמן לפי מספר פעמים ידוע מראש. למשל עמוד ושב 5 פעמים. ניתן לתת מטלה עם למידה מטרימה: ילד מחזיק דף ולפני ביצוע המטלה כותב על הדף, לתלמיד אחר, את מספר הפעמים שעליו לחזור על פעולה מסוימת ועוקב אחר הביצוע. מגוון פעולות יצירתי יאפשר להטמיע את הרעיון המורכב של מבנה לולאה.	✓		התלמידים יתנסו בפעילות שונות ויצליחו למלא הוראות מילוליות לביצוע חוזר של פעולות מספר ידוע מראש של פעמים. **2 שעות**
כתיבת אלגוריתמים ומימושם עם לולאות **'חזור_פעמים'**, **'חזור עד ש'**	חשוב לתרגל בשעות אלו את התהליך של כתיבת אלגוריתם מילולי ומימושו בקוד. יש להדגיש את ההיבט הטכני והמיומנויות של כתיבת אלגוריתם מילולי, לצד תהליך המימוש בסביבת העבודה.	✓	✓	התלמידים יתרגלו מטלות הכוללות כתיבת אלגוריתם מילולי עם מבנה של לולאה ומימושו בסביבת העבודה. **2 שעות**
מימוש אלגוריתמים עם לולאות	יש לתת מטלות העושות שימוש במרכיבים הגרפים של סביבת העבודה, זה יותר מהנה וגם מימוש גרפי של לולאה, יותר מאשר פלט חישובי, מקל על הבנת הלולאה. אפשר לשלב מרכיב של קול ומוסיקה ועוד.	✓	✓	התלמידים יבצעו מטלות הכוללות לולאות בשילוב הוראות קבוצת תנועה, והוראות קבוצת צלילים. **2 שעות**
מטלות עם לולאות	חשוב לתת גם מטלות ודוגמאות שכוללות את מרכיב המנייה של **'אירוע'**.	✓	✓	התלמידים יבצעו מטלות מהסוגים הבאים: כתוב תסריט אשר ידפיס את כל המספרים הזוגיים 0-100 כתוב תסריט שידפיס את הכפולות של 6 מ 1 עד 10. **2 שעות**

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
לולאות מקוננות	חשוב לחבר את הרעיון והיישום של המשפטים המקוננים **'אם'**, **'אם ואם לא'** עם רעיון קינון לולאות. בתכנית הוצגה דוגמה של לוח הכפל בגרסה פשוטה (רק לולאות בסיס) ובגרסה גרפית. נושא קינון לולאות אינו פשוט למעקב וכל המחשה גרפית או של צלילים תסייע לדרכי מימושו.	✓	✓	הבנה ומיומנות מעקב אחר ביצוע לולאה מקוננת. כתיבת תסריטים בסיסיים עם לולאות מקוננות. **2 שעות**
תזמון תהליכים	תכנות בסביבה מונחת אירועים אינו פשוט לעיצוב, לתכנון ולמימוש, וגם למעקב אחר הביצוע ולמציאת שגיאות. בתכנית הלימודים הוצגו שלושה כלים לתמיכה בכתיבה ובמעקב. חשוב לתרגל ולהשתמש בהם עם דוגמאות שונות. חשוב: לבחון יחד עם התלמידים את הפרויקטים המסכמים שהוצגו בפרק, להרחיב ולשפר פרויקטים אלו, לתכנן משחקים, ליצור אנימציות, לשלב בין הדמיות שקשורות לנושאים במדעים, במתמטיקה או פשוט במשחק מהנה.	✓	✓	**4 שעות**

פרק יג' פרוייקטים ותרגול מסכם

בפרק זה נציג ונדון במטלות ופרויקטים שונים. הפרויקטים יוצגו ברמות שונות ובשלבים שונים של פיתוח.

המטרה המרכזית בפרק זה היא לרכוש וליישם מיומנויות המאה 21, מיומנויות 'הנדסיות' אותן הצגנו בשער הראשון, כולל אלגוריתמים מילוליים; גישות תכנון: 'מעלה-מטה' ו'מטה-מעלה'; תרשים דמויות ותסריטים; תרשים תזמון תסריטים. בפרק זה יוצגו מספר פרויקטים ברמות שונות. כל פרויקט הוא רעיון ובסיס להרחבה וגם נקודת יציאה לפרויקטים נוספים, צריך רק לדמיין, לתכנן, לתכנת ולשתף .

היקף השעות הנדרשות בכיתה ליצירה משותפת של פרויקט הוא כ- 8 שעות עבודה, מאוד חשוב שלפחות 4 שעות יהיו מוקדשות לתכנון וכתיבה ללא מחשב, ופחות ממחצית הזמן לתכנות, הרצה, בדיקה ותיקון. מאוד חשוב לעבוד בצוות. ראוי לבחון עבודה בצוות עם תלמידים מבית ספר אחר בארץ, אולי בעולם. חשוב שכל תלמיד יתנסה גם בעיבוד גרפי של דמויות, במרכיב המוזיקה והצלילים, בתכנון וכתיבה אלגוריתמית ובתכנות קוד.



פרויקט פקמן - PackMan

פקמן הוא חלק מקבוצת המשחקים הנקראים משחקי ארקייד. משחק ארקייד הוא מכונת משחק שמופעלת על ידי מטבעות או אסימוני משחק, ולרוב היא מותקנת במקומות ציבוריים כמו משחקיות הווידאו שהיו נפוצות מאוד בשנות ה-80.

פקמן, הוא אחד המשחקים המצליחים ביותר בהיסטוריה של משחקי המחשב. המשחק פותח בתחילת שנות ה-80. בשנים הראשונות המשחק היה משחק ארקייד בלבד ובהמשך המשחק עבר לפלטפורמות מגוונות: מחשבים, טבלטים, טלפונים חכמים ועוד. בפרויקט זה נציג את המשחק – הדמויות המרכזיות ועיקרי הרעיונות שיהיו בבסיס היישום שיאפשר לנו לכתוב ולשחק במשחק שהוא חלק מרכזי בהיסטוריה של משחקי המחשב.

תיאור המשחק

המשחק 'פקמן' מתרחש בתוך מבוך גדול וכחול שברובו מלא בנקודות צהובות. מלבד הנקודות הצהובות נמצאות גם חמש דמויות: פקמן – כוכב המשחק, יצור צהוב ורעב; ארבע מפלצות צבעוניות (או 'רוחות'): אינקי, בלינקי, פינקי וקלייד. בלוח המשחק מפוזרים בונוסים שונים המאפשרים צבירת נקודות כגון דובדבנים. פקמן נע במבוך הלוח ו'אוכל' את הנקודות הצהובות ואת הבונוסים, כשהוא נזהר לא לפגוש את הרוחות – 3 מפגשים עם הרוחות מסיימים את המשחק.

רמות במשחק

אם פקמן מסיים לאכול את כל הנקודות הצהובות, המשחק עובר לרמה הבאה שהיא כמובן קשה יותר.

אתרים למשחק

קיימים אתרים רבים בהם אפשר להתנסות ולשחק במשחק. אפשר לשחזר ולחשוב אלו מרכיבים יהיו במשחק שאתם יוצרים – אי אפשר ליצור את כל המשחק, זה תהליך מאוד מורכב שדורש ידע שהוא מעבר לתכנית הלימודים ומצריך עשרות ואולי מאות שעות של עבודה.

לדוגמה אתרים למשחק פקמן על מחשב:

HTTP://GAMES.YO-YOO.CO.IL/GAMES_PLAY.PHP?GAME=140

HTTP://GAMES.YO-YOO.CO.IL/GAMES_PLAY.PHP?GAME=4284

תכנון המשחק



אנו נפשט את המשחק כך שהוא יהיה עם המאפיינים הבאים:

לוח משחק – רקע

דמות פקמן

40 דמויות – נקודות אדומות

3 דמויות – מפלצות

יצירת לוח המשחק

נייבא תמונה של לוח המשחק ובאמצעות תוכנת הצייר או העורך הגרפי ניצור את הלוח הבסיסי של המשחק (הצגת לוח המשחק בהמשך).

עתה נעצב את דמות פקמן הצהוב. הדמות תכלול 4 תלבושות שמאפשרות את הדמיית הפה שנפתח ונסגר.

התכנות הבסיסי שנתכנן הוא עבור דמות הפקמן

תנועה:

באמצעות 4 החצים

בדיקות:

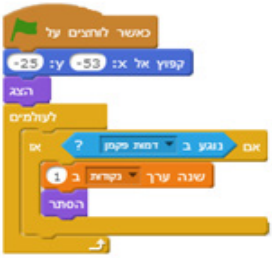
האם נוגע בגבולות המסלול (הפסים הכחולים), אם כן לא לאפשר המשך תנועה.

מנגנון 'אכילת' הנקודות האדומות ומנגנון אכילת הבונוסים הוא מנגנון מורכב. אי אפשר לשלב בתכנות של פקמן בדיקה עבור כ- 40 נקודות אדומות ו-5 בונוסים כי משמעות הדבר היא 40 תנאים לבדיקה באיזו נקודה אדומה נגעה דמות הפקמן. לכן, בכל מה שקשור לבדיקות האלו נשלב בדיקה של הדמויות שפקמן נוגע בהן.

התסריט הבסיסי של דמות הנקודה האדומה:

1. 'קפוץ' למיקום התחלתי
2. 'הצג'
3. 'לעולמים'
   - 3.1 'אם נוגע ב' פקמן
   - 3.2 נקודות + 1 ==> נקודות
   - 3.3 'הסתר' דמות

תסריט





כל התסריטים של הנקודות האדומות זהים מלבד הוראת קפוץ למיקום התחלתי ששונה בין דמות לדמות.

התסריט הבסיסי של דמות פקמן

1. נקודות < == 0
2. **'לעולמים'**
   - 1.1 התלבושת הבאה
   - 1.2 **'אם לא נוגע'** בצבע כחול וגם מקש חץ מטה נלחץ
     - 1.2.1 **'פנה לכיוון'** 180
     - 1.2.2 **'זוז'** 1 צעדים
     - 1.2.3 **'אם נוגע'** בצבע כחול // גבול המסלול
       - 1.2.3.1 **'זוז'** אחורה 2 צעדים
   - 1.3 **'אם לא נוגע'** בצבע כחול וגם מקש חץ מעלה נלחץ
     - 1.3.1 **'פנה לכיוון'** 0
     - 1.3.2 **'זוז'** 1 צעדים
     - 1.3.3 **'אם נוגע'** בצבע כחול // גבול המסלול
       - 1.3.3.1 **'זוז'** אחורה 2 צעדים
   - 1.4 **'אם לא נוגע'** בצבע כחול וגם מקש חץ ימינה נלחץ
     - 1.4.1 **'פנה'** לכיוון 90
     - 1.4.2 **'זוז'** 1 צעדים
     - 1.4.3 **'אם נוגע'** בצבע כחול // גבול המסלול
       - 1.4.3.1 **'זוז'** אחורה 2 צעדים
   - 1.5 **'אם לא נוגע'** בצבע כחול וגם מקש חץ שמאלה נלחץ
     - 1.5.1 **'פנה לכיוון'** 90-
     - 1.5.2 **'זוז'** 1 צעדים
     - 1.5.3 **'אם נוגע'** בצבע כחול // גבול המסלול
       - 1.5.3.1 זוז אחורה 2 צעדים

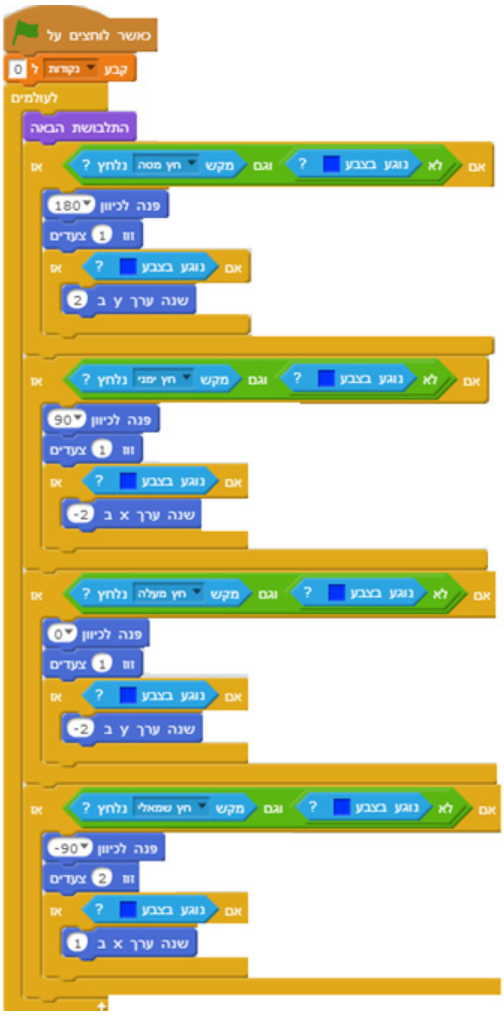
זה האלגוריתם הבסיסי עבור כל כיוון של תנועה לפי מקשי החצים.

המלצות לשיפור תנועות פקמן:

א. ניתן לשלב צלילים בתנועה

ב. ניתן לשלב צליל מיוחד כאשר פקמן 'אוכל' נקודה אדומה

תסריט בסיסי פקמן – תנועה ו"אכילת" נקודות אדומות



המלצות והצעות להמשך פיתוח המשחק

א. השלמה ל- 40 דמויות של נקודה אדומה, זכור יש רק לשנות את מיקום הנקודה בתסריט.

ב. הוספת 4 דמויות של בונוסים: לבחור 4 פירות, למקם כל דמות במיקום אחר. בתכנות הבסיסי במפגש עם דמות הפקמן הפרי נאכל והניקוד עולה.

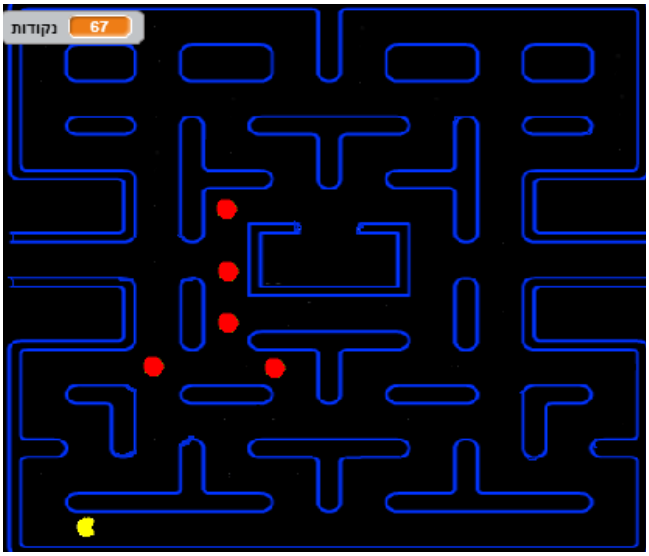
ג. יצירת דמויות אינקי, בלינקי, פינקי וקלייד – ה'רוחות': יש לתכנן את התנועה של הדמויות שאינן נשלטות על ידי המשחק אלא נעות עצמאית. ניתן ליישם את התנועה העצמאית בדרכים שונות, נציע שתיים:

- תנועה קבועה לאורך ציר אנכי או ציר אופקי. יתרונה הוא הפשטות וחסרונה שהיא צפויה מראש.
- כל דמות נעה 'עד ש' נוגעת בכחול (גבול התנועה), הדמות משנה כיוון על ידי שימוש בהוראה 'הסתובב 90 מעלות' וממשיכה לנוע. אם הכיוון אינו פנוי שוב עליה להסתובב 90 מעלות. כך הדמות מוצאת דרך להמשיך ונעה בצורה עצמאית.

ד. מפגש בין פקמן לדמות של אחת הרוחות 'מסיים' את המשחק עבור פקמן.

ה. במשחק פקמן המקורי נוספת נקודה אדומה 'גדולה' ו'אכילתה' נותנת, למשך כמה שניות, הגנה לפקמן מפני ה'רוחות'. באותן השניות פקמן יכול גם 'לאכול' את הרוחות ולזכות בנקודות בונוס, יצירת הדמויות והתסריטים שלהן בכדיי שתתאפשר תכונה זו של המשחק, אינה פשוטה כלל.

ו. סיום "אכילת" כל הדמויות ומעבר לשלבים הבאים. זו תכונה מרכזית אבל נדרשות עבודת עיצוב ותכנות של שעות ארוכות. אפשר להסתפק בשלבי הרחבה א'-ד' ולמחפשי אתגר נוסף שלב ה'.



פולשים מהחלל – Space Invaders

פולשים מהחלל הוא משחק ארקייד שציין את תחילת עידן משחקי המחשב המודרניים. עם צאתו לשוק בשנת 1978 הפך ללהיט בזכות עליונותו על משחקי מחשב קודמים שהיו דלי גרפיקה ועם אפשרויות תוכן מועטות יותר.

תאריך הפצה ראשוני: 1978 **מפתח:** TAITO CORPORATION

אנחנו נפתח גרסת סקרטץ ' מצומצמת:

תיאור המשחק

משחק פולשים מהחלל הופיע בגרסאות שונות. כיום יש משחקים רבים בהם בני אדם נלחמים בחייזרים וכמעט תמיד מנצחים. כאן בפרק על פרויקטים מסכמים אנחנו כותבים את המשחק ועדיין נשאיר אתגרים רבים להרחבה ושיפור.

משתנים

יש רק משתנה יחיד במשחק – משתנה המונה את מספר החלליות המושמדות.

רמות במשחק

יש רמת משחק אחת. כמובן שאחד האתגרים זה לפתח רמות נוספות.

אתרים למשחק

קיימים אתרים רבים בהם אפשר להתנסות ולשחק במשחק. אפשר לשחק ולחשוב אלו מרכיבים יהיו במשחק שאתם יוצרים.

דוגמה לאתרים למשחק פולשים מהחלל בסביבת סקרטץ':

HTTPS://SCRATCH.MIT.EDU/PROJECTS/14175443/

HTTPS://SCRATCH.MIT.EDU/PROJECTS/1055139/

מדריך ב- YOUTUBE למשחק

HTTPS://WWW.YOUTUBE.COM/WATCH?V=G0DBPI84VVI

תכנון המשחק

נפשט את המשחק וניתן לו את המאפיינים הבאים:

לוח משחק – רקע – תמונה שהורדה מהרשת: יש המון תמונות חופשיות לשימוש ואפשר לשנות את הרקע של המשחק.

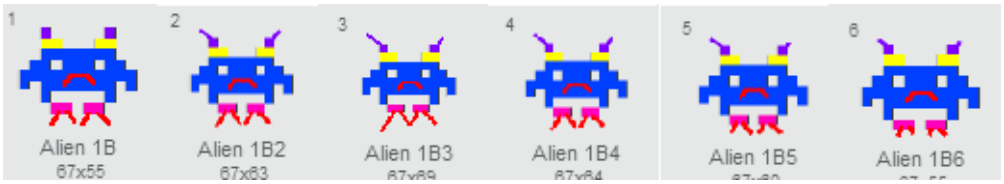
דמות חללית פולשת



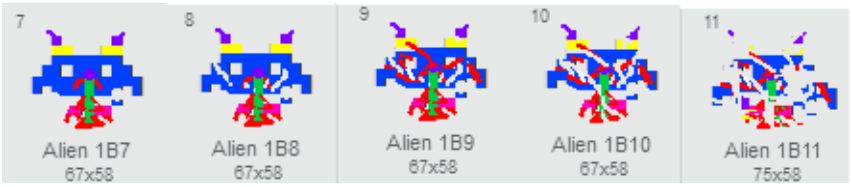
יש במשחק 6 חלליות, התסריט של כל החלליות זהה וההבדל הוא במיקום ההתחלתי של החללית.

לחללית 7 תלבושות כאשר החללית תקינה וטסה באין מפריע, ו- 5 תלבושות להדמיית פיצוץ החללית.

תלבושות חללית תקינה:

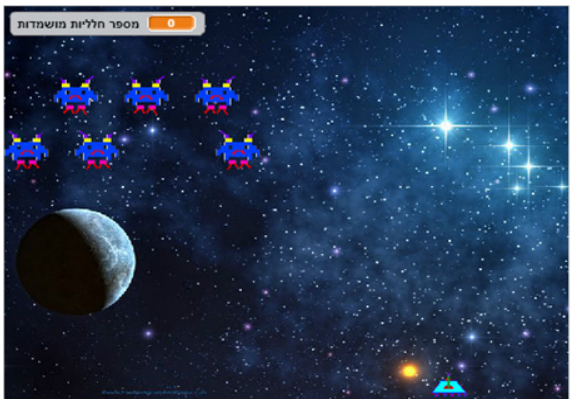


תלבושות חללית שנפגעה ומתפוצצת:



יצירת לוח המשחק

בחוברת יובאה תמונה של חלל, ניתן לייבא תמונת רקע אחרת, ואפילו תמונת רקע נוספת ולהחליף את הרקע תוך כדי המשחק לפי אירועים שמתרחשים.



תסריט דמות חללית האויב

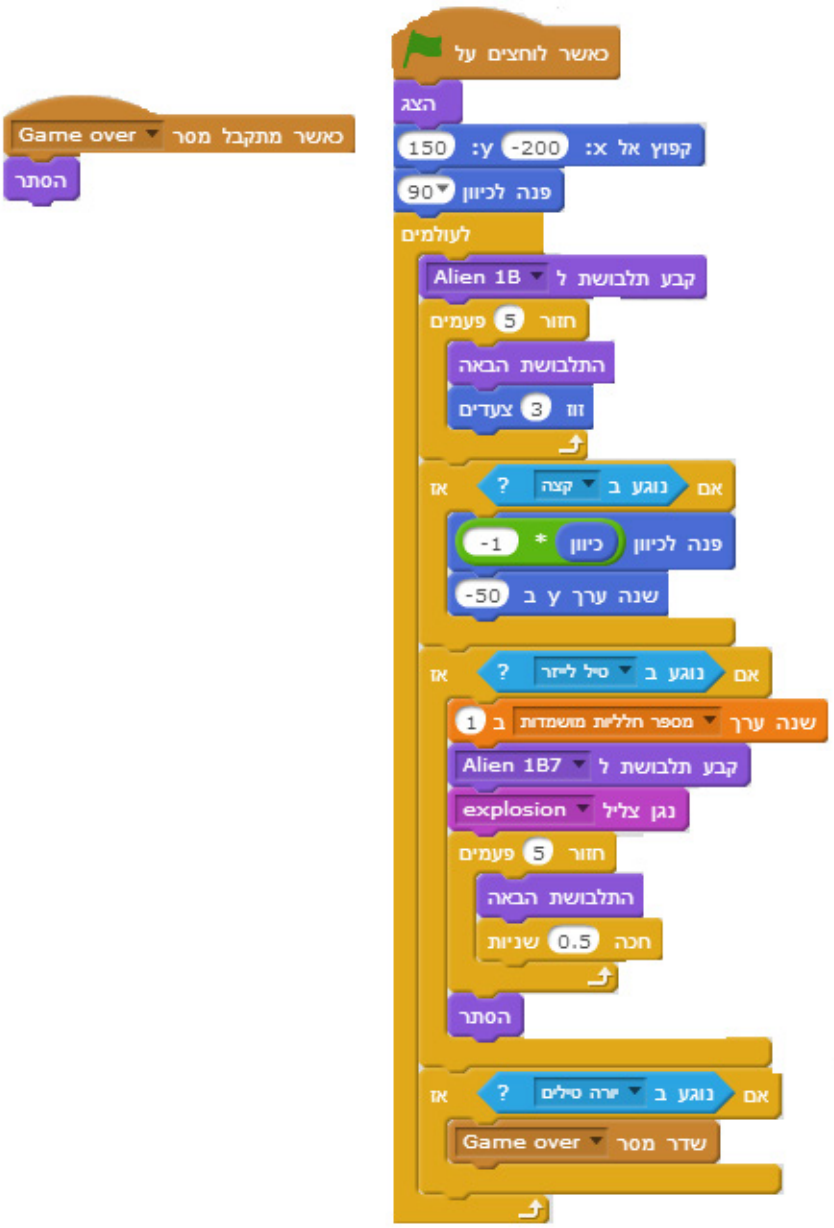
רעיון כללי: כל החלליות עם אותו תסריט, השוני הוא במיקום ההתחלתי.

אלגוריתם רעיוני:

1. מקם את החללית במיקום מסוים על הבמה
2. פנה לכיוון התנועה
3. **'לעולמים'**
4. שנה תלבושת
5. **'אם נוגע ב'** קצה שנה כיוון
6. **אם** יש פגיעה מטיל הלייזר – התפוצצי , עדכון מספר חלליות מושמדות
7. **אם** נוגע ביורה הטילים של אנשי כדור הארץ, ניצחון לפולשים

אלגוריתם מפורט

1. **'הצג'** את החללית על המסך
2. **'מקם'** במיקום שנקבע לחללית
3. **'פנה לכיוון'** 90 – כיוון התנועה בתחילת התסריט
4. **'קבע תלבושת'** ראשונה
5. **'חזור'** 5 פעמים
   - 5.1 **'עבור לתלבושת הבאה'**
   - 5.2 **'זוז'** 3 צעדים
6. **'אם נוגע'** בקצה
   - 6.1 **'הפוך כיוון'** תנועה
   - 6.2 הקטן מיקום Y ב- 50 // גורם להורדת החללית והתקדמות לעבר החלליות הפולשות כדי להשמידן
7. **אם** נפגעתי מטיל הלייזר
   - 7.1 **'עדכן מונה'** חלליות מושמדות
   - 7.2 **'קבע תלבושת'** מצב ראשון של פיצוץ
   - 7.3 **'חזור'** 5 פעמים
     - 7.3.1 **'החלף לתלבושת הבאה'** של פיצוץ חללית
   - 7.4 **'הסתר'**
8. **אם** פגעתי ביורה הטילים
   - 8.1 **'שדר מסר'** – GAME OVER



תסריט טיל הלייזר

טיל הלייזר נע כלפי מעלה לעבר החלליות הפולשות, בדיקת פגיעה מתבצעת בתסריט של החלליות.

אלגוריתם מילולי - התסריט מופעל באירוע **'כאשר נלחץ מקש ארוך'**

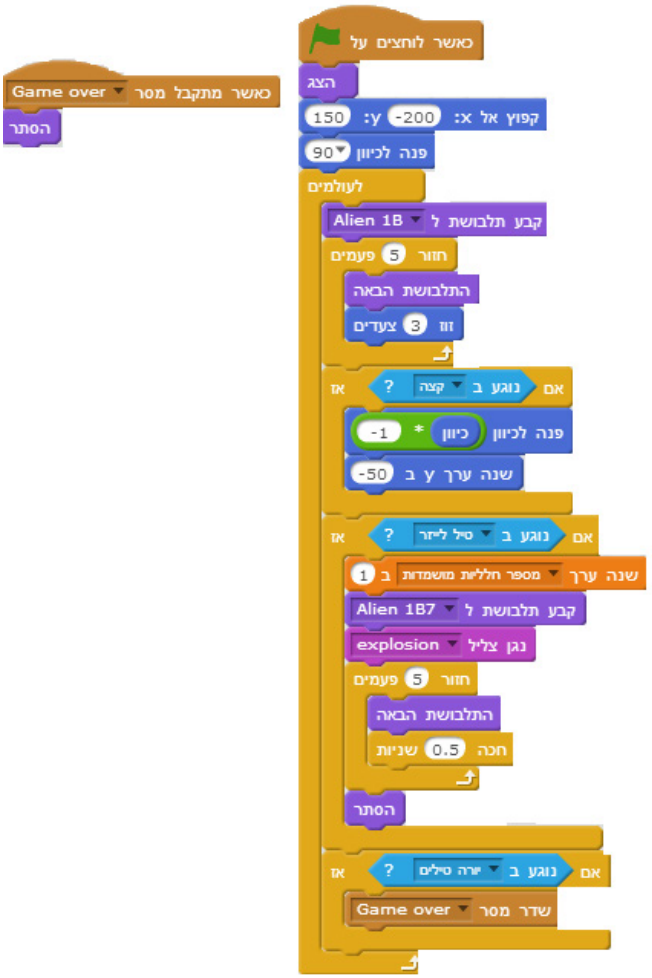
1. **'קפוץ'** אל יורה הטילים // יצירת האשליה שהטיל נורה מיורה הטילים
2. **'פנה'** לכיוון 0 // פנייה לכיוון מעלה לעבר החלליות
3. **'הצג'**// הצגת הטיל לפני ירי
4. **'חזור עד שנוגע'** בקצה // הטיל נע לעבר המטרה, אם לא פגע נפסקת התנועה בקצה הבמה העליון
   4.1 **'זוז** 10 **צעדים'**
5. **'הסתר'** // הירי הסתיים הטיל מוסתר

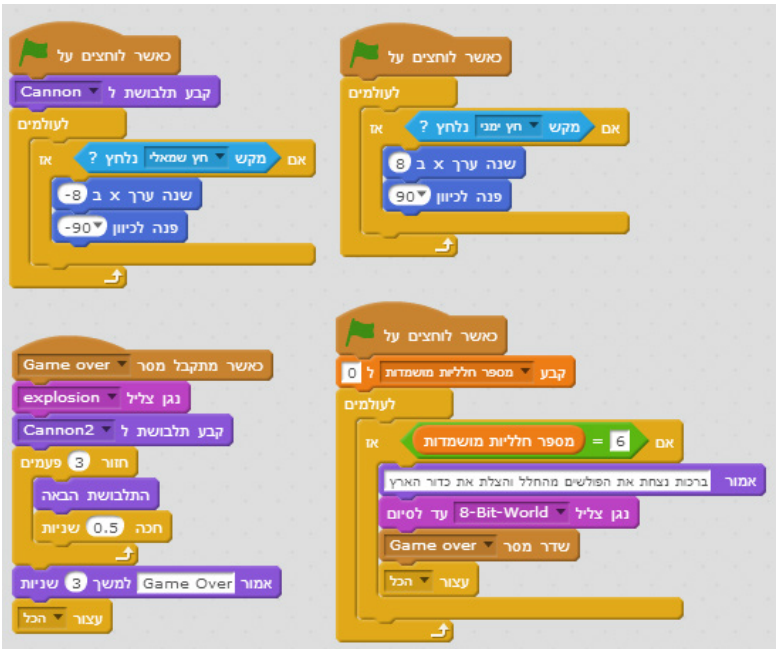
התסריטים של טיל הלייזר



תסריטי יורה טילי הלייזר

- תסריטי תנועה אופקית בסיסיים שאנחנו מכירים מיישומים קודמים
- תסריט סיום משחק: כאשר יורה הטילים נפגע מחללית אויב
- תסריט בדיקה: האם הושמדו 6 חלליות אויב





הרחבות ושיפורים למשחק

- שיפור והוספת תלבושות
- שימוש ב-2 או יותר סוגי חלליות אויב
- מתן אפשרות לחלליות האויב לירות על יורה הלייזר
- מעבר רמות עם רקע חדש, עם חלליות אויב שנעות מהר יותר ו/או יורדות מהר יותר

שער שני

מודול רובוטיקה

מבוא

לפני שני עשורים, תלמידים למדו רובוטיקה בעזרת זרוע רובוטית שהונחה על שולחן ומרבית האלגוריתמים והיישומים עסקו בהנעת הזרוע ובהרמת גוף והנחתו על גוף אחר. הרובוטים שימשו בני אדם בעיקר בתעשייה, וחלקם סייר על פני מאדים.

כיום השתנתה המציאות: רובוטים משמשים בתחומים שונים, וקיים מגוון של סוגי רובוטים זמינים ונגישים למערכת החינוך.

רובוטים משמשים ליעודים רבים: בבתי חולים כרובוטים מנתחים; בבית כרובוט השואב אבק בחדרי הבית או מנקה את הבריכה; בתעשייה במגוון רחב של יישומים; בתחבורה עם מערכות ללא נהג או בתעופה במערכות ללא טייס, אזרחיות וצבאיות.

השילוב בין מערכות תוכנה למערכות טכנולוגיות הולך ומתרחב. הלמידה של מערכות אלו דומה בחלקה ללמידה של תחום מדעי המחשב, אבל השילוב בין מערכות תוכנה וחומרה יוצר אתגרים חדשים. הפעלת יישום של תלמיד על רובוט דורשת הבנה והתמחות במערכות קלט פלט מעמיקות יותר מאלו שנדרשות ממנו במודול מדעי המחשב – הבנת הקשר בין החיישנים ופעולת הרובוט לבין הקוד הנכתב מהווה שילוב רב תחומי שאינו ברור ואינו נגזר מלימודי מדעי המחשב. בעולם של חיישנים וביצוע של יישום יש דינמיקה של REAL TIME, של דגימה רציפה, של ביצוע פעולות עם הרובוט שייתנו משוב מידי למה שקורה בסביבה בה הוא פועל, של הבנה של גדלים פיזיקליים ומערכות מכנוטרוניקה ועוד.

זה הייחוד של תחום אלגוריתמיקה ורובוטיקה בו עולם מדעי המחשב בא לידי מימוש ויישום בצורה ייחודית. זהו עולם ידע רחב גדול ורב תחומי ותכנית הלימודים במודול זה חושפת את העולם הזה בפני התלמידים.

בכתיבה נעשה מאמץ ליצור חיבור הוליסטי בין מודול מדעי המחשב למודול אלגוריתמיקה ורובוטיקה על מנת להטמיע את ההבנה שהנושאים משולבים, שהמושגים והרעיונות יונקים מאותן מקורות, מאותו מבנה תנאי, מאותו פסוק לוגי או מאותו מבנה לולאה, ומאותם משתנים – אלו אותם מושגים בשני המודולים עם אותה משמעות רעיונית ויישומית. מודול רובוטיקה מאפשר את יישום הרעיונות והאלגוריתמיקה באופן שמלהיב תלמידים, כי התוצאה של תכנית עם רובוט אינה הצגת תוצאה חישוב אלא רובוט שמבצע מטלה – את הרובוט התלמידים רואים וחווים. מודול רובוטיקה ממשש את היעד המרכזי של כלל תכנית הלימודים: חשיפה להיבטים השונים של מדעי המחשב – הפשטה, פתרון אלגוריתמי ומימושו – דרך התנסויות מהנות, מגוונות, ומעוררות עניין ומוטיבציה..

ואת המטרה הגדולה והכוללת הזו מיישמת **תכנית מודול אלגוריתמיקה ורובוטיקה** בצורה מלאה.

מודול אלגוריתמיקה ורובוטיקה

יעדי תכנית הלימודים אלגוריתמיקה ורובוטיקה –כיתות ד'-ה'

1. היכרות עם עולם מדעי המחשב בשימת דגש על פתרון בעיות באמצעות חשיבה אלגוריתמית ויישומה.
2. היכרות ראשונית עם עקרונות עבודת מערכות ממוחשבות (קלט-עיבוד-פלט) ועם מבנה המחשב.
3. הקניית מושגים בסיסיים בחשיבה לוגית וחשיבה אלגוריתמית, בצורה חווייתית בשילוב פעילויות לא ממוחשבת וממוחשבות רבות, בדגש על פעילויות חווייתיות יחד עם פעילויות תכנותיות עם מערכת רובוט. כל הפעילויות יהיו מותאמות לשכבת הגיל.
4. הבנת תהליכי פתרון בעיות ומימוש הפתרון (באמצעות שפת מחשב) תוך כדי מעבר בשלבים של תכנון; בנייה ובדיקת אלגוריתם כללי; כתיבת אלגוריתם מילולי; תכנון המרכיבים של היישום (באמצעות כלים שונים); ויישום הפתרון במערכת הרובוט.
5. פיתוח כישורי רפלקציה להבנת תהליך הפתרון ולחשיבה ביקורתית לצורך הערכת הפתרון.
6. פיתוח כישורים של עבודת צוות לצורך פתרון בעיות מורכבות ולביצוע פרויקטים בהיקפים שונים.

תכנית אלגוריתמיקה ורובוטיקה רחבה ונשענת על הפרדיגמות הבאות:

תכנות מערכות

סביבת הריצה:

- מהי סביבת הרצת התכניות
- סביבת תוכנת המערכת
- תכנית המשתמש

אלגוריתמיקה

שפת סקרטץ' בשילוב קבוצת ההוראות ROBOTS היא שפת תכנות לכל דבר ועניין.

מושגי יסוד של שפות תכנות:

- הוראה
- תחביר של הוראה (גם אם מדובר בלבנה גרפית יש להתייחס)
- פרמטרים לפקודה
- מבני תכנות, כולל: מבנה סדרתי, מבנה תנאי, מבנה לולאה
- שגיאות בתכנית: שגיאה תחבירית, שגיאת הרצה, שגיאה לוגית
- משתנים, ערכים קבועים, סוגי משתנים וערכים (לוגי, טקסט, מספר)
- הרחבת שפה (יצירת "פקודות" המשתמש)

עקרונות גישת המערכות

- רכיבי מערכת
- קלט , עיבוד ופלט במערכות
- תיכון מערכת
- יישום התיכון, בחינת התאמה לדרישות, תיקון ושיפור

עקרונות מערכות בקרה

- עיבוד ובקרה
- חיישנים
- משתנים מבוקרים
- בקרה בחוג פתוח / בקרה בחוג סגור
- מערכות מבוקרות מחשב

חיבור בין מדע לטכנולוגיה

- גדלים פיסיקליים
- נוסחאות חישוב בתחומים שונים (דוגמה: מהירות, זמן, דרך)
- עולם של טכנולוגיה: ראייה מערכתית על העולם הטכנולוגי הסובב אותנו / מערכות טכנולוגיות (מנוף, מכונית, סרט נע, מעלית, טנק)

עולם של טכנולוגיה מידע ותקשורת

- תקשורת בין רובוט למחשב
- תקשורת 'שן כחולה'
- שיתוף ידע (שיתוף רעיונות ותכניות בין הלומדים ובין הלומדים לעולם)
- חיפוש וכריית מידע (חיפוש מידע ברשת שרלוונטי למערכות שהתלמיד בונה)
- השפעת הטכנולוגיה על האדם (סביבה, כלכלה, חברה)

עקרונות פדגוגים של התכנית

התכנית תדגיש ותיישם את העקרונות הבאים:

- למידה ובנייה של ידע
- למידת חקר והתנסות
- למידת צוות
- הערכה חלופית ומעצבת
- גישה פרויקטנטית

עקרונות תכנית הלימודים רובוטיקה – הרחבה

תכנית פיתוח חשיבה לוגית-הנדסית רב תחומית באמצעות רובוטיקה כוללת מרכיבים רבים:

1. תכנות מערכות – סביבת הרצת התכניות; סביבת תוכנת הרובוט; תכניות המשתמש.
2. הנדסת תוכנה – כלים לבדיקת התוכנה (בדיקת חיישנים, בדיקה עם הוראות 'הדפסה').
3. אלגוריתמיקה וכתיבת קוד – מושגי יסוד של שפות תכנות; מהי הוראה; תחביר של הוראה; פרמטרים להוראה; מבנה תכנות: מבנה סידרתי, מבנה תנאי, מבנה לולאה; משתנים, ערכים קבועים, סוגי משתנים וערכים (לוגי, טקסט, מספר).

4. עקרונות גישת המערכות – רכיבי המערכת; קלט, עיבוד ופלט במערכות

5. עקרונות מערכות בקרה – עיבוד ובקרה; חיישנים; בקרה בחוג פתוח/בקרה בחוג סגור

6. חיבור בין מדע לטכנולוגיה – נוסחאות חישוב בתחומים שונים (דוגמאות: רדיוס וחישוב היקף גלגל, מהירות, זמן ודרך); עולם של טכנולוגיה: ראייה מערכתית על העולם הטכנולוגי הסובב אותנו.

7. עולם של טכנולוגיה, מידע ותקשורת – תקשורת בין המחשב לרובוט; תקשורת 'ישן כחולה'; שיתוף ידע (שיתוף רעיונות וידע בין הלומדים, בין הלומדים לעולם); חיפוש וכריית מידע; השפעת הטכנולוגיה על האדם .

8. עקרונות פדגוגיים של התכנית – למידת צוות; למידת חקר והתנסות; למידה עצמאית ובניית ידע; הערכה חלופית וגישה פרויקטנטית.

תכנית כוללת: רובוטיקה – 60 שעות

מסגרת התכנית מיועדת ל-70 שעות, 10 שעות יותר ממה שמוקצה בתכנון 60 שעות. החריגה בשעות מיושמת בתכנית בפרויקטי הסיכום. לקראת סיום מחצית א' וסיום שנת הלימודים מומלץ לקיים 'שבוע פרויקטים' ולערוך הקתונים מרתונים בשיתוף הורים – באירועים אלו להשלים את הפרויקטים, אלו שמופיעים בתכנית לצד רעיונות מקוריים של תלמידים.

הנחת הבסיס היא שלא ניתנים שיעורי בית לעבודה עם רובוט אולם ניתן לתת מטלות תכנון, להקדיש זמן בבית לתכנון ולחשיבה על מטלות סיכום ופרויקטים ולניהול. למידה בדרך זו (מימוש חלקי של רעיון כיתה 'הפוכה') יסייע לממש את התכנית באופן מיטבי.

בנוסף בתכנית משולב פרק העוסק בהכרה ושימוש ב'אופרטורים לוגיים', וכן לולאת '**חזור עד ש**', שני נושאים אלו מורכבים ויתכן שלא ניתן להציגם וליישמם בכל קבוצת לימוד. הסרתם מהתכנית בקבוצות לימוד מסוימות תוריד מהתכנית היקף של 10 שעות. מסגרת התכנית ויישום מלא הוא תלוי פרמטרים רבים ויש גמישות ביישום התכנית שיקבע בשטח ובכל בית ספר. לכן מוצגת תכנית ל-70 שעות.

בשעות העיוניות נכלל הזמן למבדקים, בחנים ובחינות 'מסורתיות'.

מבנה התכנית מוביל לצמצום בהערכה כמותית וחד פעמית, ועידוד הערכה חלופית מעצבת. לכן השעות שהוגדרו ויועדו בתכנית הלימודים להערכה 'מסורתית' מצומצמות.

52 שעות עבודה מעשית בחדר רובוטיקה, עבודה עצמאית ובצוותים.

18 שעות עיוניות – הרצאות קצרות, דיונים, מצגות (מצגות מורה ומצגות תלמידים). בשעות אלו נכללות 5 שעות לבחינות, בחנים, ומבדקים קצרים.

תחומים: חישוביות, אלגוריתמיקה ומיומנויות המאה 21 מרעיון לקוד

נתוני הרובוט הנדרש: דרישות שהוגדרו לרובוט בתכנית עתודה מדעית

בקר (CONTROLLER) – הכולל לפחות 3 כניסות ו-3 יציאות, תקשורת למחשב: תמיכה בסוללות AA או AAA או טעינה ישירה. אפשרות לחיווי באמצעות קול ו/או אור ו/או תצוגה.

הנעה – כולל מערכת הנעה בעלת 2 מנועים בתצורת טנק (מנוע ימין ומנוע שמאל). על מערכת ההנעה לאפשר פניה בקשת, פניה בציר ופניה סביב מרכז הכובד באמצעות שליטה על מהירות.

חיישנים (SENSORS) – חיישן אולטרה סוני (חיישן מרחק), 2 חיישני אור (למעקב אחר קו), חיישן מגע.

כוח – ערכת סוללות נטענות ומטען.

כבלים – כל הכבלים הנדרשים לחיבור הרכיבים לבקר וחיבור הבקר למחשב, כל הכבלים הנדרשים לחיבור חיישנים לרובוט (מומלץ אפשרויות לגיוון בסוגי חיישנים מעבר לדרישות הבסיס) .

בטיחות – הרובוט חייב לעמוד בתקני בטיחות בינלאומיים ובמידת הצורך תקני הבטיחות בישראל.

הערה: **התלמיד יקבל רובוט מורכב ומוכן להפעלה.** התכנית אינה מלמדת הרכבת רובוטים (בית הספר יכול להשתמש ברובוטים הדורשים הרכבה אך מפאת קוצר הזמן המורה לא יעסוק בהוראה שקשורה להרכבת הרובוט).

חיישנים ברובוט:

- שני חיישני אור
- חיישן מרחק
- חיישן מגע
- מומלץ רובוט עם אפשרויות לחיבור חיישנים נוספים כגון חיישן טמפרטורה וכדומה.

מנועים:

- מנוע שמאלי
- מנוע ימני

חיבור הרובוט למחשב

- באמצעות כבל USB מצורף לרובוט
- באמצעות רכיב בתקשורת BLUETOOTH לחיבור ללא כבלים

בחירת רובוט לתכנית הלימודים

התכנית הינה לרובוט גנרי. בכלל, כל אלגוריתם המופיע בתכנית הלימודים ניתן ליישום בכל רובוט.

להצגה מעשית ויישום האלגוריתמים אנו משתמשים בתכנית ברובוט MBOT רובוט שמבוסס על לוח ארדואינו ומרכיבים משלימים (חלקי תנועה/תמסורת, מנועים, חיישנים). ניתן לרכוש רובוט זה, מוכן ומורכב, מספקים שונים בארץ ובעולם, וניתן לרכוש את הרובוט לא מורכב ולהרכיב רובוט העונה על דרישות התכנית ועם שיפורים ותוספות מיוחדות.

הבחירה ברובוט בתכנית זו אינה חובה וגם לא המלצה, בית הספר ו/או הרשות המקומית יבחרו רובוט בהתאם לכללי ההתקשרות והרכישה של המוסד הרוכש, בתנאי שניתן ליישם עם הרובוט את תכנית הלימודים בהתאם לנתוני הרובוט הבסיסי שהוגדרו.

חשיבות התנסות מעשית עם רובוט (מתוך שמירה על אי-תלות בסביבת העבודה ובסוג הרובוט)

מודול התכנית ברובוטיקה מציג את השילוב בין הרעיונות והמושגים המרכזיים והחשובים במדעי המחשב לבין תכנית לימודים בדגש על אלגוריתמיקה, חד משמעותיות של הוראות, סדר ומבני בקרה ומימוש האלגוריתמים בשפת תכנות. כל הרעיונות הללו, בעת יישומם, באים לידי ביטוי בתנועה של הרובוט או בהדלקה ובכיבוי של נורות ובשימוש בחיישנים. ההיבט הוויזואלי והמעשי של ביצוע הפקודות באמצעות הרובוט ממחיש בצורה משמעותית יותר את רעיון הרצת תכנית – יותר מתסריט שמזיז דמות על המסך או מחשב ומציג ממוצע של שני מספרים.

כל כתיבת הקוד בתכנית הלימודים ברובוטיקה מלווה בכתיבה אלגוריתמית, בכל תסריט הוא שלב אחרון בתהליך שמתחיל ברעיון כללי וממשיך לאלגוריתם מילולי וקוד. תמיד מומלץ שתלמידים יכתבו פתרון למטלה/בעיה בתהליך מובנה, משלב הרעיון ועד לקוד בשפת התכנות בה משתמשים – במודול אלגוריתמיקה ורובוטיקה חשוב ביותר לכתוב אלגוריתם מילולי שכולל תהליך ניתוח וחשיבה משום שהתכנית תיושם בבתי ספר שונים בארץ וכל בית ספר יכול לבחור רובוט מדגם שונה ליישום התכנית, בהתאם לתקציבו ולניסיון המורים – לכן חשוב להקדים ולהציג עבור כל תסריט את האלגוריתם הכללי ואת האלגוריתם המפורט, ובשלב האחרון להתאים את הקוד לשפת התכנות בה משתמשים ולחומרה-רובוט בו עושים שימוש. התכנית של מודול רובוטיקה אחידה ואינה תלויה בסוג השפה בה משתמשים או ברובוט אותו רוכשים למימוש התכנית.

ברור שבמקביל ליעד לכתוב תכנית גנרית שאינה תלויה ברובוט הנבחר ניצבת המטרה להציג מימוש לתכנית ולשם כך בחרנו ברובוט בסיסי. ההנחה היא שכל מה שניתן לבצע עם הרובוט הבסיסי ניתן לבצע עם רובוטים מתקדמים יותר שכוללים יותר רכיבים (חיישנים, מנועים, מרכיבי בנייה ועוד).

בחירת רובוט מסוים משפיעה על יישום הקוד שבו יש להתייחס לחומרה (שם כניסה – קלט לרובוט, מאפייני החיישנים ועוד). אולם בקוד האלגוריתמי הניסוח הוא כללי וניתן ליישום בכל רובוט.

ברור שיתכנו שינויים גם בכתיבת קוד כשהרובוט וסביבת העבודה שונים, אך יחד עם זאת בכתיבת שלבי הפתרון נעשה מאמץ להקטין את תלות התכנית בסוג הרובוט ובסביבת העבודה, עד לנקודה בה אנו מממשים את הפתרון בסביבת העבודה.

התקנת סביבת העבודה וחיבור למחשב

פיסקה זו מתייחסת לרובוט mBot, במקביל להערכה שבמהלך השנים של הפעלת התכנית יהיו יותר רובוטים זמינים לשימוש בתכנית. כבר היום ישנם בשוק העולמי ובארץ רובוטים רבים שמרביתם עומדים בדרישות הבסיסיות שדורשת התכנית. יתרה מזו הרובוט המסוים בו ממומשת התכנית מבוסס על פלטפורמה של לוח ארדואינו והוא אינו הרובוט היחיד המבוסס על פלטפורמה זו, ומאחר ומדובר בלוח שהוא OPEN SOURCE אין מניעה שכל יזם בישראל ובעולם יפתח רובוט על אותה הפלטפורמה.

ככל שיהיו יותר רובוטים, מספקים שונים שמאפשר את יישום התכנית, העלויות יוזלו והנגישות לטכנולוגיה תהיה זמינה לכל בית ספר המפעיל את התכנית.

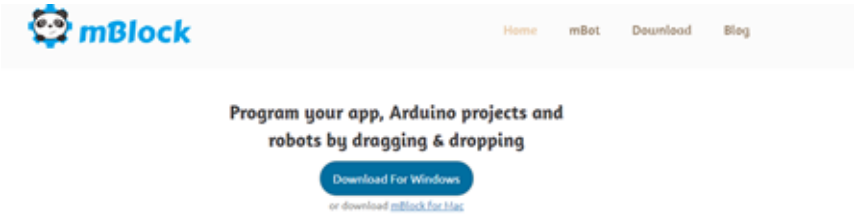
בעובדה שאנו משתמשים ברובוט מסוים אין משום המלצה על רובוט זה – כל פרויקט וכל תסריט מוצג בתיאור כללי רעיוני ועם אלגוריתם מפורט שניתן למימוש בסביבות עבודה מגוונות.

הפעלה וחיבור הרובוט mBlock

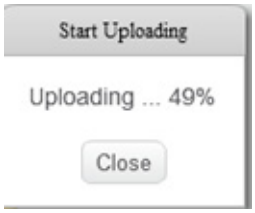
התקנת סביבת העבודה

יש להיכנס לאתר : HTTP://WWW.MBLOCK.CC/

ולפעול בהתאם להנחיות ולהתקין את התוכנה בהתאם לסוג מערכת ההפעלה



הפעל את תוכנת mBlock.



תוכנת התכנות של שפת סקרטץ׳ המוכרת לנו תופיע על המסך.
זו סביבת עבודה המאפשרת לעבוד בשפת סקרטץ׳ ולתכנת את הרובוט.

שפה

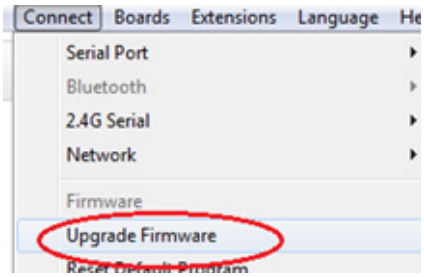
ניתן לשנות את השפה ל-׳עברית׳, אבל יש לשים לב שהשינוי משנה את סדר כתיבת ההוראות על הבלוקים לסדר שחלקו חסר משמעות, לכן מומלץ להישאר בשפת הסביבה ׳אנגלית׳. בתהליך למידה לא ארוך תלמידים יזהו פקודות ויבינו מה הן מבצעות (וילמדו וישפרו את הידע באנגלית).

שלבים בחיבור הרובוט למחשב באמצעות כבל USB:

א. בתפריט ראשי יש לבחור CONNECT

A. בתת התפריט יש לבחור SERIAL PORT ולבחור את היציאה COM4, חשוב: מספר הכניסה תלוי במערכת המחשב שלכם והציוד ההיקפי המחובר למערכת

ב. בתפריט CONNECT יש לבחור UPGRADE FIRMWARE



ג. המחשב יחל לטעון את התוכנה לרובוט ועל המסך תופיע תיבת מידע

ד. בסיום הטעינה (100%) הרובוט ישמיע ״ביפ״ קצר שמשמעותו מוכן לפעולה.

חיבור הרובוט למחשב באמצעות Bluetooth

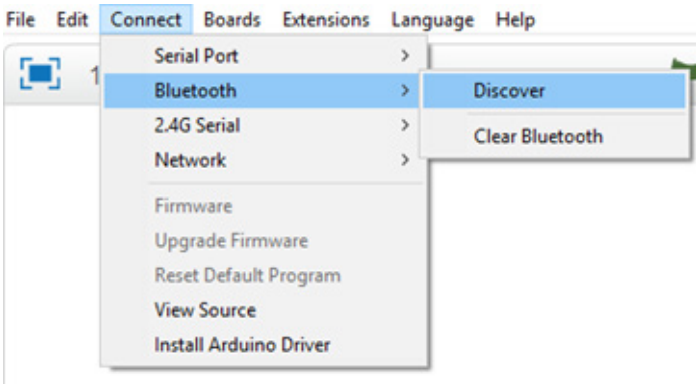
חיבור באמצעות BLUETOOTH

בתפריט הראשי בחר בלשונית CONNECT

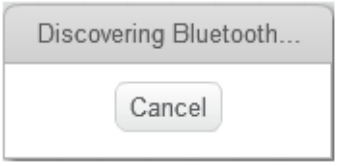
בחר באפשרות BLUETOOTH

בחר באפשרות DISCOVER

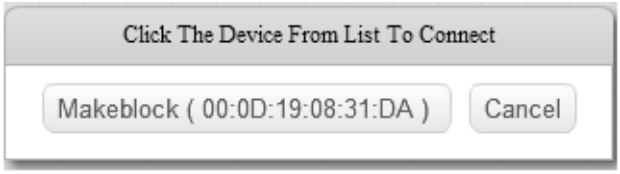
הפעולה שבחרנו גורמת למחשב שלנו 'לחפש' מכשיר (פועל) עם חיבור BLUETOOTH.



התקדמות תהליך החיפוש מוצגת בתיבה הבאה על המסך



עם זיהוי הרובוט עם חיבור BLUETOOTH תופיע על המסך רשימת המכשירים שזוהו



נאשר את המכשיר – רובוט זוהה ואנו מקבלים הודעת אישור



בכך חיברנו בין הרובוט למחשב וכל תכנית שנפעיל, תטען על הרובוט ותחל לפעול.

חשוב לציין שהקשר בין המחשב לרובוט לא נפסק גם במהלך הריצה של התכנית: אם יש הוראת אירוע כגון **'כאשר נלחץ מקש'** התכנית המופעלת תגיב גם לכל הקשת מקש במחשב בהתאם לתסריט שמופעל. זהו מצב שמזמן הזדמנויות ליצור יישומים מדהימים ויצירתיים.

הרצת תכניות עם הרובוט

הוצגו שתי דרכים להעלאת התוכנה לרובוט והרמת התוכנה:

א. באמצעות כבל USB

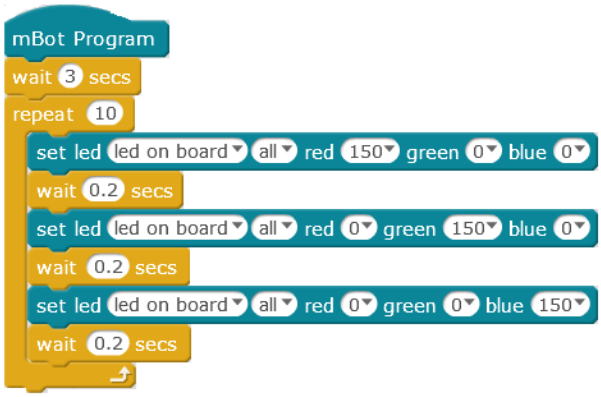
ב. באמצעות BLUETOOTH

כאשר אנו מפעילים תסריט באמצעות BLUETOOTH זה דורש קיום של חיבור כזה במחשב או לחילופין הימצאותו של רכיב USB ייחודי שתומך בחיבור BLUETOOTH. ישנה דרך נוספת להעלאת תכנית לרובוט ולהפעלת הרובוט ללא חיבור למחשב עם כבל וללא חיבור BLUETOOTH – באמצעות העלאת התסריט לרובוט דרך הלוח של הרובוט ARDUINO – יש לחבר את הרובוט למחשב באמצעות כבל USB עד לסיום שלב העלאת התכנית.

הפעלת תכניות עם הרובוט באמצעות לוח הארדואינו

אופן הביצוע:

כתוב תסריט כאשר האירוע שמתחיל את התסריט הוא הבלוק: 



התסריט מבצע הדלקת נורות על הרובוט (העניין עתה הוא לא התסריט אלא העלאתו לזיכרון הרובוט)

העלאת התסריט ללוח ארדואינו:

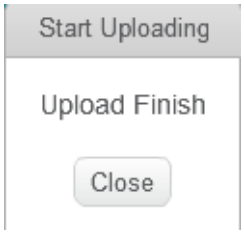
הקלק על הבלוק 

בחלק העליון של התסריט

נקבל מצד ימין מסך עם התוכנה כפי שהיא נכתבת ללוח של הרובוט, עם הצעת האפשרות להעלות את התוכנה לרובוט: הקלק על כפתור זה



עם ההקלקה על הכפתור UPLOAD TO ARDUINO: במשך כמה שניות מתבצע תהליך בדיקת התוכנה והעלאתה לזיכרון הלוח ואנו מקבלים את ההודעה:



ניתן לנתק את כבל USB והתוכנה תרוץ ללא חיבור למחשב.

במצב הזה אנו עובדים רק עם התסריט עם הבלוק 

לא ניתן להפעיל תסריטים אחרים.

אם יש תוכנה שפועלת על הלוח ורוצים לעצור את פעולת התוכנה יש לבצע RESET וזאת עושים על ידי הטענת תוכנה ללא הוראות בצורה פשוטה: מעלים ללוח את התוכנה של ההוראה 

וזו תכנית ללא הוראות – התכנית הקודמת שפעלה על הלוח-רובוט מתבטלת ולא פועלת.

חשוב! כאשר עובדים בתכנות הרובוט עם לוח הבקר, רק 4 קבוצות של שפת סקרטץ' פעילות וזמינות לכתיבת הקוד:

- קבוצת הוראות לטיפול במשתנים – DATA & BLOCK 

- קבוצת הוראות בקרה - CONTROL 

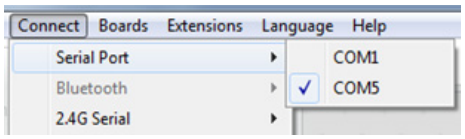
- קבוצת הוראות מפעילים – OPERATORS 

- קבוצת הוראות של הרובוט - ROBOTS 

חזרה למצב תכנות באמצעות סקרטץ' ולא ישירות עם הכרטיס

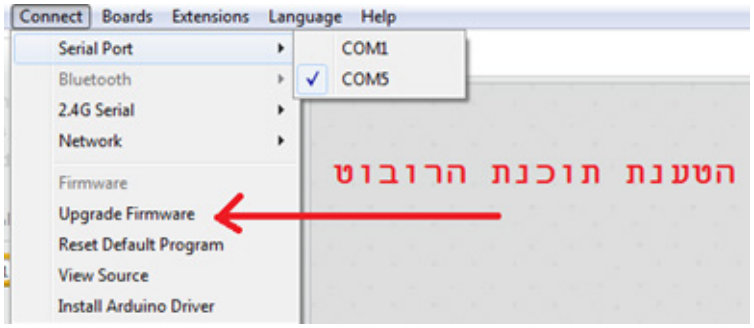
א. מחברים את כבל USB

ב. מוודאים שהרובוט מחובר.



רואים סימן אישור לחיבור הרובוט באמצעות חיבור סריאלי

ג. עתה מעלים חזרה את התוכנה שמחברת בין הרובוט לסקרטץ', תוכנה שנקראת FIRMWARE. הפעולה נמצאת באותו תפריט של קביעת החיבור:



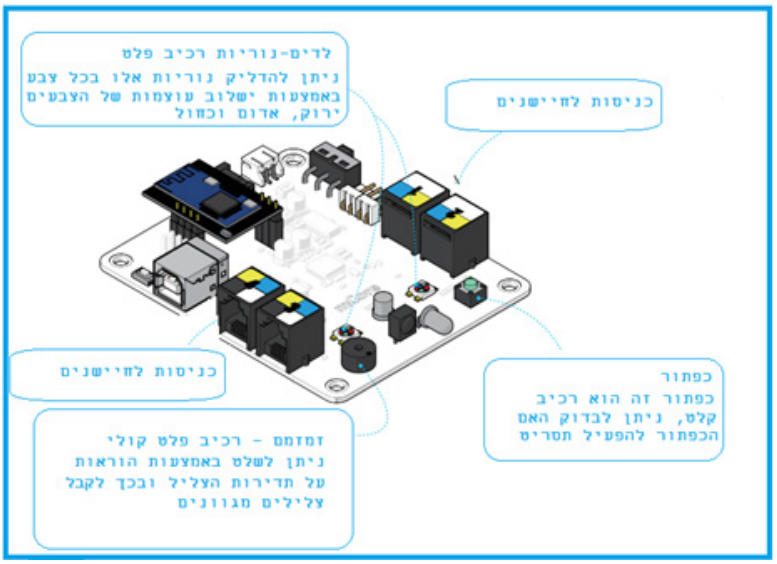
ד. בתום הטעינה נקבל אישור:



זמינות קבוצות ההוראות בסביבת התכנות סקרטץ'

בתכנות בסביבת סקרטץ' כל קבוצות הפקודות של שפת סקרטץ' זמינות ונוספות להן קבוצות ההוראות Robots . השילוב מאפשר תכנות של הרובוט בעולם 'הפיזי' כשניתן ליצור הדמיות בעולם של הדמויות בסקרטץ' ובבמה.

תרשים של הלוח המרכזי
המפעיל את הרובוט
באמצעות התכנות



תכנית כוללת, רובוטיקה – 60 שעות
(+ 10 שעות הרחבה, פרויקטים קבוצתיים)

פרק	תוכן הפרק	שעות עיוני	שעות מעשי	שעות סה"כ	שעות סה"כ בתכנית
א	מערכת מחשב-רובוט קלט, פלט, מעבד סוגי מערכות מחשב ושימוש במחשבים	1	1	2	2
ב	מבוא למערכות • מהי מערכת • מרכיבי מערכת • סוגים ודוגמאות למערכות • מרכיבי רובוט • תרגול בסיסי בהפעלת הרובוט	2	2	4	6
ג	סקירת מרכיבי הרובוט, תנועה בקו ישר ופניות • מרכיבי הרובוט: מנועים, בקר, חיישנים • תנועה בקו ישר • פניות	1	4	5	11
ד	תנועה ופניות עם שליטה על כל מנוע בנפרד הרחבה – מרעיון ליישום בקוד, תהליכי כתיבת קוד מובנים פניות שונות: קשת, רגל ציר, במקום. שילוב הוראות מקבוצת 'אירועים' ביישומים מורכבים	2	6	8	19

פרק	תוכן הפרק	שעות עיוני	שעות מעשי	שעות סה"כ	שעות סה"כ בתכנית
ה	תהליכי עיצוב תוכנה, מרעיון לקוד אלגוריתם מילולי כללי אלגוריתם מילולי מפורט גישת 'מעלה-מטה'	6	שעות אלו כלולות ונפרשות בכל פרקי התכנית		18
ו	בקרה בחוג פתוח, בקרה בחוג סגור • בקרה בחוג פתוח – הגדרת מושגים • בקרה בחוג סגור • חיישנים • לולאה אינסופית (לולאת **לעולמים** בסביבת סקרטץ')	2	6	8	27
ז	הרחבה בנושא מרכיבי שפת תכנות בסיסיים משתנה – ייצוג ושימוש משפט תנאי – '**אם**' משפט תנאי '**אם ואם לא**' שילוב חיישנים במטלות פשוטות ואתגרים	2	5	7	34
ח	קלט ממקלדת (ההוראה '**שאל**' בסביבת סקרטץ') הרחבת מבני לולאה – לולאה לא מותנת ('**חזור_פעמים**' בסביבת סקרטץ') חיישן אור **עוקב פס שחור** (חיישן אור לקבלת ערך אור חוזר) שילוב ביישומי בקרה בחוג פתוח (ללא חיישנים) ובחוג סגור (עם חיישנים)	2	8	10	44
ט	תנאים לוגיים מורכבים – שילוב בתכנית לפי שיקול דעת של המורה. 'וגם' - 'או' - 'לא' לולאת '**חזור עד ש**' - שילוב בתכנית לפי שיקול דעת מורה	1	6	8	52
י	מערכות בקרה בחוג סגור – המשך חיישן מגע – מימוש מערכת בקרה בחוג סגור חיישן טמפרטורה – מימוש מערכת בקרה בחוג סגור חיישן אור – מימוש מערכת בקרה בחוג סגור כל התרגול בשילוב משפט תנאי ולולאות (לעולמים, חזור_פעמים)	2	8	10	62
יא	מטלות סיכום ופרויקטים	2	6	8	70

פרק א' מבוא למערכות מחשב

תחומי תוכן: מיומנויות המאה 21



מחשב

מחשב הוא מכונה המעבדת נתונים על פי תכנית. למכונה ולרכיבים שנכללים בה אנחנו קוראים חומרה. לתכניות שמפעילות את המחשב ואת היישומים בהם אנו משתמשים, אנחנו קוראים תוכנה.

רכיבי קלט

המרכיבים במערכת מחשב שמאפשרים העברת נתונים ומידע למחשב. לדוגמה: מקלדת, עכבר, ידית היגוי, מיקרופון.

רכיבי פלט

המרכיבים במערכת מחשב שמאפשרים קבלת תוצאות העיבוד מהמחשב. לדוגמה: מסך, מדפסת, רמקול.

מקלדת

רכיב שמאפשר לנו להעביר קלט למחשב באמצעות הקלדתו.

עכבר

רכיב המאפשר לנו, לסמן, לבחור ולעשות פעולות נוספות שתוצאתן שפעולה שבחרנו באמצעות העכבר מתבצעת.

ידית היגוי

רכיב קלט למחשב המשמש בעיקר במשחקי מחשב.

מדפסת

רכיב פלט שמאפשר קבלת תוצאות עיבוד על דף נייר.

עיבוד מידע

תהליך בו המחשב מבצע פעולות על הנתונים כדי לקבל תוצאות. לדוגמה: חיפוש מידע ב'גוגל' כרוך בעיבוד המידע שנתנו למחשב. ודוגמה נוספת: בתהליך הדפסת תעודה לתלמיד המחשב מחשב את ציוני התלמיד ויוצר את התעודה שנשלחת להדפסה.

רובוט

רובוט הוא מכונה הכוללת מרכיבים אלקטרונים ומכניים ומסוגלת לבצע משימות באופן עצמאי וגמיש. בדרך כלל לרובוט

יש יכולת תנועה, יכולות חישה, יכולת הפעלה של זרועות מכניות ושל מערכות מכניות נוספות בהתאם לייעדו של הרובוט. לרובוט יש יכולת בקרה. להנחיה והפעלת הרובוט יש לכתוב תכניות מחשב.

רובוטיקה

רובוטיקה היא תחום בהנדסה שנוגע לבניית רובוטים – מכשירים היכולים לנוע ולהגיב על פי מידע שמתקבל מחיישנים. רובוטיקה היא גם ענף בחקר הבינה המלאכותית. רובוטיקה הוא נושא רב-תחומי ונוגע בתחומי הנדסה רבים כמו הנדסת מכונות ,הנדסת אלקטרוניקה, הנדסת תוכנה והנדסת ייצור. (ויקיפדיה)

דרכי הערכה

1. מטלת חקר מצומצמת על מחשב: סוגי מחשבים ושימושים במחשבים.
2. מטלת עבודה עיונית שכוללת משימת איסוף מידע על מערכת מחשב: יעודה, ההתפתחות של מערכת זו בעתיד ועוד.
3. מטלת כיתה: הדגמה של שימוש במחשב.

מטרות ביצועיות

1. התלמיד ידע לציין לפחות 3 שימושים למחשב.
2. התלמיד ידע למנות 3 רכיבי קלט ופלט ולתאר מהו ייעודו של כל רכיב.
3. התלמיד יתאר מערכת מחשב שבה עושים שימוש בבית.
4. תלמידים יצלמו בבית או במקום העבודה של ההורים, מרכיב שהוא קלט או פלט ויכתבו למה הוא משמש, מי משתמש בו ועוד.

מטרות ביצועיות – דוגמאות

1. התלמיד ידע לציין 3 שימושים למחשב; תלמידים יציינו שימושים שנחשפו להם, כגון: תוכנת מצגת, מעבד תמלילים ועוד. יש להוביל אותם לציון שימושים כגון: תוכנת עיבוד תמונות, תוכנת עיבוד וידאו, תוכנה לרובוט, תוכנה לניהול חשבונות, וגם לקבל בברכה ציון אפליקציות של טלפונים חכמים.
2. התלמיד ידע למנות 3 רכיבי קלט ופלט ולתאר מהו ייעודו של כל רכיב; תלמידים יציינו: מדפסת, מסך, עכבר. חשוב להרחיב את היריעה ולציין או לעודד לחשוב על רכיבים נוספים כגון: קורא ברקוד בסופרמרקט, מדפסת תלת-ממד, ידית היגוי למשחקים, זרוע של רובוט.
3. תלמיד יציג מערכת מחשב הנמצאת בביתו; גם במטרה זו יש להוביל את התלמידים לחשיבה על מהו מחשב, ובדיון מונחה להוביל להבנה שגם טלפון חכם זה מערכת מחשב, גם טבלט זה מערכת מחשב, גם הרובוט שמנקה את האבק זה מערכת מחשב וגם שעון חכם הוא מערכת מחשב.
4. תלמידים יצלמו בבית או במקום העבודה של ההורים מרכיב שהוא קלט או פלט, ויכתבו למה הוא משמש, מי משתמש ועוד. חשוב לעודד לצלם סרט קצר מעניין על מערכת מחשב ייחודית ועל שימוש לא שגרתי ולשתף. מומלץ לעודד שיתוף של סרטים הקשורים לרובוטיקה (רובוט צעצוע, רובוט מנקה וכדומה).

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
מהו מחשב ומה הם שימושיו.	דיון כללי ודוגמאות ביצוע המטרות הביצועיות בכיתה.	✓		מאפייני מחשב כולל דיון הצגת עבודות תלמידים. **1 שעות**
מבנה מחשב – רמת מרכיבים ותפקיד מקלדת מסך מדפסת	דיון כללי ודוגמאות . חשיבה על דוגמאות נוספות, עם מה משתמשים כאשר משחקים משחק מחשב : המחשב אצל הקופאית בסופרמרקט ועוד.	✓		מבנה מחשב כללי ומכללים כולל הצגת סרטי תלמידים (סרטים קצרים 2-4 דקות). **1 שעות**

פרק ב' מבוא למערכות



מערכת

מערכת כוללת קבוצת רכיבים שמבצעים יחד את מטרת המערכת.

מערכות שונות

מערכת הדלק במנוע רכב

- מטרת המערכת לספק דלק למנוע
- מרכיבי המערכת : מיכל דלק, צינורות, משאבת דלק, מסננים

מערכת העיכול

- מטרת המערכת לספק אנרגיה לגוף באמצעות המזון שנכנס לגוף
- מרכיבי המערכת : פה, וושט, קיבה, מעי דק, מעי גס

מערכת מחשב

- מטרת המערכת הרצת תכניות מחשב
- מרכיבי המערכת: זיכרון, מעבד, דיסק, מקלדת, עכבר, מסך

מערכת מכונת כביסה

- מטרת המערכת לכבס את הכביסה המלוכלכת
- מרכיבי המערכת: מנוע חשמלי, תוף, תא חומרים, גוף חימום

מה משותף לכל המערכות:

- לכל המערכות יש קלט, פלט ומעבד
- כל המערכות כוללות מרכיבים רבים וחלק מהמרכיבים הם תת-מערכת ומבצעים משימות מיוחדות.
- לכל המערכות יש תהליך או תהליכים אותם הן מבצעות

מה הקלט למערכות?

הקלט למערכת הוא אחד מ-3 הסוגים הבאים: **אנרגיה, חומרים, מידע**

מערכת מחשב

קלט למערכת: חשמל, נתונים, תכנית

פלט המערכת: תוצאות התכנית על המסך, או בנייר המדפסת או צליל מהרמקולים

מערכת מכונת כביסה

קלט למערכת: חשמל, מים, אבקת כביסה, כביסה מלוכלכת, הוראות כביסה

פלט המערכת: כביסה נקייה, מים אפורים (מלוכלכים)

מערכת דלק במכונית

קלט למערכת: דלק, מידע

פלט המערכת: סיבוב המנוע, עשן

למערכות מורכבות יש תתי מערכות

מערכת מכונית כוללת: מערכת דלק, מערכת בלימה, מערכת קירור, מערכת הנעה, מערכת מיזוג

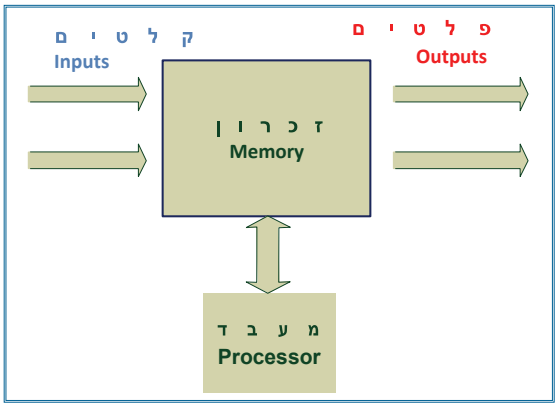
מערכת מכונת כביסה כוללת: מערכת חימום מים, מערכת ניקוי הבגדים, מערכת סחיטה

למרבית המערכות יש מערכת חשובה שאינה מבצעת פעולות באופן ישיר אלא מבקרת את פעולת תתי המערכות והמערכת כולה. מערכת זו כוללת חיישנים שמעבירים לה מידע בנוגע לאופן שבו מתקדם תהליך העיבוד. **במכונת הכביסה** ישנו חיישן המודד את טמפרטורת המים ומעביר מידע לבקרה של המכונה כשהטמפרטורה של המים הגיעה ליעד. **במכונית** ישנו חיישן 'רוורס' שמתריע בפני הנהג הנוסע לאחור על מכשול. **בגוף האדם** יש 'חיישן' שמתריע שהגוף רעב או שהגוף כבר שבע.

רישום מודל של מערכת – תרשים מלבני

ניתן לייצג מערכת באמצעות 'תרשים מלבנים', התרשים מציג מרכיבים של מערכת וקלטים ופלטים של מערכת.

מערכת מחשב כללית (מחשב, טבלט, טלפון חכם, רובוט)



התנסות בסביבת עבודה

פסקה זו משתנה בהתאם לסביבת העבודה של הרובוט בו בוחרים ללמד את התכנית.

יש להכיר את הסביבה ברמת הפעלתה כולל:

- פתיחת קובץ חדש לעבודה ושמירת הקובץ.
- כתיבת תכנית לרובוט, עריכה (במידה ויש מרכיבים טקסטואליים שיש לכתוב), בחירת לבנת הוראה (בשפות ויזואליות מהן אנו בוחרים לבנה שמייצגת הוראה)
- הרצת הוראה בסיסית
- הרצת תכנית / תסריט
- יצירת תסריט חדש בסביבת סקרטץ'

איך יוצרים תסריט חדש

יצירת תסריט היא פשוטה ביותר (מההיבט הטכני), כל שעלינו לעשות הוא הצעדים הבאים:

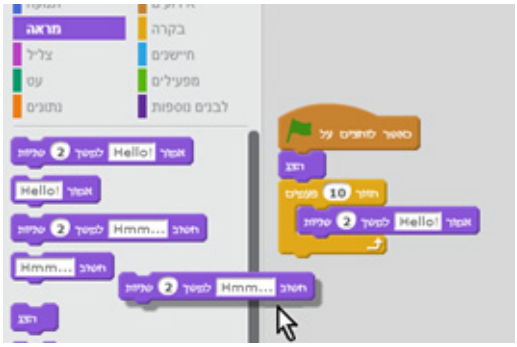
1. בחר את הלשונית 'תסריטים' והפעל אותה
2. בחר את קבוצת הבלוקים ממנה אתה מעוניין לבחור בלוק (הוראה)
3. גרור את הבלוק בו אתה מעוניין אל חלון התסריטים

כל הוספת הוראה/בלוק דורשת את הצמדת ההוראה החדשה למיקום הנכון לפי בחירתכם.

השלבים לביצוע הוספת הוראה/בלוק לתסריט

1. בחר את קבוצת הבלוקים ממנה ברצונך לבחור בלוק
2. בחר את הבלוק שברצונך להוסיף
3. גרור את הבלוק אל המיקום המדויק בתסריט ושחרר את הבלוק. ודא שהבלוק אכן נצמד לתסריט במיקום הנכון.

איור: בחירת בלוק והוספה לתסריט



בסביבת העבודה עם הרובוט בסקרטץ' אין צורך ליצור דמויות, אנחנו עובדים וכותבים תסריטים רק לדמות אחת.

ברירת המחדל היא הדמות :  כל התסריטים נכתבים לדמות זו.

M-Panda

ח ש ו ב !

כפי שנלמד בהמשך וגם במודול מדעי המחשב, ניתן לכתוב תסריטים לדמות – תסריטים רבים, ובמקביל לכתוב תסריטים לדמויות אחרות. בתכנית הלימודים אלגוריתמיקה ורובוטיקה אנחנו כותבים תסריטים רק לדמות אחת. אם מעוניינים ניתן להחליף את הדמות לדמות יותר מתאימה.

זה אפשרי ולדוגמה נבחרה דמות של הרובוט 

MBOT

כשאנחנו כותבים הוראות, כל התסריט הוא לדמות אבל חלק מההוראות ישפיעו על הדמות שעל הבמה, וחלק מהוראות ישפיעו על הרובוט האמיתי שנמצא על השולחן או על הרצפה.

לדוגמה ההוראות : 

יזיזו את הדמות על המסך והדמות תאמר(בתיבת דיאלוג) HELLO

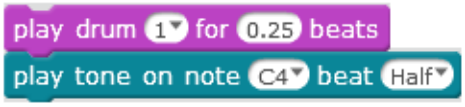
ההוראות  יזיזו את הרובוט האמיתי.

ח ש ו ב !

הרובוט בקטע זה לא עוצר. אם אתם מפעילים את הרובוט על שולחן הוסיפו הוראות עצור (בהתאם לסוג הרובוט ושפת התכנות) או הפעילו את הרובוט על הרצפה. ההוראות שכתבנו יזיזו את הרובוט על המשטח עליו הוא ניצב ושני הלדים (נוריות) על הלוח הראשי ידלקו באור אדום

כדי לעצור את הרובוט יש להוסיף הוראה RUN FORWARD עם ערך 0

ההוראות הבאות משלבות שילוב מעניין ומבלבל:



ההוראה בסגול היא מקבוצת ההוראות SOUND והיא הוראה לדמות ותוצאתה צליל תופים קצר.

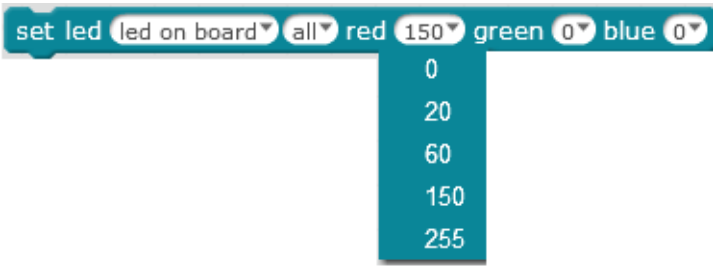
ההוראה בכחול היא הוראה השייכת לקבוצת ההוראות ROBOTS ותוצאתה שהרובוט ישמיע צליל קצר.

שילוב הוראות שבעיקר משפיעות על הבמה של סקרטץ' ועל הדמויות של סקרטץ' עם הוראות המשפיעות רק על התנהגות הרובוט הוא מבלבל אבל יש בו יתרון. אנו מחברים את העוצמה והיכולות של שפת סקרטץ' לתכניות עבור הרובוט. השילוב יוצר הזדמנויות גם ללמידה מעניינת וקלה יותר וגם לשילוב מעניין כגון: הרובוט על המשטח נע סביב ריבוע דמיוני, הרובוט על המסך נע סביב ריבוע שאותו הוא גם מצייר. אם נכתוב מטלה זו בשני תסריטים נראה מאפיינים דומים בשני התסריטים וזה יסייע לנו להבין טוב יותר וליהנות מהלמידה.

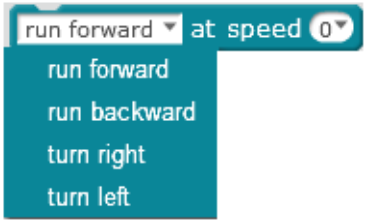
תרגילים ומטלות

נשתמש בהוראות:

- **'נוע'** (קדימה, אחורה, פנייה לימין, פנייה לשמאל)
- **'המתן __ שניות'**
- השמעת צליל של הרובוט
- הדלקת נורות על הרובוט וכיבוי הנורות (חקור כיצד ניתן לכבות את הנורה ואיך לבחור צבע)



ההוראה **'נוע'** עם האפשרויות שניתן לבחור



ההוראה **'השמע צליל'**



שלב הוראות אלו עם הוראת **'המתן __ שניות'**, הקלק על קבוצת ההוראות והרובוט יתעורר לחיים. אם תשלב הוראת תנועה, הוראת צליל או הוראת תאורה תקבל רובוט זז מאיר ושר.

דרכי הערכה

ההערכה בפרק זה תתבצע בשני ממדים:

הראשון בחדר מחשבים: מתן מטלות ליצירת תסריטים קצרים שמשלבים: צלילים, אורות, תנועה. אפשר לשלב צלילים ותנועה של הדמות על הבמה במקביל לרובוט.

השני ממד מילולי: תלמיד ייתן לחבריו תיאור מילולי מדויק וחד משמעי ככל האפשר והם יצטרכו לבצעו. תרגול התלמידים במתן הוראות לחבריהם ובחינה האם נתנו הוראות מדויקות והאם התלמידים הבינוי וביצעו את ההוראות כפי שנותן ההוראות ביקש.

מטלות

א. העבר לקוד את האלגוריתם הבא:

- נורה ירוקה למשך 1 שנייה
- כיבוי הנורה
- נורה אדומה למשך 1 שנייה
- כיבוי הנורה
- נורה כחולה למשך 1 שנייה
- כיבוי הנורה

ב. כאשר בוחרים לקבוע ערך עבור שלוש הנורות מקבלים את כל הצבעים.

הערכים עבור כל נורה מתחילים בערך 0 וממשיכים עד הערך 255

בדוק איזה צבע נקבל עם ההוראה הבאה 

חשוב! יישום האלגוריתמים נכתב עבור גרסת תוכנה 3.3.1. כעת זמינה גרסת 3.3.2. ההבדלים קטנים. הרובוט והיישום בסביבת mBlock מהווים חלופה אחת מיני רבות ביישום תכנית הלימודים, ומאחר שמתרחשים שינויים טכנולוגיים מהירים, חשוב לתת את הדעת למטרות ולרעיונות שניתנים ליישום בכל סביבת למידה בהתאם לבחירת בית הספר.

ג. הנע את הרובוט קדימה 1 שנייה

נסה לתכנן ולבצע פנייה ב-90 ימינה של הרובוט

הנע את הרובוט קדימה 1 שנייה

ד. כתוב תסריט קצר שיניע את הרובוט לפי הצורה של האות T

ה. כתוב מספר הוראות שיכללו הדלקת נורות והשמעת צלילי מוסיקה

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
מבוא למערכות מערכת כללית – הגדרה אינטואיטיבית מערכת – הגדרה מדויקת	הבנת מרכיבי מערכת כללית. לחבר את המושג למה שתלמידים מכירים משיעורי מדעים: מערכת הדם, מערכת העיכול ועוד.			✓	1. התלמיד יכיר את המושג 'מערכת'. 2. התלמיד יכיר את מרכיבי מערכת המחשב: קלט, פלט, עיבוד. 3. התלמיד יכיר את תהליך עיבוד המידע במערכת מחשב. **2 שעות**
סקירת מרכיבי הרובוט כתיבת הוראות בסיסיות בתסריטים קצרים	להציג את הרובוט, להסביר את מרכיביו, לחפש ברשת סרטים המציגים את הרובוט בפעולה ולקיים דיון בנושא. כתיבת הוראות בסיסיות להנעת הרובוט קדימה ואחורה, התנסות וחקר ללא למידה מקדימה, היכרות עם סביבת סקרטץ'.			✓	התלמיד יפעיל את הרובוט ויכתוב אלגוריתם מילולי עם הוראות בסיסיות. **2 שעות**

מושגים

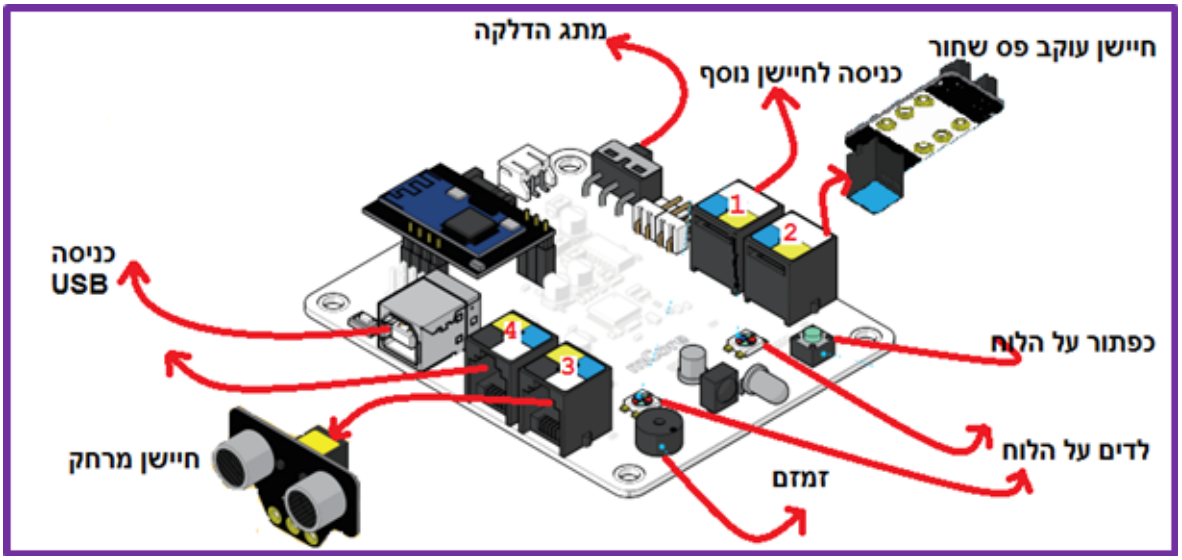
רכיבי פלט ברובוט:

- ✓ מנועים:
- ✓ מנוע שמאלי – M1
- ✓ מנוע ימני – M2
- ✓ נוריות (לדים)

רכיבי קלט ברובוט:

- ✓ חיישנים:
- ✓ חיישני אור
- ✓ חיישן מרחק- אולטראסוניק
- ✓ חיישן מגע

הלוח המרכזי של הרובוט



פירוט הרכיבים והוראות בסיסיות:

מנוע שמאלי

ברובוט המנוע השמאלי מוגדר כמנוע M1

חשוב! בכל רובוט בו נשתמש ישנם לפחות שני מנועים המשמשים לתנועה של הרובוט: מנוע שמאלי; מנוע ימני. בכל רובוט ובתוכנה המפעילה את הרובוט יש מזהה שמאפשר גישה למנוע מסוים והפעלתו. בחלק מהמערכות המזהה תלוי באופן החיבור בין המנוע לבקר הרובוט (לכניסה של הבקר), במערכות אחרות אין אפשרות לחבר את המנועים לכניסות שונות והחיבור נקבע ומובנה ברובוט.

במערכת בה אנו משתמשים בתכנית הלימודים, המנועים מורכבים ומחוברים לבקר. לא ניתן לשנות את החיבור והגישה והזיהוי של המנועים הוא: M1 , M2.

M1 מנוע שמאלי

M2 מנוע ימני

ההוראה הבסיסית להפעלת מנוע מאפשרת קביעת ערך למהירות המנוע וקביעת כיוון סיבוב המנוע על ידי קביעת ערך מהירות מספר חיובי או מספר שלילי.

חשוב! מספרים שליליים

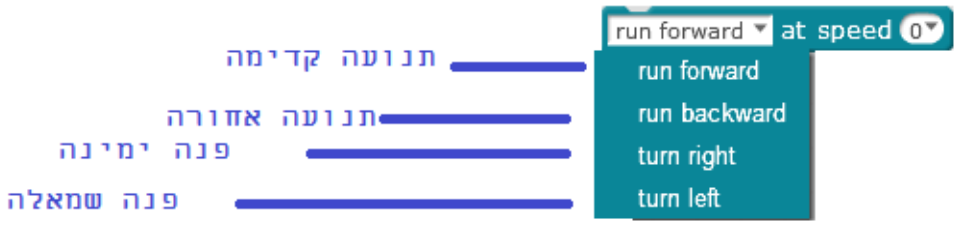
בהוראות הפעלת מנועים ניתן לשלב ערכים שליליים. תלמידי בית ספר יסודי אינם מכירים ואינם לומדים על מספרים שליליים. יש להסביר את הסימן מינוס כמייצג היפוך בכיוון סיבוב המנועים כשבכך אנו גורמים להיפוך בכיוון התנועה של הרובוט. הסבר כזה יאפשר לנו שימוש בערכים השליליים מבלי להיות בתחום התוכן של מספרים מכוונים במתמטיקה.

בכל מקרה מומלץ בהוראת תנועה לשני המנועים להשתמש בהוראה **'נוע'** עם כיוון "BACKWARD" במקום השימוש בערך שלילי שכדאי לדחות ככל האפשר, ותמיד לתת את ההסבר שערך שלילי זה כיוון הפוך.

כדי לסיים את פעולת המנוע יש להפעיל הוראה המפעילה את המנוע עם מהירות 0 ובכך מסתיימת פעולת המנוע.

בקבוצת ההוראות ROBOTS Robots יש הוראה המאפשרת להפעיל את שני המנועים לתנועה קדימה, והוראה המאפשרת להפעיל את שני המנועים לתנועה אחורה. הוראה זו מאפשרת לבצע פנייה ימינה או פנייה שמאלה.

כל האפשרויות של ההוראה **'נוע'**



חשוב! נסיעה קדימה אחורה ופניות

בפרק ד' נפרט ונלמד באופן מעמיק על תנועה של הרובוט ופניות של הרובוט ימינה ושמאלה. אנחנו נשתמש בהוראות של קביעת מהירות למנוע שמאל ומנוע ימין בנפרד.

קבוצות ההוראות ROBOTS של mBOT מאפשרת גם הנעת הרובוט קדימה ואחורה ופניות באופן פשוט עם הוראה ישירה הכוללת את האפשרויות:

- נוע קדימה
- נוע אחורה
- פנה ימינה
- פנה שמאלה

הוראות אלו מאפשרות לנו לכתוב אלגוריתמים לנסיעה קדימה ואחורה ולפניות מבלי לחשב את המהירות של כל מנוע בנפרד – גם לגבי התנועה בקו ישר וגם לגבי פניות. כמו כן ההוראות הללו מאפשרות לנו שינויים בתנועה של הרובוט קדימה ואחורה וכן פניות של הרובוט, מבלי להזדקק לשימוש בערכים שליליים (שנעשה

בהמשך) עבור ביצוע פניות של הרובוט. כך ניתן לתכנת תנועה של הרובוט קדימה אחורה ופניות בצורה יותר פשוטה. כפי שצוין למעלה השימוש בערכים שליליים הוא בעייתי ויש להציג את ההסבר לסימן 'מינוס' כסימן שמייצג שינוי כיוון (ולא ערך מספרי שלילי), השימוש בהוראה אחת לתנועה קדימה או אחורה או לפניות, ולאחר מכן בשלב שני ביצוע שינוי כיוון התנועה ופניות באמצעות ערכים שליליים עם הסבר חלקי, מאפשר לנו יישום ושימוש הדרגתי במונחים, ו'התחמקות' מדיון מעמיק או מדיון בכלל במספרים מכוונים.

חשוב! כל הוראת המתנה - WAIT מקבוצת הבקרה שמופיעה אחרי הוראת תנועה, משמעותה שהתנועה נמשכת למשך פרק הזמן של ההמתנה, כלומר הרובוט ינוע בכיוון שנקבע בהוראה לפני ההמתנה לאותו פרק זמן של ההמתנה.

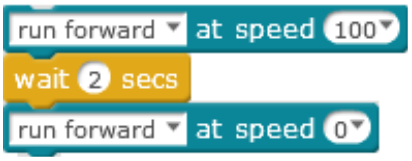
לעצירת הרובוט אחרי התנועה יש **תמיד!** להשלים הוראה עם השמה של הערך 0. עבור שני המנועים או עבור כל מנוע בנפרד או עבור ההוראות RUN FORWARD או BACKWARD – יש לכתוב את ההוראה עם מהירות 0 כדי שהרובוט יפסיק את תנועתו.

דוגמאות:

אלגוריתם לתנועה קדימה

1. **'סע קדימה'**
2. **'המתן'** 2 שניות

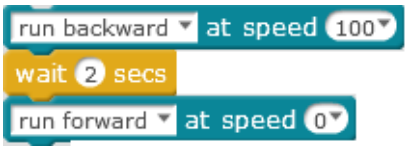
התסריט:



אלגוריתם לתנועה אחורה

1. **'סע אחורה'**
2. **'המתן'** 2 שניות

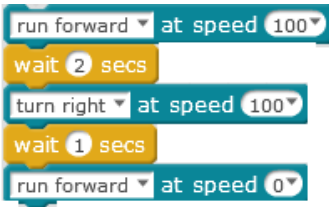
התסריט:



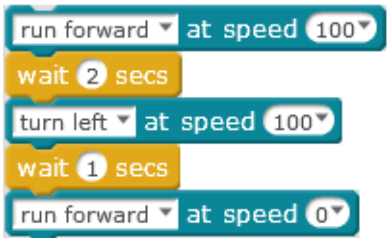
אלגוריתם לתנועה קדימה 2 שניות ופנייה ימינה 1 שנייה

1. **'סע קדימה'**
2. **'המתן'** 2 שניות
3. **'פנה ימינה'**
4. **'המתן'** 1 שנייה

התסריט:



אלגוריתם לתנועה קדימה 2 שניות ופנייה שמאלה 1 שנייה



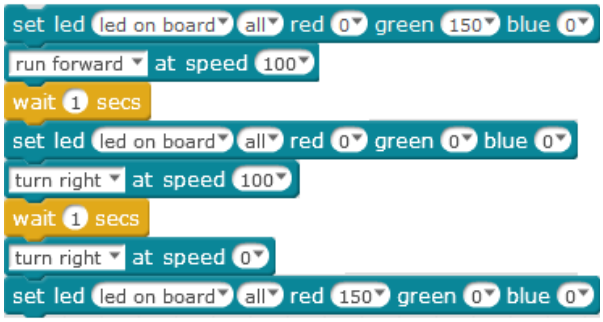
מ ט ל ה

- רובוט מדליק שתי נורות על הלוח הראשי בצבע ירוק
- רובוט נוסע קדימה 1 שנייה
- פונה ימינה 1 שנייה
- מדליק שתי נורות על הלוח הראשי בצבע אדום

אלגוריתם:

1. **'הדלק נורות'** על הלוח בצבע ירוק
2. **'נוע'** קדימה במהירות 100
3. **'המתן'** 1 שנייה
4. **'כבה נורות'** על הלוח
5. **'פנה ימינה'** במהירות 100
6. **'המתן'** 1 שנייה
7. **'עצור'** פנייה ימינה
8. **'הדלק נורות'** על הלוח בצבע אדום

התסריט:



ה ת נ ס ו ת :

תלמידים יכתבו תסריטים קצרים ויתנסו בהוראות:

'נוע קדימה'

'פנייה ימינה'

'פנייה שמאלה'

'הדלק' שתי **נורות'** על הלוח בצבעים שונים.

הוראת הפעלת מנוע בודד

דוגמה להוראת הפעלת מנוע:

מנוע שמאלי- מוגדר כללית כמנוע L (ללא קשר לסוג הרובוט)

מנוע שמאלי מקבל ערך



מנוע שמאלי מקבל ערך 100-



מנוע ימני - מוגדר כללית כמנוע R (ללא קשר לסוג הרובוט)

ברובוט המנוע הימני מוגדר כמנוע M2

ההוראה הבסיסית להפעלת מנוע מאפשרת קביעת ערך לעוצמת הפעולה, וקביעת כיוון סיבוב המנוע – מספר חיובי או מספר שלילי. כדי לסיים את פעולת המנוע יש להפעיל הוראה שמשימה ערך 0 בעוצמת המנוע ובכך מסתיימת פעולת המנוע.

מנוע ימני מקבל ערך 100



מנוע ימני מקבל ערך 100-



מנוע ימני מקבל ערך 0 – עצירה



ההוראות למנוע הימני יפעלו באותו אופן כמו ההוראות למנוע השמאלי.

תרגול

יש לאפשר לתלמידים להתנסות בהוראות להפעלת המנועים. בשלב ראשון מעין תרגול 'ללא מטרה': לבחור הוראה של הפעלת מנוע ולהעביר אותה לאזור התסריטים, להציג כיצד בוחרים ערך מהערכים בברירות המחדל בהוראה וללחוץ לחיצה כפולה על ההוראה לשם הפעלתה. תלמידים יכולים לחבר עד 4 הוראות לשם תרגול בסיסי.

חשוב! מנוע המופעל באמצעות ההוראה SET MOTOR ימשיך לפעול עד שנפעיל הוראה SET MOTOR עם הערך 0.

תנועה קדימה

כאשר שני המנועים מקבלים הוראה לתנועה עם מהירות חיובית זהה הרובוט ינוע קדימה

תנועה אחורה

כאשר שני המנועים מקבלים הוראה לתנועה עם מהירות שלילית זהה הרובוט ינוע אחורה

מטלות דוגמה:

מטלה א'

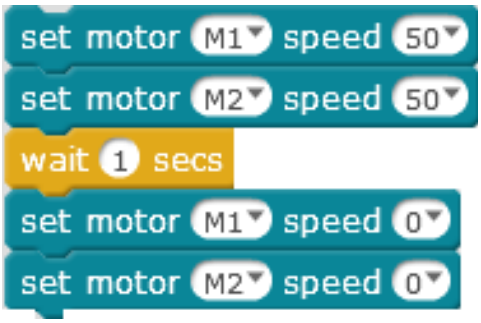
רובוט ינוע קדימה במשך 1 שנייה

פתרון:

אלגוריתם כללי

קבע מהירות שווה לשני המנועים

המתן שנייה

אלגוריתם מפורט	תסריט בסביבת העבודה
1. L <==50 2. R <==50 3. **'המתן'** 1 שנייה 4. L <==0 5. R <==0	

מטלה ב'

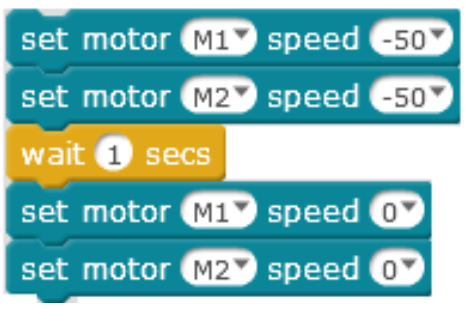
רובוט ינוע אחורה במשך 1 שנייה

פתרון:

אלגוריתם כללי

קבע עוצמה שווה לשני המנועים

המתן שנייה

אלגוריתם מפורט	תסריט בסביבת העבודה
1. L <== 50- 2. R <== 50- 3. **'המתן'** 1 שנייה 4. L <== 0 5. R<== 0	

דרכי הערכה

ההערכה בפרק ג' תתבסס על מתן מטלות ומתן ציונים על ביצוען.

המטלות יתמקדו בכתיבת סדרת הוראות מילוליות ממוספרות שמורות מה אנו מבקשים שהרובוט יבצע.

על התלמידים לכתוב קוד לכל קבוצת הוראות.

יש לתת מטלות עם הוראות ברורות ומדויקות ולצדן לתת מטלות עם הוראות יותר מעורפלות ופחות חד משמעותיות.

יש לקיים דיון על ההבדלים בין ההנחיות שהתלמידים קיבלו לבין מה שבוצע בפועל.

דוגמה להוראה חד משמעית:

הדלק נורה אחת ירוקה למשך 1 שנייה

דוגמה להוראה לא ברורה וחד משמעית:

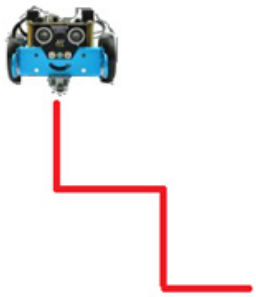
הדלק מספר נורות

מטלות

א. כתוב תסריט שיכלול תנועה של הרובוט במהירות (עוצמה) מסוימת למשך 5 שניות. מדוד את המרחק שהרובוט נסע.

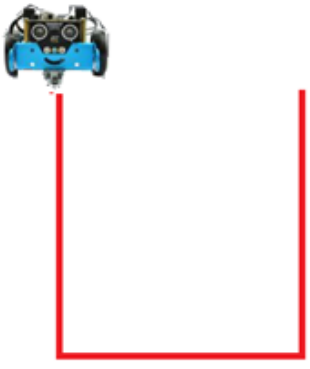
חשב מהי המהירות של הרובוט ביחידות של ס"מ/שנייה

ב. כתוב קבוצת הוראות תנועה שתניע את הרובוט בצורה של שתי מדרגות דמיוניות:



ג. כתוב קבוצת הוראות שתניע את הרובוט סביב ריבוע דמיוני.

חשוב אלו הוראות מתבצעות מספר פעמים וכמה פעמים ההוראות מתבצעות.



נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
תנועה של רובוט קדימה אחורה	בשלב ראשון ההתנסות צריכה להיות 'טכנית' תרגול בבחירת הוראה, בחירת ערכים לביצוע והקלקה על הוראה.		✓	✓	יישום אלגוריתמים פשוטים של תנועה עם מימוש של תסריט קצר. **1 שעות**
תנועה קדימה אחורה ופניות	בשלב שני חשיבה על רעיון: מהו מסלול הנסיעה הנדרש. כתיבת אלגוריתם מילולי בסיסי ויישומו רק עם ההוראה RUN.	✓	✓	✓	יישום אלגוריתמים עם מורכבות הכוללים את תכנון מסלול הנסיעה של הרובוט. **1 שעות**
תנועה קדימה/ אחורה ופניות, עם ההוראה SET MOTOR כל התסריטים יכללו הוראה מקבוצת ההוראות 'אירועים' ויופעלו על ידי האירוע שיבחר	זהו השלב השלישי בתרגול תנועה בסיסי. השלב יותר מורכב כי יש הוראה נפרדת לכל מנוע ויש גם מספרים שליליים . חשוב שתלמידים יכתבו (עדיין בצורה חופשית) את: מטרות אוסף ההוראות – התסריט שהם כותבים. כתיבה עם מספור אלו הוראות הם מורים לרובוט לבצע. בתום שלב זה התלמידים יכתבו קוד.	✓	✓	✓	יישום אלגוריתמים שכוללים הפעלה נפרדת של 'יציאות פלט', כלומר הפעלת מנועים באופן נפרד, הפעלת 'יציאות פלט' של נורות לד ומוסיקה. **3 שעות**

פרק ד' תנועת הרובוט ופניות - הרחבה

חשוב! עבור רובוט כללי, הפעלת מנועים והמתנה משמעותה שהמנועים יעבדו לפי העוצמה שנקבעה להם בפרק הזמן לפי ערך ההמתנה. מיד בתום ההמתנה יש לתת לשני המנועים ערך עוצמה 0 כדי שהתנועה תעצור.

הרעיון החשוב הזה מובנה בתכנית הלימודים ברובוטיקה בחטיבת הביניים. בהיבט היישום לא כל הרובוטים וסביבת התכנות שלהם פועלים בהלימה לרעיון הזה. הרובוט mBot בסביבת סקרטץ' פועל בדיוק באופן הזה. וכך יש הלימה בין האלגוריתמים המילוליים לבין היישום.

תכונות הרובוט ופניות

מבנה הרובוט שהצגנו כלל שני מנועים:

מנוע שמאלי L – מנוע M1 ברובוט בו השתמשנו ליישום התכניות

מנוע שמאלי R – מנוע M2 ברובוט בו השתמשנו ליישום התכניות

חשוב: הרובוט יבצע פנייה תמיד לכיוון המנוע היותר איטי

תכונות הרובוט והתנהגותו מסוכמים במידע הבא ובטבלה המסכמת:

לרובוט התכונות הבאות:

שני מנועים (מנוע L, מנוע R)

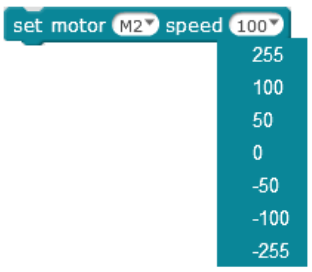
לכל מנוע ערך מספרי המייצג את מהירות המנוע

0= עצירה

ערך >0: מהירות התנועה קדימה

ערך <0: מהירות התנועה אחורה

חשוב להתייחס לברירות המחדל של הפקודה שניתנת למנוע, אלו הברירות שנשתמש בהן במרבית התכניות.



הטבלה הבאה מציגה את התנהגות הרובוט לפי ערכים שונים למנועים. חשוב לציין שבחרנו ערכי דוגמה, וחשוב להכליל משום שהתנהגות הרובוט תהיה דומה עבור ערכים שונים (דומה אך לא זהה)

עבור ערכים שווים וחיוביים הרובוט ייסע קדימה.

עבור ערכים שווים ושליליים הרובוט ייסע אחורה.

עבור ערך חיובי במנוע אחד, וערך שלילי (שווה בגודלו) במנוע השני הרובוט יפנה פנייה במקום. זמן הפעלת המנועים לקבלת פנייה במקום של- 90 מעלות תלוי בעיקר במהירות המנועים ותלוי גם בתנאי 'השטח'

עליהם נע הרובוט: עוצמת הסוללות. אנו נבחר את הזמן 0.5 שנייה כזמן הנדרש לפנייה של- 90 מעלות. עבור שני ערכים שונים מאפס ולא שווים הרובוט יעשה פנייה בקשת לכיוון המנוע האיטי. הטבלה הבאה מציגה דוגמאות לערכים ואופן תנועת הרובוט.

התנהגות הרובוט	מהירות מנוע ימין (M2)	מהירות מנוע שמאל (M1)
קדימה בקו ישר הערכים הם דוגמה לשני מספרים שווים.	100	100
אחורה בקו ישר הערכים הם דוגמה לשני מספרים שווים ושליליים	-100	-100
פנייה שמאלה במקום (סביב נקודה במרכז הרובוט) הערכים הם דוגמה לשני מספרים שווים בגודל אבל מספר אחד שלילי והשני חיובי	100	-100
פנייה ימינה במקום (סביב נקודה במרכז הרובוט) הערכים הם דוגמה לשני מספרים שווים בגודל אבל מספר אחד שלילי והשני חיובי	-100	100
פנייה שמאלה רגל ציר (גלגל ימני לא נע) הערכים הם דוגמה לשני מספרים שווים בגודל אבל מספר אחד חיובי והשני 0	100	0
פנייה ימינה רגל ציר (גלגל שמאלי לא נע) הערכים הם דוגמה לשני מספרים שווים בגודל אבל מספר אחד חיובי והשני 0	0	100
פנייה שמאלה בקשת בכיוון קדימה מנוע ימני במהירות גדולה ממנוע שמאלי, שתי המהירויות > 0	80	20
פנייה שמאלה בקשת בכיוון אחורה מנוע ימני במהירות גדולה ממנוע שמאלי, שתי המהירויות < 0	-80	-20
פנייה ימינה בקשת בכיוון קדימה מנוע ימני במהירות גדולה ממנוע שמאלי, שתי המהירויות > 0	20	80
פנייה ימינה בקשת בכיוון אחורה מנוע ימני במהירות גדולה ממנוע שמאלי, שתי המהירויות > 0	-20	-80

בהתאם לנתוני הרובוט שמופיעים בטבלה נתרגל את התנועות והפניות הבאות:

- תנועה קדימה
- תנועה אחורה
- פנייה ימינה
- פנייה שמאלה
- תנועה קדימה ופנייה ימינה
- תנועה קדימה ופנייה שמאלה
- תנועה אחורה ופנייה ימינה
- תנועה אחורה ופנייה שמאלה

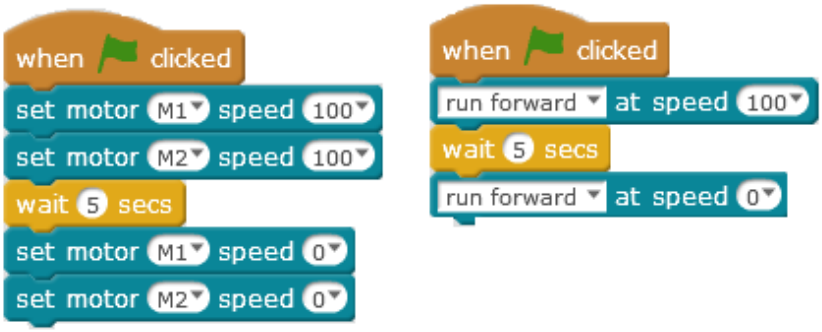
תנועה קדימה/אחורה בקו ישר

אם שני המנועים השמאלי והימני מקבלים ערך חיובי שווה הרובוט ינוע בקו ישר קדימה

1. L<== 100
2. R<== 100
3. **'המתן 5 שניות'**
4. L <== 0
5. R <==0

התכנית בסביבת סקרטץ'

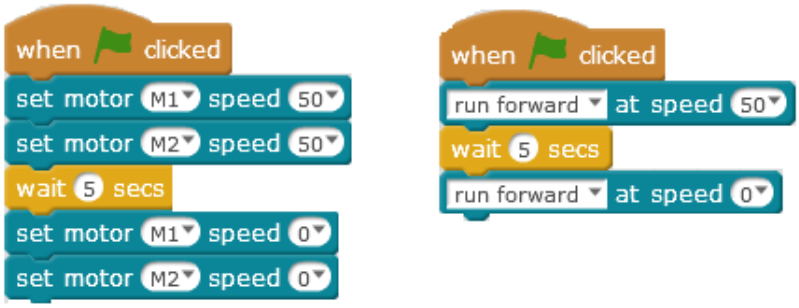
שתי תוכניות אפשריות למימוש האלגוריתם, איזו תכנית מתבצעת בדיוק לפי האלגוריתם המילולי?



שים לב, העוצמה/מהירות של המנועים בהוראות נקבעה ל- 100

הרץ את התכנית עם עוצמה שונה מ-100 אך עם עוצמה שווה בשני המנועים, וראה שהתנועה של הרובוט עדיין בכיוון ישר קדימה.

דוגמה לפתרון המטלה עם מהירות 50 בכל אחד מהמנועים

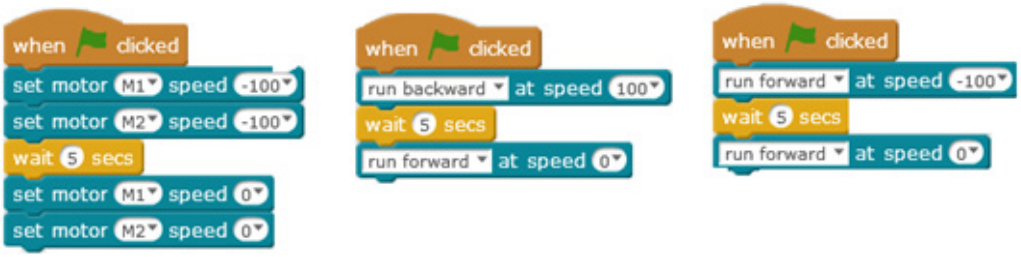


אם מעוניינים שהרובוט ייסע לאחור, זו התכנית בסביבת הרובוט הדמיוני שלנו:

1. L <== -100
2. R <== -100
3. **'המתן 5 שניות'**
4. L <== 0
5. R <== 0

התכנית בסביבת סקרטץ'

את האלגוריתם הזה ניתן לממש בסביבת סקרטץ' בשלושה תסריטים



תרגיל

א. שנו את התכניות של התנועה קדימה ואחורה על ידי שינוי עוצמות המנועים, שימו לב שהעוצמה תהיה שווה בשני המנועים.

ב. שנו את התכניות של התנועה קדימה ואחורה על ידי שינוי זמן ההמתנה, בדקו מהו המרחק שהרובוט שלכם מבצע בשנייה.

ג. שנו את התכניות של התנועה קדימה ואחורה עם שינוי של עוצמות המנועים וגם של הזמנים של התנועה, בדקו מהו המרחק שהרובוט שלכם מבצע בכל תכנית ורשמו את המרחק.

שאלת אתגר

האם אתם יכולים לחשב את מהירות הרובוט (מהירות היא: סנטימטר לשנייה)

פניות

אפשרויות היגוי – פנייה בקשת

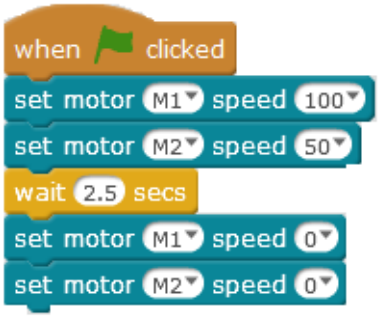
פנייה בקשת שמאלה	פנייה בקשת ימינה
מנוע שמאלי בעוצמה 50 מנוע ימני בעוצמה 100	מנוע שמאלי בעוצמה 100 מנוע ימני בעוצמה 50

אלגוריתמים לפניית קשת

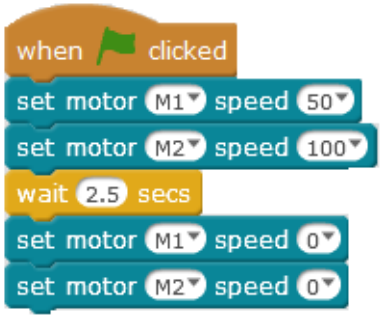
פנה קשת ימינה	פנה קשת שמאלה
1. L< == 100 2. R<== 50 3. **'המתן 1 שניות'** 4. L<== 0 5. R<==0	1. L<==50 2. R <==100 3. **'המתן 1 שניות'** 4. L <==0 5. R <==0

יישום הדוגמאות:

פנייה בקשת ימינה

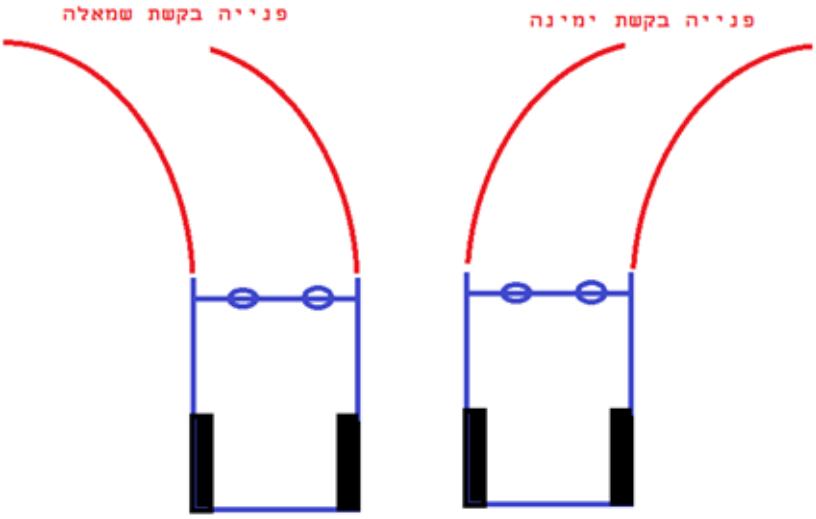


פנייה בקשת שמאלה



סיכום כללי פנייה בקשת

בכל תנועה בה יש פער בין הגודל של העוצמות של המנועים (M1,M2) , ואף עוצמה אינה שווה ל-0, הרובוט יעשה פנייה בקשת – כיוון הפנייה יהיה לכיוון הצד של הגלגל שמחובר למנוע האיטי



חשוב ביותר!! כיוון הפניה הוא תמיד לצד של המנוע האיטי יותר, כלומר המנוע שמבצע את הקשת הקטנה יותר

תרגיל

כתוב אלגוריתם בו הרובוט:

ינוע בקו ישר במשך 3 שניות

ינוע בקשת ימינה במשך 2 שניות

ינוע בקו ישר במשך 2 שניות

ינוע בקשת שמאלה במשך 4 שניות

העבר את האלגוריתם לתכנית בסביבת העבודה והרץ את התכנית.

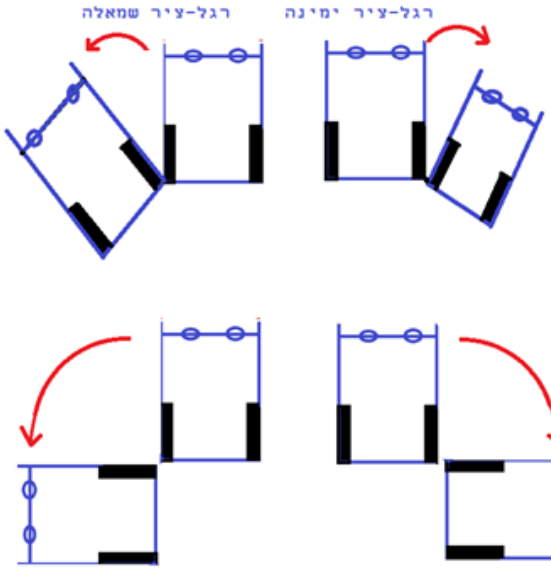
אפשרויות היגוי – פניית רגל ציר

אלגוריתמים לפניית רגל-ציר

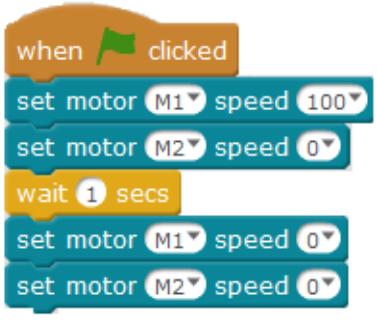
פנה רגל ציר ימינה	פנה רגל ציר שמאלה
1. L< == 100	1. L<== 0
2. R<== 0	2. R <==100
3. **'המתן 1 שניות'**	3. **'המתן 1 שניות'**
4. L<== 0	4. L <==0
5. R<==0	5. R <==0

סיכום כללי פניית רגל ציר

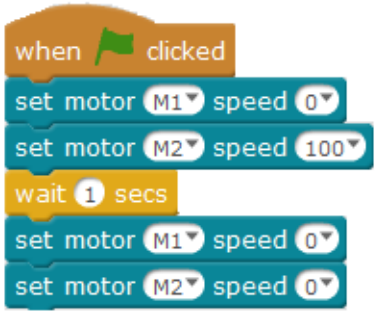
אם עוצמתו של אחד מהמנועים שווה 0, הפנייה נקראת פניית 'רגל ציר'. הפנייה מתבצעת סביב הגלגל שאינו נע. כיוון הפנייה הוא לכיוון המנוע שעוצמתו שווה ל-0



פנייה רגל ציר ימינה



פנייה רגל ציר שמאלה



הערה: אם מעוניינים שהרובוט יפנה ב-90 מעלות שמאלה או ימינה יש לשנות את הערכים של העוצמות בשני המנועים ואת משך זמן הפעלת המנועים.

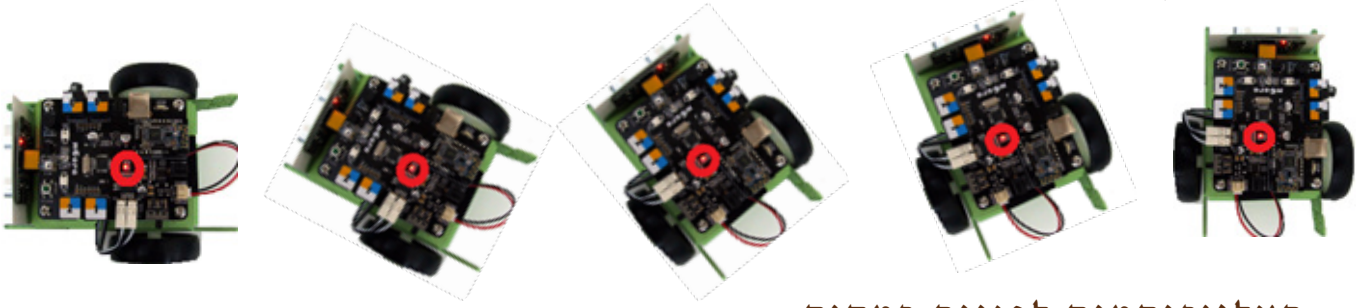
בדוגמה עבור עוצמה 50 במנוע אחד ו- 100 במנוע שני והמתנה של 1 שנייה – קבלנו פנייה כמעט מדויקת של 90 מעלות.

אפשרויות היגוי – פנייה במקום

עבור רובוט סיבוב במקום הוא סיבוב סביב נקודת המרכז – בקדמת הרובוט מעל חיישן אור עוקב פס שחור.

כך הסיבוב נראה בשלבים:

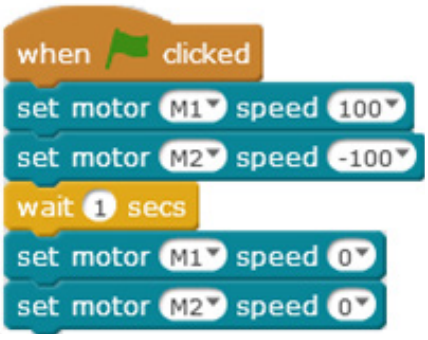
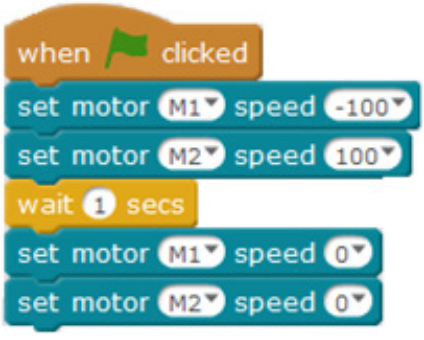
- הנקודה האדומה היא הנקודה סביבה הרובוט יבצע את הסיבוב
- שינוי כיוון בסיבוב סביב מרכז הרובוט: שני המצבים לפני ואחרי

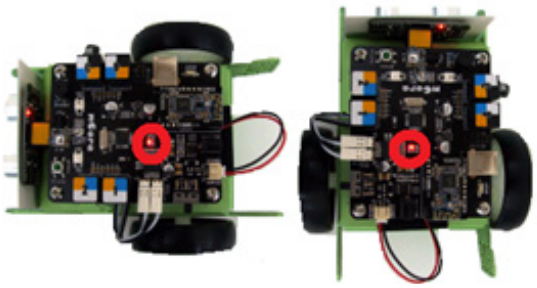


האלגוריתמים לפנייה במקום:

פנה במקום ימינה	פנה במקום שמאלה
1. L< == 100	1. L<==-100
2. R<== -100	2. R <==100
3. **'המתן 1 שניות'**	3. **'המתן 1 שניות'**
4. L<== 0	4. L <==0
5. R<==0	5. R <==0

התכנית בסביבת העבודה

פנה במקום ימינה	פנה במקום שמאלה
	



הערה: ככל שעוצמות המנועים יהיו עם ערך גבוה יותר, מהירות הפנייה תהיה גבוהה יותר. עבור הערכים בתסריטים שכתבנו הפנייה היא כמעט 90 מעלות, אם מכניסים ערכים כגון: 255, 255- הרובוט מבצע מספר סיבובים במקום. ולכן ביישום מעשי יש לבחון את הערכים שאנו מכניסים לפקודה הפעלת המנוע, בשילוב הערך של ההוראה **'המתן _שניות'**

מטלות

מטלת אורות וצלילים

כתוב תסריט שמניע את הרובוט קדימה 3 שניות, עם סיום התנועה הרובוט משמיע צליל, ומדליק את שני הלדים על הלוח בצבע ירוק למשך 2 שניות.

אלגוריתם פתרון כללי

- **'סע קדימה'** 20 סנטימטר
- **'עצור'**
- **'השמע צליל'**
- **'הדלק נורות'** על הלוח בצבע ירוק

ניתוח תתי-המשימות שיש לבצע:

- נסיעה קדימה 3 שניות:
  - שני מנועים במהירות 100
  - המתנה של 3 שניות
  - שני מנועים במהירות 0 (עצירה)
- השמעת צליל:

בקבוצת ההוראות של הרובוט יש הוראה להשמעת צליל, נשתמש בהוראה הזו

- הדלקת שתי נורות (לדים) על הלוח בצבע ירוק:

יש לנו הוראה בקבוצת ההוראות של הרובוט שמדליקה את שתי נורות הלוח בצבע הנדרש

אלגוריתם מפורט:

R<===100

L<===100

'המתן' 3 שניות

R<===0

L<===0

'השמע צליל'

'הדלק נורות' בצבע ירוק למשך 2 שניות

'כבה נורות'

תזכורת: הוראת השמעת צליל בקבוצת ההוראות ROBOTS 

(מומלץ לשנות את ברירת המחדל לצליל הרצוי לך).

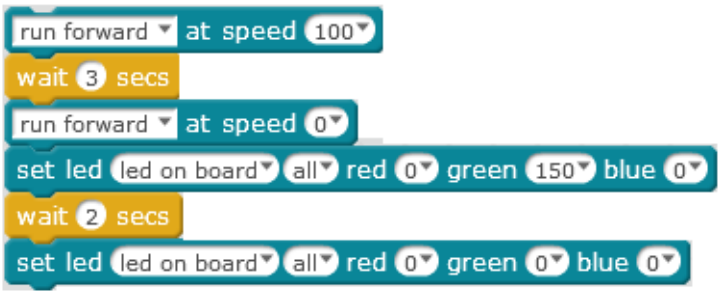
הוראת הדלקת לדים (נורות) על הלוח הראשי של הרובוט





(שינוי ברירות המחדל קובע את הצבע של האור ; ערך המספר קובע את עוצמת הצבע)

התסריט הסופי – השתמשנו בהוראה אחת לתנועה קדימה במקום הוראה לכל מנוע



בעיית חשיבה

המטלה שבצענו שונתה ועתה הרובוט צריך לנוע למרחק מסוים – במקום תנועה קדימה במשך 3 שניות עליו לנוע למרחק 20 סנטימטר.

רמזים לפתרון

- חקור, במהירות שתבחר, בכמה זמן נסיעה הרובוט מבצע סיבוב אחד של הגלגל
- חשב מהו היקף הגלגל – ניתן לחשב את ההיקף בעזרת נוסחה או לבצע מדידה עם חוט מה היקפו של הגלגל
- חשב כמה סיבובים של גלגלי הרובוט נדרשים כדי לעבור מרחק של 20 ס״מ
- חשב מהו מספר השניות שיש לכתוב בתסריט בעבור זמן הנסיעה של הרובוט – כדי שהרובוט יבצע נסיעה של 20 ס״מ

כתיבת יישום עם הרובוט וקבוצת הוראות 'אירוע' של סקרטץ'

עתה נכתוב יישומים המשתמשים בשאר קבוצות ההוראות של סקרטץ' בשילוב עם קבוצת ההוראות ROBOTS

מטלה – 'מכונית על שלט'

שליטה על תנועת הרובוט באמצעות חצים :

חץ למעלה – רובוט נע קדימה

חץ מטה – רובוט נע אחורה

חץ ימינה – רובוט פונה ימינה

חץ שמאלה – רובוט פונה שמאלה

נשתמש בהוראת אירוע של סקרטץ' **'כאשר נלחץ מקש'** :

אלגוריתמים כללים של המטלה:

'**כאשר נלחץ מקש**' חץ מעלה

'**נוע קדימה**'

'**המתן** 1 **שנייה**'

'**כאשר נלחץ מקש**' חץ מוטה

'**נוע אחורה**'

'**המתן**' 1 שנייה

'**כאשר נחלץ מקש**' חץ שמאלה

'**פנה ימינה**'

'**המתן** 0.2 **שנייה**'

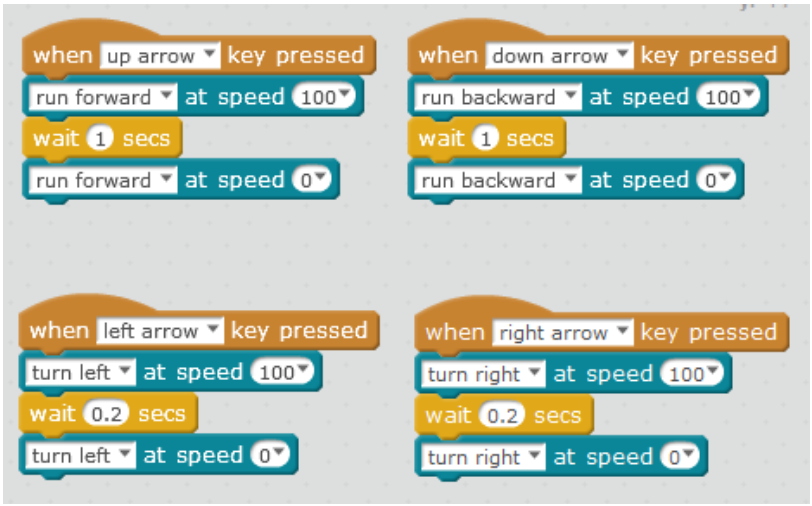
'**כאשר נלחץ מקש**' חץ ימינה

'**פנה ימינה**'

'**המתן** 0.2 **שנייה**'

הערה: מהירות המנועים בפנייה ומשך זמן הנסיעה אחורה, קדימה ובפניות – יקבע על פי התנסות ולפי העדפות של כל תלמיד שמריץ את היישום.

התסריטים בסביבת העבודה:



הפעלת התסריטים:

עתה יש לנו 'מכונית על שלט', כל לחיצה על אחד ממקשי החצים גורם לרובוט לנוע בהתאם למקש חץ שנלחץ.

אם עובדים בסביבת BLUETOOTH, הרובוט נע חופשי על ריצפת הכיתה ללא כבל המתחבר למחשב.

שיפורים

שפר את הפרויקט באמצעות הוספת הוראת השמעת צליל הזהרה כאשר הרובוט נוסע אחורה או הדלקת לדים ירוקים כאשר הרובוט נוסע בבטחה קדימה.

שילוב תנועת הדמות על הבמה עם תנועת הרובוט

המטרה בשיפור זה היא לעקוב אחר מסלול הנסיעה של הרובוט. נעשה זאת על ידי שילוב של תנועת הרובוט עם תנועת דמות במסך (עם ההוראה עט מטה), כך נוכל לדעת מהו מסלול הנסיעה שביצע הרובוט.



התסריטים המשולבים

התסריטים לתנועה של הרובוט משולבים עם תנועה של דמות על הבמה. הדמות שנבחרה היא דמות של mBot רובוט. הפעלת התסריטים יוצרת שילוב מעניין בין הדמות על המסך לתנועת הרובוט האמיתי.

כל תלמיד יכול לשפר את הפרויקט על ידי הוספת צלילים של הרובוט האמיתי וצלילים של הדמות על המסך או עם הוספת אורות לרובוט האמיתי בעת התנועה.

פרויקט זה מסיים בצורה נפלאה ומהנה את הפרק על תנועת הרובוט ופניות.

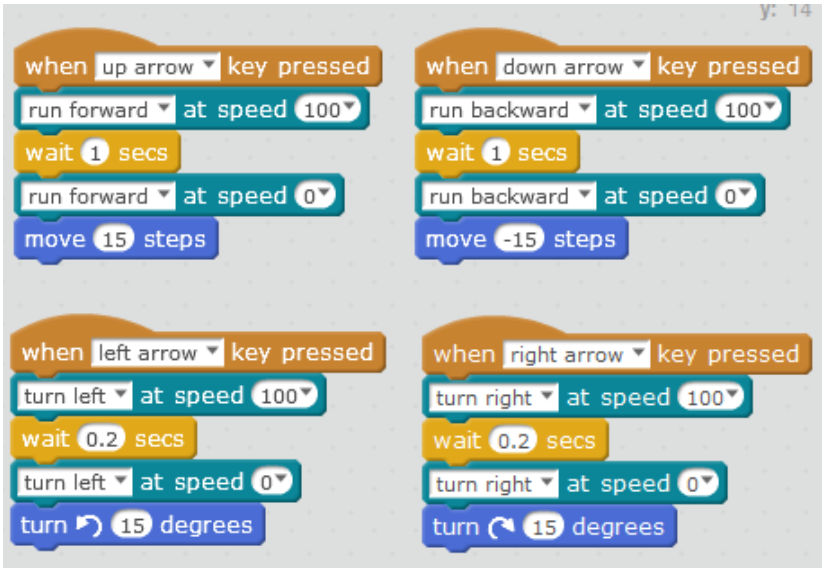
הדמות של הרובוט על הבמה (דמות של סקרטץ')



MBOT

התסריטים של הדמות:





הבמה לאחר הפעלה קצרה של התסריטים:

הדמות של הרובוט בסביבת סקרטץ' נעה בהתאם להקשה על החצים. הרובוט "האמיתי" נע על משטח במקביל. אנחנו רואים על המסך איזה מסלול מבצע הרובוט "האמיתי".

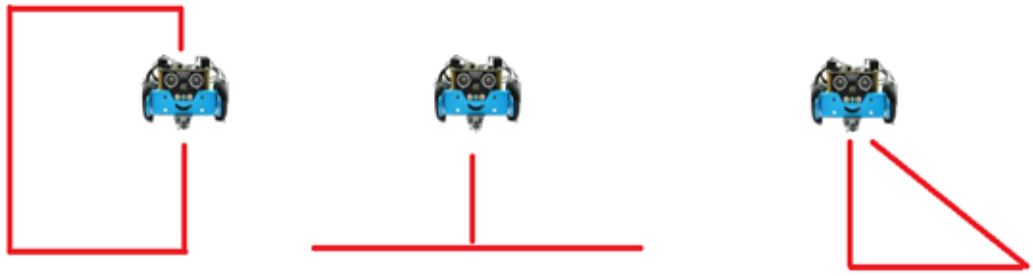
דרכי הערכה

א. מאוד חשוב שהתלמידים יבינו ויפנימו את הרעיונות סביב פניות הרובוט. יש לבחון את רכישת המיומנות לביצוע פניות עם הרובוט – איך פונים עם הרובוט לכיוון הרצוי ואופן הפנייה.

לכתוב מבדקים עם תסריטים קצרים: תחילה להציג את כיוון הרובוט ואחר לשאול את התלמידים מה יהיה כיוון הרובוט לאחר ביצוע התסריט. הדרישה היא לזהות את סוג הפנייה – לא נדרש לזהות את הכיוון המדויק של הרובוט כי זה תלוי במהירות המנועים ובזמן ההפעלה.

ב. לתת מטלות של מסלולי נסיעה שנדרשים מהרובוט, ולבקש לתכנן את המסלול, לכתוב אלגוריתם למסלול הנסיעה ולממש אותו בסביבת העבודה.

מסלולי נסיעה לדוגמה:



ג. חשוב לשלב בתסריטים הוראות 'צלילים' ו'אורות'. כמו כן מומלץ להשתמש בהוראות מקבוצות 'אירועים' בסקרטץ' לאתחול התסריטים (**'כאשר לוחצים על דמות'**, **'כאשר נלחץ מקש'**)

ניתן ליצור תסריט מאתחל שמשדר מסר לתסריט אחר שמניע את הרובוט.

כללי

השימוש היותר מקיף בהוראות מקבוצות ההוראות הכלליות של סקרטץ' תלוי בהתקדמות הלמידה במודול מדעי המחשב.

במודול רובוטיקה התכנית נכתבת מתוך הנחה שתלמידים אינם לומדים במקביל את 'מודול מדעי המחשב', ולכן ניתן הסבר על 'הוראות' ובמיוחד על מבני הבקרה: **'אם'**, לולאת **'לעולמים'**, לולאת **'חזור'**. אנחנו משתמשים במבנים אלו ברמה בסיסית: שימוש מצומצם ב'משתנים' וגם זאת רק בשלבים מאוחרים, וכך גם השימוש בתנאים לוגיים עם קמטים לוגיים (וגם ;או ; לא), והשימוש בלולאה **'חזור עד ש'**. כל שימוש במבנה תכנות וכתיבה אלגוריתמית מוסבר כאן ב'תכנית רובוטיקה' על בסיס ההנחה שיש לעשות זאת 'לאט ובזהירות', אולם אם התלמידים שאתם המורים מלמדים למדו או לומדים במקביל ללמידת רובוטיקה נושאים מתקדמים ב'מודול מדעי המחשב', אתם רשאים ויתרה מכך מומלץ לאפשר לתלמידים ליצור רעיונות ואתגרים יותר מורכבים הניתנים למימוש בעזרת מבני תכנות ואלגוריתמיקה יותר מורכבת.

כך שניתן ללמד את התכנית במדרג של רמות: מורה שמעוניין ברמת בסיס יעביר את התכנית עם יישום מינימלי של משתנים, ללא אופרטורים לוגיים ומומלץ לוותר על האופרטור **'לא'** ועל לולאת **'חזור עד ש'**. ברמת הביניים וברמה הגבוהה ישולבו מרכיבים אלו אבל גם במידה, רק אם נדרש ולא ניתן להסתדר ללא שילובם. יש בתכנית דוגמאות רבות וכל מורה יבחר להציגן בהתאם לרמת הכיתה. היקף התכנית והעברתה באופן מלא עם כל הדוגמאות חורג מ-70 שעות, ולכן לכל מורה בכל בית ספר ישנן אפשרויות לבחירה ולהתאמה של התכנית להיקף 60 שעות.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
תנועה קדימה ; תנועה אחורה ; פנייה בקשת ; פנייה שמאלה וימינה ; פנייה במקום ; פניית רגל ציר	הרעיון המרכזי בתנועה בקו ישר או פנייה בכל רובוט הוא שינוי בעוצמה/ מהירות המנועים. בתכנית הלימודים יש שתי שיטות לתנועה ופניות : הוראה לשני המנועים ללא הפרדה וללא ערכים שלילים ; הוראות נפרדות לכל מנוע עם שילוב ערכים שליליים. ערך שלילי יוצג כשינוי בכיוון ולא בהיבט של מספרים מכוונים.	✓	✓	✓	הנעת הרובוט במסלולים שונים ; שליטה על פניות של הרובוט לפי סוגי הפניות ; כתיבת אלגוריתמים ומימושם בסביבת העבודה. **2 שעות**
שילוב הוראות מקבוצת ROBOTS (השמעת צלילים, הדלקה וכיבוי לדים)	שילוב הוראות של צלילים והדלקת נורות מציג לתלמידים פלטים מסוגים שונים כולל : פלט קולי ; פלט אור ; פלט סיבוב מנוע. גיוון הפלטים מסייע להבנת הרעיון של מערכת עם קלט. עיבוד ופלט.	✓	✓	✓	שילוב פלט, כולל צלילים והדלקת נורות בתסריטי תנועה של הרובוט. **2 שעות**

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
שילוב הוראות מקבוצת 'אירועים' בתסריטים לתנועה של הרובוט	הרחבה ובנייה של יישומים בגישה כוללת; כתיבת אלגוריתמים מורכבים לפני מימוש; שילוב תכנות מונחה אירועים בתכנות הרובוט.	✓	✓	✓	תכנון ומימוש יישומים לכלל פרויקט מורכב ומעניין שכוללים: הוראות מקבוצת אירועים; הוראות פלט; הוראות צלילים ואורות; הוראות תנועה. **2 שעות**
שילוב הוראות לדמות של סקרטץ' עם הוראות לרובוט	בתסריטים המשלבים הוראות לרובוט והוראות לדמות(למשל מקבוצת תנועה) אנו יכולים ליצור הדמיה גרפית של התנהגות ותנועה של הרובוט.	✓	✓	✓	כתיבה בצוות של יישום מבוסס אירועים שעושה שימוש משולב בהוראות לרובוט ובהוראות לדמות. **2 שעות**

פרק ה' כתיבה של אלגוריתם מילולי, תכנון ויישום (מרעיון לקוד)

תחומי תוכן: חישוביות, אלגוריתמיקה ומיומנויות המאה 21, כתיבת קוד

מושגים

- ✓ אלגוריתם
- ✓ פסיאודו -קוד
- ✓ משפט קלט
- ✓ משפט תנאי מצומצם
- ✓ משפט תנאי מורחב
- ✓ לולאה מבוקרת
- ✓ מודל 'תכנון 'מעלה-מטה'
- ✓ מודל תכנון 'מטה-מעלה'

אלגוריתם

אלגוריתם הוא דרך שיטתית (כלומר כזו שצעדיה מוגדרים היטב) לביצוע של משימה מסוימת במספר סופי של צעדים.

פסיאודו - קוד

כתיבת הוראות לתכנית מחשב שאינן כתובות בשפת תכנות מסוימת אלא בניסוח כללי בעברית (תלמידים באנגליה יכתבו באנגלית, בצרפת בצרפתית וכדומה). כאשר מעבירים את התכנית-אלגוריתם למחשב מבצעים זאת באמצעות שפת התכנות בה אנו משתמשים.

משפט קלט

הוראת קלט שמטרתה לקבל קלט מהמשתמש בתכנית המחשב, בדרך כלל ברירת המחדל היא קלט מהמקלדת.

משפט תנאי מצומצם

משפט תנאי ללא האפשרות 'אחרת'.

משפט תנאי מורחב

משפט הכולל את ההוראות לשתי תוצאות התנאי: 'אם הוא מתקיים' ו'אם לא מתקיים'.

לולאה מבוקרת

לולאה שמתבצעת מספר פעמים מוגדר מראש

מודל תכנון 'מעלה-מטה' (הרחבה בהמשך)

במודל 'מעלה-מטה' מנוסחת תחילה סקירת מערכת בקווים כלליים בלי להיכנס לפרטים של שום חלק ממנה. לאחר מכן מנוסח כל חלק של המערכת לפרטיו.

מודל תכנון 'מטה-מעלה' (הרחבה בהמשך)

במודל 'מטה-מעלה' מוגדרים תחילה החלקים הפרטיים של המערכת לפרטיהם, ואז מצורפים החלקים יחדיו לרכיבים גדולים יותר.

מבוא

כל נושא אלגוריתמיקה מילולית מופיע בפרק זה כולל נושאים שלא נלמדו. ניתן ללמד חלק מנושאי הפרק גם בשלבים יותר מוקדמים וניתן ללמד את הפרק על פסיאודו-קוד עם לולאות לקראת סוף תכנית הלימודים (סוף שנת הלימודים של כיתה ד'), או בשנת הלימודים של כיתה ה'. חשוב לרכז את הנושא בפרק אחד ולא לפרוס את הנושא לפרקים שונים בתכנית הלימודים – הלמידה והיישום של הנושאים צריכה להיות מתואמת ומקבילה להתקדמות בלמידת הנושאים ברובוטיקה ובשילוב של נושאים אלו עם מבני בקרה.

אלגוריתמיקה מילולית

כתיבת אלגוריתמים מילוליים לפני כתיבת קוד מהווה תהליך תומך בעבודת עיצוב וניתוח מטלה לקראת כתיבת קוד. היתרון הגדול של כתיבת אלגוריתם בפסיאודו-קוד הוא באי התלות בשפת התכנות – כתיבת תכנית בפסיאודו-קוד אינה תלויה בשפה בה ממשים את הקוד ומהווה רכיב חשוב בתהליכי תכנון ועיצוב האלגוריתם לקראת מעבר לקוד. זה חלק מהיעד שהוגדר לתכנית הלימודים הכללית – כתיבת קוד בתהליך מובנה ופשוט.

הכלי הוא כלי בסיסי בתהליכי התכנון ומשתלב עם התפיסה הכוללת שתהליכי הניתוח ועיצוב הפתרון בשילוב גישת 'מעלה-מטה' הם החשובים ביותר, בהיותם תלויים בשפה ולא תלויים בהיבטים הטכנולוגים של מימוש הפתרונות.

'מטה-מעלה' - שילוב כלים בסביבת סקרטץ'

'מעלה-מטה': שילוב בתכנית רובוטיקה

המודל הזה יתאים למימוש בתכנית הלימודים עבור בעיות מורכבות יותר ועבור פרויקטים מסוגים שונים שכוללים יותר מ-2 תסריטים לדמות. לדוגמה: רובוט נע ועוקף מכשולים, המטלה תחולק ל-3 תתי-מטלות: נסיעה, זיהוי מכשול, עקיפה. חלוקה זו מאפשרת את חלוקת תתי-המטלות בין חברי צוות; חשיבה על כל תת-מטלה בנפרד; פתרון רעיוני לכל אחת מתתי-המטלות; כתיבה אלגוריתמית; ובשלב האחרון כתיבת קוד ואינטגרציה של המרכיבים. וזו הגישה המשלבת בין מודל 'מעלה-מטה' למודל 'מטה-מעלה'.

הגישה המשולבת

אפשר לאמץ גישה משולבת שבה מתחילים את שלב התכנון במודל 'מעלה-מטה'; מגדירים את המשימה הכללית; מפרקים את המשימה הכללית לתתי משימות. בשלב התכנוני – בו כמעט כל המרכיבים הנדרשים ידועים – נוקטים בגישת 'מטה-מעלה', כותבים כל מרכיב בצורה מפורטת וכך יוצרים את היישום בצורה מפורטת.

במודל 'מעלה-מטה' מנוסחת תחילה סקירת מערכת בקווים כלליים מבלי להיכנס לפרטים של שום חלק ממנה. לאחר מכן מנוסח כל חלק של המערכת לפרטיו, ואז כל פרט חדש עשוי להיות מנוסח שוב תוך שמגדירים אותו לפרטים נוספים – עד שכל המפרט מפורט דיו כדי לתקף את המודל.

בניגוד לזאת, במודל 'מטה-מעלה' מוגדרים תחילה החלקים הפרטיים של המערכת לפרטיהם, ואז מצורפים החלקים יחדיו לרכיבים גדולים יותר שבתורם מצורפים גם הם עד להשגתה של המערכת השלמה.

מיומנויות המאה 21 ושילוב מודלים של פיתוח

ככל שגדלה מורכבות היישום (ברמת הילדים) ואנחנו נדרשים למודל בגישת 'מעלה-מטה' אנחנו גם נדרשים לתהליכי יישום מיומנויות המאה 21: מורכבות גדולה יותר של הבעיה/מטלה; יותר תסריטים; יותר משתנים וכדומה – משמעותם יותר תלמידים שמשתתפים בכתיבת היישום ולכן דרישות סביבתיות לרכוש וליישם מיומנויות המאה 21:

פירוט חלקי ומרכזי:

- יצירתיות וחדשנות
- חשיבה ביקורתית ופתרון בעיות
- תקשורת ושיתוף פעולה
- אוריינות מידע, אוריינות מדיה (שימוש במדיות שונות) – ICT

נושאי משנה בפרק:

כתיבת אלגוריתם מילולי – הרעיון המרכזי

כללי כתיבה מוסכמים

מספור הוראות, מספור משנה, הזחה

מעבר מובנה מאלגוריתם לקוד

פסיאודו-קוד - מבוא

פסיאודו-קוד (פסיאודו-אלגוריתם) – תיאור מצומצם ולא רשמי לאלגוריתם של תכנית מחשב. מיועד לקריאה על ידי **בני אדם** ולא לקריאה על ידי מחשב. לאחר כתיבת פסיאודו-קוד, ניתן לכתוב את הקוד האמיתי בשפות תכנות שונות (עם שינויים המתאימים לשפת התכנות). אין דרך פורמלית יחידה לכתוב פסיאודו-קוד – חשוב שהקורא יבין את הקוד ויוכל לתכנת בפשטות, שתישמר עקביות במהלך הכתיבה, שלכל הוראה יינתן מספר ושהמספור יתחיל ב-1.

ניסוח מוסכם של פקודות באלגוריתם מילולי בתכנית הלימודים זו:

משפט השמה:

ביטוי== > משתנה

משפט קלט:

'**קלוט**' == > משתנה

משפט תנאי מצומצם:

'**אם**' (תנאי) בצע

משפט ביצוע

משפט תנאי מורחב

'**אם**' (תנאי) בצע

משפט ביצוע א

'**אחרת**'

משפט ביצוע ב

לולאה מבוקרת

'**חזור** (ביטוי) **פעמים**'

משפט ביצוע

לולאה מותנת

'**חזור עד ש**' (תנאי)

משפט ביצוע

לולאה **לעולמים**

'**בצע לעולמים**'

משפט ביצוע

משפט ביצוע

משפט ביצוע פשוט – רק הוראה אחת

משפט ביצוע מורכב – מספר הוראות עם מספור פנימי

דוגמה לתנאי עם: משפט ביצוע פשוט; משפט ביצוע מורכב

משפט ביצוע פשוט

1. '**קלט**' ==> ציון
2. '**אם**' (ציון>90)
   2.1 '**אמור**' "כל הכבוד"

משפט ביצוע מורכב

1. **'קלט'** ==> ציון
2. **'אם'**(ציון < 56)
   2.1 ציון + 10 ==> ציון
   2.2 **'אמור'** "קבלת תוספת לציון"
3. המשך האלגוריתם.....

דוגמה ללולאה עם משפט ביצוע פשוט; משפט ביצוע מורכב

משפט ביצוע פשוט

1. **'קלט'** ==> צלילים
2. **'חזור'** (צלילים) **פעמים'**
   2.1 **'השמע צליל'**

משפט ביצוע מורכב

1. 0 ==> סכום
2. **'קלט'** ==> שחקנים
3. **'חזור'** (שחקנים) **פעמים**
   3.1 **'קלט'** ==> נקודות שחקן
   3.2 סכום + נקודות שחקן ==> סכום
3. **'הצג'** (סכום)

דרכי הערכה

1. כתיבת אלגוריתם מילולי לבעיה (קטנה).
2. מעבר מאלגוריתם מילולי לקוד.

מטרות ביצועיות

1. התלמיד ידע לרשום הוראות ביצוע פשוטות באלגוריתם מילולי
2. התלמיד ידע לכתוב אלגוריתם מילולי שכולל תנאי פשוט עם משפט ביצוע פשוט.
3. התלמיד ידע לכתוב אלגוריתם מילולי עם **'אם כן'** ו**'אם לא'**
4. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים פשוטים שכתב (לכתוב קוד לאלגוריתמים מילוליים בסיסיים).
5. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה ועם הוראה אחת שכלולה בלולאה (משפט ביצוע פשוט).
6. התלמיד ידע לכתוב אלגוריתם מילולי עם הוראה ועם מספר הוראות שכלולות בהוראה זו.
7. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים שכוללים לולאות ומשפטי ביצוע מורכבים.

8. התלמיד ידע לכתוב, בהינתן בעיה, אלגוריתם מילולי עם הוראות תנאי עם הוראות לולאה ועם משפטי ביצוע למבנים אלו.

9. התלמיד ידע ליישם בסביבת העבודה אלגוריתמיים מילוליים.

מטרות ביצועיות-דוגמאות

א. התלמיד ידע לכתוב אלגוריתם מילולי עם **'אם כן'** ו**'אם לא'**

1. קלט ==> מספר א
2. קלט ==> מספר ב
3. **'אם'** (מספר א)> (מספר ב)
   - 3.1 **'אמור'** (מספר א)
   - 3.2 **'אחרת'**
   - 3.3 **'אמור'** (מספר ב)

ב. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים פשוטים שכתב (לכתוב קוד לאלגוריתמים מילוליים בסיסיים)

האלגוריתם:	**העברה לקוד בסביבת העבודה – SCRATCH**
1. קלט== > מספר א 2. קלט ==> מספר ב 3. **'אם'** (מספר א) >(מספר ב) 3.1 **'אמור'** (מספר א) 3.2 **'אחרת'** 3.3 **'אמור'** (מספר ב)	

ג. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה ועם הוראה אחת שכלולה בלולאה (משפט ביצוע פשוט)

1. 0==> מונה
2. **'חזור לעולמים'**
   - 2.1 **'אמור'** (''אני בלולאה אינסופית'')
   - 2.2 מונה + 1 ==> מונה
   - 2.3 **'אמור'** (מונה)

ד. התלמיד ידע לכתוב אלגוריתם מילולי עם לולאה ועם מספר הוראות שכלולות בהוראה זו.

1. מונה <== 0

2. **'חזור** (4) **פעמים'**

2.1 **'אמור'** ("אני בלולאה")

2.2 מונה <== מונה + 1

2.3 **'אמור'** (מונה)

ה. התלמיד ידע ליישם בסביבת העבודה אלגוריתמים מילוליים שכוללים לולאות ומשפטי ביצוע מורכבים.

אלגוריתם מילולי :	תכנית בסביבת העבודה
1. עכברים <== 0 2. **'חזור עד ש'** (עכברים=10) 2.1 **'אם'** (חתול 'נוגע ב"עכבר) 2.1.1 עכברים <== עכברים + 1 2.1.2 **'אמור'** ("תפסתי את העכבר") 3. **'אמור'** ("תפסתי עשרה עכברים")	

ה ע ר ה : שעות הלמידה שמתוכננות לפרק זה אינן שעות 'אמיתיות'. אלו שעות שמשולבות במסגרת למידת כל פרק בתכנית. תכנית הלימודים עוסקת בשלב המרכזי והחשוב של התהליך 'מרעיון לקוד'. למידת הרעיון שיש לתכנן ולעצב את הפתרון לבעיה לפני המימוש שלה בסביבת העבודה אינה למידה ישירה וגם אינה חד פעמית – היא מתבצעת במקביל ללמידה של כלל הנושאים, בצורה מדורגת ובגישה טבעית, כחלק אינטגרלי מתהליך פתרון הבעיות.

ולכן השעות המוקצות בטבלה משולבות בפרקי הספר, וציוני הדרך משולבים לאורך פרקי הספר.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
כתיבת אלגוריתם מילולי הרעיון המרכזי	חשוב להדגיש כי מדובר בתהליך. אנו לא עוברים ישירות מתיאור בעיה לכתיבת תכנית מחשב, יש לנתח את הדרישות ולנסח באופן מילולי את הפתרון.	✓			התלמיד ינסח פתרון מילולי כללי עבור בעיה מסוימת בשפה העברית ללא כללים ברורים. **1 שעות**
כללי כתיבה מוסכמים	הצגת כללי הכתיבה ומתן דגש על ניסוח ההוראות המרכזיות כפי שמופיע במבוא לפרק.	✓		✓	התלמיד ידע לכתוב הוראות בשפת האלגוריתמיקה הכללית לפי ההנחיות שנתנו. **1 שעות**
מספור הוראות; מספור משנה; הזחה	הצגת רעיון המספור והמספור המשני.		✓		התלמיד יקבל תיאור מילולי כללי של אלגוריתמים ויעביר את האלגוריתמים לקוד. **2 שעות**
מעבר מאלגוריתם לקוד באופן מובנה	שילוב תנאים מגוונים; יחסים בין מספרים ויחסים בין דמויות. הדגשת השימוש בשפה ללא קשר לסביבת העבודה.	✓	✓	✓	התלמיד יבצע את חמשת השלבים בפתרון בעיה. א. הבנת הבעיה ב. ניתוח וחשיבה על דרכי פתרון ג. כתיבת אלגוריתם מילולי ד. העברה לקוד ה. הרצה, בדיקות ושיפורים עבור מטלות ובעיות שונות. **2 שעות**

פרק ו' בקרה בחוג פתוח ; בקרה בחוג סגור - חיישנים, לולאת 'לעולמים'

בפרק זה נתחיל להיחשף לרעיון מרכזי וחשוב ברובוטיקה – חיישנים.

אנחנו כותבים תכנית לרובוט לתנועה ועצירה ליד מכשול באופן הבא: מודדים את המרחק בין הרובוט למכשול ; מחשבים את הזמן הנדרש לרובוט להיות בתנועה ולעצור בדיוק לפני המכשול ; מריצים את התסריט ומשנים מהירות מנועים ו/או זמן תנועה עד שאנחנו מגיעים למטרה – הרובוט מתחיל בתנועה ונעצר בדיוק לפני המכשול.

שאלה

נציב את הרובוט במיקום שהצבנו (על קו המטרה) ועתה נרחיק את המכשול מרחק כלשהו, האם התכנית שכתבנו תתאים למיקום החדש של המכשול?

תשובה

לא. חישבנו את מהירות הנסיעה ואת זמן הנסיעה בדיוק לפי מרחק מסוים שהיה ידוע לנו, מרגע ששונה המרחק בין הרובוט למכשול צריך לתכנן ולכתוב תסריט חדש.

בקרה בחוג פתוח

'בקרה בחוג פתוח' – כאשר הפעולה של הרובוט מתרחשת בסביבה שבה הרובוט אינו מקבל כל קלט מהסביבה, ואופן הביצוע של התכנית שהרובוט מפעיל אינו תלוי בקבלת משוב מהסביבה. המשמעות המעשית היא שאיננו משתמשים בחיישנים של הרובוט לצורך חישוב ערכים לביצוע פעולות בתכנית שהרובוט מפעיל.

בקרה בחוג סגור

בקרה בחוג סגור – כאשר בכל פעם שהתכנית מופעלת כתוצאה מקלט המתקבל מהסביבה, תכנית הרובוט מתבצעת באופן שונה.

בבקרה בחוג סגור הרובוט מקבל 'מידע' על סביבתו מאחד או יותר מהחיישנים המחוברים לרובוט, וכתוצאה מהקלט שמגיע מהחיישנים, הרובוט פועל באופן שונה בכל פעם שהתכנית מופעלת.

הוראת נושא חיישנים לתלמידי כיתות ד', ה'



כל התייחסות לקלט לתכנית מחשב אינה עניין טריוויאלי לתלמידים בגילאים אלו. העובדה שהרובוט יפעל בצורה שונה כתוצאה מקלט אינה קלה להבנה; יש קושי לתפוס מונחים שקשורים לחיישנים; קושי בהבנת המעבר מערך אנלוגי לערכים דיגיטליים ; ההבנה שהערכים שהחיישן דוגם מתקיימים לאורך זמן פעולת התכנית בצורה רציפה היא הבנה מאתגרת ; וגם התהליך המקביל שבודק כל העת האם ערך מסוים מקיים תנאי כלשהו, אינו פשוט. ולכן ההוראה של שילוב חיישנים ובאופן כללי עבודה בסביבה של בקרה בחוג סגור, היא מורכבת להבנה ומורכבת יותר להוראה וללמידה.

נושא החיישנים נלמד באופן משולב עם רעיונות מרכזיים במדעי המחשב – בתהליך התכנון של הפתרון וביישום בשלב האחרון בקוד. אנו מתוודעים יחד עם התלמידים למבני הבקרה של משפט תנאי פשוט ושל משפט תנאי מורכב, למבנה

לולאה (**'לעולמים'**, **'חזור** מספר **פעמים'**, **'חזור עד ש'**), ולשילוב מבני בקרה אלו יחד עם נושא ומושג מרכזי בכל שפת תכנות – משתנה. ובנוסף עולם החיישנים יוצר קשיים פדגוגיים ודידקטיים לא קלים. האסטרטגיה הפדגוגית-דידקטית של התכנית היא להמעיט בשימוש בתנאי **'אם ואם לא'** ולהעדיף משפטי תנאי בסיסיים נפרדים, להמעיט בשימוש בתנאים עם אופרטורים לוגיים 'וגם', 'או' ולדחות שימוש זה ככל האפשר, להמעיט בשימוש במשתנים בשילוב חיישנים וגם זאת רק אם חייבים ובשלב מאוחר ככל האפשר של התכנית. במקום שימוש במשתנים נשתמש בערך המתקבל ישירות מחיישן ולא נעביר אותו למשתנה. כמו כן נעשה שימוש בלולאת **'לעולמים'**, ורק בשלב מתקדם נשתמש בלולאת **'חזור'**, ושימוש מותנה ותלוי ברמת הקבוצה בלולאה **'חזור עד ש'**.

חיישן מרחק

חיישן המרחק מעביר לנו מידע על המרחק בין הרובוט לבין עצם כלשהו. במרבית הרובוטים חיישן המרחק ממוקם בחזית הרובוט כך שהמרחק המתקבל מהחיישן מייצג את המרחק בין הרובוט לעצם.

ברובוט mBot חיישן המרחק ממוקם בחזית הרובוט. אנחנו יכולים להבחין ב'שתי עיניים' שהן הרכיבים המרכזיים של החיישן.

חיישנים הם רכיבי קלט – מעבירים לרובוט מידע מ'העולם' שמחוץ לרובוט, כל חיישן מחובר לכניסה של קלט, כלומר חיישנים מחוברים עם כבל מיוחד לרובוט. כל כבל מחובר לכניסה של קלט של הרובוט ולכל כניסה יש מזהה-שם. כשרוצים לדעת מהו המרחק בין הרובוט לעצם צריך 'לקרוא' את המידע מהכניסה המסוימת אליה חובר החיישן.

ערוצי הקלט לחיישנים ברובוט – כללי

עלינו לבדוק כל חיישן שמחובר לרובוט שבו אנחנו משתמשים בתכנית הלימודים ולבדוק לאיזו כניסה מחובר החיישן. שפת התכנות של הרובוט כוללת הוראות לקריאת הנתונים המגיעים מהחיישן וכל הוראה של חיישן מתייחסת לסוג החיישן ולזיהוי הכניסה-ערוץ ברובוט אליו מחובר החיישן המסוים.

חיבור חיישן מרחק- אולטראסוני

החיישן ברובוט שלנו מחובר לערוץ הכניסה: PORT3

תסריט ניסוי קצר

- נעמיד את הרובוט לפני עצם
- נקרא את המרחק של הרובוט מהעצם ונבקש מהדמות באמצעות ההוראה **'אמור'** להציג את המרחק
- נשתמש עם מקש חץ למעלה ונקדם את הרובוט לעבר העצם
- נקרא שוב את המרחק של הרובוט מהעצם ונבקש להציגו

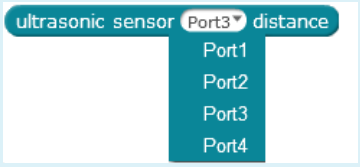
אופן הקריאה של המרחק מעצם באמצעות חיישן המרחק 

באמצעות 'המפעיל' שכלול בקבוצת ההוראות ROBOTS אנחנו מקבלים את המרחק בין עצם לרובוט. יש לשים לב למרכיבים במפעיל:

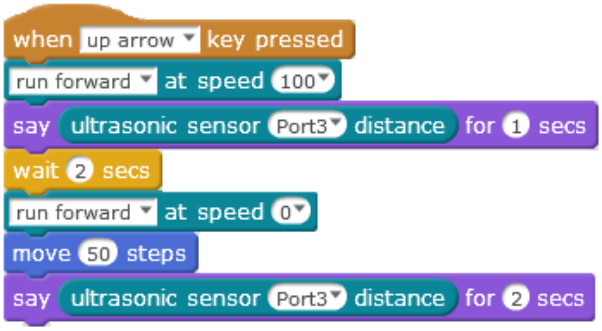
שם החיישן - ULTRASONIC SENSOR DISTANCE

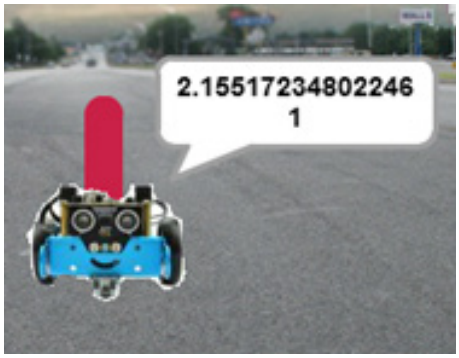
שם מזהה הכניסה על הרובוט אליה מחובר החיישן - PORT3

חשוב! יתכן וברובוט שקבלתם בבית הספר החיישן מחובר לערוץ כניסה אחר. ניתן לעקוב אחרי כבל החיבור בין החיישן לערוץ ולראות לאן החיישן מחובר, ובהתאם לבחור בחיישן שמפעיל את הערוץ הנכון – באמצעות שימוש בגלילה שבהוראה לבחירת ערוץ:



אם ברשות בית הספר רובוט של ספק אחר צריך לבדוק כיצד משנים את זיהוי הערוץ , וכן איך מחברים את החיישן לערוץ שונה במידה ומעוניינים בכך.

אלגוריתם להצגת המרחק בין הרובוט לעצם	התסריט
1. **'הצג'** את המרחק בין הרובוט לעצם 2. **'סע קדימה'** 3. **'עצור'** 4. **'הצג'** את המרחק בין הרובוט לעצם	

המרחק של הרובוט מהעצם בתחילת התסריט	המרחק של הרובוט מהעצם בסוף התסריט
	

חשוב! בתסריט שולבה תנועה של דמות הרובוט על המסך (עם יצירת קו התנועה) יחד עם תנועה של הרובוט האמיתי.

מושגים

לולאה 'לעולמים' – מבנה בקרה המאפשר ביצוע חוזר ללא סיום של קבוצת פקודות

לולאות הרעיון וייישומו – הרחבה

לולאה היא מבנה בקרה, כלומר מבנה המארגן את זרימת ביצוע פקודות בתכנית מחשב/תסריט. הלולאה מאפשרת חזרה על קבוצת פקודות מספר פעמים:

לולאה '**לעולמים**' – הלולאה תתבצע אינסוף פעמים ובסביבת סקרטץ' ניתן לעצור את הלולאה בלחיצה על 'עצור' שמופיע בחלק העליון של הזירה. בדרך כלל מדעני מחשב ומתכנתים לא משתמשים במבנה של לולאה אינסופית- '**לעולמים**', לעיתים זה אירוע של שגיאה שכותב הקוד לא הבחין שתנאי הלולאה לא מתקיים ויש ביצוע אינסופי. בשפת סקרטץ' ובמיוחד בתכנות משחקים ואנימציות יש שימוש יותר תדיר בלולאה '**לעולמים**'. אנחנו נשתמש בלולאת 'לעולמים' במסגרת תכנית רובוטיקה ובמיוחד בהקשר לנושא חיישנים.

דוגמאות לולאה 'לעולמים'

רובוט דמיוני / רובוט אמיתי

בסביבת הרובוט mBot ובסביבת סקרטץ' אפשר ליישם מרכיב מעניין באופן הבא:

בוחרים דמות בסביבת סקרטץ' מתוך תמונות של הרובוט – זו תהיה הדמות שלה נשייך את התסריטים: גם את תסריטי הסקרטץ' ללא שימוש בקבוצת ההוראות לרובוט, וגם תסריטים עם הוראות לרובוט.

עתה כאשר נותנים הוראות תנועה לדמות מתוך קבוצת ההוראות 'תנועה', הדמות נעה על המסך. אותו אלגוריתם-קוד נכתב עבור תנועה של הרובוט על הרצפה, אלא שבמקום שימוש בהוראות מקבוצת 'תנועה' משתמשים בהוראות למנועים – כך יש לנו תסריט המניע דמות על המסך ותסריט המניע רובוט על הרצפה!

זה רעיון רב עוצמה כי תנועת הדמות על המסך היא סוג של הדמיה. התלמידים יכולים לכתוב אלגוריתם מילולי וליישם אותו עבור הדמות בסביבת סקרטץ' ואם התכנית מתבצעת לפי התכנון, אפשר לשנות אותה על בסיס האלגוריתם והקוד – לקוד עבור הרובוט.

יש ליתרון זה השלכות פדגוגיות נרחבות: הרובוט על המסך יקרא 'הרובוט הדמיוני' והרובוט על הרצפה (או כל משטח אחר) יקרא הרובוט האמיתי. הרעיון וסביבות עבודה לתכנון ותכנות רובוטים באמצעות סביבה וירטואלית קיימים היום גם ברובוטים בהם עושים שימוש כללי, וגם ברובוטים לצורכי למידה.

כאן היישום פשוט יותר וללא תשלום.

הרובוט הדמיוני מתרוצץ על הבמה ללא מנוחה ומשרטט את תנועתו:

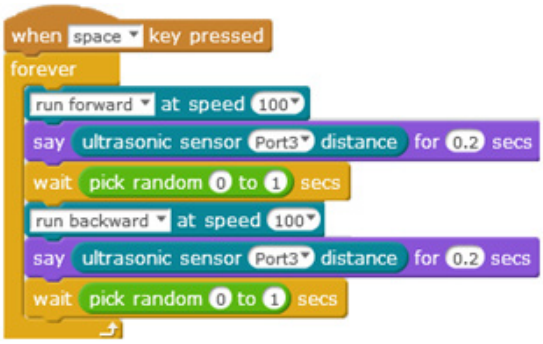


חשוב! השתמשנו בפקודה 'גלוש' במפעיל, ליצירת מספרים אקראיים,



הערכים שבחרנו מזיזים את הדמות לכל נקודה על הבמה.

עתה נניע את הרובוט האמיתי שלנו בתנועה אינסופית קדימה ואחורה אל מול עצם, ונבקש מהדמות הדמיונית של הרובוט "לדווח" לנו מהו המרחק של הרובוט מהעצם.

האלגוריתם	התסריט בסביבת העבודה:
1. **'לעולמים'** 1.2 **'נוע קדימה'** במהירות 100 1.3 **'אמור'** מרחק הרובוט מהעצם 1.4 **'המתן'** זמן אקראי (בתחום 0-1) 1.5 **'נוע אחורה'** במהירות 100 1.6 **'אמור'** מרחק הרובוט מהעצם 1.7 **'המתן'** זמן אקראי (בתחום 0-1)	

מטלות

א. אנחנו רוצים להכין מופע אינסופי של אורות וצלילים, נשתמש ליישום הרעיון ב:

1. לולאת **'לעולמים'.**
2. על המסך הוראות הדלקת נורה על הרובוט.
3. על המסך הוראות השמעת צלילים של הרובוט.
4. על המסך מפעיל ליצירת מספרים אקראיים.

האלגוריתם הכללי:

'לעולמים'

'הדלק נורה' אקראית / או נורות אקראיות

'השמע צלילים'

התסריט



הפעלת התסריט עם הקלקה על הדמות ברקע של 'מסיבה': האורות המתחלפים על הרובוט וצלילים עזים יוצרים תחושה מהנה של מסיבה.

התלמידים יכולים להשתמש בתסריט זה כבסיס לשינויים, שילוב מוסיקה וצלילים לפי בחירתם, כתיבת תסריט מקביל לדמות שמניעה את הדמות על המסך וכדומה.

ב. תנועה אינסופית

בתסריט זה הרובוט ינוע באופן אקראי למרחקים שונים ויעשה פניות אקראיות:

האלגוריתם הכללי

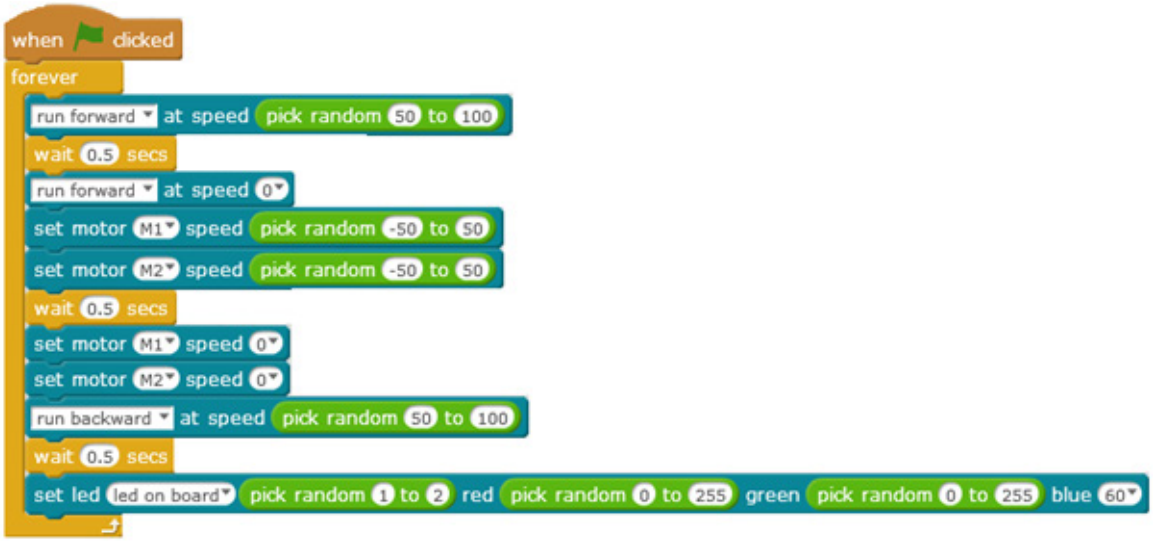
1. **'לעולמים'**

1.1 **'נוע'** למרחק אקראי

1.2 **'פנה לכיוון'** אקראי

1.3 **'הדלק נורה'** אקראית

התסריט



הערה: יש המון ערכים אקראיים בתסריט אבל יש גם, לכאורה, קשר אקראי בין האלגוריתם הכללי לתסריט שיושם – אבל רק לכאורה.

מאוד מאתגר לתת לתלמידים את האלגוריתם הכללי, ואפשר גם לתת להם עזרה נוספת בצורת התחלת התסריט עם לולאת 'לעולמים':





עתה האתגר של התלמידים ליצוק תוכן למסגרת המוצעת. זה גם מלמד על צורת התכנון של 'מלמעלה-למטה' – הנה הרעיון המרכזי בואו נרד לפרטים.

יש גם מתן מרחב יצירתי לתלמידים כך שהתוצאה הסופית אצל כל תלמיד תהיה שונה ובחלל חדר הרובוטיקה ירקדו רובוטים ביחד – אחד ירקוד וולס שקט והשני בקצב רוק מטורף.

תלמיד שירצה לשלב צלילים באמצעות קבוצת הצלילים של שפת סקרטץ' יוכל לייבא ממש צליל של ריקוד וכך הוא יתמודד עם מטלה יצירתית צבעונית ומלאת רעשים.

דרכי הערכה

בפרק זה ההערכה חייבת להיות הערכה של מטלות סיכום יצירתיות, במסגרת ההערכה חייבים להעריך יישום של מיומנויות המאה 21. יש לתת מטלות שדורשות:

- שיתוף פעולה של צוותים.
- מיומנות מדיה – שימוש בצלילים, חיפוש אחר קבצי קול, עריכה של קבצי קול, שילוב ידע מוסיקלי – יתכן שלא כל התלמידים יכולים אבל דרישה זו נותנת במה לתלמידים שיש להם יכולות מוסיקליות.

יש לוודא עם התלמידים שמתקיימים תהליכי תכנון. מומלץ לעודד לכתוב תסריטים שיתבצעו במקביל עם אחת הדמויות הכלולות בספריית סקרטץ' – דמות הרוקדת. דמות זו עם תסריט המשלב לולאת 'לעולמים' והוראת החלפת תלבושת תיצור שילוב מדהים מעניין ומלהיב.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
בקרה בחוג סגור; בקרה בחוג פתוח;	הסבר על הרעיון; הדגמה מעולם החושים של בני האדם; הצגת תסריט ללא חיישן כמו נגיעה ועצירה לפני מכשול ללא חיישן מרחק.			✓	הבחנה בתיאור של עולם בקרה בחוג סגור לעולם בקרה בחוג פתוח. **2 שעות**
חיישן מרחק; כניסה ראשונה לעולם של בקרה בחוג סגור	הצגת בעיות שניתן לפתור רק באמצעות שילוב חיישן. חשוב! אנחנו משתמשים ישירות בערך המגיע מהחיישן ללא משתנים.		✓	✓	כתיבת תסריט המשלב חיישן מרחק. **2 שעות**
הרחבת יישום היעד המרכזי של התכנית; מרעיון לקוד בתהליך מובנה	הצגת אלגוריתמים כרעיון כללי; פירוק הרעיון לצעדים קטנים; יישום בסביבת העבודה.	✓	✓	✓	עבודה בצוות, חלוקת משימות, כתיבה ויישום מטלות. **2 שעות**
לולאת **'לעולמים'** ושילובה בפרויקטים	החיבור של לולאת **'לעולמים'** עם סביבת העבודה, הן של הרובוט והן של שפת סקרטץ', והיכולות של השפה כולל תכנות מונחה אירועים, שילוב אנימציה וצלילים – יש בהם כדי ליצור התלהבות, עידוד היצירתיות והנאה מהלמידה.	✓	✓	✓	פרויקט משלב הרעיון עד ליישום. **2 שעות**

פרק ז' משתנים , משפטי תנאי 'אם', 'אם ואם לא'

מ ו ש ג י ם

- ✓ משתנה
- ✓ יחס לוגי, פסוק לוגי, ערך אמת, ערך שקר
- ✓ משפט **'אם'**; משפט **'אם ואם לא'**

מ ש ת נ ה

כל משתנה הוא 'קופסה' שבה ניתן לאחסן ערכים. כל משתנה מאוחסן בזיכרון המחשב, למקום האחסון יש כתובת בזיכרון אולם כדי שנוכל להתייחס למשתנה בקלות אנחנו נותנים לו שם ייחודי. מותר לתת שם בעברית או באנגלית.



משתנה משנה את ערכו לפי ההוראות שהמתכנת משלב בתסריטים (קבוצת הוראות 'משתנים').

נניח שיש לנו משתנה הנקרא פרי, הוא מכיל עצמים מסוג פירות. המשתנה יכול להכיל ערכים שונים בזמנים שונים בעת הרצת תסריט:



עתה נכתוב את ההוראה – **'שים'** את העצם בננות במשתנה 'פרי' ונקבל:



אנחנו רואים שערך התוכן של המשתנה 'פרי' השתנה ועתה הוא מכיל בננות.

בנספח ההוראות על קבוצת משתנים הצגתי את ההוראות לטיפול ושינוי ערכי משתנים כולל ההוראות:



סוגי ערכים המאוחסנים במשתנים – Data, Type

מילה – מחרוזת

אוסף של תווים המורכב מאותיות, ספרות וסימנים נקרא מחרוזת או בשם המוכר בסביבת סקרטץ' 'מילה'. ישנן פעולות מיוחדות שניתן לבצע על משתנה המכיל 'מחרוזת' כמו לחשב את אורך המחרוזת-מילה, או לגשת לתו במיקום מסוים במחרוזת-מילה.

ערך מספרי

ערך מספרי הוא ערך שמורכב מספרות אם מדובר בערך שלם, ויכלול גם נקודה עשרונית אם מדובר בערך ממשי. מספרים הם חלק מביטויים מתמטיים שאנו משתמשים בהם בתסריטים שאנחנו כותבים. מספר יכול להיות ערך שמוגדר **קבוע**, לדוגמה 7, או 9.787, או ערך שמאוחסן במשתנה.

ערך לוגי

ערך לוגי יכול לקבל אחד משני ערכים: **אמת** (TRUE), **שקר** (FALSE). יש לנו גם ביטויים לוגיים שחישובם יהיה אחד מהערכים TRUE או FALSE. לדוגמה ביטוי לוגי: 5 < 8 , 8>3+X, או לדוגמה החיישנים בסקרטץ' **'נוגע בצבע'** ו**'נוגע ב'** כשהם מחושבים תוצאתם תהיה TRUE או FALSE (אמת או שקר)

סיכום משתנים והוראות :

משתנה , אתחול ערך, שינוי ערך

משתנה (VARIABLE) הוא תא זיכרון בתכנית שמכיל נתון שיכול להשתנות בזמן הריצה. משתנה בשפת תכנות מקבל שם – במרבית השפות שם משתנה מכיל תווים מסוג: אותיות, מספרים וסימנים נוספים (כמו _), בשפת סקרטץ' ניתן לתת שמות בעברית וניתן להתחיל שם משתנה בספרה אך זה לא מומלץ. חלק מהשפות כוללות בהגדרה של המשתנה את הסוג של הערכים שהמשתנה יכול להכיל (שלמים, ממשיים, תווים, בוליאני), בשפת סקרטץ' אין צורך להכריז על סוג הערכים – הסוג נקבע לפי הערכים המושמים במשתנה: אפשר להשים ערכים שלמים, ממשיים, וטקסטים (מחרוזת – STRING).

אתחול ערך משתנה, שינוי ערך

אתחול ערך משתנה הוא לעיתים קרובות חלק מתחילת ביצוע אלגוריתם. מומלץ לאתחל משתנים ולא להניח שיש בהם ערך התחלתי שניתן באופן אוטומטי (לדוגמה במשתנה המייצג ערך מספרי להשים את הערך 0): דוגמה: ההוראה אתחל ערך משתנה בשפת סקרטץ' - 

הערך עשוי להשתנות במהלך ריצה באמצעות הוראת השמה – הוראה המשימה ערך חדש במשתנה. השמת הערך דורסת את הערך שקיים במשתנה.

דוגמאות למשפטי השמה בשפת סקרטץ': 

יחס לוגי, פסוק לוגי, ערך אמת, ערך שקר

יחס לוגי, פסוק לוגי

יחס לוגי הוא אופרטור של פעולה שמחבר בין שני ביטויים ויוצר פסוק לוגי.

לדוגמה: 3<6 , < הוא היחס הלוגי. קבלנו פסוק לוגי שנוכל להגיד האם הוא **'אמת'** או **'שקר'**.

דוגמאות נוספות: 4>=D התוצאה של פסוק לוגי היא **'אמת'** או **'שקר'**.

שפת סקרטץ' מזמנת פסוקים-מפעילים לוגיים שקיימים בכל שפות תכנות, לצד מפעילים לוגיים ייחודיים לסביבת הלמידה.



ערך אמת , ערך שקר

כל ביטוי לוגי יקבל את אחד הערכים: ערך **'אמת'**; ערך **'שקר'**.

אם התנאי **'האם נוגע בדמות'**, ואכן דמות אחת נוגעת בדמות האחרת, הערך של הביטוי יהיה אמת.

דוגמאות:

האם המכפלה של 4 ב-2 גדולה מ-12 – 12<4x2



האם מנת החלוקה של 15 ב-5 היא ללא שארית



האם הסכום של 3 והמכפלה 5 ב-12 > 63









מאוד חשוב לשים לב שבכתיבת ביטויים מתמטיים יש סדר פעולות ויש סוגריים, ואלו באים לידי ביטוי בתחביר של סקרטץ' באמצעות השילוב שעושים בין המפעילים כך שהשילוב של המפעילים המתמטיים, בתוך המפעיל הלוגי, יהיה זהה לפסוק הלוגי שהתלמידים כותבים. המטלה הזו מחברת בין הבנה ושליטה בסדר פעולות ומשמעות של סימני היחס =,>,< להעברה של הביטויים לשפת תכנות בהתאם לתחביר השפה.

משפט **'אם'**; משפט **'אם ואם לא'** – שילוב עם רובוטיקה

משפט 'אם'

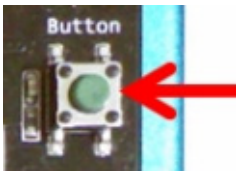
משפט **'אם'** הוא מבנה בקרה על זרימת ביצוע תסריט. כשהמחשב מבצע קבוצת פקודות בכל שפת תכנות הוא מבצע את הפקודות באופן סדרתי, כלומר פקודה אחר פקודה. אולם לעיתים צריך לבדוק האם התרחש מצב מסוים ולפעול בהתאם לתוצאת הבדיקה. משפט **'אם'** מאפשר, בשילוב ביטוי לוגי, לבצע קבוצת פקודת מסוימות רק אם התנאי שנבדק הוא 'אמת'.

משפט 'אם ואם לא'

מבנה בקרת זרימת ביצוע תכנית של משפט זה מאפשר כתיבת קבוצת פקודות אם התנאי אכן התקיים (אמת), ואם לא (שקר) אז מתבצעת קבוצה אחרת של פקודות.

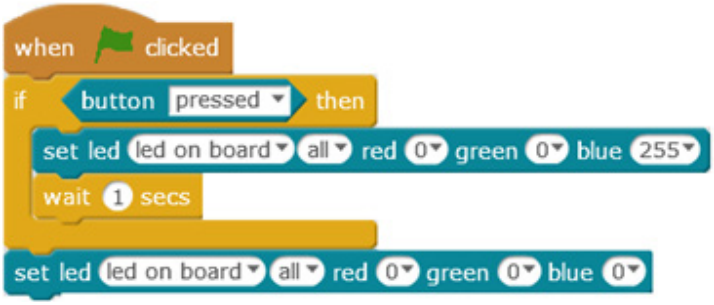
מבנה **'אם'** ומבנה **'אם ואם לא'** מאפשרים לנו לקבל תשובה 'אמת' או 'שקר' למצבים שנבדקים בהוראות אלו, ואנחנו משלבים הוראות ביצוע בהתאם לתשובה מתנאי הבדיקה.

דוגמה לשימוש בהוראה 'אם':



אם אנו לוחצים על הכפתור הממוקם על הבקר של הרובוט , הלדים על הלוח ידלקו למשך שנייה בצבע כחול.

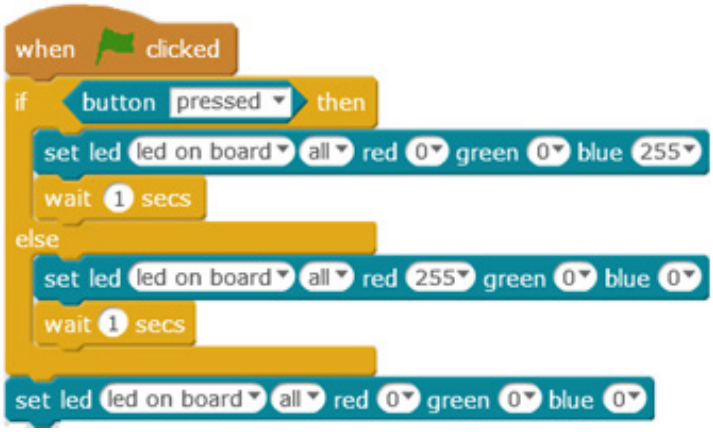
אם לא לחצנו על הכפתור בתחילת התסריט לא יתרחש דבר.



דוגמה לשימוש בהוראה 'אם ואם לא':

בקטע זה נשפר את התסריט הקודם:

אם לחצנו על הכפתור ידלקו הלדים בצבע כחול למשך שנייה, אחרת ידלקו הלדים בצבע אדום למשך שנייה



בתכנית הלימודים ברובוטיקה נשתמש בהוראה **'אם'**, ו**'אם לא'** בעיקר בהקשר של ערכי חיישנים. באמצעותם ובשילוב של מפעילים לוגיים נבדוק את ערך התנאי עבור המידע המגיע מהחיישנים השונים, כדי לתת הוראות מתאימות למצב משתנה של הרובוט. אם יש קושי פדגוגי בהעברה של משפט **'אם ואם לא'** מומלץ ליישם את האלגוריתם בשני משפטי **'אם'**.

דוגמאות

תנועה לעבר עצם ועצירה לפני העצם במרחק 10 ס"מ

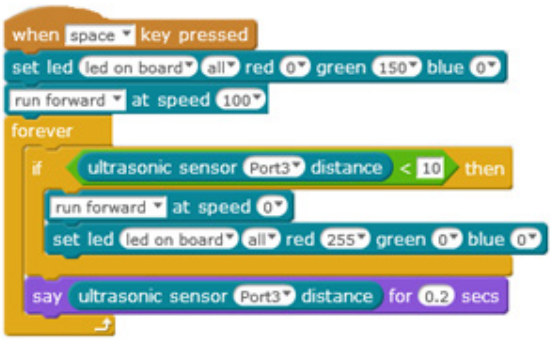
באתגר זה אנו משלבים את מה שלמדנו כולל:

- שימוש בחיישן מרחק
- שימוש בלולאת **'לעולמים'**
- שימוש במשפט **'אם'**

המטרה:

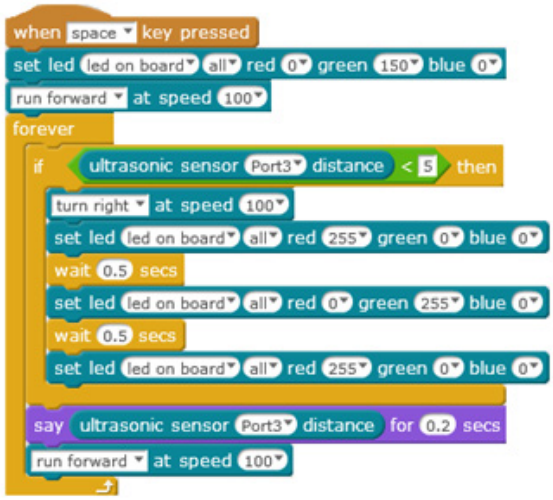
רובוט נוסע לכיוון עצם ועוצר בדיוק 10 ס"מ לפני העצם. מעמידים את הרובוט בנקודה אקראית לפני העצם, כלומר התכנית חייבת לעבוד בכל מיקום של הרובוט מהעצם.



האלגוריתם	התסריט:
1. 'הדלק נורות' ירוקות 2. 'נוע קדימה' במהירות 100 3. 'לעולמים' 4. 'אם' מרחק מעצם < 10 4.1 'עצור' 4.2 'הדלק נורות' אדומות 5. 'הצג' את המרחק מהעצם	

מטלה רובוט נוסע על שולחן ולעולם לא נופל מהשולחן

מה הרובוט יבצע? הרובוט ייסע על שולחן (או על כל משטח גבוה), וכאשר יתקרב לקצה ואוטוטו.. הוא נופל, אנחנו עם כף היד מול חיישן המרחק 'מעבירים לו מסר' שלא כדאי להמשיך. הרובוט מזהה את כף היד ופונה לכיוון אחר. כך הרובוט לא ייפול מהשולחן.

האלגוריתם	התסריט:
1. 'הדלק נורית' ירוקה 2. 'סע קדימה' במהירות 100 3. 'לעולמים' 3.1 'אם ערך' חיישן מרחק< 5 3.1.1 'פנה ימינה' 3.1.2 'הדלק נורית' אדומה 3.1.3 'הדלק נורית' ירוקה 3.1.4 'הדלק נורית' אדומה 3.2 'אמור' ערך חיישן מרחק 3.3 'סע קדימה' במהירות 100	

אלגוריתם מילולי

אלגוריתם מילולי נכתב כחלק מתהליך פתרון המטלה ומציג את הרעיון של הפתרון בשפה כללית ולא תלויה בשפת תכנות.

בפרקים הראשונים של תכנית הלימודים השתמשנו באלגוריתם מילולי כללי, אך לא התייחסנו ולא פירטנו כיצד כותבים הוראה באלגוריתם מילולי ואיך יודעים מהו סדר הביצוע. ההצגה של אלגוריתם כללי מתבססת על הבנת הוראות בשפה העברית, כמו קטע המציג מתכון לאפיית עוגה. אולם בהמשך הפרקים נזקקנו לכתיבה יותר מסודרת של אלגוריתם מילולי. באלגוריתם אנחנו צריכים להציג מבני תנאי **'אם'** ו**'אם לא'**, לולאת **'לעולמים'** ולולאות מסוגים נוספים. עלינו לדעת אלו הוראות כלולות במסגרת מבנה התנאי שאנו כותבים ואלו הוראות כלולות בלולאה בה אנו משתמשים. לכן השורות באלגוריתמים ממוספרות והשורות בתוך לולאה או במשפט תנאי ממוספרות ב'מיספור משנה'. בצמתים אלו של כתיבה אלגוריתמית יותר מורכבת חשוב לשלב את הלמידה עם תוכן הפרק בתכנית העוסק ומרחיב בנושא.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
משתנים	נושא משתנים מוצג בתכנית הלימודים לאחר כ-25 שעות. הנושא חשוב אך מעורר, בקרב ילדים צעירים, קשיים בהבנה של מושג, מה הוא מייצג ומה משמעות הקשר בין משתנה לבין מצב אותו המשתנה מייצג. בפרק אנחנו מציגים את הרעיון, כל שימוש במשתנה בהמשך התכנית יהיה רק כאשר הוא ממש נדרש, וגם יעשה מאמץ לקשר את המשתנה לפעילות הרובוט כך שהתלמיד יכול לראות את ההשפעה.	✓	✓		הבנה ויישום בדוגמאות מגוונות. **2 שעות**
פסוק לוגי	המונחים בתכנית הם מדעיים ומנוסחים בצורה לא פשוטה לתלמידים. ההסבר על פסוקים לוגיים נועד למורה ולא לתלמיד – לתלמידים יש להציג את הרעיון; לתרגל כתיבה תמטית של פסוקים לוגיים ולהשתמש במפעילים מתאימים בשפת סקרטץ' כדי ליצור אותם; וכמובן לשלבם עם חיישנים. זו דרך מעשית ויעלה ללמוד תנאים לוגיים כי קיום התנאי הוא 'פיסי' – בעולם ה'אמיתי'.	✓	✓		היכרות והבנה של פסוק לוגי; יכולת ליישם ולעבור מניסוח מילולי של פסוק לוגי לייצוג שלו בסביבת העבודה. **2 שעות**
משפט **'אם'** משפט **'אם ואם לא'**	הצגת מבנה הבקרה החשוב ביותר בתכנית. יש להציג יישומים של שימוש במשפט תנאי עם שימוש בחיישנים. זהו המרכיב והמוטיבציה החשובה ביותר לשילוב בתכנית בשלב זה.	✓	✓	✓	יישום משפט תנאי במטלות; שילוב משפטי תנאי עם קריאת ערכים מחיישנים. **3 שעות**

פרק ח' לולאות וחיישנים - הרחבה ; קלט מהמקלדת

קלט ממקלדת

המשתנה 'תשובה' בסקרטץ'

בשפת סקרטץ' מוגדרים משתני מערכת. אלו משתנים שאין צורך להגדירם והם חלק מסביבת העבודה. משתנה מיוחד הוא 'תשובה' 

משתנה זה מכיל את הקלט המתקבל באמצעות הפקודה 'שאל'. הערך ממשתנה 'תשובה' ניתן להעברה למשתנה אחר שהוגדר על ידי כותב התסריט לפי הצורך.

לולאת 'חזור _ פעמים'

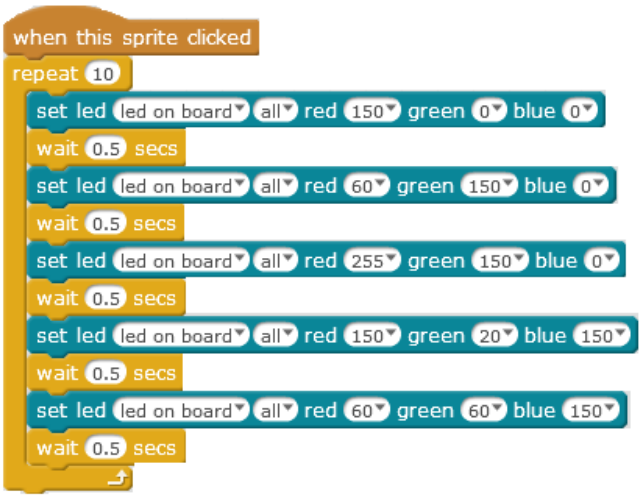


לולאת **'חזור _ פעמים'** – REPEAT

לולאת **'חזור _פעמים'** היא לולאה המתבצעת מספר פעמים ידוע מראש.

דוגמה:

הרובוט מדליק ומכבה נורות בצבעים שונים



דוגמה:

הרובוט עושה תנועה סביב ריבוע דמיוני. משך זמן הנסיעה הקובע את אורך הצלע של הריבוע נקבע ע"י מספר המתקבל מהקלט.

תסריט בסיסי לביצוע המטלה

תסריט עם קלט של זמן התנועה – אורך צלע הריבוע משתנה

תסריט בו הרובוט מבצע את תמונת הריבוע הדמיוני – במקביל מצויר ריבוע על המסך

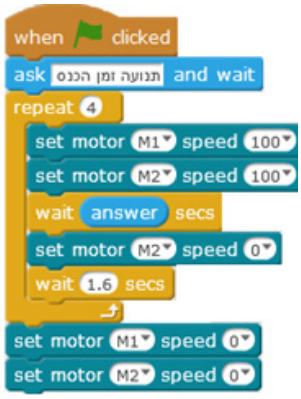
תנועה סביב ריבוע דמיוני – תסריט בסיסי

אלגוריתם כללי

'חזור' 4 פעמים

'סע'

'פנה' ב-90 מעלות

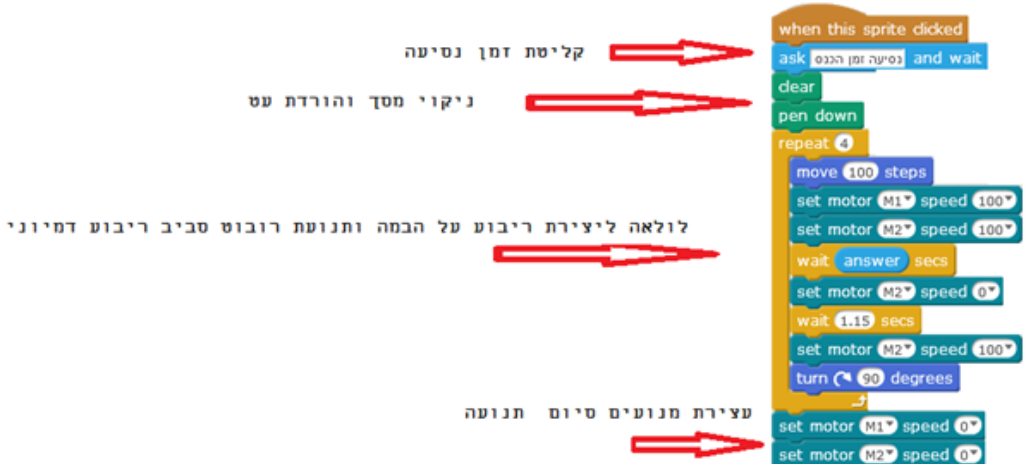
אלגוריתם מילולי מפורט	קוד בסביבת סקרטץ
1. קלט ==> זמן נסיעה 2. **'חזור'** (4) פעמים 2.1 R מנוע ==>100 2.2 L מנוע ==>100 2.3 **'המתן'** זמן_ תנועה 2.4 R מנוע ==>0 2.5 **'המתן'** 1.6 שנייה	

תנועה סביב ריבוע דמיוני עם זמן נסיעה מהקלט, ובמקביל ציור ריבוע על המסך.

אלגוריתם מילולי מפורט – (**חשוב!** זמני הנסיעה משתנים ותלויים בפרמטרים רבים, ולכן התאימו אותם לסביבה בפועל ולסוג הרובוט)

1. קלט ==> זמן נסיעה
2. **'נקה'** מסך
3. **'הורד עט'**
4. R מנוע ==> 100
5. **'חזור'** (4) פעמים
   - 5.1 **'זוז'** (100) צעדים // הוראה לדמות על הבמה לזוז 100 צעדים
   - 5.2 L מנוע ==>100 // הכנה לתנועה קדימה של הרובוט
   - 5.3 **'המתן'** (זמן נסיעה) **שניות'** // קביעת זמן נסיעה קדימה של הרובוט
   - 5.4 R מנוע ==>0 // הכנה לפנייה ב-90° ימינה
   - 5.5 **'המתן'** 1.6 שניות' // זמן לפנייה, חשוב שצריך לחשב זמן עבור פנייה ב-90° מדויקת הערך תלוי במהירות מנוע L
   - 5.6 R מנוע ==> 100 // עדכון לקראת הנסיעה קדימה במחזור הבא
   - 5.7 'פנה ימינה' (90°) // פנייה של הדמות על המסך
6. R מנוע ==> 0
7. L מנוע ==>0

המימוש בסביבת העבודה



חשוב! המטלה שביצענו מחברת בין שני ממדים: ממד הבמה בה נעה הדמות ומציירת ריבוע וממד ה'עולם האמיתי' בו הרובוט מבצע תנועה סביב ריבוע דמיוני.

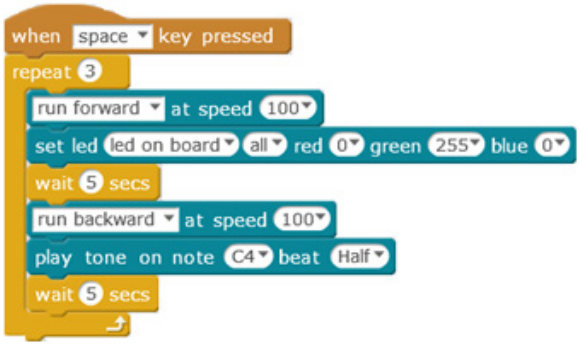
החיבור בין שני ממדים אלו מסייע לתלמידים להבין מה הרובוט האמיתי מבצע על פי הממד של הדמות על הבמה, שהוא אותו תהליך של תנועה ופנייה ב-90 מעלות. התלמידים מבינים ומודעים לחשיבות הדיוק ולמורכבות של פרמטרים 'סביבתיים' ופרמטרים הקשורים לרובוט בכדי לקבל ריבוע מדויק בתנועת הרובוט (עוצמת סוללות, המשטח עליו הרובוט נע, דיוק התיאום בין שני המנועים וכדומה).

על הבמה דמות הרובוט ביצעה מסלול סביב ריבוע וציירה את הריבוע



מטלה נסיעה ישראלית (מקור רחל פרלמן – בלוג מדעי המחשב בחטיבת ביניים)

יצרן מכוניות 'נסיעה ישראלית' מעוניין לייצר מכונית לפי המפרט הבא: בנסיעה קדימה – תודלק נורה; כדי לראות את השטח בנסיעה אחורה – יושמע זמזם כדי להזהיר את הסביבה. כתבו תכנית לנסיעה של: 3 פעמים קדימה 5 שניות, ולאחר מכן אחורה 5 שניות, לפי המפרט המתואר.

אלגוריתם	התסריט
1. **'חזור 3 פעמים'** 1.1 **'סע קדימה'** 1.2 **'הדלק נורות'** (צבע ירוק) 1.3 **'המתן'** 5 שניות 1.4 **'סע אחורה'** 1.5 **'השמע צליל בזמזם'** 1.6 **'המתן'** 5 שניות	

מ ש י מ ה

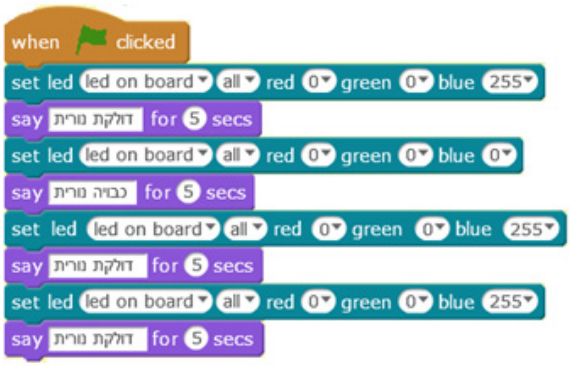
שנה את התכנית כך שמספר הפעמים שהלולאה מתבצעת ייקלט מהקלט.

מטלה כיבוי והדלקת נורות והודעה על מצב הנורה
(מקור רחל פרלמן – בלוג מדעי המחשב בחטיבת ביניים)

מ ש י מ ה **נתון אלגוריתם להעברה לקוד:**

אלגוריתם להעברה לקוד	
1. **'הדלק נורית'** 2. הדפס 'נורית דולקת' 3. **'המתן 5 שניות'** 4. **'כבה נורית'** 5. הדפס 'נורית כבויה' 6. **'המתן 5 שניות'** 7. **'הדלק נורית'** 8. הדפס 'נורית דולקת' 9. **'המתן 2 שניות'** 10. **'כבה נורית'** 11. הדפס 'נורית כבויה'	

- יש להעביר את התסריט לקוד.
- יש לשלב לולאת **'לעולמים'** בה ייכלל כל התסריט.

אלגוריתם להעברה לקוד	תסריט
1. **'הדלק נורית'** 2. הדפס 'נורית דולקת' 3. **'המתן 5 שניות'** 4. **'כבה נורית'** 5. הדפס 'נורית כבויה' 6. **'המתן 5 שניות'** 7. **'הדלק נורית'** 8. הדפס 'נורית דולקת' 9. **'המתן 2 שניות'** 10. **'כבה נורית'** 11. הדפס 'נורית כבויה'	

שולבו הוראות לרובוט והוראות לדמות של סקרטץ':

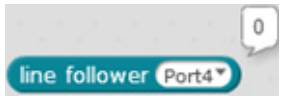
- הרובוט ידליק ויכבה נורית
- הדמות תאמר "נורית כבויה"; "נורית דולקת"

חשוב! לא שולבה הוראת זמן המתנה כי הוראת **'אמור'** כוללת זמן המתנה שחופף לדרישות.

חיישן אור – line follower – חיישן פס שחור

בחזית הרובוט ממוקם חיישן אור כפול, לחיישן שני צדדים ימני ושמאלי

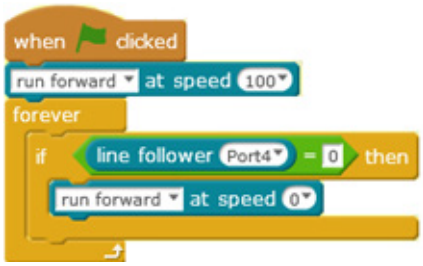


כאשר החלק הימני של החיישן מעל פס שחור החיישן מחזיר את הערך 2		
כאשר החלק השמאלי מעל פס שחור החיישן מחזיר את הערך 1		
כאשר שני הצדדים הימני והשמאלי מעל פס שחור החיישן מחזיר 0		
כאשר שני הצדדים הימני והשמאלי מעל משטח לבן החיישן מחזיר 3		

תכניות ואתגרים עם חיישן עוקב פס שחור

הרובוט נוסע קדימה ונעצר על קו שחור

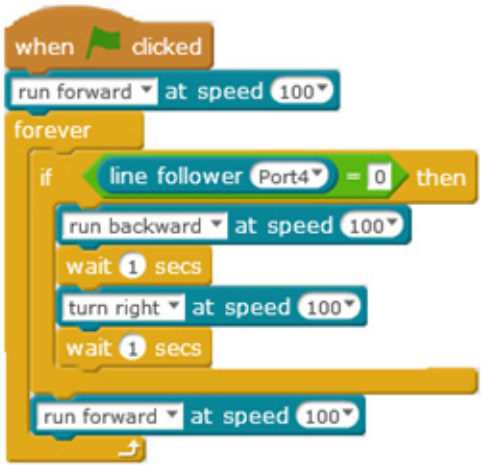
לפי נתוני החיישן כאשר שני החיישנים מעל קו שחור ושתי הנורות הכחולות של החיישן אינן דולקות החיישן מחזיר ערך 0. ולכן נוכל לרשום את האלגוריתם הבא:

אלגוריתם	התסריט
1. 'סע קדימה' במהירות 100 2. 'לעולמים' 2.1 'אם חיישן עוקב קו' = 0 2.1.1 'עצור' את הרובוט	

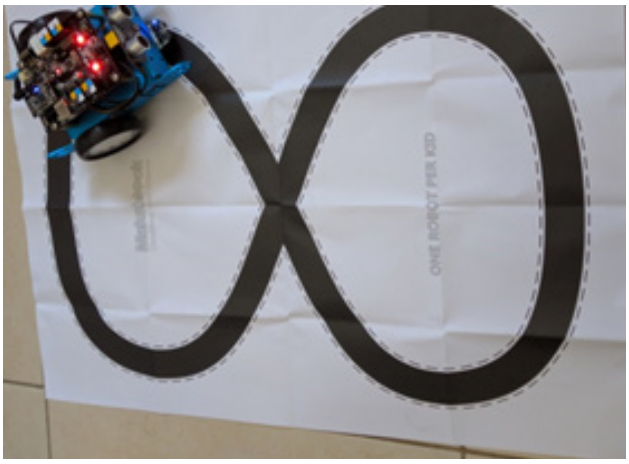
רובוט לא נופל מהשולחן

חיישן האור מורכב משני חלקים, חלק המאיר על המשטח וחלק שמקבל את האור החוזר. כשהחיישן מעל משטח לבן הוא מחזיר את הערך: שונה מ-0. כך גם כאשר החיישן אינו נמצא מעל משטח (למשל מחוץ לשולחן החיישן יחזיר 0).

על בסיס תכונה זו נוכל לכתוב אלגוריתם בו הרובוט משוטט על השולחן וכשהוא מגיע לקצה וחיישן האור מחזיר 0, הרובוט משנה כיוון נסיעה ופונה וחוזר למרכז השולחן

אלגוריתם	התסריט
1. **'סע קדימה'** 2. **'לעולמים'** 2.1 **'אם חיישן אור'** =0 2.1.1 **'סע אחורה'** 2.1.2 **'המתן'** 1 שנייה 2.1.3 **'פנה ימינה'** 2.1.4 **'המתן'** 1 שנייה 2.2 **'סע קדימה'**	

מעקב אחר פס שחור



אלגוריתם כללי -

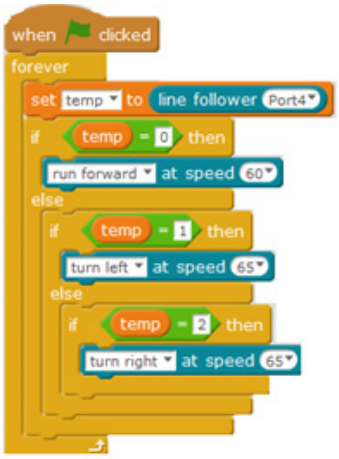
'לעולמים'

'קרא ערך' חיישן עוקב-פס-שחור

'אם הערך' = 0 **'סע ישר'** \\ שני החיישנים הימני והשמאלי ממוקמים על הפס השחור

'אם הערך' = 1 **'פנה שמאלה'** \\ הרובוט צריך לתקן כיוון הפעלה קצרה של מנוע שמאל

'אם הערך' = 2 **'פנה ימינה'** \\ הרובוט צריך לתקן כיוון הפעלה קצרה של מנוע ימין

התסריט	אלגוריתם מפורט
	1. **'לעולמים'** 1.1 ערך חיישן עוקב-פס-שחור==> TEMP 1.2 **'אם'** TEMP=0 סע ישר 1.3 **'אם'** TEMP=1 פנה שמאלה 1.4 **'אחרת'** 1.5 **'אם'** TEMP=2 פנה ימינה

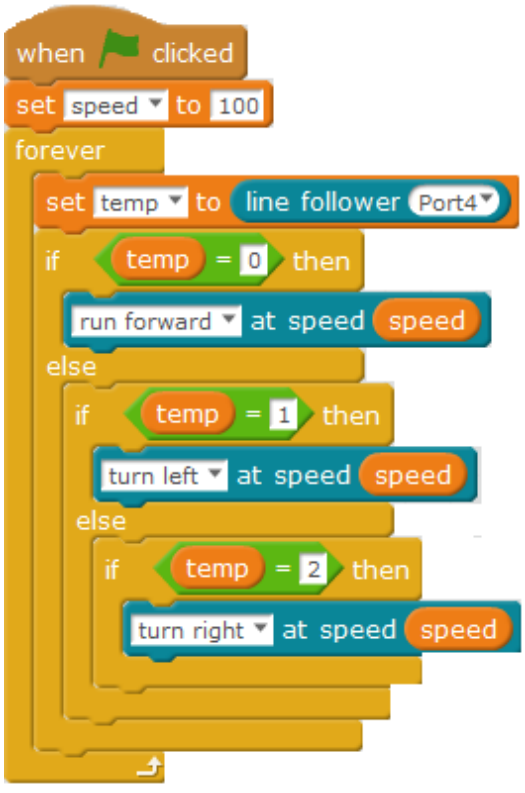
חשוב!

א. ניתן לכתוב את התכנית עם משתנה נוסף SPEED ובמקום לרשום מהירות מסוימת בהוראה, נשלב את המשתנה שיקבל ערך התחלתי רצוי.

ב. ניתן לוותר על משתנים כולל המשתנה TEMP ולשלב בביטויי התנאי : 'אם' המפעיל המחזיר ערך מחיישן פס שחור

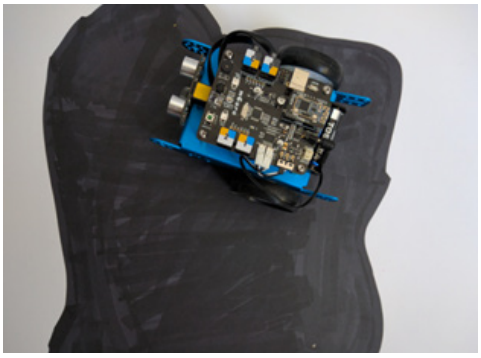
ג. המהירויות אינן קבועות ותלויות בעוצמת הסוללות, במיקום המשטח מתחת לדף הנייר, בקפל או קמטים בנייר, בתאורת החדר, בתנועה ישרה של הרובוט עם מנועים המתפקדים בצורה מושלמת ועוד. ולכן יש לנסות את התכנית ולתקן מהירויות בהתאם. מניסיון והרצת התכנית גם לאחר הפעלה מוצלחת יתכן והרובוט לפעמים 'יחמיץ' את הפס השחור.

תכנית מעקב אחר פס שחור עם משתנה לשמירת ערך המהירות ולשימוש בכל התסריט



מטלה – לבד בחשיכה

הרובוט נמצא על משטח שחור, נוסע הלוך ושוב ואינו יוצא מהמשטח השחור



חקר המטלה

חיישן פס שחור מחזיר את הערכים הבאים:

0; 1; 2 אם החיישן קורא צבע שחור בצד שמאל; ימין; בשני הצדדים; 3 אם החיישן על משטח לבן.

נוכל לפתור את הבעיה באופן פשוט: אם חיישן פס שחור מחזיר ערך 3 – הרובוט יוצא מתחום המשטח השחור – יש להחזירו לאחור ולהפנות את הרובוט לכיוון אחר.

הרעיון הכללי של האלגוריתם:

1. **'לעולמים'**

1.1 **'אם הרובוט יוצא'** מהמשטח השחור

1.1.1 **'השמע צלילים'**

1.1.2 **'הדלק אורות איתות'**

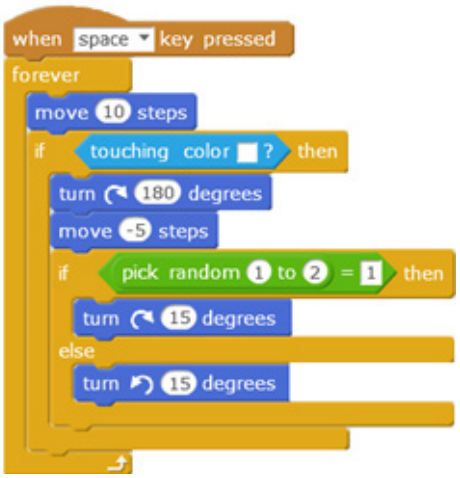
1.2 **'החזר את הרובוט'** למשטח

שלב הדמיה

לפני המעבר לתכנות הרובוט, נוכל להציג הדמיה של הפתרון על דמות של סקרטץ'. להדמיה 'הוירטואלית' יש פוטנציאל ללמידה מטרימה, להבנה טובה יותר של הרעיון וכהכנה להבנת השלב שמקדם לקראת תכנות הרובוט בעולם האמיתי.

הזירה דומה: רובוט ממוקם על משטח שחור

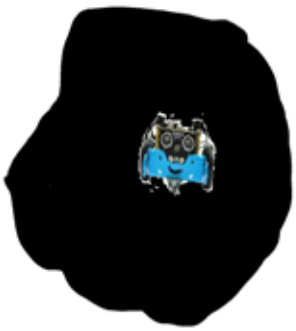
המימוש של הפתרון:



התסריט עבור דמות בסקרטץ' פשוט יותר ומאפשר מעבר כמעט ישיר מאלגוריתם כללי ליישום. הרעיון המרכזי באלגוריתם לרובוט הוא זהה ועובדה זו נותנת לנו כלים פדגוגיים ודידקטיים איכותיים – כלים שמשתלבים בתיאוריות של ויגודסקי לגבי הפיגומים ההדרגתיים שאנחנו מניחים לפני התלמיד במעבר הלמידה: מלמידה של הידע שמוכר, נהיר וברור לתלמיד, אל למידה של ידע חדש ולא מוכר. זה משתלב עם רעיון האינטליגנציות המרובות של גארדנר – יש כאן סביבה גרפית אחרת, הוראות שקשורות לשינוי במצב גרפי, אבל בהיבט הלוגי של זרימת האלגוריתם זה מחובר לבעיה של הרובוט על המשטח: שני עולמות דואליים שמאפשרים למידה ויישום בצורה טובה יותר.

האלגוריתם המפורט עבור הרובוט:

1. **'לעולמים'**
   - 1.1 **'הדלק נורות'** ירוקות
   - 1.2 **'סע קדימה'**
   - 1.3 **'אם חיישן פס שחור'** = 3
     - 1.3.1 **'הדלק נורות'** אדומות
     - 1.3.2 **'השמע צליל'** הזהרה
     - 1.3.3 **'סע לאחור'**
     - 1.3.4 **'המתן** 0.5 **שניות'**
     - 1.3.5 **'פנה לכיוון'** שמאל או ימין באופן אקראי



יישום תהליך מובנה: 'מרעיון לקוד':

שאלה הנובעת מהאלגוריתם:

איך ניישם פנייה שמאלה או ימינה באופן אקראי?

פתרון: (זו הצעה ואפשר לפתור אחרת)

נגריל מספר בתחום 1-2, אם הוגרל 1 הרובוט יפנה ימינה, אם הוגרל 2 הרובוט יפנה שמאלה.

יישום בסביבת העבודה - שלבים:

האלגוריתם הועבר לתסריט; הרובוט לא פעל כמצופה והוא יצא באופן חריג מתחום המשטח השחור; הפניות התבצעו ברדיוס גדול מדי; הרובוט כמעט עשה פניית 200 מעלות; הנסיעה לאחור הייתה למרחק גדול מהנדרש.

התיאור כאן הוא תיאור 'נורמלי' של כל תהליך כתיבת ויישום אלגוריתם – לעיתים קרובות מה שתכננו אינו פועל כפי שנדרש ואנחנו מגדירים זאת כבאג-תקלה בתוכנה – תקלה בתוכנה צריך לתקן.

התיקון מלווה בשאילת שאלות וחשיבה על פתרון:

מדוע הרובוט יוצא מהמשטח למרות שזיהה שהוא מחוץ למשטח?.

התשובה תהיה באותו הסדר של מה שבצענו לאחר זיהוי המצב הלא תקין – הרובוט חרג מגבולות המשטח – בצענו שלוש פעולות ובסדר הבא:

1. הדלקת נורות אדומות
2. השמעת צליל
3. נסיעה לאחור

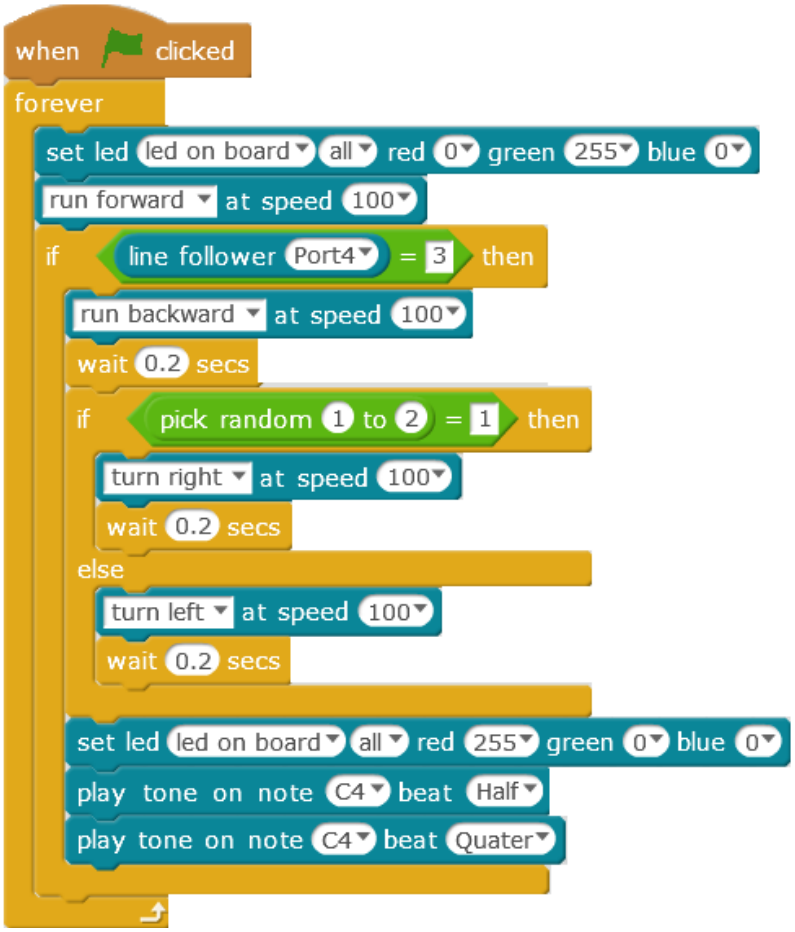
ברור שבעת הביצוע של ההוראות: **'הדלקת נורות'** אדומות, **'השמעת צליל'**, הרובוט המשיך לנסוע קדימה ולכן כדי לתקן

את הבאג הזה, שינינו את הסדר:

1. נסיעה לאחור
2. הדלקת נורות אדומות
3. השמעת צליל

הבאג השני של נסיעה לאחור ופניות החורגות ממה שנדרש תוקנו על ידי הקטנת זמן הפנייה וזמן הנסיעה לאחור.

התסריט הסופי:



דוגמה זו היא רעיון לתהליך המדמה עבודת מהנדסי תוכנה:

א. מתכננים

ב. מיישמים

ג. בודקים נכונות ותקינות הפתרון

ד. מתקנים

ה. ושוב ניתן לחזור על השלב של הבדיקה

לכן, צרו עם התלמידים את האלגוריתם המקורי על פי הרעיון הכללי; יישמו יחד עם התלמידים את הפתרון; התלמידים יריצו את התסריט ויעלו השאלות מדוע התכנית לא פועלת באופן מדויק ולפי הנחיות המטלה; תנו זמן לחשיבה ולדיונים בתוך צוותי התלמידים ובמליאת הכיתה – עד לקבלת פתרון.

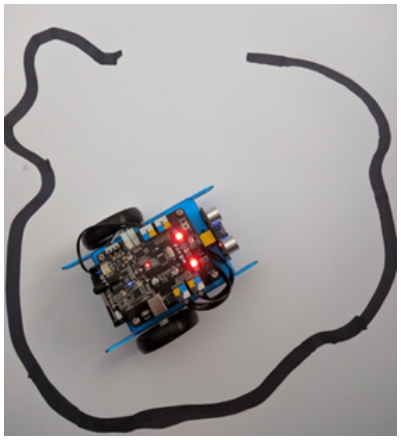
חשוב! רק בסוף, ללא קשר לפתרונות שנתנו – כי התהליך הוא החשוב – ניתן להציג את הפתרון שפותר את הבעיה. אם תלמידים הציעו רעיונות אחרים שונים שגם הם פותרים את הבעיה, חשוב להדגיש שיש לעיתים קרובות כמה פתרונות לאותה הבעיה.

מטלה - Free Willi

סרט ידוע על לוויתן קטלן שהיה כלוא במפרץ סגור. הלוויתן הצליח לצאת מפתח מאוד קטן במפרץ.

עתה הרובוט שלנו נמצא באותה בעיה: הוא לכוד במפרץ סגור בפס שחור ויש רק פתח יציאה יחיד.

המטרה להוציא את הרובוט מהכלא – באמצעות אלגוריתם חכם.



חקר המטלה

חיישן פס שחור מחזיר את הערכים הבאים:

0; 1; 2 אם החיישן קורא צבע שחור בצד שמאל; בצד ימין; בשני הצדדים;

3 אם החיישן על משטח לבן.

נוכל לפתור את הבעיה באופן פשוט:

הרובוט נוסע ישר

אם הוא נתקל בפס שחור – הוא יבצע נסיעה קצרה לאחור ופנייה אקראית ימינה או שמאלה.

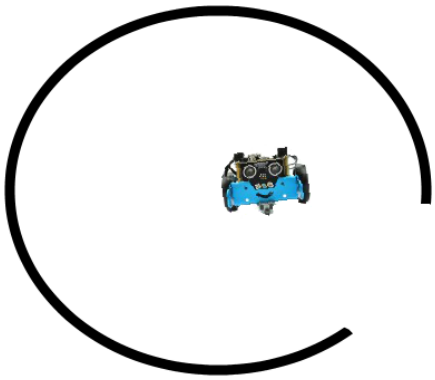
בהנחה שבתהליך של תנועה וחיפוש הרובוט ימצא את דרכו אל החופש.

נפתור את הבעיה עבור דמות על הבמה של סקרטץ'

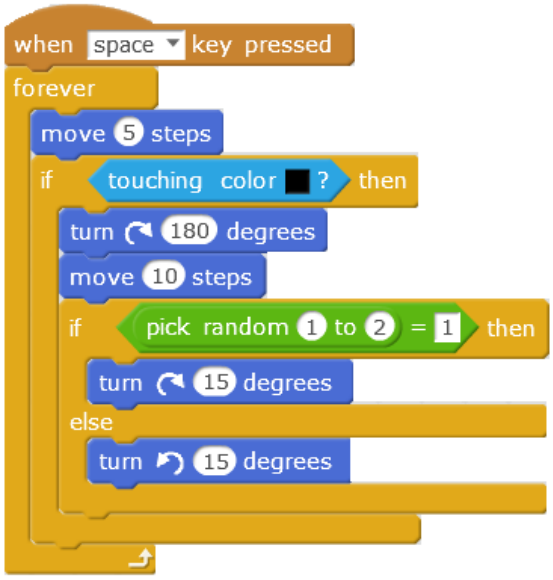
1. **'לעולמים'**
    1.1 **'נוע 5 צעדים'**
    1.2 **'אם נוגע בצבע'** שחור
        1.2.1 **'הפוך כיוון תנועה'**
        1.2.2 **'נוע 10 צעדים'**
    1.3 **'אם ערך אקראי'** = 1
        1.3.1 **'פנה ימינה'** 15 מעלות
        1.3.2 אחרת **'פנה שמאלה'** 15 מעלות

ההדמיה בעולם של דמות על הבמה בסביבת סקרטץ'

הבמה והדמות:



התסריט עבור דמות הרובוט על הבמה בסביבת סקרטץ'



הרעיון המפורט של האלגוריתם

1. **'לעולמים'**

1.1 **'הדלק נורה'** אדומה

1.2 **'סע קדימה במהירות'** 75

1.3 **'אם חיישן פס שחור'** < 3 // המשמעות רואה שחור באחד מהחיישנים או בשני החיישנים

1.3.1 **'נוע אחורה במהירות'** 100

1.3.2 **'המתן** 0.5 **שניות'**

1.3.3 **'אם מספר אקראי'** = 1

1.3.3.1 **'פנה ימינה'**

1.3.3.2 **'המתן** 0.3 **שניות'**

1.3.4 **'אחרת'**

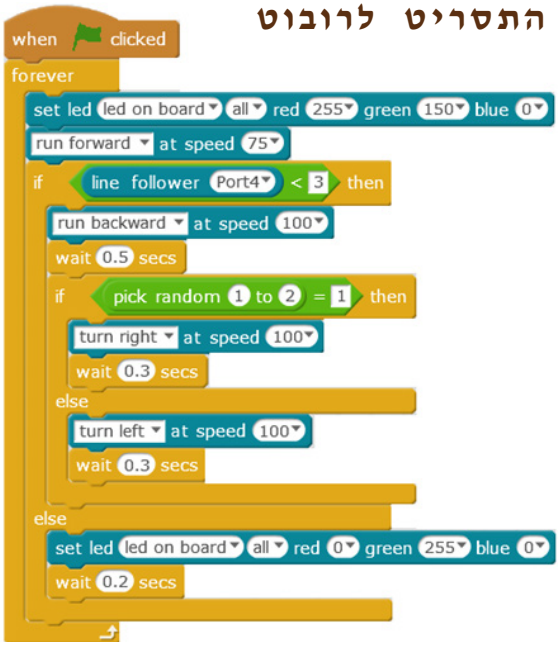
1.3.4.1 **'פנה שמאלה'**

1.3.4.2 **'המתן** 0.3 **שניות'**

1.4 **'אחרת'**

1.4.1 **'הדלק נורה'** ירוקה // בדרך אל החופש

התסריט לרובוט



חשוב! האורות והצלילים הם לא חלק מהרעיון המרכזי והם אינם מרכיב בפתרון הלוגי – הם נועדו ליצור עניין. אולם אם נשלב הוראת **'השמע צליל'** או הוראת **'הדלק נורה'** בצבע מסוים, נוכל לראות האם מתבצעת לולאה או האם תנאי מתקיים באמצעות שילוב הוראות אלו בלולאות ובמבני תנאי – במידת הצורך אנחנו יכולים לעקוב אחר הביצוע.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
קלט מהמקלדת	בתכנית רובוטיקה אנחנו עוסקים בקלטים מסוגים שונים, במיוחד באמצעות חיישנים; קלט מסורתי הוא קלט מהמקלדת המאפשר קלט מחרוזת וערך מספרי. חשוב לעשות בו שימוש – כך תסריט הופך בהרצות חוזרות עם ערכים שונים לסביבת חקר.		✓		תלמיד ישלב בתסריטים פקודות קלט, התלמיד יריץ הרצות שונות את אותו תסריט עבור ערכים שונים. **2 שעות**
חיישן אור	מהחיישנים החשובים ביותר. קיים שימוש בחיישני אור במציאות אותה התלמיד מכיר כמו חיישן בדלת אוטומטית או במעלית, ותלמיד יכול לחבר את המציאות היום יומית ללמידה בכיתה. ניתן לכתוב תכניות מעניינות ומאתגרות בשילוב חיישן זה.	✓	✓	✓	כתיבת תכנית קצרה בשילוב חיישן אור; שימוש מתקדם עם חיישן עוקב פס שחור. **4 שעות**
לולאות הרחבה לולאת **'לעולמים'** לולאת **'חזור _ פעמים'**	בפרק זה אנו משלימים למבני הבקרה מרכיב חשוב: לולאת **'חזור _ פעמים'**. לולאה זו תאפשר לנו לכתוב יישומים מעניינים ומאתגרים, בשילוב מרכיבים חשובים נוספים של שפת התכנות – קלט וחיישנים.	✓	✓	✓	היכרות ושימוש פשוט בלולאת חזור; יישום הלולאה במטלות עם חיישנים ותנועה של הדמות על הבמה. **4שעות**

פרק ט' לוגיקה – תנאים מורכבים , לולאת 'חזור עד ש'

ביטויים לוגים מורכבים, שילוב ביטויים לוגיים מורכבים בתסריטים, קינון משפטי תנאי, הקבלה בין קינון משפטי תנאי לשימוש באופרטורים לוגים.

עיון ולמידה של נושאים בפרק זה מתבצעים בהתאם לתזמון ולהחלטות של המורים. במידה ויש קושי, ניתן לא להשתמש באופרטורים לוגיים בתכנית וגם לא להשתמש במשפט התנאי **'אם ואם לא'**. כמו כן לולאת **'חזור עד ש'** הינה מבנה מורכב: יצירת התניה בלולאה, יצירת מנגנון המקדם את 'הסיכוי' שהתנאי בלולאה יתקיים (אחרת זו תהיה לולאה אינסופית) – אינם פשוטים להבנה ויישום. גם בנושא של לולאת **'חזור עד ש'** ההחלטה האם ללמד או לא היא של המורים בשטח – אפשר להרחיב את הדוגמאות ולתת פרויקטים ומטלות בהם אין צורך להשתמש בנושאים אלו, ו/או להציג קוד חלופי דוגמת משפט תנאי מקונן במקום שימוש באופרטור AND. ברור שגם לשימוש בתנאי מקונן יש את המורכבות שלו ויתכן שעבור תלמידים מסוימים שימוש באופרטור AND יותר פשוט ממשפט תנאי מקונן וההיפך. התכנית מאפשרת יישום מותאם ואדפטיבי להתקדמות של התלמידים.

מושגים

- ✓ האופרטור AND;
- ✓ האופרטור OR;
- ✓ האופרטור NOT;
- ✓ קינון משפטי תנאי

האופרטור AND

אופרטור זה משמש לשילוב שני ביטוים לוגיים, כאשר התוצאה הסופית של חישוב הביטוי הלוגית מוגדרת באופן הבא עבור שני תנאים שנקרא להם תנאי א' ; תנאי ב'.

טבלת אמת AND

תנאי א'	תנאי ב'	תנאי א' AND תנאי ב'
שקר	שקר	שקר
שקר	אמת	שקר
אמת	שקר	שקר
אמת	אמת	אמת

האופרטור OR

אופרטור זה משמש לשילוב שני ביטוים לוגיים, כאשר התוצאה הסופית של חישוב הביטוי הלוגי מוגדרת באופן הבא עבור שני תנאים שנקרא להם תנאי א' ; תנאי ב'.

טבלת אמת OR

תנאי א'	תנאי ב'	תנאי א' OR תנאי ב'
שקר	שקר	שקר
שקר	אמת	אמת
אמת	שקר	אמת
אמת	אמת	אמת

האופרטור לא - NOT

אופרטור השלילה: אם ערך הביטוי הלוגי 'אמת', התוצאה של הפעלת האופרטור היא 'שקר'; אם ערך הביטוי הלוגי 'שקר' התוצאה של הפעלת האופרטור היא 'אמת'.

טבלת אמת NOT

תנאי	NOT
אמת	שקר
שקר	אמת

קינון משפטי תנאי

מבנה תכנותי בו אנחנו משלבים משפט ביצוע שהוא משפט תנאי בתוך משפט תנאי וכך אנו יוצרים קינון של משפטי תנאי. יש חשיבות לבחון אלו מצבים אנו מכסים במבני קינון שונים.

נושאי משנה:

ריענון משפט התנאי **'אם ואם לא'**

האופרטור AND

יישום משפט תנאי עם האופרטור AND

יישום תסריט עם אופרטור AND ויישום אותו תסריט עם קינון משפטי תנאי

האופרטור OR

יישום תסריט עם האופרטור OR ויישום אותו תסריט ללא האופרטור OR

דרכי הערכה

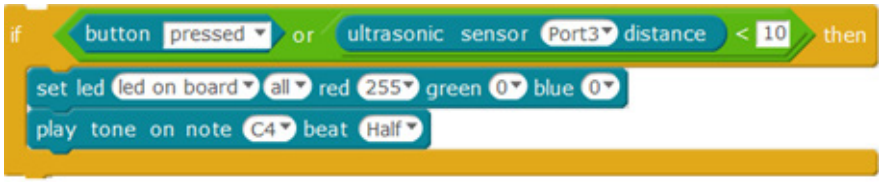
1. תרגול בניסוח ובכתיבה מילולית של טענות לוגיות.
2. תרגול ביישום טענות לוגיות מילוליות באמצעות קטעי תכנות בסיסיים.
3. ניסוח סיפור עם התניות ויישומו בתסריט עם אופרטורים לוגיים וללא אופרטורים לוגיים.

דוגמאות

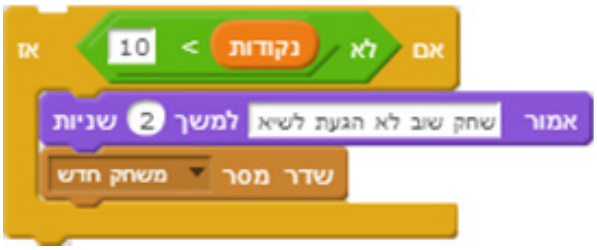
דוגמה: שימוש ב: AND - וגם



דוגמה: שימוש ב: OR - או

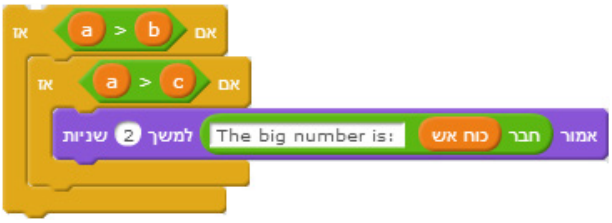


דוגמה: שימוש ב: NOT - לא



דוגמה: משפטי תנאי מקוננים:





דוגמאות סיכום: הדמיה על במת סקרטץ'

החיפושית הקופצת

דוגמה זו מציגה מצבים שונים: חיפושית קופצת על משטח משובץ בריבועים אדומים וכחולים, על המשטח יש חיפושית נוספת ובננה.

החיפושית הקופצת תנחת בכל קפיצה:

על משבצת אדומה

או

על משבצת כחולה

או

על בננה

או

על החיפושית השנייה

או

על חיפושית וגם על משבצת אדומה

או

על בננה וגם על משבצת כחולה

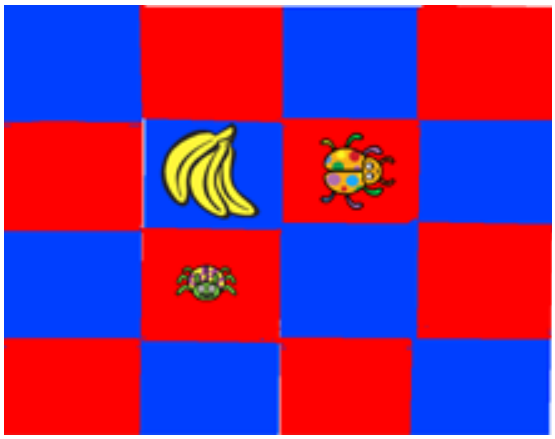
ועוד אפשרויות שמומלץ לבחון בהרצת היישום ולענות על השאלה: האם נבדקו כל המצבים של החיפושית הקופצת ביחס לרקע ולדמויות האחרות?

הרצת היישום פשוטה: כל הקשה על מקש רווח מקפיצה את החיפושית למיקום הבא.

הדוגמה מאפשרת מניפולציות שונות עם התנאים הלוגיים ועם האופרטור AND והאופרטור OR. אפשר ומומלץ לשנות את המשטח המשובץ, לשנות צבעים, להחליף דמויות על המשטח, להוסיף דמויות על המשטח.

הערה: המשטח נוצר כתמונה של הרקע

הבמה של המשחק



התסריטים של החיפושית הקופצת





יישום עם אופרטור or ואופרטור not

בדוגמה זו נשתמש בשני חיישנים:

חיישן מגע המחובר לכניסה 1 – חיישן זה מזהה נגיעה ומחזיר 'אמת' אם נוגעים בו ו'שקר' אם אחרת: כניסה 1

חיישן מרחק שמחשב את המרחק בין הרובוט לעצם : כניסה 3

הרעיון הכללי

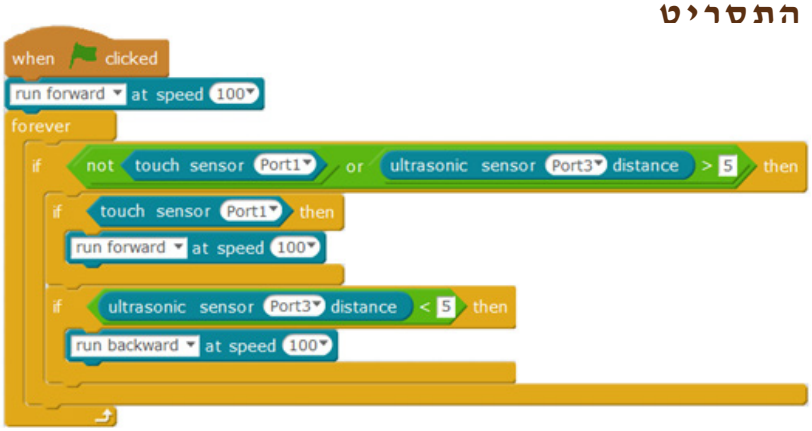
1. **'לעולמים'**

1.1 **'סע קדימה'**

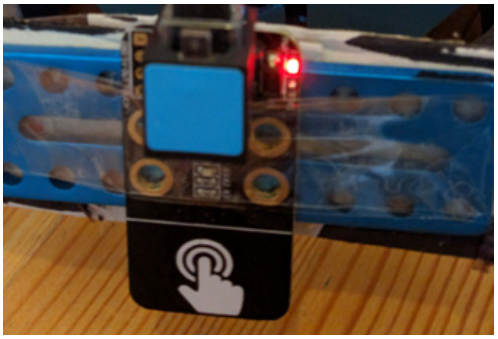
1.2 **'אם יש נגיעה'** בחיישן מגע 'סע קדימה'

1.3 **'אם המרחק'** מעצם<5 'סע אחורה'

התסריט



חיישן מגע



לולאת 'חזור עד ש'

מבוא וסיכום של מה שלמדנו על לולאות:

מבנה **'חזור עד ש'** הוא המבנה השלישי של לולאה, עד כה הכרנו את:

מבנה לולאת **'לעולמים'**



מבנה זה הוא הפשוט ביותר להבנה על ידי התלמידים. התלמיד אינו עוסק בשאלה: כמה פעמים באלגוריתם הוא נדרש לבצע את הלולאה – הבחירה בבלוק **'לעולמים'** פשוטה. הוא לא 'מוטרד' בשאלה מתי הלולאה מסתיימת, כך שהוא אינו

עוסק בשאלות של תנאים לוגיים שיתנו את המשך הלולאה; אין עיסוק עם ביטויים לוגיים פשוטים או מורכבים שיבקרו את הלולאה ולכן מבנה לולאה זה היה ברירת המחדל בתכנית בעת שבחרנו את מבנה הלולאה הראשונה בה נשתמש.

אם לא משלבים משתנה למונה – תבנית מנייה או משתנה לסכום: תבנית סכום – דרגת הקושי להבנה וליישום נמוכה ומאפשרת לנו לשלב את מבנה לולאת **'לעולמים'** בשלב מוקדם בתכנית.

בסביבות העבודה בהם הבחירה בלולאה היא בחירה של בלוק מתאים, מומלץ לשלב את המבנה מוקדם ככל האפשר. בסביבות עבודה בהם התלמידים נדרשים ליצור את התנאי הבסיסי של לולאת 'לעולמים' ישנה מורכבות מסוימת וצריך לכתוב ערך שיבטיח שהלולאה לא תפסק

לדוגמה: `While (true){`

.....

}

שזה יותר מורכב לתלמידים בבית ספר יסודי בהשוואה לבחירת בלוק של לולאת 'לעולמים'.

מבנה לולאה **'חזור _ פעמים'**



מבנה זה מעורר אצל תלמידים שתי שאלות חשיבה:

- האם אני נדרש להשתמש באלגוריתם במבנה לולאה
- האם אני צריך מבנה לולאה של **'חזור __ פעמים'**

בכל הפעמים בהן השתמשנו בדוגמאות בלולאת חזור נעשה מאמץ לבחור דוגמאות שיקלו על ההתמודדות של התלמידים עם הדילמות הללו. בחרנו דוגמאות כמו 'רובוט ינוע סביב ריבוע דמיוני' וחיברנו את האלגוריתם עבור הרובוט אל האלגוריתם לדמות בסקרטץ' שתצייר ריבוע. כאן אנחנו מתחברים לידע קודם של התלמידים לגבי תכונות ריבוע – הרעיון של ארבע צלעות שוות וזוויות של 90 מעלות שוות מאפשר את קפיצת החשיבה לרעיון של לולאה שתתבצע 4 פעמים, וזה נעשה קל יותר אם מחברים את סביבת סקרטץ' עם דמות שמציירת ריבוע על הבמה. זהו מעבר מחשבתי כמעט ישיר לאלגוריתם עבור התנועה של הרובוט סביב ריבוע דמיוני, ולשימוש במבנה לולאה **'חזור __ פעמים'**, ולהבנה שצריך לנוע קדימה (אפשר גם אחורה) ולפנות 90 מעלות. לאחר שפתרנו את הבעיה עבור הדמות על הבמה נשארו רק השאלות הטכניות לגבי מהי או מה הן ההוראות לתנועה קדימה ברובוט, ומה הן ההוראות לפנייה ב-90 מעלות.

מבנה לולאה **'חזור עד ש'**



לולאת **'חזור עד ש'** מובילה אותנו לקושי אמיתי מול תלמידי יסודי.

מבנה זה מעורר שתי שאלות חשיבה:

- האם אני נדרש להשתמש באלגוריתם במבנה לולאה.
- האם אני צריך מבנה לולאה של **'חזור עד ש'**

א. בחשיבה האלגוריתמית והלוגית כדי לענות על השאלה הראשונה התלמיד צריך לחשוב על הבעיה שהמטלה מעמידה בפניו: מה נדרש ממנו ולמה רעיון של לולאה מקדם אותו לפתרון. אם נניח שהרובוט

צריך להסתובב בחדר ולחפש נר דולק. התלמידים נדרשים להגיע להבנה שהחיפוש הוא סדרה של תנועות עם תנאים מסוימים שאמורה להניע ולקדם את הרובוט לקרבת הנר – צריך לעבור את המשוכה של 'הרובוט יחפש עד שימצא את הנר'. זהו אתגר לוגי ומחשבתי לתלמיד צעיר ובוודאי שהאתגר מתעצם אם יש התניות נוספות בתהליך.

ב. לאחר ההבנה של התלמיד שאכן יש כאן לולאה שמסתיימת רק כאשר מתרחש אירוע מסוים (נניח הרובוט הגיע לקרבת הנר), עליו לנסח את ההוראות 'שמקרבות' את הרובוט לנר, כלומר עליו לחשוב אלו הוראות מקדמות את הרובוט אל היעד שהוגדר. במקרים פשוטים כמו:

'שחקן במשחק מחשב מתחיל עם עוצמה 10, אם העוצמה מגיעה ל-0 הסתיים המשחק', החשיבה שהורדת עוצמה מקדמת את התהליך היא ברורה יותר מאשר במקרה: 'חיישן טמפרטורה ימני מחזיר טמפרטורה יותר גבוהה מחיישן טמפרטורה שמאלי ולכן יש לפנות ימינה'. בהפעלת לולאת **'חזור עד ש'** תלמיד צריך להתמודד עם השאלה האלגוריתמית הזו בהתאם למטלה שקיבל ומתוכה לנסח את תנאי העצירה כמו בדוגמה של הנר: **'חזור עד ש'**'חיישן טמפרטורה >200'.

דוגמאות ללולאת 'חזור עד ש'

א



ב



ג



דוגמאות בסיסיות ללולאת 'חזור עד ש'

רובוט נע קדימה ועוצר על פס שחור

1. **'חזור עד ש'** חיישן פס שחור = 0

   1.1 **'נוע קדימה'**

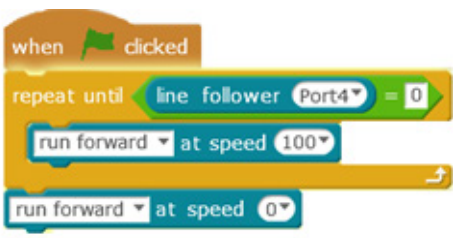
2. **'עצור'**

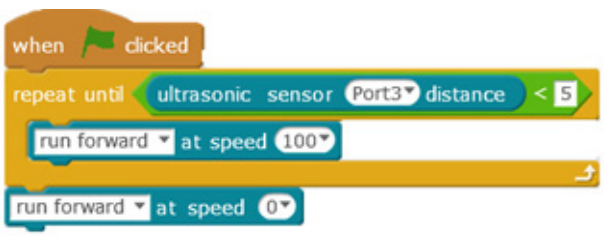
רובוט נע קדימה ועוצר 10 ס"מ לפני מכשול

1. **'חזור עד ש'** חיישן מרחק < 10

   1.1 **'נוע קדימה'**

2. **'עצור'**





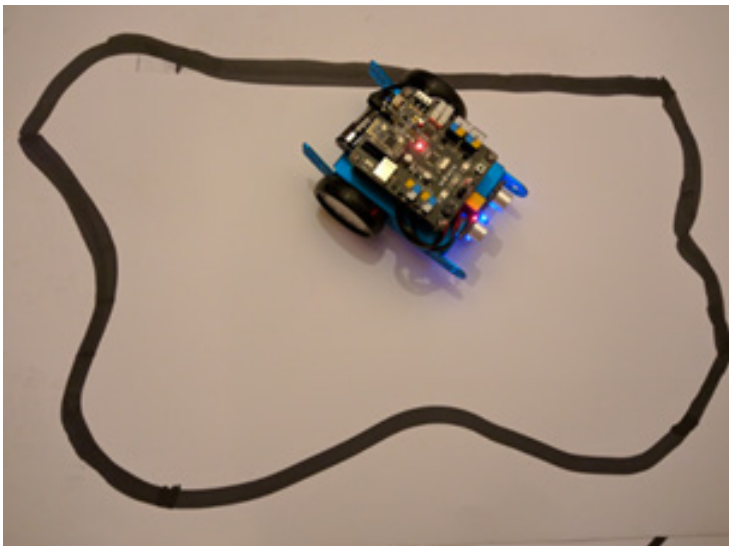
נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
ריענון משפט תנאי **'אם ואם לא'**	הסבר באמצעות דוגמאות פשוטות כמו $x>0$ ואם לא; לדבר על מה קורה עם $x=0$, לעיתים תלמידים לא מבחינים. להראות דוגמאות עם שני תנאים פשוטים, ואת אותה דוגמה עם משפט תנאי מורכב.		✓		תלמיד יכתוב קטעי תסריטים עם המשפט **'אם ואם לא'.** ואותו קטע עם אותה מטרה עם שני משפט **'אם'** פשוטים. **1 שעות**
האופרטור AND	מתן דוגמאות ללא תכנות, 'מהחיים', לתנאים עם DNA; הצגת טבלת האמת של DNA; להשתמש בייצוג בסיסי ואינטואיטיבי .	✓			התלמיד יכתוב קטעי קוד עם האופרטור AND ויכתוב קטעים שקולים עם שני משפטי 'אם' מקוננים, חשוב לשלב בתנאים ערכי חיישנים. **1 שעות**
האופרטור OR	מתן דוגמאות ללא תכנות, 'מהחיים', לתנאים עם OR; הצגת טבלת האמת של OR; להשתמש בייצוג בסיסי ואינטואיטיבי.	✓			התלמיד יכתוב קטעי קוד עם האופרטור OR ויכתוב קטעים שקולים עם שני משפטי 'אם', חשוב לשלב בתנאים ערכי חיישנים. **1 שעות**

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	רובוטיקה	ציוני דרך
יישום משפט תנאי עם האופרטורים AND ; OR	דוגמאות בסיסיות עם שני ביטויי תנאי, אפשר רק עם קבועים, אפשר גם עם משתנים, למתקדמים אפשר עם שילוב ביטויים מתמטיים.		✓	✓	התלמיד יכתוב מספר קטעי קוד שונים, עם משפט תנאי בשילוב האופרטורים AND ; OR התלמיד יכתוב קוד שקול ללא שימוש באופרטור AND ; OR למימוש בסביבת הרובוט מומלץ לשלב תנאי AND הבודקים ערך של חיישן עם ערך משתנה וכך גם לגבי האופרטור OR ; אפשר לשלב הוראות תנועה, הוראות צלילים, אורות ומשתנים. **2 שעות**
לולאת **'חזור עד ש'**	להציג את מרכיבי הלולאה, את רעיון שצריך להיות מנגנון המקרב את סיום הלולאה, וכן את החשיבה על תנאי העצירה.		✓	✓	תרגול דוגמאות עם מבנה לולאה **'חזור עד ש'** ; יישומים עבור סביבת סקרטץ' עם דמות ; יישומים עבור הרובוט. **3 שעות**

פרק י' מערכות בקרה בחוג סגור - הרחבה

בפרק זה נעמיק את הידע שלנו ונשכלל את יכולות היישום של הוראות תכנות הרובוט. נשלב את ההוראות בסיפורים מעניינים על ההרפתקאות של הרובוט. ונראה כיצד התכניות שאנחנו כותבים שומרים עליו מכל פגיעה. זהו מודול הרחבה לתכנית הבסיס של 60 שעות. ישנם מספר אתגרים בפרק זה והמורים מוזמנים להוסיף ולחדש, ולתת לתלמידים אתגרים בצוותי חשיבה. מאוד חשוב לספר סיפור סביב המטלה, כגון: רובוט המגן על גדר בגבול, וכדומה. ניתן לחשוב על רעיונות המשלבים אינטראקציה של שני רובוטים ויותר.

הרובוט שומר על בסיס החלל



המשימה של הרובוט היא לשמור על הבסיס ולסייר לאורך הגבול, אם הרובוט מרגיש תנועה חשודה לפניו הוא מדליק נורות אדומות ומשמיע קול אזעקה.

רעיון כללי:

1. '**המתן**' ללחיצה על כפתור
2. '**לעולמים**'
   - 2.1 '**סייר**' לאורך גבולות המחנה
   - 2.2 '**אם זוהה עצם**' לפני הגבול
     - 2.2.1 '**הדלק אורות**' חירום אדומים
     - 2.2.2 '**השמע קול**' הזעקה

גישת 'מלמעלה – למטה'

המרכיבים שיש ביישום:

סיור לאורך הגבול ושמירה;

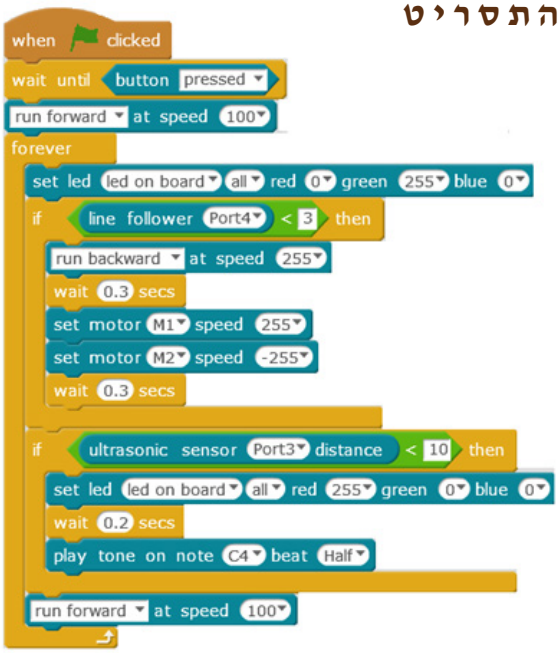
חיפוש עצמים ליד הגבול;

הפעלת נוהל חירום במקרה של זיהוי עצם.

האלגוריתם המפורט:

1. המתן ללחיצה על כפתור (הכפתור על לוח הבקרה)
2. **'נוע קדימה'**
3. **'חזור לעולמים'**
   - 3.1 **'הדלק אור ירוק'** בשתי הנורות
   - 3.2 **'אם רואה פס שחור'**
     - 3.2.1 **'פנה'** ו**'חזור לאחור'**
   - 3.3 **'אם רואה עצם'**
     - 3.3.1 **'השמע צלילים'** ו**'הדלק אורות'** אדומים

התסריט



חשוב! הערכים של זמן נסיעה לאחור, של הפנייה במקום, של מהירויות של המנועים – תלויים במאפיינים רבים כולל: סוג הגלגלים, סוג המשטח עליו נוסע הרובוט, תאורת החדר, עובי הפס השחור ועוד.

מטלה: אורות ומרחק מהרובוט

במטלה זו ניצור קשר בין חיישן המרחק לתדירות הדלקת נורות הלד על הלוח.

הרובוט יהיה בתנועה וככל שיתקרב לעצם נורות הלד על הלוח יהבהבו בתדירות גבוהה יותר

הרעיון הכללי:

1. **'סע ישר'**
2. **'חזור עד ש'** מרחק מהעצם <5
   - 2.1 **'הדלק לדים'** על הלוח בצבע אקראי
   - 2.2 **'המתן'** (20/מרחק מהעצם) שניות
   - 2.3 **'כבה נורות'**
3. **'עצור תנועה'**

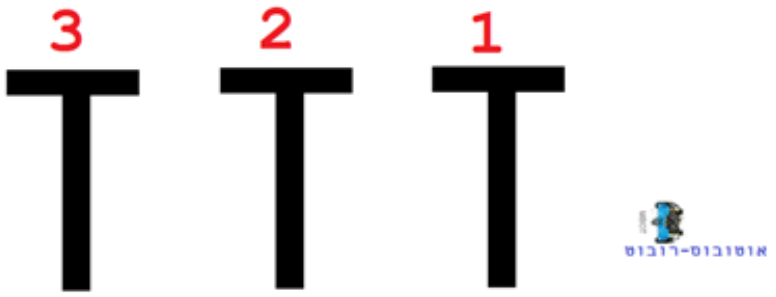


פרק יא' שילוב מערכות ומבנה בקרה מתקדמים – פרויקטים

מטלת אוטובוס-רובוט

במטלה זו הרובוט שלנו הוא אוטובוס. האוטובוס ממתין לנוסעים בנקודת היציאה.

לפני תחילת הנסיעה האוטובוס מקבל נתון – מספר התחנה שעליו להגיע, יוצא לנסיעה ונעצר כשהוא מגיע לתחנה כדי להוריד נוסעים. בקו של האוטובוס יש 3 תחנות.



עלינו לכתוב תכנית לרובוט שתכלול:

קליטת מספר התחנה

נסיעה לתחנה המבוקשת ועצירה

אלגוריתם כללי – עם הרעיון הבסיסי

1. **'קלוט מספר'** תחנה מבוקשת // קליטת מספר התחנה שיש להגיע
2. תחנה נוכחית – 0 // האוטובוס נמצא בתחילת הדרך
3. **'קלוט'** תחנת יעד של הנוסעים
4. **'סע קדימה'**
5. **'כבה נורות'**
6. קלט מחיישן פס שחור===> חיישן אור
7. **'חזור עד ש'** תגיע תחנה נוכחית = תחנת היעד של האוטובוס // האוטובוס נוסע עד לתחנת היעד
   - 7.1 קלט מחיישן פס שחור===> חיישן אור
   - 7.2 **'אם חיישן פס שחור'** הגיע לתחנת אוטובוס // בדיקה האם החיישן רואה שחור
   - 7.3 **'השמע צליל'** "הגעתי לתחנה "

7.4 **'סע קדימה'** 0.2 שניות// נסיעה קדימה אחרי הפס השחור

7.5 קלט מחיישן פס שחור===> חיישן אור

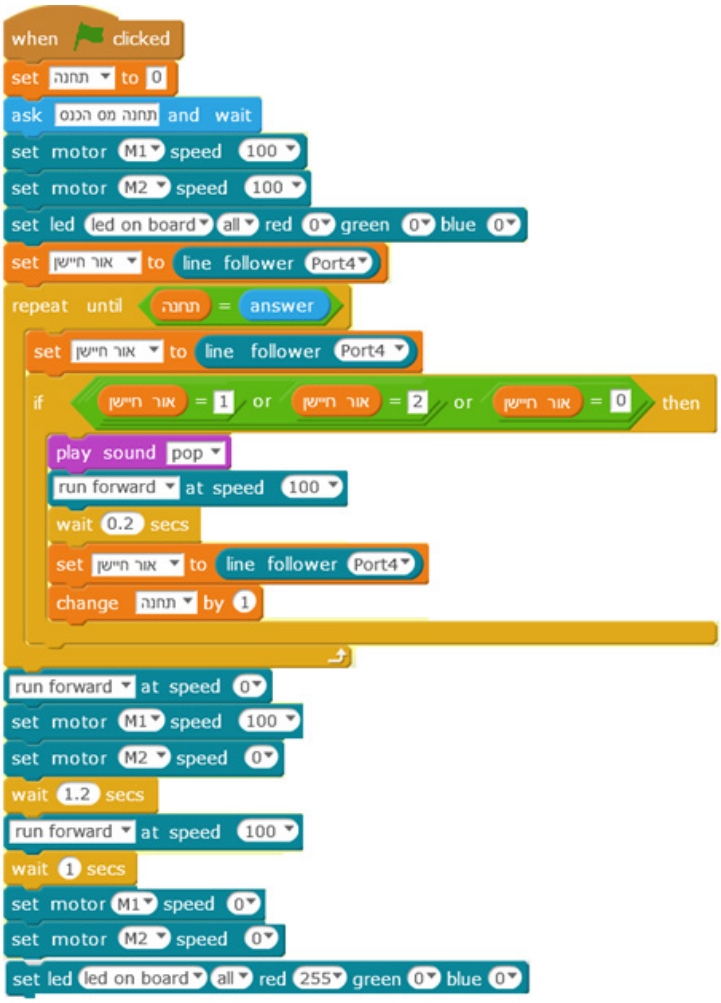
7.6 תחנה נוכחית + 1===> תחנה נוכחית // האוטובוס נוסע לתחנה הבאה

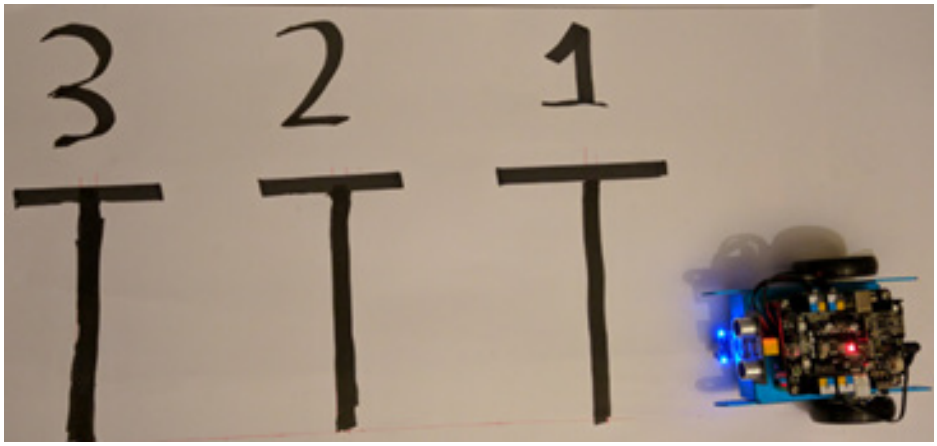
7.7 **'עצור'**

7.8 **'פנה ימינה' 'סע לתחנה'**

7.9 **'הדלק אורות'** אדומים

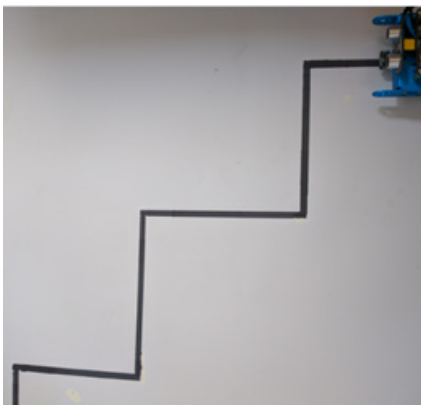
תסריט – אוטובוס-רובוט נוסע בקו נסיעה עם שלוש תחנות





מטלה - תנועה 'במדרגות'

הרובוט נע לאורך מסלול מדורג ואינסופי, אורך כל צלע במסלול אינו ידוע ועל הרובוט לפנות שמאלה וימינה כדי להמשיך על הקו.



הרעיון המרכזי :

1. **'לעולמים'**

1.1 **'חזור עד ש'** חיישן עוקב פס שחור = 1

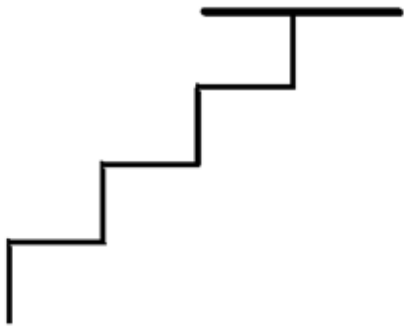
1.1.1 **'נוע קדימה'**

1.2 **'פנה שמאלה'**

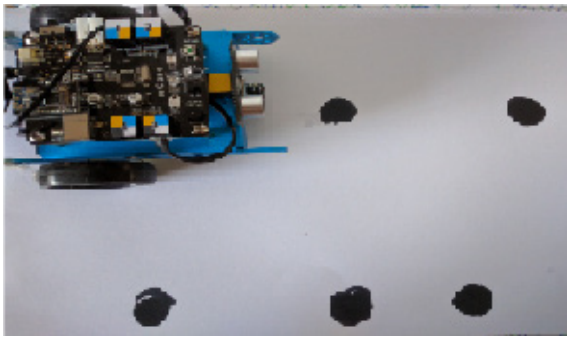
1.3 **'חזור עד ש'** חיישן עוקב פס = 2

1.3.1 **'נוע קדימה'**

אפשר להמיר את ההוראה לעולמים עם: **'חזור עד ש'** חיישן עוקב פס = 0 שהמשמעות שלה שיש סוף לקו המדורג בשילוב קו סיום במדרגה האחרונה.



מטלה – חיפוש מוקשים (מקור: השראה מתרגיל של אהוד עמיחי – הבלוג של איריס)



הצגת המטלה

הרובוט מחפש מוקשים בשדה מוקשים. הרובוט צריך לנוע ולסרוק את השדה 'רצועה' אחר 'רצועה' 'תלם' אחר 'תלם' כמו שטרקטור חורש שדה: משלים תלם ומבצע סיבוב 180 מעלות לתלם הבא.

שאלה לדיון

כיצד הרובוט ידע שהגיע לקצה השדה ועליו לבצע סיבוב?

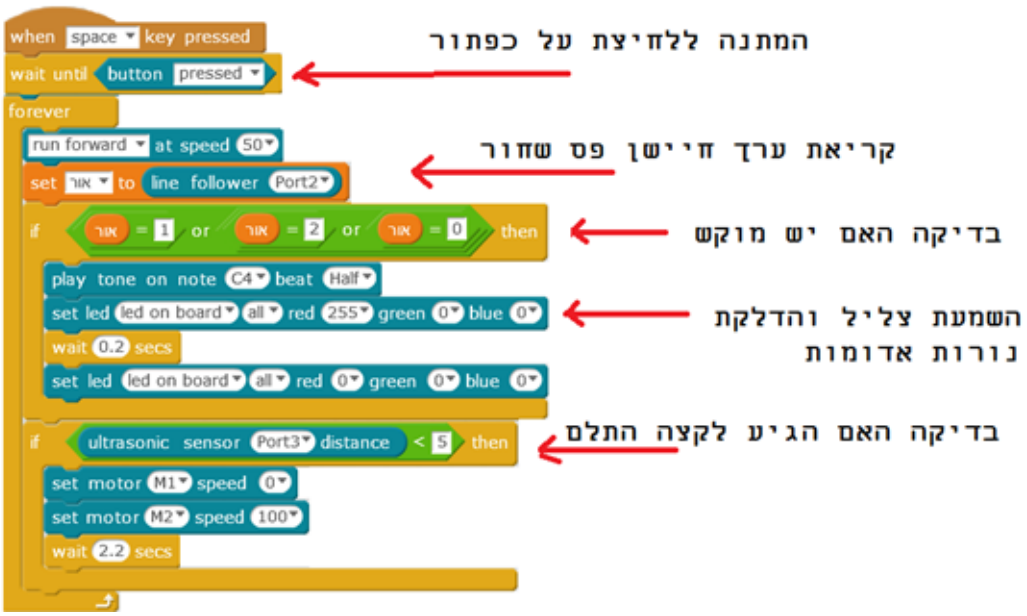
התשובה

בכל צד של שדה המוקשים יהיה עצם המהווה את גדר השדה, הרובוט יבצע סיבוב 180 מעלות לתלם הבא, כשהמרחק של הרובוט מהעצם המגביל את השדה יהיה קטן מהערך שבחרנו.

אלגוריתם:

1. **'המתן'** ללחיצה על כפתור
2. **'לעולמים'**
   - 2.1 **'סע קדימה'**
   - 2.2 קריאת חיישן אור==> אור
   - 2.3 **'אם'** זיהוי מוקש
     - 1.9.1 **'השמע צליל'**
     - 1.9.2 **'הדלק נוריות'**
     - 1.9.3 **'המתן** 0.5 **שניות'**
     - 1.9.4 **'כבה נוריות'**
   - 2.4 **'אם מרחק'** < 5 // הרובוט הגיע לקצה שדה המוקשים
     - 2.4.1 **'פנה 180 מעלות שמאלה'** // פנייה לתלם הבא

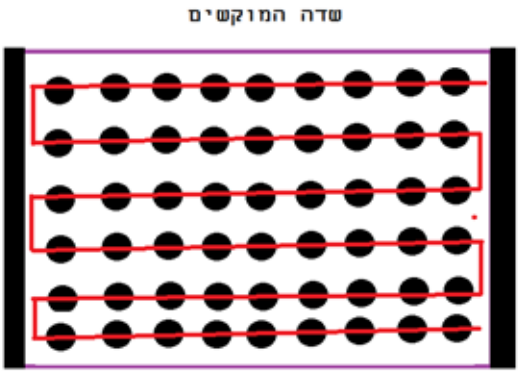
התסריט



לדיון

האם האלגוריתם פותר בעיית סריקת שדה מוקשים עם שתי שורות של מוקשים? חשוב! לדון על הנושא של נכונות פתרון.

הפתרון הוא רק לשתי שורות משום שבכל פנייה יש להחליף את כיוון הפנייה: פעם פונים ב- 180 מעלות שמאלה, ופעם פונים ב- 180 מעלות ימינה. רעיון נכונות של תכנית הוא מרכזי במדעי המחשב, וחשוב מידי פעם להציג או לפתח עם התלמידים את פתרון הבעיה, להריץ את הפתרון ולתת לתלמידים לחקור ולשאול שאלות לגבי נכונות הפתרון. נכונות תכנית ברמה התיאורטית מוגדרת כתכנית שפועלת עבור כל קלט ולכן תכנית שפועלת רק על שדה מוקשים עם שתי שורות אינה נכונה באמת – והדיון הזה חשוב יותר מפתרון הבעיה.



אנחנו רואים שהפניות הם שמאלה וימינה

אתגר

איך אפשר לפתור את שינוי הפנייה?

לסיכום

מטלה זו מציגה אתגרים רבים בנוגע לערכים של מהירות המנועים בתסריט ולזמן הפנייה. הם תלויים בתנאים המסוימים של הסביבה בה מופעלת התכנית וגם בעוצמת הסוללות ופרמטרים סביבתיים נוספים.

שער שלישי

תכנית חלופה המוצעת לכיתה ו'

חלק שלישי של מודול מדעי המחשב

היקף 30 שעות.

חלק זה כולל:

- פרק על לבנת משתמש : הגדרה ושימוש
- פרק ביצירת סביבות חקר במתמטיקה (הסתברות)
- פרק על שיבוט ורשימות

התכנית בשער שלישי מהווה את אחת החלופות לתוכנית כיתה ו.

פרק א' לבנות משתמש – תכנון מודולרי ומובנה



בלוק משתמש

לבנה אישית היא אוסף הוראות שמתכנת יוצר על מנת לבצע מטלה מסוימת. אוסף ההוראות מקבל שם ייחודי ונוצרת הוראה אחת שמאגדת את כל ההוראות שנכתבו, כך שבכל פעם שהמתכנת רוצה להשתמש ולהפעיל את אוסף ההוראות הוא בוחר רק בהוראה החדשה שיצר.

יצירת הוראות חדשות על ידי המתכנת בסביבת סקרטץ' מהווה רעיון מתקדם נוסף שמשמעותו מעבר להיבט הטכני היא 'הרחבת השפה' – יצירת סביבת עבודה עם קבוצת הוראות שאנחנו יכולים להשתמש בהן, כשבאפשרותנו גם ליצור הוראות חדשות.

זימון

הפעלת בלוק משתמש.

ארגומנט

השם המקובל לערך אותו מעבירים לבלוק משתמש (אם הוגדרו כאלה).

פרמטרים

הערכים המועברים לבלוק משתמש בעת הזימון של הבלוק מועברים למשתנים שמופיעים בכותרת הבלוק. אנו יכולים להשתמש במשתנים אלו בתוך בלוק המשתמש. השם המקובל למשתנים שמכילים ערכים אלו הוא פרמטרים.

הסבר: כיצד מגדירים בלוק משתמש ואופן השימוש בו

כותבים תסריט ונותנים לו שם מיוחד. כך יוצרים בלוק חדש שמופיע בקבוצת ההוראות

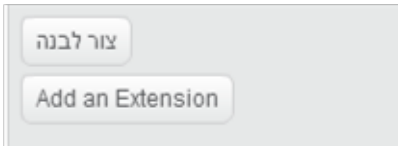


אופן יצירת לבנה – בלוק חדש

נבחר את קבוצת ההוראות ללבנים נוספות



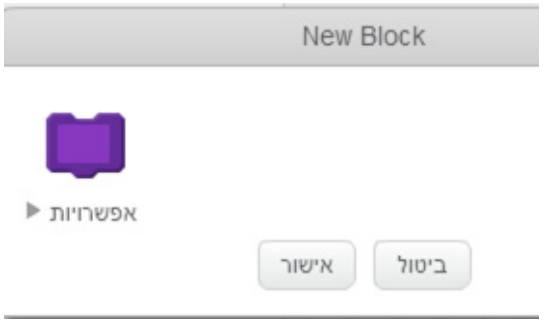
ונקבל את המסך הבא:



לפנינו שתי אפשרויות. האפשרות השנייה אינה נכללת בנושאי התכנית, ונתמקד באפשרות הראשונה



לחיצה על כפתור אפשרות זה יציג לנו את תיבת השיחה הבאה:



נעביר את הסמן מעל התיבה הסגולה ונלחץ על הכפתור השמאלי. הקלקה זו מאפשרת להקליד את שם הלבנה שאנחנו רוצים ליצור, נקליד את המילה ריבוע.

הערה: בשלב זה נתעלם מהתפריט של אפשרויות שיוסבר בשלב מאוחר יותר.

אישור השם יוצר בלוק הוראות חדש בשם 'ריבוע' ומאפשר להכניס את ההוראות השייכות לבלוק זה.



יצרנו את הבלוק החדש שהפעלתו מציירת ריבוע על הבמה. בשלב הבא צריך להציג את שילוב הבלוק-לבנה בתסריט. כך נוכל להבין את היתרון בשימוש בבלוקים של המשתמש.



שילוב הבלוק בתסריט



התסריט שמופעל כאשר לוחצים על דגל ירוק ברור מאוד – לולאה שמתבצעת 12 פעמים – בכל מחזור מציירים ריבוע, ולאחר מכן פנייה שמאלה.

והתוצאה:



זימון/הפעלת בלוק משתמש (הרחבת ההסבר על המושג)

שילוב הוראה שנכתבה על ידי המשתמש נקרא 'זימון' כי התהליך שמתרחש הוא הפסקת התסריט שמבוצע ומעבר אל תסריט אחר הנושא את הכותרת של הבלוק שזומן. עם סיום התסריט הכלול בהוראה של המשתמש ממשיך התסריט המקורי שזימן את הפעולה מהוראה אחרי זימון.

המחשת המושג זימון:

ניצור בלוק משתמש פשוט שנקרא לו 'זימון'.



התסריט הראשי, כולל הוראה המזמנת את הבלוק



בסיום ביצוע הבלוק 'זימון', חוזרת השליטה לתסריט הראשי, והוא ממשיך להתבצע מהוראה אחרי זימון (המחשת הזרימה של הביצוע).

איור: תהליך ביצוע זימון



פרמטרים וארגומנטים

תכונה חשובה נכללת במנגנון כתיבת בלוקים של המשתמש. בעת זימונו אפשר להעביר לבלוק שיצרנו ערכים של מספרים ו/או מחרוזות לבלוק, כך שבאופן הזה הבלוק עובד שונה בכל פעם שהוא מזומן והתכונה הזו מאפשרת להגדיר בלוקים עם גמישות. לערכים שאנו מעבירים קוראים **ארגומנטים**, ולמשתנים בבלוק שהוגדר אנו קוראים **פרמטרים**.

הערכים (הארגומנטים) שאנו מעבירים לבלוק מאוחסנים במשתנים (הפרמטרים) שמוגדרים בתוך הבלוק. משתנים אלו אינם מוכרים מחוץ לבלוק, הם מוכרים רק בעת הרצת הבלוק.

דוגמה בסיסית להמחשת רעיון העברת ארגומנטים לבלוק אישי:

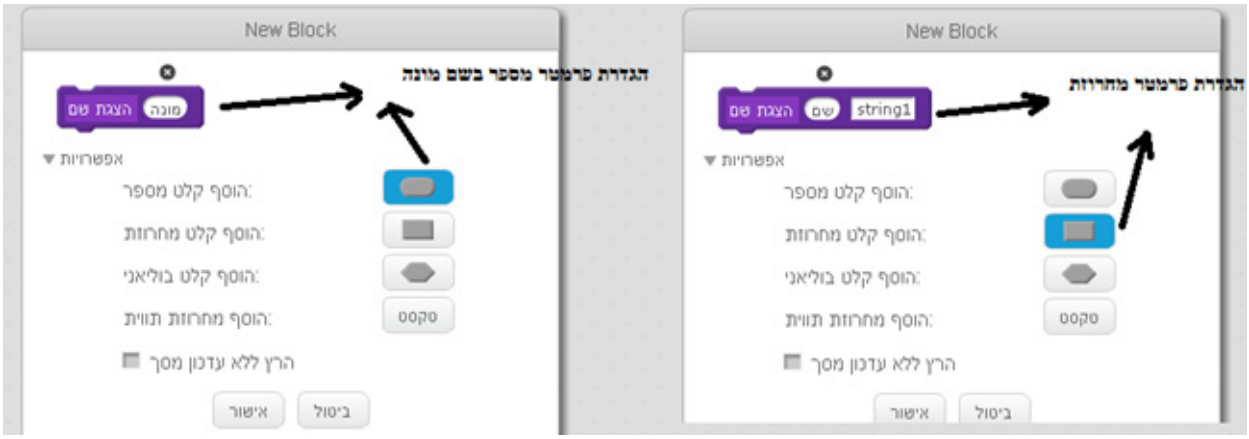
נגדיר בלוק שמקבל מחרוזת ומספר (כלומר שני פרמטרים) ונציג את המחרוזת מספר פעמים לפי ערך המספר.

אופן ההגדרה של פרמטרים ללבנה האישית:

בוחרים צור לבנה

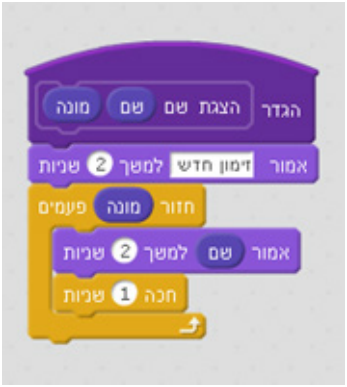


אם בחרנו להוסיף פרמטרים וסימנו עם העכבר על **אפשרויות** נקבל את תיבת השיחה הבאה:



תיבת השיחה מאפשרת הוספת פרמטרים ללבנה כך שאפשר לזמן את ביצוע הלבנה בכל זימון עם ערכים אחרים. בדוגמה הלבנה מקבלת מחרוזת (נניח שם של תלמידה), ומספר שלם, הפעולה מבצעת את ההוראה '**אמור**' עם המחרוזת שקיבלה כפרמטר – המשתנה 'מונה', והשם שהדמות תאמר מועבר עם הפרמטר 'שם'.

איור: בלוק אישי – הצגת שם



העובדה שהגדרנו את שני הפרמטרים הללו – משתני הבלוק – מאפשרת לנו לזמן את הלבנה 'הצגת שם' עם שני ערכים שונים בכל זימון.

תסריט עם שני זימונים והעברת ערכים (ארגומנטים) שונים לבלוק:

איור: זימון בלוק



פרוצדורה/ שגרה /רוטינה / שיטה

בסביבת סקרטץ' כאשר מוגדרים בלוקים חדשים על ידי כותב התוכנה הם מופיעים בקבוצת הוראות הנקראת 'לבנים'. חשוב לציין ש'לבנה חדשה' היא לבנה של דמות ואי-אפשר להשתמש בה בתסריטי דמות אחרת בפרויקט, אלא רק בדמות אחרת שנוצרה על ידי שיכפול. הרעיון של כתיבת ו"המצאת" הוראה חדשה בה ניתן לאחר מכן להשתמש בתסריטים שונים של הדמות מהווה כלי רב עוצמה שדרכו ניתן להרחיב את שפת התכנות בה אנו משתמשים וליצור הוראה שכוללת קבוצת הוראות. בנוסף ניתן לזמן אותה מכל מקום בכל תסריטי הדמות וזאת עם היכולת להעביר ערכים (ארגומנטים) שונים בעת הביצוע – כלי שחשיבותו מבוטאת בכך שהוא מסייע לתכנת תכנות מודולארי שמאפשר חסכון בכתיבת קוד חוזר.

לאורך שנות התפתחות שפות התכנות נקרא הכלי הזה בשפות שונות ובשמות שונים: פרוצדורה, שגרה, רוטינה, פעולה ובשנים האחרונות המילה **שיטה** הפכה להיות מושג נפוץ יותר מהמושגים האחרים.

בתכנית הוקפד שלא להשתמש במילה שיטה או בכל מילה אחרת, ולהישאר צמוד להגדרה יותר ארוכה ומסורבלת כמו: בלוק חדש המוגדר על ידי המשתמש (או לבנה), והסיבה היא פדגוגית – הרצון להישאר עם המונח בלוק או לבנה שהוא מונח שקשור למהותה של סביבת סקרטץ' כשפה ויזואלית.

בלוק בהגדרת משתמש או שיטה; בלוק בהגדרת משתמש או פרוצדורה – המשמעות זהה ומתייחסת לרעיון מרכזי בשפות תכנות – האפשרות להרחיב את השפה ובכך לתרום לתוכנה שנכתבת באופן ברור יותר, שניתנת לתחזוקה, שמאפשרת ותורמת לעבודת צוות, שחוסכת שורות קוד, זמן ושגיאות.

רעיון ויישום בלוקים של משתמש מאפשר לנו יישום של מודלי הפיתוח לכתיבת תוכנה בגישת 'מעלה-מטה' או תוכנה בגישת 'מטה-מעלה'. היישום של המודל אינו פשוט אולם כדאי להשקיע זמן ושעות למידה לתרגול פיתוח של יישומים (לא מורכבים) בדרך של תכנון, חלוקה למשימות, בניית בלוקים של המשתמש לפי המשימות שהוגדרו וחיבור כל המרכיבים לתסריט אחד שמפעיל ומקשר בין חלקי היישום. אפשר, ניתן ורצוי להשתמש באסטרטגיות ומודלים אלו בכל מטלה.

השילוב הכולל של הכלים שהוצגו בתכנית הלימודים:

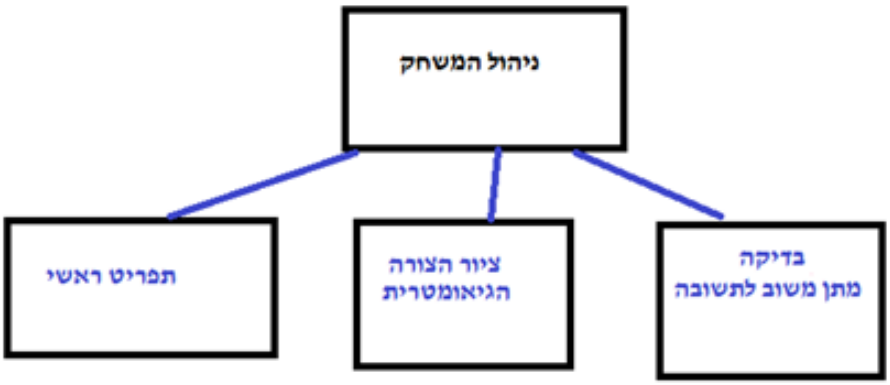
- אלגוריתמים מילוליים
- טבלת דמויות ואירועים
- תרשים תזמון תסריטים
- מודל פיתוח מעלה-מטה/מטה-מעלה

נותנים כלי תכנון לתלמידים והופכים את העבודה שלהם למאורגנת ללא ויתור על יצירתיות. כלים אלו משתלבים ומתכתבים עם מיומנויות המאה 21 בהתייחסות לאסטרטגיות בפתרון בעיות; שיתוף פעולה בצוותי עבודה ולמידה; חלוקת מטלות בצוות; שילוב ואינטגרציה של רעיונות. הכלים האלו מתאימים גם לרעיון של הטקסונומיה של בלום – יש רמות חשיבה שונות בכל יישום של הכלים וברמת ובמורכבות היישום.

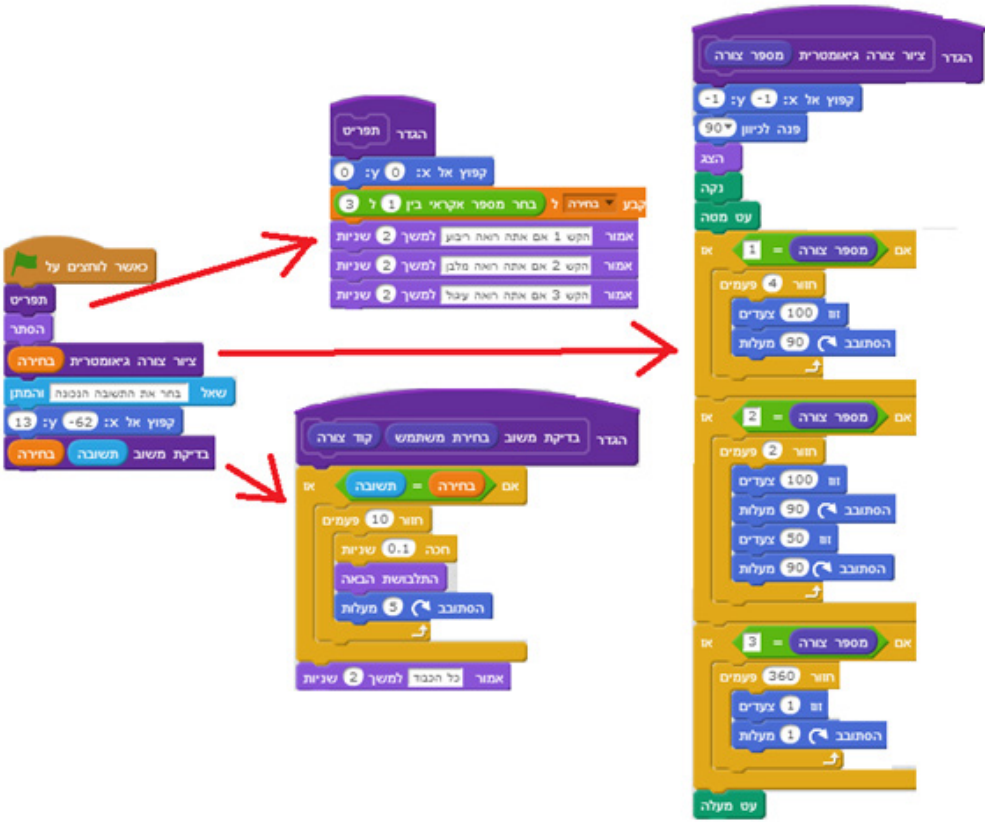
דוגמאות – תכנון יישום בשיטת מעלה-מטה

משחק זיהוי צורות

- על המסך מוצגות בצורה רנדומלית צורות הנדסיות: מלבן, ריבוע, עיגול
- על המשחק להשיב על השאלה: מה הוא רואה? תשובת המשחק נבדקת והוא מקבל משוב

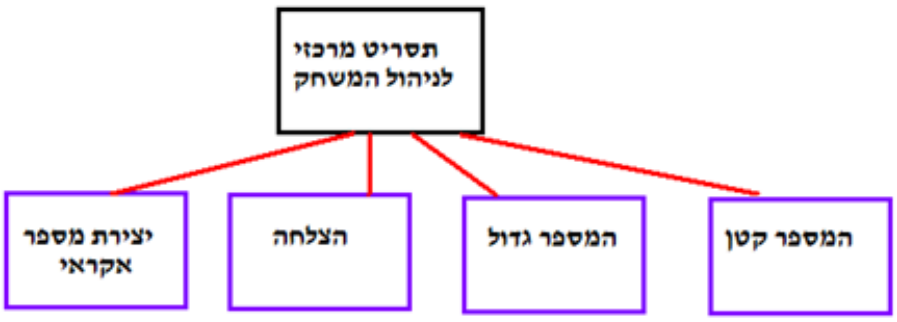


תסריטי לבנות המשתמש ותסריט הניהול של המשחק

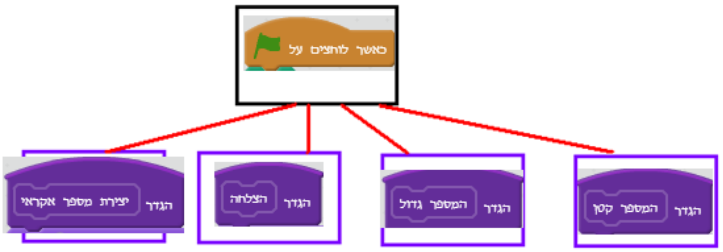


משחק - נחש את המספר

תיאור קצר של המשחק: המחשב "חושב על מספר" והמשחק מנסה לנחש מהו המספר. מתקבל משוב אם הניחוש של המשחק גדול מהמספר ואם הוא קטן ממנו, עד לניחוש הנכון.



מימוש המשחק לפי התכנון עם בלוקים של המשתמש



התסריטים של המשחק

התסריט המרכזי

- קובע ערכים התחלתיים למשתנים
- מפעיל את הבלוק 'יצירת מספר אקראי'

לולאת 'חזור עד ש' - בכל מחזור הלולאה קולטת את הניחוש של השחקן ובודקת את הניחוש

- אם קטן מהגודל של פיקו, מופעל (זה נקרא זימון) הבלוק המספר קטן
- אם גדול מהגודל של פיקו, מופעל (זה נקרא זימון) הבלוק המספר גדול
- אם שווה לגודל של פיקו, מופעל (זה נקרא זימון) הבלוק הצלחה

הצגת מבנה יישום באופן הזה ללא פתרון מלא, מאפשר לך המורה ליצור דיון על מרכיבי הפרויקט; להניח לצוות לעבוד כאשר כל חבר צוות בונה מרכיב ביישום וביחד מבצעים אינטגרציה של הפתרון כמו צוות פיתוח אמיתי – תרגול ויישום של מיומנויות המאה 21.

נושא משנה	רעיונות והדגשים	חישוביות, אלגוריתמיקה ומיומנויות המאה 21	כתיבת קוד	ציוני דרך
בלוק משתמש רציונל	יש להדגיש בפני התלמידים את היתרונות הבאים: • חסכון בכתיבה קוד • טיפול יותר שיטתי וקל במשימות מורכבות • מתאים לעבודת צוות	✓		הצגת מטלות (לא ממוחשבות) ודיון וביצוע של חלוקה לתתי משימות. **1 שעות** מבט על מטלות שבוצעו בעבר ושאילת שאלות האם אנחנו רואים איך מפרקים את המטלה למטלות משנה. **1 שעות**
הצגת מטלות וחלוקה למטלות משנה	יוצגו מטלות בכיתה, דיון על תתי המטלות, חלוקה לצוותי תלמידים הכותבים מרכיבים של היישום. ביצוע אינטגרציה של המרכיבים, יישום גישת מטה-מעלה, מעלה-מטה.	✓	✓	ביצוע כל שלבי התהליך ממטלה, ייצוג תרשים למעלה-למטה, פירוק לתתי מטלות, ניתוח וייצוג תרשים תזמון תסריטים וטבלת דמויות ואירועים. **4 שעות**

פרק ב' תכנית הסתברות, אלגוריתמיקה וכתיבת קוד - 10 שעות

"THERE'S NO SENSE IN BEING PRECISE WHEN YOU DON'T EVEN KNOW WHAT YOU'RE TALKING ABOUT"

(ג'ון פון נוימן)

חשיבות הנושא: למה חשוב ללמוד מושגים בסיסיים בהסתברות?
הסתברות וסטטיסטיקה מהווים הכנה ובסיס ללימוד מכונה ומדע נתונים.

חלון לנושא: ההסתברות שמאורע יקרה:

ההסתברות שמאורע יקרה היא ערך שנע בין 0 ל-1; ההסתברות שמאורע יקרה שווה למספר האפשרויות שבהן המאורע קורה לחלק במספר האפשרויות הכללי; המושג למספר האפשרויות הכללי הוא מרחב המדגם; מרחב המדגם הוא סך כול המאורעות האפשריים; מרחב מדגם אחיד הוא מרחב בו ההסתברות היא שווה לכל אפשרויות של מאורע בסיסי.

הטלת קוביה

מרחב המדגם הוא {1,2,3,4,5,6} מספר האפשרויות הוא 6,

לכן ההסתברות שנקבל את אחת התוצאות היא 6/1.

עתה המשך הדיון הוא הרלוונטיות של הנושא לתכנית הלימודים של כיתה ה' במדעי המחשב.

נושא הסתברות כרוך בזיהוי ייצוג וביצוע פעולות חשבון על שברים פשוטים. חישוב הערכים של ההסתברויות במחשב או בתהליך חישוב ידני שמשנה את הערכים לערכים עשרוניים. אלו נושאים שתלמידים צריכים לדעת עד סוף כיתה ד' ולהשתמש בידע בהמשך הלימודים בבית הספר היסודי.

מהו האתגר וההזדמנות שמזמנת תכנית הלימודים במדעי המחשב? אנחנו יכולים להנגיש רעיונות מתחומי ההסתברות על ידי שילוב של אלגוריתמיקה וכתיבת קוד שמציגים לנו ניסויים במרחבי הסתברות שיוצרים סביבת חקר של נושא עם הדמיה. באמצעות הניסויים מחשבים את ההסתברויות למאורעות שונים וממרחב הניסוי אנחנו יכולים לקיים דיון יותר כללי, לבצע הכללה של התוצאות ולהגיע לחישוב של הסתברות למאורע כלשהו בדרך הניסוי. הצגה מקבילה של חישוב מאורע הסתברותי באמצעות שברים ופעולות על שברים יוצרת את הסינרגיה של חיבור בין היבט מתמטי לבין היבט של אלגוריתמיקה וכתיבת קוד.

כתיבת קוד ע"י התלמידים מאפשרת יצירת שדה ניסויים וחקר בו משתלבים רעיונות מתחומים שונים בצורה ויזואלית ומעניינת.

אם נתחבר לאמירה של ג'ון פון ניומן בנוגע לכמה חשוב לדעת על מה מדובר לפני שמנסים להיות מדויקים, סביבת החקר מאפשרת לתלמיד לחקור מרחב מדגם, לבצע הדמיות ולקבל מושג על ההסתברות של האירוע, נניח תוצאה 3 או מספר זוגי במרחב קובייה, ובשלב השני לחשב במדויק את ההסתברות לאירוע. מלבד המושגים הבסיסיים והראשוניים, אין מעורבות של הוראה ויישום של מתמטיקה גבוהה, וכל החישובים שתלמידים יבצעו במסגרת יחידה זו הם חישובים עם ארבע פעולות החשבון על מספרים עשרוניים ושברים פשוטים.

הדמיה של הטלת מטבע

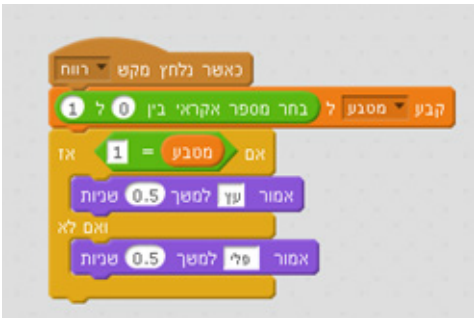
התלמידים יגדירו משתנה אחד, יכתבו תסריט קצר ויקבלו את הסימולציה הבסיסית לקבלת 1 או 0.



חשוב! לשוחח עם התלמידים: מדוע בחרנו ייצוג 1 או 0?, האם זה הייצוג היחידי? אפשר לדון בשאלה: האם נוכל לשנות את התסריט כך שחתול יגיד עץ או פלי במקום 0 או 1?

שאלה: האם נוכל לשנות את התסריט כך שחתול יגיד עץ או פלי במקום 0 או 1?

והתשובה שזה אתגר קצר וקל לתלמידים:



שיפור ההדמיה

ניצור דמות מטבע עם שתי תלבושות וברגע שנלחץ על הדמות היא תופיע עם הצד עץ או עם הצד פלי.

התסריטים הבאים מבצעים את ההדמיה של זריקת מטבע "למעלה" שינוי הצדדים עד "לנפילה" למטה.

שתי תלבושות לדמות:



התסריט הבא מבצע אנימציה של זריקת מטבע, להפעלה מקליקים על הדמות שעוברת מעץ לפלי וחזור עד להגעה למצב סופי שהוא עץ או פלי.

הדרך בה אנו מבצעים היא בחירת מספר בתחום 1-0, אם נבחר מספר 0 הבחירה תהיה עץ, אם נבחר 1 הבחירה תהיה פלי.

עריכת ניסוי

עריכת הניסוי באמצעות תסריט הכולל לולאה שתתבצע מספר פעמים לפי קלט. התסריט ימנה את מספר הפעמים שהתקבלה התוצאה עץ/פלי. בתום הניסוי יוצג היחס בין מספר התוצאות שהתקבלו, פלי/עץ, למספר ההטלות.

עלינו:

- ליצור משתנה שיספור את מספר הפעמים
- ליצור משתנה שיספור כמה פעמים קבלנו עץ
- ליצור משתנה שיספור כמה פעמים קבלנו פלי (שאלה לחשיבה: האם אפשר לחשב זאת?)

התסריט הבא מציג את ההדמיה עבור מספר הטלות מטבע עם הודעות, שינוי בתצוגה של המטבע והצגת התוצאות בסוף התסריט.

למען פשטות התהליך לא נציג את ההדמיה של חילוף שני צדי המטבע, אלא רק את התוצאה הסופית של "גלגול" המטבע ואת התוצאות הסופיות של כל הניסוי. כל תלמיד רשאי לשנות ולשפר את האנימציה.

התסריט המלא



- הקלקה על הדגל מפעילה את התסריט;
- המשתמש נשאל כמה ניסיונות ברצונו לבצע;
- בכל מחזור בלולאה מתבצע ניסוי הטלת מטבע:
- אם התוצאה 0, כלומר עץ (כך בחרנו):
- מוצגת תלבושת העץ;
- מתעדכן מונה התוצאה עץ;
- והדמות אומרת 'עץ'.

ואם לא:

- מוצגת תלבושת פלי;
- מתעדכן מונה התוצאה פלי;
- והדמות אומרת 'פלי'.

סיום הלולאה:

הדמות אומרת כמה פעמים התוצאה הייתה עץ,

וכמה פעמים התוצאה הייתה פלי.

שיעור ניתוח תוצאות ודיון - 3 שעות (שיעור פעיל)

הרצת התכנית מספר פעמים הניבה את התוצאות הבאות:

מספר הטלות	סה"כ עץ	סה"כ פלי	יחס בין תוצאה עץ למספר הזריקות	יחס בין תוצאה פלי למספר הזריקות
10	6	4	6/10	4/10
20	9	11	9/20	11/20
100	52	48	52/100	48/100
200	98	102	98/200	102/200

מטלה

כל צוותי התלמידים מפעילים את התסריט 'ניסוי הטלת מטבע':

- 50 פעמים
- 100 פעמים
- 200 פעמים
- 400 פעמים

הערה: רצוי להקטין את זמני ההמתנה בהוראה **'אמור'** ל- 0.1 שניות.

יש לבנות טבלה זהה לטבלה שהוצגה.

התלמידים יחשבו עם מחשבון את היחס המוצג עבור כל תוצאה בכל ניסוי.

דיון: מה קבלנו ומה לדעתכם ההסתברות לקבל תוצאה עץ או תוצאה פלי בניסוי זריקת מטבע.

סיכום ודיון

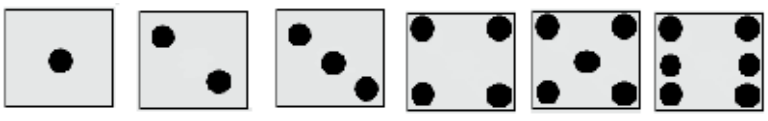
- מרחב המדגם של הטלת מטבע הוא (פלי ; עץ)
- ההסתברות לקבלת תוצאה עץ היא 0.5
- ההסתברות לקבלת תוצאה פלי היא 0.5

שיעור מרחב מדגם קובייה – 3 שעות

הדמית הטלת קובייה

קוביית משחק כמו כל קובייה כוללת 6 פאות, על כל פאה יש מספר נקודות שונה בתחום 1-6.

שש תוצאות המדגם:



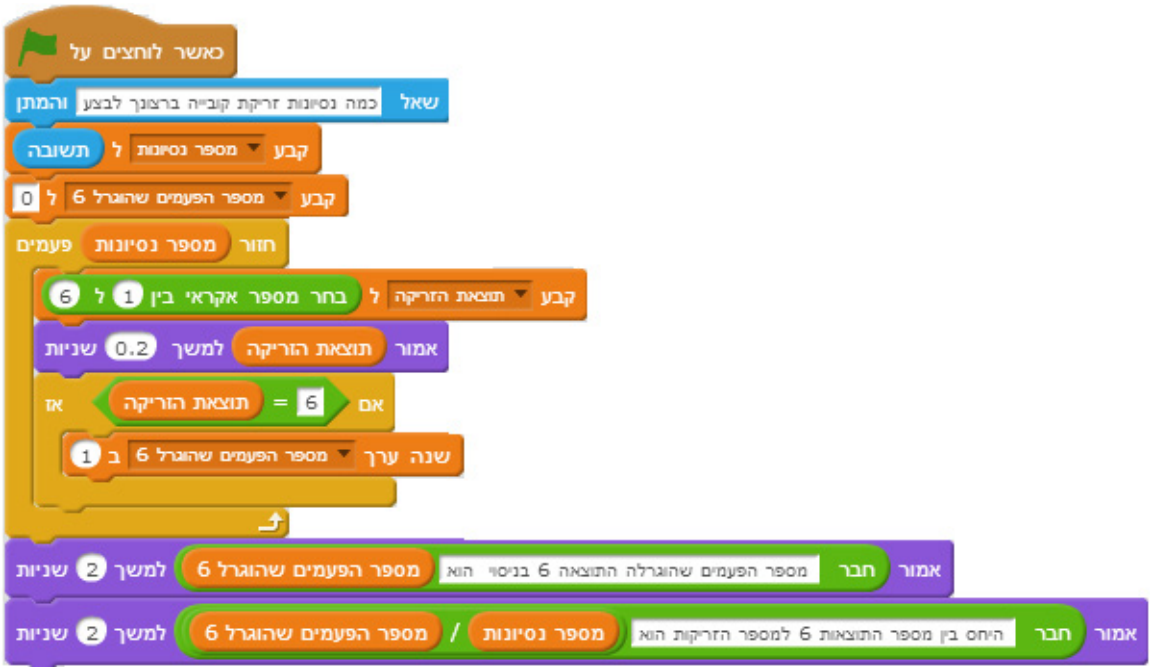
הדמיה

נבחר הדמיה בסיסית ללא אנימציה. תלמידים יכולים לבצע פרויקט וליצור הדמיית ניסוי 'הטלת קובייה' עם 6 דמויות מתחלפות. אם תלמיד בעל מיומנויות וידע בגרפיקה מעוניין ליצור הדמיה עם קובייה תלת ממדית אפשר לעודד זאת עם מגבלה על זמן ההשקעה בגרפיקה בשיעורים.

תסריט ניסוי 'זריקת קובייה'

משתני היישום

- מספר ניסיונות – מייצג את מספר הניסיונות שאנו זורקים את הקובייה
- תוצאת הזריקה – מייצג את תוצאת הזריקה של הקובייה, ערך בתחום 1-6



הרצת התסריט בניסוי:

עבור 500 ניסיונות זריקה: ב- 75 ניסיונות התוצאה הייתה 6.

היחס בין התוצאה 6 למספר הניסיונות הוא: 0.15

מטלות

הפעילו את התסריט עבור 100; 200; 600 ניסיונות. רשמו את התוצאות בטבלה.

מה המסקנה מהתוצאות של ניסוי 'הטלת קובייה'? מהי ההסתברות לקבלה התוצאה 6 בניסוי 'הטלת קובייה'?

שנו את התסריט לבדיקת ההסתברות לקבלת התוצאה 3 בניסוי או קבלת התוצאה 4.

שאלה:

תלמיד בכיתה אמר לחברו: בוא נשנה את התסריט ונבצע שני ניסויים:

- **ניסוי א:** חישוב ההסתברות לקבלת תוצאה >4 בניסוי 'הטלת קובייה'
- **ניסוי ב:** חישוב ההסתברות לקבלת תוצאה זוגית.

אם התוצאה של הניסוי הראשון גבוהה יותר מהתוצאה של הניסוי השני אני אתן לך לרכוב על האופניים החדשים שלי.

מה דעתכם: האם הסיכוי שהתלמיד ייתן לחברו לרכוב על אופניו גדול מהסיכוי שלא ייתן לחברו לרכוב על האופניים?

בצעו את הניסויים ובהתאם לניסוי חוו דעתכם.

האם תוכלו להציג נימוק מתמטי משכנע ללא צורך בשני הניסויים?

שיעור מרחב מדגם שתי קוביות – 3 שעות, פרויקט סיכום ודיון

ניסוי 'הטלת שתי קוביות'

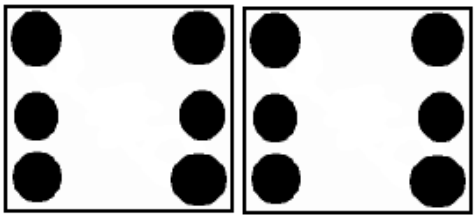
אם מטילים שתי קוביות, כמה תוצאות אפשריות מקבלים.

- התלמידים יכתבו תסריט קצר המבצע 'ניסוי הטלת שתי קוביות'
- התלמידים יעקבו אחר התוצאות המתקבלות משתי הקוביות, וירשמו בטבלה:

ניסיון מספר	תוצאת הטלת קובייה א'	תוצאת הטלת קובייה ב'

- התלמידים יקיימו דיון על התוצאות: כמה תוצאות יש במרחב המדגם? מהי ההסתברות של כל תוצאה?
- התלמידים יבדקו ברשת האינטרנט האם יש מידע שתומך במסקנה שלהם.
- התלמידים ינמקו את המסקנה בנימוק מתמטי המתבסס על הניסוי במחשב, על חקר שבוצע ואולי, לבעלי יוזמה מבין התלמידים, על ניסוי 'הטלת שתי קוביות' אמיתי עם קוביות משחק.
- התלמידים ישנו את התסריט כדי לקבל את ההסתברות לקבלת תוצאה שסכום הקוביות יהיה זוגי.
- התלמידים ישנו את התסריט כדי לקבל את ההסתברות לקבלת תוצאה שסכום הקוביות יהיה אי זוגי .
- התלמידים ישנו את התסריט כדי לקבל את ההסתברות לקבלת תוצאה ששתי הקוביות מראות את אותו מספר.

כל שחקן מתחיל במשחק שש-בש יתעניין לדעת: מהי ההסתברות לקבלת התוצאה:



תוצאה זו בתדירות גבוהה מסייעת לניצחון במשחק!

התלמידים יכתבו מסמך מסכם על נושא ההסתברות, יציגו את ההיבט המתמטי מדעי, יציגו את התסריטים שכתבו ואת התוצאות של הפעלת תסריטים אלו, ויעלו את המסקנות שלהם מלמידת הנושא.

פרק ג' שיבוט דמויות

שיבוטים בשפת סקרטץ' – Clone

גרסת סקרטץ' 2.0 כוללת חידושים רבים. אחד החידושים המאוד משמעותיים הוא שיבוט של דמות, רעיון רב עוצמה מההיבט הרעיוני ומההיבט של מכלול האפשרויות שהחידוש הזה מאפשר בכתיבת יישומים.

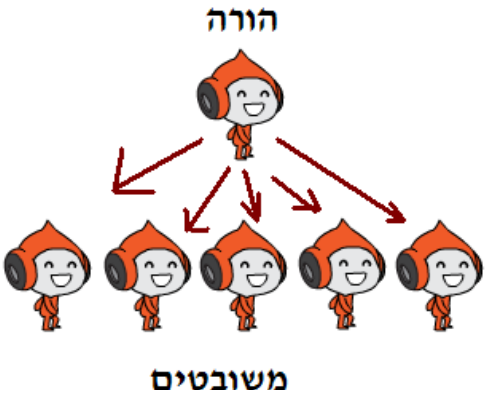
ראינו בתכנית הלימודים את הבעייתיות בהתמודדות עם כתיבת משחק מחשב שכולל דמויות רבות – לכל דמות יש כמה תלבושות, צלילים, משתנים ותסריטים כך שבמשחק בו יש 5 דמויות אנו עלולים לטפל בעשרות מרכיבים של הפרויקט.

ברור שלא ניתן לוותר על עושר של דמויות ביישום גדול, אולם נוצרת בעיה מרכזית במיוחד במשחקים בהם יש דמויות שאנו יוצרים מהם המון "יחידות".

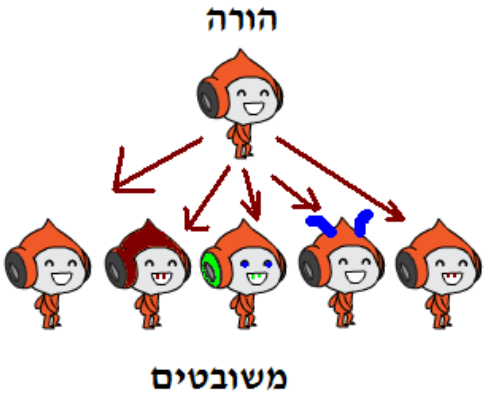
נניח שאנו כותבים משחק SPACE INVADER – פלישה מהחלל. ללא הרף יורדים אלינו עשרות חלליות או שועטים לעברנו אנדרואידים ששים אלי קרב. גדל הקושי להתמודד עם תסריטים ותלבושות שמשתנים בעשרות דמויות, העניין הופך לבעייתי ולמורכב לשליטה ועולות בעיות רבות של תכנון ויישום התכנון.

כאן בדיוק הזמן להשתמש בשיבוט של הדמות – יצירת העתק של הדמות. השיבוט יורש מדמות 'ההורה' את התסריטים (על פיהם הוא יפעל), את הצלילים, ואת מכלול התכונות של 'ההורה'.

ניתן לנסח בשפה יותר מקצועית את הרעיון של דמות והמשובטים של הדמות: הדמות מהווה במידה מסוימת עצם שיש לו תכונות ותסריטים שנכנה אותם 'התנהגות'. העצמים הם חלק מרכזי בכל יישום בשפות תכנות שנקראות שפות מונחות עצמים (OBJECT ORIENTED LANGUAGE). סביבת סקרטץ' מעט מזכירה את הרעיונות של תכנות מונחה עצמים – ישנן דמויות ולדמויות יש תכונות (מיקום, מראה וכו') והתנהגות שהם התסריטים שאנו כותבים לדמות. חשוב לציין שזה לא תכנות מונחה עצמים – סקרטץ' אינה שפה המאפשרת תכנות מונחה עצמים באותה מידה שסקרטץ' אינה מריצה תהליכים במקביל; יש מעין פסיאודו הרצה במקביל כמו פסאודו הורשה באמצעות שיבוט. זה לא "הדבר האמיתי" אבל זה כן כולל רעיונות המאפשרים הוראה ולמידה מטרימה של רעיונות אלו לקראת תכנית הלימודים בחטיבה העליונה.



הדרך בה סקרטץ' 2.0 תומכת במנגנון השיבוט נקראת "BYOB - STYLE CLONING", שהמשמעות המעשית של מנגנון זה היא שהשיבוט הוא ממש דמות חדשה ברשימת הדמויות של הפרויקט; הדמות יכולה להיות שונה בחלק מהתכונות מדמות "ההורה" אפילו במראה החיצוני; וזה מאפשר גמישות גדולה יותר.



הוראות בסיסיות – שיבוט דמות-כפיל

הוראה – 

ההוראה יוצרת כפיל של הדמות לה שייך התסריט. אם רוצים ליצור כפיל של דמות אחרת, לוחצים על החץ הקטן בהוראה ומתפריט הגלילה בוחרים את הדמות האחרת ממנה רוצים ליצור כפיל.

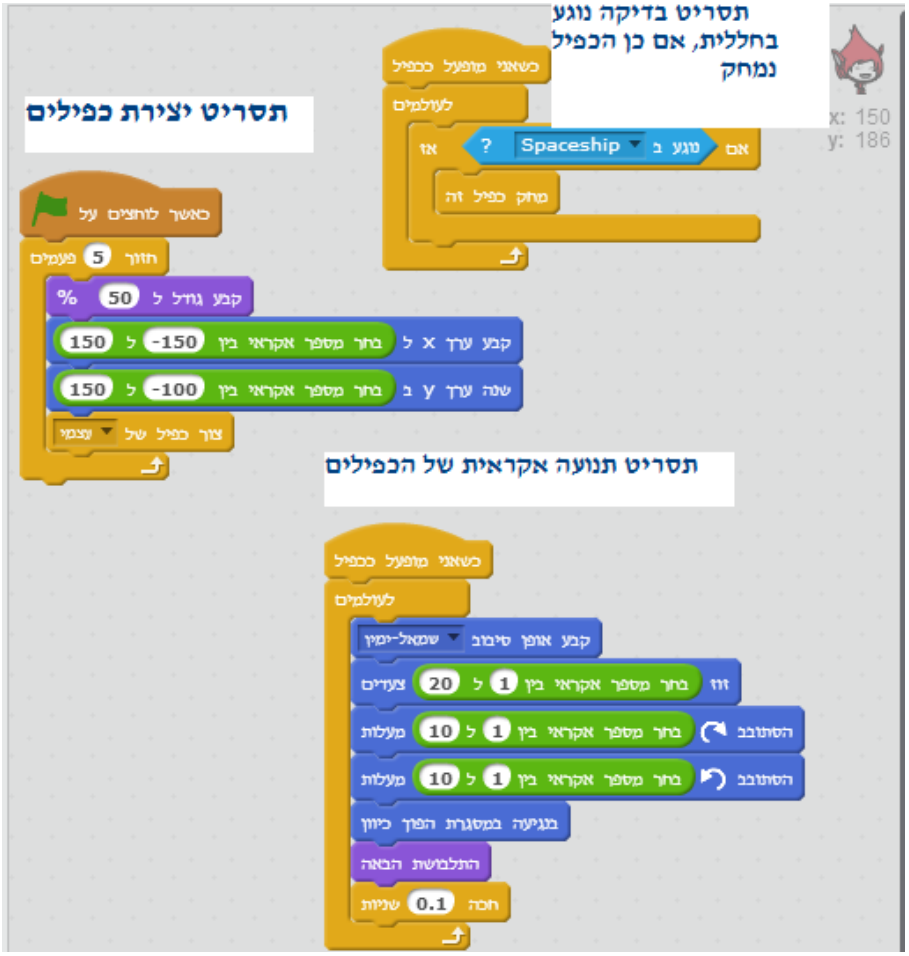
הוראה – 

הוראה זו הכלולה בתסריט ששייך לדמות ההורה מורה לכפיל לבצע את התסריט, כלומר מורה לכל הכפילים של הדמות.

הוראה – 

הוראה זו מוחקת את הכפיל. למשל אחד הכפילים נפגע מירי, כלומר הוראת החיישן 'נוגע ב' התקיימה עבורו ואנחנו רוצים למחוק אותו.

יוצגו שלושה תסריטים של הדמות GIGA WALK: תסריט שיוצר את הכפילים, תסריט שיוצר את התנועה של הכפילים על הבמה, ותסריט שבודק אם הכפיל נוגע בדמות SPACESHIP – הכפיל המסוים שנגע (ולא שאר הכפילים) נמחק.



במת המשחק בשלבים שונים ממחישה תנועה עצמאית של הכפילים והעלמות כל כפיל כאשר **'נוגע ב'** - SPACESHIP

איור: הצגת שלבים בהדמיית שימוש בכפילים



ההדמיה כללה העמדה של SPACESHIP במרכז הבמה ; הכפילים נעים אקראית (עם הדמיית הליכה) ; כל נגיעה בדמות SPACESHIP מעלימה כפיל שנגע בהוראת מחיקה.

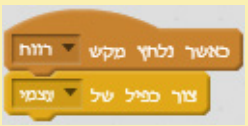
נקודות חשובות

• על הבמה יש דמות שישית : ההורה של הכפילים. תסריט התנועה לא מתייחס אליו כי הוא לא מבצע תסריט שתחילתו היא 'כשאני מופעל ככפיל'. אם רוצים שגם דמות 'הורה' תנוע צריך להעתיק את תסריט התנועה, לתת לו כותרת אירוע שונה נניח - 

ודמות ההורה תנוע כמו המשובטים. חשוב לשים לב, שאם לא ניצור עבור דמות ההורה את התסריט הבודק אם הדמות נוגעת בחללית, יתרחש מצב בו כל המשובטים ימחקו ודמות ההורה תישאר, ואז אפשר ליצור ממנה עוד משובטים.

• כל התסריטים ששייכים לדמות המקורית שייכים גם לשיבוטים של הדמות, כלומר דמות משובטת מבצעת כל תסריט ששייך לדמות המקורית וגם את התסריטים המתחילים באירוע 

לעומת זאת דמות ההורה מבצעת את כל התסריטים ששייכים לה, אבל לא מבצעת את התסריטים שמתחילים עם האירוע 

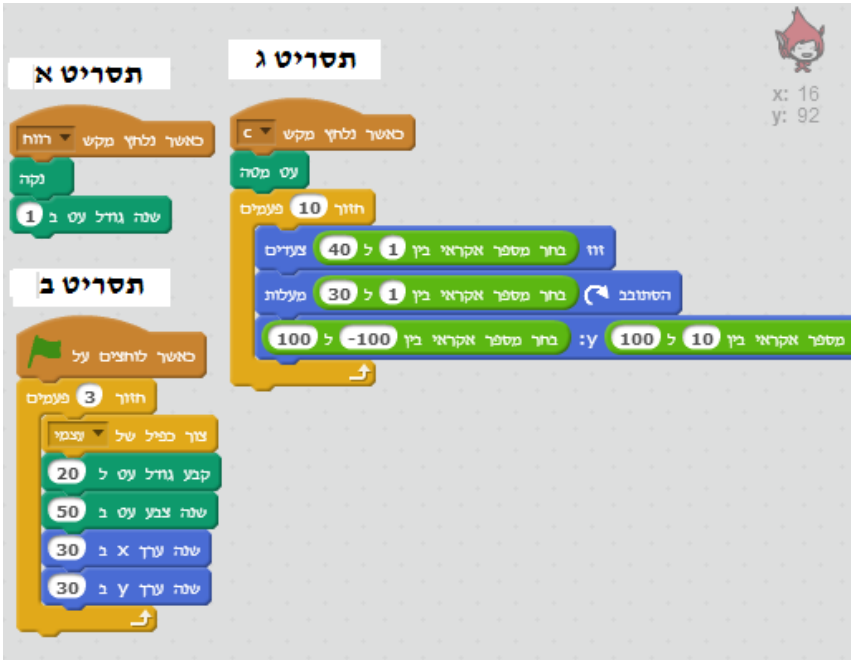


• לא מומלץ לבנות תסריט שכולל הוראת יצירת שיבוט עם האירוע . התהליך שעלול להתרחש הוא אינפלציה של שיבוטים – לחיצה ראשונה על מקש רווח יצור מהמקור שיבוט אחד, הלחיצה הבאה תיצור כבר 4, כי גם השיבוט יוצר שיבוט וגם הדמות המקורית יוצרת שיבוט, לחיצה שלישית 8 שיבוטים, לחיצה רביעית כבר 32 שיבוטים, זה גדל בפונקציה של 2^n נסה לשלב את התסריט הקצר ולחץ על מקש רווח מספר פעמים. לכן מומלץ לחבר פקודת יצירת שיבוטים לאירוע 

• בהקדמה ציינתי שהשיבוט יורש את כל הנכסים של ההורה: התלבושות, המיקום על הבמה, צבע העט, גודל העט, מצב הסתר או הצג של הדמות, האפקטים הגרפים ועוד, כך שאם אנחנו יוצרים שיבוטים בנקודות זמן שונות בפרויקט, כל שיבוט יהיה שונה ויהיה תלוי בתכונות ומצב ההורה ברגע השיבוט.

שלושת התסריטים הללו ממחישים את הרעיון שתכונות השיבוט נקבעות לפי ערכי התכונות של הדמות ברגע השיבוט.

- תסריט א – פועל כאשר נלחץ מקש רווח – מבצע ניקוי מסך.
- תסריט ב – פועל כאשר לוחצים על דגל ירוק, יוצר 3 שיבוטים בלולאה כשבכל מחזור שלה משתנה תכונת צבע העט.
- תסריט ג – פועל כאשר לוחצים על מקש c ושלושת השיבוטים והדמות המקורית מבצעים אותו, ולשני השיבוטים שנוצרו במחזור השני והשלישי של הלולאה יש צבע עט שונה.



הבמה עם השיבוטים והצבעים השונים של העט



הורשה של משתנים

התסריטים של נושא זה נמצאים ביישום: תכונות של משובטים.

כל שיבוט יורש את המשתנים של ההורה אולם הערכים של משתנים אלו יכולים להיות שונים עבור כל שיבוט.

בתסריט זה ניצור רשימה עם שלוש מחרוזות. כל מחרוזת תייצג שם של חתול.

לדמות החתול ניצור משתנה מקומי בשם NAME שיכיל את שם החתול.

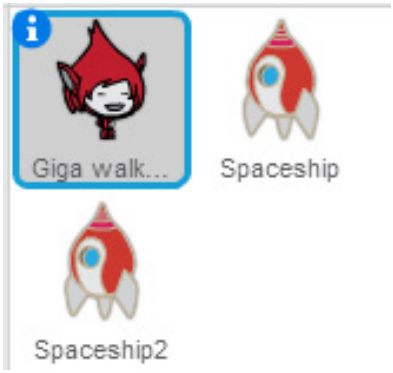
בהמשך התסריט ניצור שלושה שיבוטים של דמות החתול שכוללת תכונה: שם החתול, ונציג איך בפועל אצל כל שיבוט שיורש את התכונה NAME, הערך של התכונה, כלומר שם החתול המשובט, יהיה שונה.

מנגנון זה של ירושה של משתנה שבפועל הערכים שלו יהיו שונים עבור כל שיבוט, מאפשר הבחנה בין השיבוטים ומתן מטלה מסוימת לשיבוט מסוים אותה לא יעשו השיבוטים האחרים.

העתקת דמות מול שיבוט דמות



חשוב לא להתבלבל! אנו יכולים ליצור העתקים של דמות על ידי בחירת דמות ולחיצה על הכפתור הימני בעכבר לבחירת האפשרות **שכפל**.



כאשר אנו משכפלים דמות מדובר בדמות עצמאית לחלוטין. היא אינה קשורה לדמות ממנה הועתקה ונקראת בשם שונה (שניתן כמובן לשנות). ברירת המחדל של סביבת סקרטץ' היא לתת את אותו שם של הדמות המקורית עם תוספת מספור.

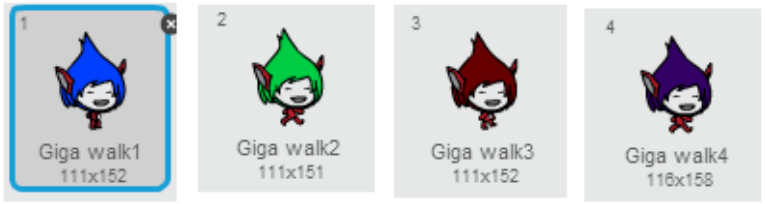
אנחנו רואים כי בהעתקה נוצרו שתי דמויות של חללית, אולם כל אחת היא דמות עצמאית הנקראת בשם שונה ואין לשתי הדמויות שום דבר משותף מלבד הזהות הגרפית, לפחות לאחר העתקה שגם אותה ניתן לשנות. חשוב לציין שכאשר משכפלים דמות שיש לה תסריטים, משוכפלים גם התסריטים מהדמות המקורית לדמות המשוכפלת, וכמובן שניתן לשנות תסריטים לאחר השכפול בכל אחת מהדמויות המשוכפלות.

דוגמה נוספת הממחישה את הרעיון ששיבוט יורש את התכונות של דמות ההורה וכך אם התכונות ישתנו יהיו לנו שיבוטים שונים.

א. בוצע שינוי בארבע תלבושות הדמות GIGA WALKING, השינוי הוא בצבעי הדמות.

ב. נכתב תסריט שמחליף תלבושות של דמות המקור לאחר כל הוראה של יצירת שיבוט. כך מתקבלים ארבעה שיבוטים עם מראה שונה.

ארבע התלבושות של הדמות



התסריט ליצירת 4 שיבוטים. בכל מחזור בלולאה משתנה התלבושת הנוכחית של דמות המקור ונוצרים ארבעה שיבוטים שונים במראה. בשילוב התסריט המשנה את צבע העט שהוצג באיור קודם, מקבלים ארבעה שיבוטים עם צבע עט שונה ומראה שונה והתוצאה של הפעלת התסריט **'כאשר נלחץ מקש** C**'** מוצגת באיור הבא:



שער רביעי

נספחים

נספח א: סיכום מקוצר של הוראות שפת סקרטץ'

נספח ב: סיכום מורחב של הוראות שפת סקרטץ'

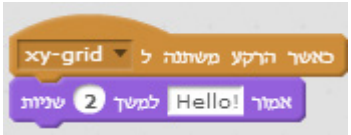
נספח ג: סיכום סביבת התכנות של שפת סקרטץ'

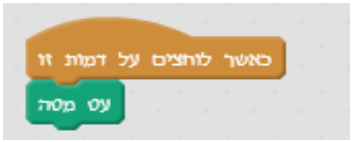
נספח ד: סביבת התכנות – תפריטים, תגיות, כלים

נספח ה: ידע במתמטיקה תלמידי כיתות ד', ה'

נספח א' סיכום מקוצר של הוראות שפת סקרטץ'

קבוצת - אירועים 

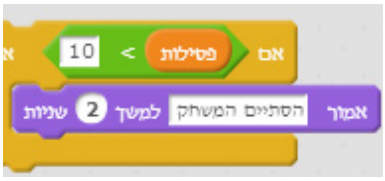
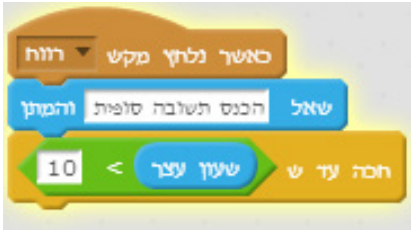
הוראה	משמעות	דוגמה
	התסריט יתחיל עם לחיצה עם העכבר על הדגל הירוק	
	התסריט יתחיל עם הקשה על מקש כלשהו לפי בחירה	
	התסריט יתחיל עם שינוי ברקע הבמה לפי הבחירה	

הוראה	משמעות	דוגמה
	התסריט יתחיל כאשר התנאי מתקיים (יש כמה חלופות)	
	התסריט יתחיל כאשר לוחצים על הדמות עם סמן העכבר	
	התסריט יתחיל עם קבלת מסר מסוים	
	על הדמות לשדר מסר. או מסר שכבר קיים או מסר חדש שנוצר (יש לבחור מסר או לכתוב חדש)	
	הדמות משדרת מסר והתסריט נכנס למצב המתנה	

קבוצת – בקרה

בקרה

	התסריט ממתין פרק זמן בשניות **דוגמה:** לולאה המציירת ריבוע, בתום ציור של צלע יש המתנה של שנייה.	
	**הגבלה מראש:** סדרת ההוראות תתבצע בדיוק כ- **מספר הפעמים** המבוקש	

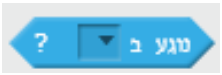
הוראה	משמעות	דוגמה
	**ללא הגבלה:** ביצוע **אינסופי** של סדרת הוראות דוגמה: הדמות תנוע במעגל לעולמים	
	**הגבלה מותנית:** סדרת ההוראות תתבצע כל עוד התנאי **מתקיים** דוגמה: הדמות תנוע ימינה כל עוד **אינה** נוגעת במסגרת	
	עצירת כל התסריטים, או לפי בחירה	
	משפט תנאי מצומצם. **אם** התנאי מתקיים **אז** מתבצעת קבוצת ההוראות בתוך מבנה הקשתי בבלוק. דוגמה: **אם** התנאי מתקיים תוצג הודעה	
	משפט תנאי מורחב: **אם** התנאי מתקיים תתבצע קבוצת ההוראות הראשונה **אם** אחרת קבוצת ההוראות השנייה	
	התסריט ממתין עד שיתקיים התנאי דוגמה: התסריט ממתין 10 שניות לתשובת המשתמש	
	התסריט מבוצע רק על ידי כפילים (שיבוטים) של דמות.	הרחבה ודוגמאות בפרוייקטי העשרה

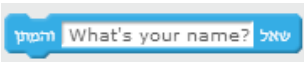
• שימו לב: לולאת 'לעולמים' היא תמיד הוראה אחרונה בתסריט, ולא ניתן לשלב אחריה הוראה נוספת!

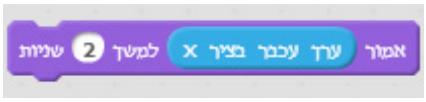
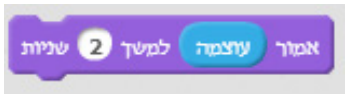
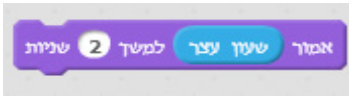
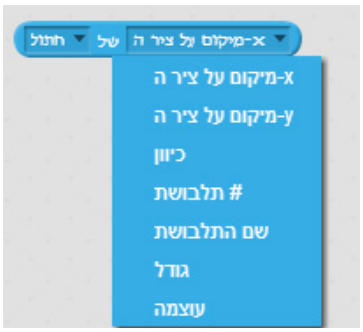
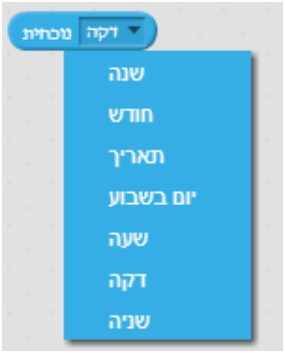
הוראה	משמעות	דוגמה
	יצירת כפיל (שיבוט) של הדמות לה שייך התסריט, או של כל דמות אחרת בפרויקט לפי בחירה.	הרחבה ודוגמאות בפרוייקטי העשרה
	מחיקת כפיל (שיבוט).	הרחבה ודוגמאות בפרוייקטי העשרה

קבוצת – חיישנים



חיישן	משמעות	דוגמה
	חיישן בדיקה – האם הדמות נוגעת בקצה – סמן העכבר או דמות מסוימת. כשמתקיים התנאי החיישן מחזיר **'אמת'**; אחרת החיישן מחזיר **'שקר'**. הכלב שלנו מסתובב ונע, אם הוא נוגע בחתול הוא אומר "תפסתי אותך"	
	חיישן בדיקה האם הדמות נוגעת בצבע מסוים. בפרויקט בו נוצרו ארבעה עטלפים שחורים, החתול עוקב אחרי סמן העכבר המוביל אותו לעטלפים לפי צבע העטלף. כך נוכל להתייחס לכל העטלפים.	

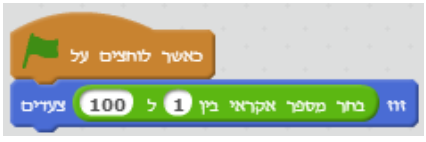
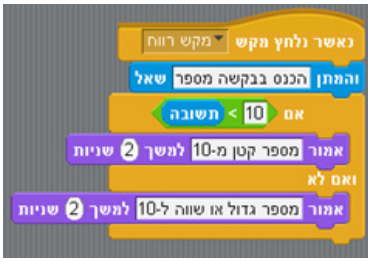
חיישן	משמעות	דוגמה
	חיישן הבודק האם צבע מסוים נוגע בצבע אחר. צבעתי את ידיו ורגליו של החתול באדום, ועתה ברצוננו לבדוק מתי החתול תופס עטלף בידיו וברגליו:	  חתול  Bat2
	חיישן המחשב את המרחק בין הדמות לסמן העכבר או לאחת הדמויות.	
	חיישן הקולט קלט של המשתמש (מספר או מחרוזת) ומאחסן את הערך הנקלט במשתנה המערכת '**תשובה**'.	
	משתנה המערכת המקבל את הקלט של המשתמש.	
	בדיקה האם נלחץ מקש. **חשוב:** הבדיקה מתבצעת ללא המתנה, אם משלבים את התנאי בלולאה הבדיקה מתבצעת במחזוריות. דוגמה: הנעת החתול לצד ימין בכל פעם שנלחץ מקש חץ ימני.	

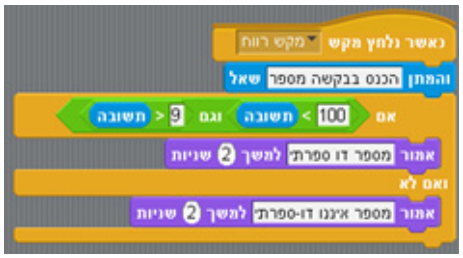
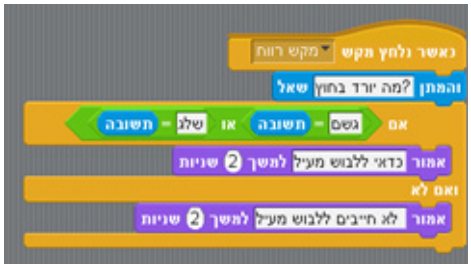
חיישן	משמעות	דוגמה
	בדיקה האם נלחץ העכבר דוגמה: בכל פעם שנלחץ העכבר מוחתמת הדמות על הבמה.	
	חיישן שמחזיר לנו את ערך X של מיקום העכבר.	
	חיישן שמחזיר לנו את ערך Y של מיקום העכבר.	
	חיישן שמחזיר את עוצמת המיקרופון.	
	משתנה מערכת שמאחסן את ערך שעון העצר.	
	איפוס שעון העצר.	
	קבלת נתונים שונים על הדמויות בפרויקט.	
	חיישן שמחזיר מידע מפורט על התאריך והזמן של שעון המערכת.	

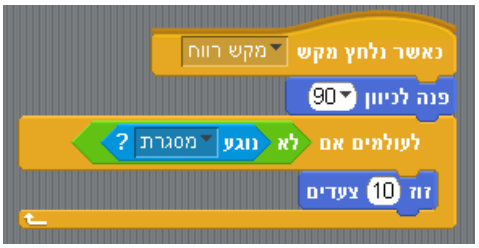
חיישן	משמעות	דוגמה
	חיישן שמחזיר את מספר הימים שחלפו מאז 1/1/2000	
	חיישן שמחזיר את שם המשתמש- בעל החשבון באתר סקרטץ'.	

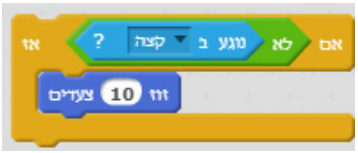
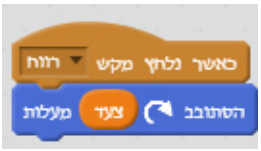
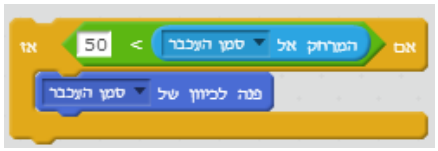
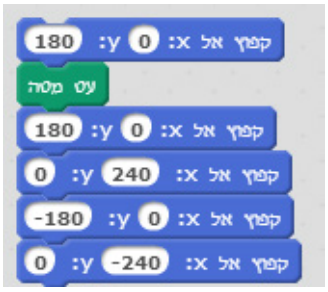
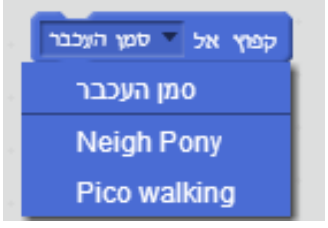
קבוצת – מפעילים

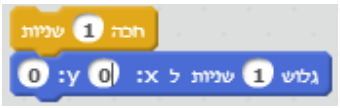
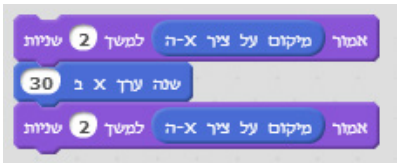
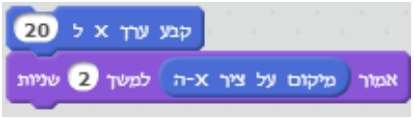
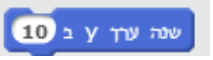
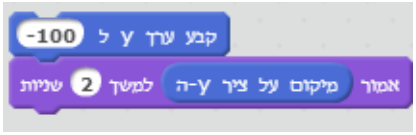
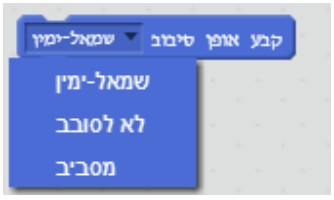
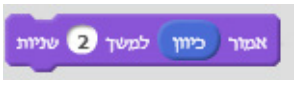


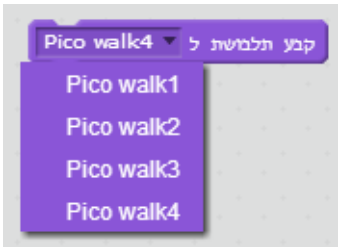
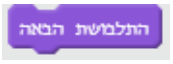
מפעיל	משמעות	דוגמה
	ביטוי המחבר את הערך באגף שמאל עם הערך באגף ימין.	
	ביטוי המחסר את הערך באגף ימין מהערך באגף שמאל.	
	ביטוי המכפיל את הערך באגף שמאל בערך באגף ימין.	
	ביטוי המחלק את הערך באגף שמאל בערך באגף ימין.	
	יצירת מספר אקראי בתחום המצוין בהוראה.	
	תנאי המתקיים רק אם אגף שמאל קטן מאגף ימין. דוגמה: ייבדק המספר שהוכנס לתוך משתנה 'תשובה'. • אם קטן מ-10: הדמות תאמר "מספר קטן מ-10" • אחרת: הדמות תאמר "מספר גדול או שווה ל-10"	

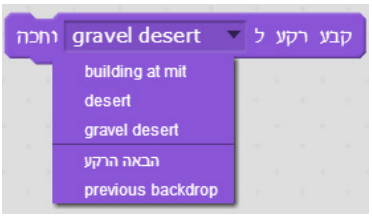
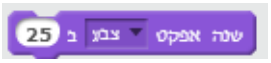
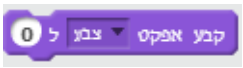
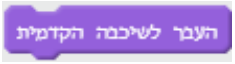
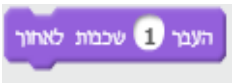
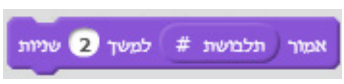
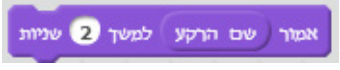
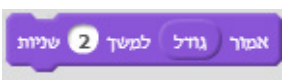
מפעיל	משמעות	דוגמה
	**תנאי שוויון כללי:** מתקיים רק אם שני האגפים שווים. ייבדק המספר שהוכנס לתוך משתנה 'תשובה': • אם הוכנס 2: הדמות תאמר "המספר הוא 2". • אחרת: הדמות תאמר "המספר איננו 2".	
	תנאי שמתקיים רק אם אגף שמאל **גדול** מאגף ימין.	
	התנאי כולו מתקיים **רק אם שני התנאים** הפנימיים מתקיימים גם יחד. ייבדק המספר שהוכנס לתוך משתנה 'תשובה': • אם קטן מ-100 וגם גדול מ-9: הדמות תאמר "המספר דו-ספרתי". • אחרת: הדמות תאמר "המספר איננו דו-ספרתי".	
	התנאי כולו מתקיים אם לפחות אחד מהתנאים הפנימיים מתקיים. תיבדק התשובה לשאלה "מה יורד בחוץ?": • אם התשובה היא 'גשם' או 'שלג': הדמות תאמר "כדאי ללבוש מעיל". • אחרת: הדמות תאמר "לא חייבים ללבוש מעיל".	

מפעיל	משמעות	דוגמה
	התנאי כולו מתקיים רק אם התנאי הפנימי **אינו** מתקיים. דוגמה: הדמות תוסיף להתקדם כל עוד **אינה** נוגעת במסגרת.	
	מצמיד שתי מחרוזות האחת לשנייה דוגמה: החתול יבקש שם, ואז יאמר "שלום <שם>, מה שלומך?". **שימו לב!** בפקודה  החתול יאמר "23" ולא "5"!	
	החזרת תו במיקום מסוים (לפי ערך) של מחרוזת-מילה.	
	החזרת אורך מחרוזת-מילה.	
	שארית החלוקה השלמה של שני מספרים. דוגמה: החתול יבקש מספר ויאמר לנו האם הוא זוגי או אי-זוגי	
	עיגול ערך מספרי.	
	קבוצת פונקציות מתמטיות.	

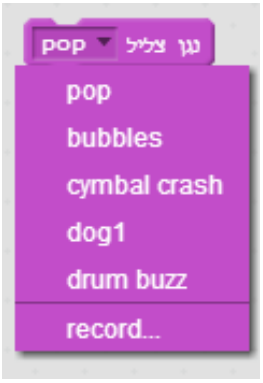
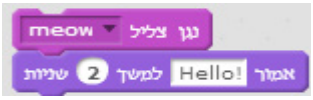
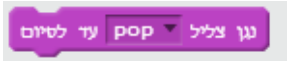
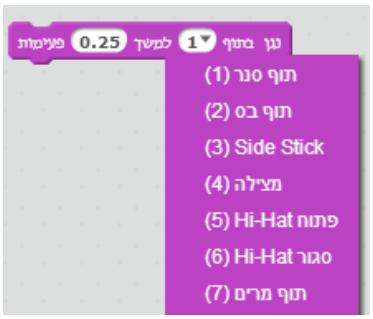
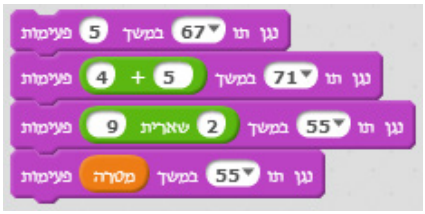
הוראה	משמעות	דוגמה
	הדמות נעה למרחק לפי הערך בהוראה, בכיוון הנוכחי של הדמות.	
	הדמות מבצעת סיבוב ימינה יחסי, לפי הערך.	
	הדמות מבצעת סיבוב שמאלה יחסי, לפי הערך.	
	הדמות משנה את כיוונה לפי אחד מ-4 הערכים של כיווני הבמה.	
	מפנה את הדמות לכיוון סמן העכבר או לכיוון אחת הדמויות.	
	משנה מיקום למיקום על הבמה לפי הערכים בהוראה.	
	קופץ אל המיקום של סמן העכבר או למיקום של אחת הדמויות.	

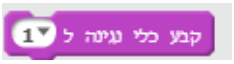
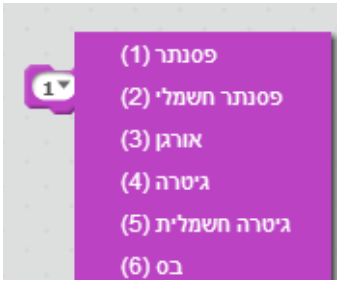
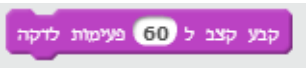
הוראה	משמעות	דוגמה
	הדמות נעה (גולשת) בקצב הנקבע בהוראה, לעבר מיקום בבמה לפי הערכים בהוראה.	
	משנה את הערך X של הדמות לפי הערך הנוכחי בתוספת הערך בהוראה.	
	קובע את ערך X של הדמות לפי הערך בהוראה.	
	משנה את הערך Y של הדמות לפי הערך הנוכחי בתוספת הערך בהוראה.	
	קובע את ערך Y של הדמות לפי הערך בהוראה.	
	אם הדמות נוגעת במסגרת-קצה, כיוון הדמות מתהפך.	
	קובע את כיוון הסיבוב של הדמות כאשר הדמות משנה כיוונים.	
	מחזיר את מיקום X של הדמות.	
	מחזיר את מיקום Y של הדמות.	
	מחזיר את ערך הכיוון של הדמות.	

הוראה	משמעות	דוגמה
	מציג הודעה לפי בחירה למשך פרק זמן מסוים.	
	מציג הודעה לפי בחירה למשך פרק זמן קצר ביותר.	
	מציג הודעה לפי בחירה למשך פרק זמן מסוים, תיבת הדיאלוג מעל הדמות כמו בתהליך 'חשיבה'	
	מציג הודעה לפי בחירה, למשך זמן קצר מאוד.	
	מעביר את הדמות למצב **הצג**, כלומר רואים את הדמות על הבמה.	
	מעביר את הדמות למצב **הסתר**, כלומר **לא** רואים את הדמות על הבמה.	
	קובע תלבושת של הדמות לפי הערך שבהוראה.	
	משנה את הדמות לתלבושת הבאה (המעבר בין התלבושות מעגלי).	

הוראה	משמעות	דוגמה
	קובע את הרקע לפי הערך שבהוראה.	
	משנה את אפקט הצבע לפי הערך שבהוראה.	
	קובע את ערך אפקט הצבע לפי הערך בהוראה, ללא קשר לערך הנוכחי.	
	מבצע ניקוי אפקטים, הדמות חוזרת לתצוגה רגילה.	
	משנה את הגודל **הקיים** של הדמות בערך שבהוראה.	
	קובע את אחוז גודל הדמות, 100% זה גודל ברירת המחדל.	
	מעביר את הדמות לחזית הבמה לשכבה הקדמית.	
	מעביר את הדמות לשכבה מסוימת לאחור, לפי הערך שבהוראה.	
	מחזיר את שם התלבושת הנוכחית.	
	מחזיר את שם הרקע הנוכחי.	
	מחזיר את גודל הדמות.	

קבוצת – צליל

הוראה	משמעות	דוגמה
	מנגן צליל לפי הבחירה בהוראה מתוך **קבוצת** הצלילים של הדמות. **חשוב!** הצליל יכול להימשך וההוראה הבאה בתסריט תחל להתבצע. אם נשתמש בהוראה ולאחר מכן בהוראת **'אמור'**, הצליל ימשך והדמות תאמר את מה שיש בהוראה **'אמור'**. הצליל נמשך וההוראה הבאה מתבצעת לפי תום הצליל.	 
	מנגן צליל מקבוצת הצלילים של הדמות, עד תום הצליל לא יהיה מעבר להוראה הבאה. דוגמה: השמעת צליל לאורך כל זמן ריצת הפרויקט.	
	מנגן באחד מכלי הנגינה לפי בחירה לפרק זמן מסוים לפי בחירה.	
	השהייה בנגינה לפרק זמן לפי הערך שבהוראה.	
	מנגן תו מסולם התווים המוסיקליים לפי בחירה ולפרק זמן לפי בחירה.	

הוראה	משמעות	דוגמה
	בוחר את כלי הנגינה הפעיל מתוך אוסף כלי הנגינה.	
	משנה את עוצמת הקול לפי הערך הנבחר.	
	קובע את עוצמת הקול באחוזים, 100% זו ברירת המחדל.	
	מחזיר את עוצמת הקול.	
	משנה את קצב הנגינה לפי הערך שנבחר, בתוספת לערך הנוכחי.	
	קובע את קצב הפעימות ללא קשר למצב הנוכחי.	
	מחזיר את קצב הפעימות הנוכחי.	

קבוצת – עט

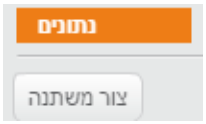
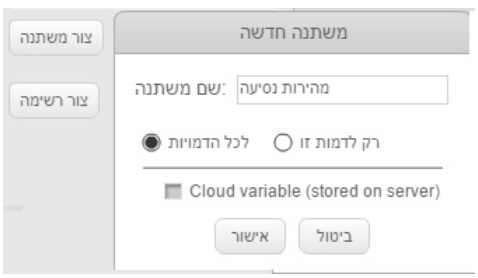


הוראה	משמעות	דוגמה
	ניקוי הבמה מכל הגרפיקה שנוצרה על ידי הדמויות (הדמויות לא נעלמות).	
	מחתים את הדמות על הבמה במיקומה הנוכחי.	
	קובע את העט למצב מטה, כעת כל תנועה של הדמות תשאיר קו.	
	קובע את העט למצב מעלה, כעת כל תנועה של הדמות לא תשאיר קו .	

הוראה	משמעות	דוגמה
	קובע את צבע העט ללא קשר למצב הנוכחי.	
	משנה את צבע העט בגודל מסוים בהשוואה למצב הנוכחי.	
	קובע את צבע העט.	
	קובע את גודל העט בערך מסוים ללא קשר לגודל הנוכחי.	
	משנה את גודל העט בערך מסוים שמתווסף לגודל הנוכחי.	
	משנה את הצללת העט ללא קשר לגודל הנוכחי.	
	משנה את הצללת העט ביחס למצב הנוכחי.	

קבוצת – נתונים



הוראה	משמעות	דוגמה
	**יצירת** משתנה **חדש**	יצירת משתנה חדש בשם 'מהירות הנסיעה': 

הוראה	משמעות	דוגמה
	**קביעת ערך המשתנה .** אם היה ערך קודם הוא "נמחק".	
	**שינוי תוכן המשתנה:** הערך נקבע לפי ערך קיים והעידכון	
	**שימוש בערך** של המשתנה בביטוים שונים	
	**יצירת** רשימה חדשה בעלת שם שנקבע	
	**הוספת** ערך לרשימה.	
	מחיקת פריט מהרשימה	
	**מספר** הערכים ברשימה. **גישה** לפריט האחרון ברשימה, גישה לפי ערך מיקום	

קבוצות של בלוקים – הוראות הסבר מפורטות



הבלוקים מחולקים ל- 10 קבוצות וכל בלוק מותאם בצבעו לקבוצה לה הוא שייך.

להלן הקבוצות וצבען:

- **אירועים:** בקבוצה זו נמצאים כל הבלוקים שמהווים התחלת ביצוע תסריט.

לדוגמה: כאשר לוחצים על דגל ירוק. הוראות קבוצה זו הן בצבע חום.

- **בקרה:** בקבוצה זו נמצאים כל הבלוקים האחראים לבקרה על ביצוע הוראות.

לדוגמה: 'חכה __ שניות' (הקו התחתי מייצג ערך מספרי או מילולי שכותבים בתוך המקום). הוראות קבוצה זו הן בצבע חום

- **תנועה:** בקבוצה זו נמצאים כל הבלוקים הקשורים לנקודה בה נמצאת הדמות והכיוון אליו היא פונה.

לדוגמה: 'הסתובב __ מעלות נגד כיוון השעון'. הוראות קבוצה זו הן בצבע כחול.

- **חיישנים:** בקבוצה זו נמצאים כל הבלוקים הקשורים לבדיקות של מצבים שונים, הבלוקים הקשורים למגע ולמרחק בין הדמות לדבר מה. לדוגמה: 'המרחק אל __' , הוראות קבוצה זו הן בצבע תכלת.

- **מראה:** בקבוצה זו נמצאים כל הבלוקים הקשורים למראה הדמות, למה שהדמות "חושבת" או "אומרת" ולגודל.

לדוגמה: 'אמור __ למשך __ שניות'. הוראות קבוצה זו הן בצבע סגול.

- **מפעילים:** בקבוצה זו נמצאים כל הבלוקים הקשורים לערכים מספריים שאינם קשורים לקבוצה אחרת.

לדוגמה: '__ + __' (בתור ערך מספרי). הוראות קבוצה זו הן בצבע ירוק בהיר.

- **צליל:** בקבוצה זו נמצאים כל הבלוקים הקשורים לצלילים המופקים עקב פעולה כלשהי של הדמות.

לדוגמה: 'הפסק את כל הצלילים'. הוראות קבוצה זו הן בצבע ורוד.

- **נתונים:** בקבוצה זו נמצאים כל הבלוקים הקשורים למשתנים או לרשימות המשתתפים בתכנית.

לדוגמה: 'קבע __ ל-__' (במקרה זה הקו התחתי הראשון מייצג שם של משתנה). הוראות קבוצה זו הן בצבע כתום.

- **עט:** בקבוצה זו נמצאים כל הבלוקים הקשורים למה שמציירת הדמות.

לדוגמה: 'קבע צבע עט __' (במקרה זה הקו התחתי מציין צבע שבוחרים מתוך מגוון אפשרויות של צבעים). הוראות קבוצה זו הן בצבע ירוק כהה.

- **לבנים נוספות:** בקבוצה זו יש הוראות שהגדיר המשתמש. הקבוצה מאפשרת ליצור הוראה של המשתמש שתכלול כמה הוראות מקבוצות אחרות.

קבוצת ההוראות אירועים



ההוראה – 

תסריט שמתחיל עם הוראת בקרה זו יבוצע לאחר לחיצה על הדגל הירוק. בדרך כלל נעשה שימוש בהוראה הזו לצורך אתחול, כלומר קביעת ערכים ומצבים שונים בתחילת משחק או כל יישום אחר. לעיתים האתחול הוא עבור חלק מהדמויות ולעיתים הוא עבור כולן.

ההוראה – 

התסריט שמתחיל בהוראה זו יופעל כאשר לוחצים עם העכבר על הדמות. לדוגמה: נניח שאנו מציגים משחק עם ארבע דמויות שאחת מהם שונה, המשחק בוחר את הדמות היוצאת דופן באמצעות העכבר והתסריט יופעל.

ההוראה – 

התסריט מתחיל כאשר יש מעבר לרקע שונה. לדוגמה: נניח שאנו עוברים לרקע חדש שמשמעו שינוי ברמת המשחק, בדרגות הקושי ובעוד מאפיינים, הוראת אירוע זו תכלול את התסריט עם כל ההוראות של שינוי רמת המשחק.

ההוראה – 

העברת מסרים היא אחד הכלים החשובים בדיאלוג בין דמויות; כלי מרכזי בשיתוף מידע בין דמויות שונות; והמסר מאפשר להפעיל תסריט של דמות אחרת באמצעות העברת המסר. ההוראה הזו תתחיל תסריט אם יתקבל מסר מסוים שנבחר – זוהי הוראת ההפעלה של התסריט עם קבלת המסר.

ההוראה – 

הוראה זו מתחילה תסריט כאשר נלחץ מקש כלשהו, לדוגמה מקש רווח, שמשמש במשחקים רבים כבחירה למקש "ירי". כמו כן השימוש בהוראה זו נפוץ להתחלת תסריט בהמתנה למקשי חיצים, למקש ירי וכדומה.

ההוראה – 

ההוראה מעבירה הודעה לדמויות אחרות ביישום. ההוראה הזו ביחד עם ההוראה 'כאשר מתקבל מסר' יוצרות את המנגנון המרכזי לדיאלוג בין דמויות ביישום ובהפעלת תסריטים באמצעות מסרים.

ההוראה – 

ההוראה דומה להוראה 'שדר מסר', אלא שבהוראה הזו התסריט ממתין ולא ממשיך את הביצוע עד תום ביצוע התסריט שמקבל מסר.

ההוראה – 

ההוראה מאפשרת ביצוע של תסריט כששלושה סוגי אירועים מתרחשים וקשורים ל:



- וידאו
- שעון העצר
- עוצמת הקול

ניתן לבחור את אחת האפשרויות הללו, ולבחור את הערך שברצוננו לבחון האם הייתה מערך זה חריגה.

קבוצת ההוראות בקרה



ההוראה –



הוראה זו יוצרת השהייה של שניות בביצוע התסריט, לפי ערך שאנחנו קובעים (מספר קבוע או ערך של ביטוי)

ההוראה –



לולאה ליישום מבנה של לולאה, מספר הפעמים נקבע לפי הערך שאנו קובעים (מספר קבוע או ערך של ביטוי).

ההוראה –

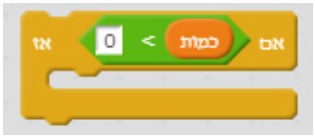


מבנה לולאה לביצוע אינסופי של קטע של תסריט.

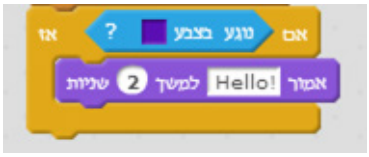
ההוראה –



מבנה למימוש תנאי בדיקה. לשם כתיבת התנאי משתמשים ב'מפעילים' כמו בדוגמה הבאה:



בתנאי הבדיקה ניתן להשתמש גם בחיישנים כמו בדוגמה הבאה:



בתוך מבנה ההוראה אנחנו מכניסים את ההוראות שיתבצעו אם ערך התנאי 'אמת'

ההוראה –



מבנה תנאי עם שני המצבים לביצוע:

מצב א': התנאי '**אמת**' ואז מבוצעות ההוראות לאחר המילה '**אז**'

מצב ב': התנאי '**שקר**' ואז מבוצעות ההוראות לאחר המילה '**ואם לא**'



ההוראה –



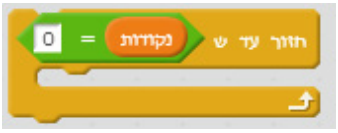
זוהי הוראת המתנה מותנת בדומה להוראה 'חכה', אולם ההמתנה בהוראה כאן אינה מוגבלת בזמן ותיפסק כאשר תנאי יתקיים. לדוגמה:



ההוראה –



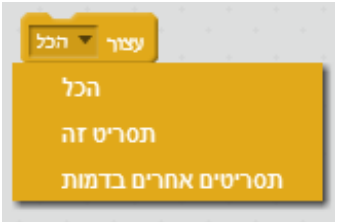
לולאה ליישום מבנה של לולאה (חזרה על תסריט מספר פעמים), אולם בניגוד להוראה 'חזור', בה נקבע מספר הפעמים, בהוראה 'חזור עד ש' הלולאה תתבצע עד שיתקיים תנאי כלשהו.



ההוראה –



הוראה זו נועדה לעצור תסריט או תסריטים בצורה "ישירה", כלומר לא כתוצאה מסיום "טבעי" של תהליך. להוראה מספר אפשרויות:



ההוראות ליצירת ולשימוש ב'כפילים'



רעיון הכפילים CLONE קיים רק בגרסת 2.0 של סקרטץ'. המנגנון הזה מאפשר ליצור העתקים מדמות כשכל העתקים מבצעים את אותו תסריט. עקב חשיבות הרעיון והפוטנציאל בשימוש וביישום שלו ביישומים, יוקדש פרק מיוחד שיכלול אפשרויות והוראות חדשות שמאפיינות את גרסת סקרטץ' 2.0.

ההוראה – 

הדמות נעה מספר פיקסלים לפי הערך הנבחר (מספר קבוע או ביטוי חשבוני), בכיוון הנוכחי של הדמות.

ההוראה – 

דמות מבצעת סיבוב לפי הערך הנבחר (מספר קבוע או ביטוי חשבוני). הפנייה היא פנייה יחסית. לדוגמה: אם נניח שהדמות נמצאת בכיוון 90, לאחר הפעלת 'הסתובב 30' הדמות תהיה בכיוון 60.

דוגמה נוספת:

לפני



ההוראה - 

אחרי



הדמות עתה נמצאת בכיוון 150 (90+60)

ולזה אנו קוראים 'סיבוב יחסי'

ההוראה – 

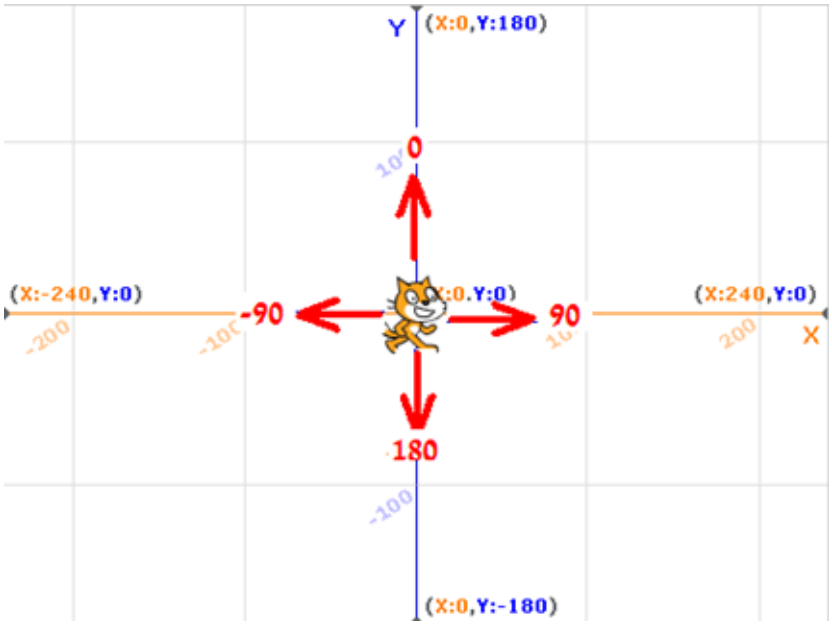
פועלת כמו הוראה הסתובב ימינה, כשהסיבוב היחסי בהוראה זו הוא שמאלה.

ההוראה – 

הסיבוב בהוראה זו הוא מוחלט, כלומר אינו תלוי בכיוון הנוכחי של הדמות.

האיור הבא מתאר את הכיוונים המוחלטים של הבמה.

האפשרויות של הוראה זו:

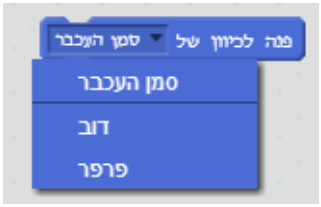


ההוראה –



ההוראה מאפשרת הפניית כיוון הדמות לכיוון של אחת הדמויות בפרויקט או לכיוון סמן העכבר.

האפשרויות עבור פרויקט עם הדמויות חתול; פרפר; דוב: עבור כל דמות ניתן להפנות את הדמות לכיוון של אחת משתי הדמויות האחרות או לכיוון העכבר.

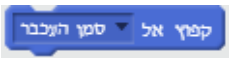


ההוראה –

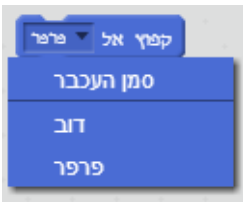


ניתן להזיז כל דמות לכל מיקום על הבמה. ההוראה משנה את מיקום הדמות מהמיקום בו היא ממוקמת אל המיקום שנבחר בהוראה. חשוב לציין שאם מצב העט הוא **למטה**, הקפיצה תיצור קו בתנועת הדמות.

ההוראה –



מזיזה את הדמות למיקום העכבר או למיקום של אחת הדמויות בפרויקט. בפרויקט בו יש את דמות החתול, הדוב והפרפר אפשרויות ההוראה עבור החתול יהיו:



שימו לב, הדמות קופצת בדיוק למיקום הדמות שנבחרה. הדמות תהיה בשכבה מתחת לדמות שנבחרה (על שכבות יוסבר בהמשך), במידה והדמות קטנה משמעותית מהדמות אליה "קפצה" יתכן שהיא תוסתר.

דוגמה:

עבור התסריט:





נקבל את דמות החתול "מסתתרת" מאחורי דמות הדוב.

ההוראה –



הדמות גולשת מהמיקום בו היא ממוקמת למיקום המצוין בהוראה. בהוראה ניתן לקבוע את מיקום ה-X ואת מיקום ה-Y, אולם ניתן גם לקבוע כמה זמן תימשך הגלישה. חשוב לציין שאם מצב העט הוא **למטה** הגלישה תיצור קו בתנועת הדמות.

ההוראה –



לכל דמות יש ערכי X ו-Y של מיקום הדמות על במה. ההוראה **שנה ערך X** , משנה את מיקום הדמות על ידי הוספת 10 פיקסלים למיקום ה-X של הדמות, כלומר הדמות תנוע 10 פיקסלים לצד ימין של במה.

ההוראה – 

ההוראה קובעת את ערך ה-X של הדמות באופן מוחלט ללא קשר למיקום הדמות. מיקום ה-Y של הדמות לא ישתנה, והשינוי יהיה במיקום ה-X.

ההוראה – 

ההוראה קובעת את ערך ה-Y של הדמות באופן מוחלט ללא קשר למיקום הדמות. מיקום ה-X של הדמות לא ישתנה, והשינוי יהיה במיקום ה-Y.

ההוראה – 

הוראה זו משנה את מיקום הדמות על ידי שינוי בערך X של מיקום הדמות, אבל בשונה מההוראה **קבע ערך X**, השינוי במיקום של הדמות הוא יחסית למיקום לפני ביצוע ההוראה. הערך 10 בהוראה המופיע בדוגמה יחובר לערך של X במיקום הדמות וזה יהיה X המיקום החדש של הדמות.

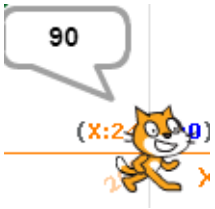
ההוראה – 

הוראה זו משנה את מיקום הדמות על ידי שינוי בערך Y של מיקום הדמות, אבל בשונה מההוראה קבע ערך Y, השינוי במיקום של הדמות הוא יחסית למיקום לפני ביצוע ההוראה. הערך 10 בהוראה המופיע בדוגמה יחובר לערך של Y במיקום הדמות וזה יהיה Y המיקום החדש של הדמות.

ההוראה – 

הוראה זו משנה את כיוון הדמות כאשר היא נעה ונוגעת במסגרת הבמה, כיוון הדמות מתהפך לכיוון הנגדי לכיוון המקורי לפני נגיעה במסגרת. אם הדמות הייתה בכיוון 90, ובתנועה נגעה במסגרת, הכיוון עתה יהיה (90-) .

דוגמה: הדמות בכיוון 90 ונעה לצד ימין, מצב כיוון הדמות מוצג באיור הבא:



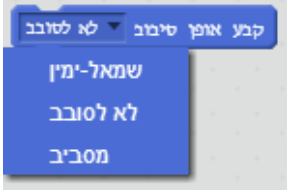
לאחר מפגש עם המסגרת וביצוע ההוראה 

כיוון הדמות מתהפך. כפי שמציג האיור הבא:



שים לב! כיוון הדמות שונה ל 90-, אולם הדמות גם בצעה "סיבוב" סביב נקודת ציר מרכזית במרכז הדמות, אם ברצוננו שהדמות לא תבצע את הסיבוב הזה בשינוי הכיוון סביב ציר הדמות, נשתמש בהוראה הבאה שיש לה 3 אפשרויות.

ההוראה – 

אפשרויות ההוראה: 

האפשרות 'שמאל-ימין' משנה את הדמות וצד ימין הופך לצד שמאל, וההיפך.


דוגמה לפני ביצוע שינוי הכיוון:

הדמות בכיוון 90 – הצד הימני זה חזית החתול, והצד השמאלי הזנב של החתול.

לאחר שינוי כיוון הדמות ל - 90 עם ההוראה **קבע את אופן הסיבוב למצב שמאל-ימין** נקבל את הדמות במצב הבא:



הדמות בכיוון (90-) – הזנב מצד ימין וחזית הדמות-חתול מצד שמאל.

האפשרות 'לא לסובב'

מצב לפני שינוי כיוון:



החתול בכיוון 90 – הזנב מצד שמאל והחזית מצד ימין.

לאחר שינוי כיוון הדמות על-ידי פגיעה במסגרת
ובחירת האפשרות לא לסובב נקבל את המצב:



החתול שינה כיוון (90-) אולם הדמות לא הסתובבה סביב צירה – הזנב נשאר מצד שמאל ותנועה לכיוון (90-) נראית כאילו החתול הולך לאחור.

האפשרות 'מסביב'



נראה עתה שמצב החתול לפני נגיעה במסגרת הוא:

עם האפשרות 'מסביב', כאשר החתול פגע במסגרת הוא שינה כיוון אבל הסתובב סביב צירו. כדי להפוך כיוון מצב החתול יהיה:



הקביעה איזו אפשרות לבחור תלויה בהקשר ובמטרת התסריט.

הוראות קבלת מידע על מיקום הדמות

קבוצת ההוראות הבאות מאפשרת לקבל את ערכי התכונות של מיקום X; של מיקום Y; ושל כיוון הדמות. ניתן לשלב ערכים אלו בתסריטים ולהכניסם למשתנים (הכול על משתנים בהמשך).

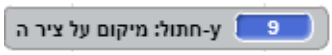
ההוראה – 

מאפשרת קבלת ערך מיקום ה X של הדמות.

שים לב! המסגרת הקטנה בצד שמאל מאפשרת לך להציג ערך זה בבמה כפי שיודגם בהוראה הבאה.

ההוראה – 

מאפשרת קבלת ערך מיקום ה Y של הדמות.

שים לב! המשבצת הקטנה מסומנת ולכן נראה בבמה את ערך ה- Y של הדמות: 

ההוראה – 

מאפשרת את קבלת ערך הכיוון של הדמות ואת הצגתו.

קבוצת ההוראות **מראה**

ההוראה – 

ההוראה מציגה אמירה באמצעות הדמות. את האמירה ניתן להציג לפרק זמן מסוים.

אם משתמשים בהוראה המקוצרת 

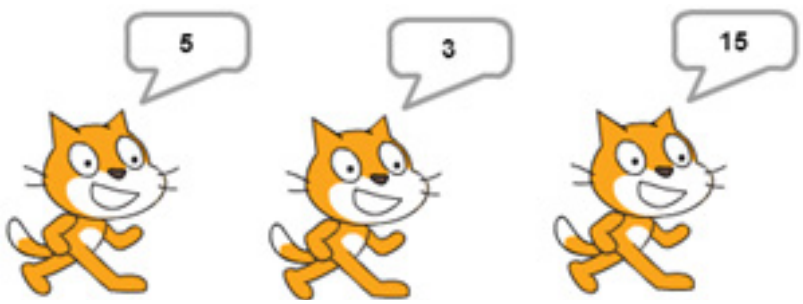
הפעולה משתנה בהתאם לתסריט. הוראה מקוצרת אחת של **'אמור'** תשאיר את ההודעה ללא שינוי מעל הדמות. אם מכניסים הוראת **'אמור'** נוספת, תימחק ההודעה הראשונה ותופיע ההודעה החדשה.

ההוראה 'אמור' והצגת ערכי משתנים וביטויים

ניתן באמצעות ההוראה **'אמור'** להציג ערכי משתנים וביטויים. התסריט הבא מציג שימוש בהוראה להצגת ערכי משתנים וביטוי מתמטי.



תוצאות התסריט מוצגות באיור הבא (הביצוע משמאל לימין).



ההוראה – **'חשוב'**



ההוראה זהה ומבצעת בדיוק את מה שההוראה **'אמור'** מבצעת, ההבדל הוא בתיבת השיחה המתנוססת מעל הדמות בעת ביצוע ההוראה. בהוראה **'חשוב למשך'** ובהוראה **'חשוב'**, תיבת השיחה היא בדומה לתיבת שיחה בקומיקס כאשר הדמות "חושבת".

ביצוע ההוראה **'חשוב'**



ביצוע ההוראה **'אמור'**



ההוראה – **'הצג'**



ההוראה קובעת שהדמות בתסריט תוצג. בתסריטים רבים ובמיוחד משחקים אנחנו מציגים דמויות על הבמה – בשלבים מסוימים מעוניינים שהדמות תוצג; בשלבים אחרים שהדמות תעלם. ההוראה **'הצג'**, מעבירה את הדמות למצב **'הצג'**.

ההוראה – **'הסתר'**



ההוראה קובעת שהדמות לא תוצג כלומר היא תהיה במצב **'הסתר'**. חשוב מאוד לדעת: דמות במצב הסתרה אינה "אומרת" דבר בפעולה **'אמור'**; וכן הבדיקה לדוגמה, **אם דמות נוגעת בדמות**, מחזירה **שקר** כאשר הדמות השנייה היא במצב **'הסתר'**, כלומר אין יכולת לבדוק האם נוגעים בדמות מוסתרת.

התסריטים הבאים של החתול והעטלף מציגים את המשמעות של **'הסתר'**

תסריט העטלף	תסריט החתול
	
החתול במצב **'הסתר'** וכאשר העטלף נוגע בחתול לא מתבצע משפט התנאי הבודק נגיעה בין הדמויות כי החתול מוסתר.	

ההוראה – **'תלבושת'**

סקרטץ' מאפשרת ליצור לכל דמות מספר 'תלבושות'. תלבושת של דמות היא ייצוג גרפי שונה של הדמות שאנו בוחרים. שינוי התלבושות מאפשר לנו ליצור אנימציה ולייצג את הדמות באופן שונה בהתאם למצב הדמות.

דוגמה: לדמות העטלף יש שתי תלבושות:

שם של תלבושת 1: כנפיים למעלה

שם של תלבושת 2: כנפיים למטה\





שימוש בשתי התלבושות בלולאה יאפשר לנו ליצור אנימציה של נפנוף כנפיים של דמות העטלף.

התסריט הבא משתמש בהוראה 'התלבושת הבאה', ובכך נוצר שינוי בכל 0.25 שניות במצב הכנפיים ויוצר אשליה של נפנוף כנפיים של העטלף.



אופן בחירה ויצירת תלבושות

במסך סקרטץ' יש שלוש לשוניות לבחירה:



בחירת האפשרות '**תסריטים**' מאפשרת לנו לכתוב תסריטים לדמויות.

בחירת האפשרות '**תלבושת**' מאפשרת לנו לטפל בתלבושות הדמות (לערוך, להוסיף, למחוק).

בחירת האפשרות '**צלילים**' מאפשרת לטפל בצלילים שילוו את התסריטים של הדמות.

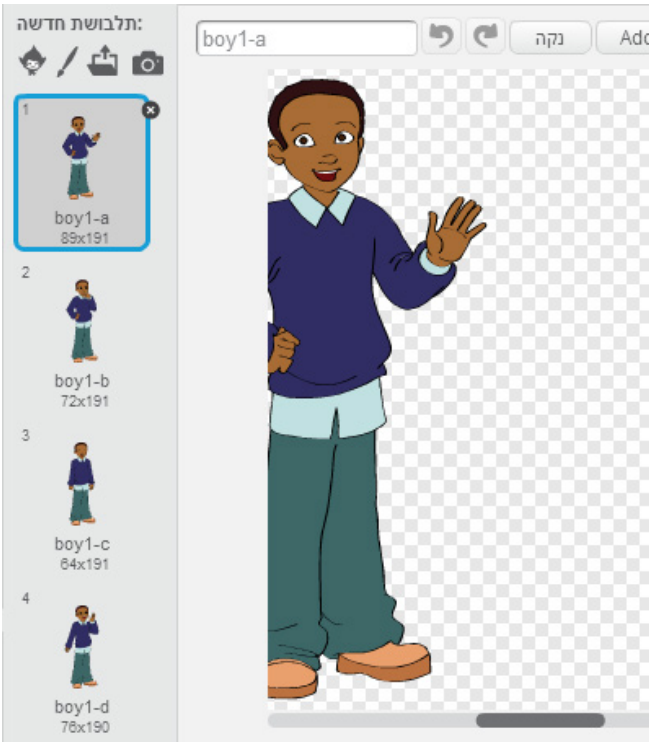
האפשרות 'תלבושת'

נבחר דמות חדשה מספריית הדמויות שנקראת BOY1

נעבור למצב עריכת תלבושת על ידי בחירת הלשונית '**תלבושת**' – 

, ונוכל לראות שבגרסת סקרטץ' 2.0 יש מספר תלבושות לדמות שנבחרת – לדמות BOY1 יש 4 תלבושות שמאפשרות, בדומה לתסריט שכתבתי עבור העטלף, ליצור אנימציה מהירה. העתיקו את התסריט של העטלף לדמות ה- BOY1 על ידי בחירת התסריט וגרירתו מעל לדמות BOY1.

מסך העריכה של הדמויות והתלבושת דומה במידה רבה למסך של כל תוכנת עריכה גרפית בסיסית כמו PAINT, ואפשר להתנסות בעריכה של דמויות קיימות או דמויות מקוריות, וכמובן בעריכה גרפית של התלבושת של כל דמות.

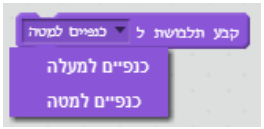


אנו רואים את הדמות BOY1 ואת 4 התלבושות הזמינות לנו: BOY-1A; BOY-1B; BOY-1C; BOY-1D

ניתן לערוך כל תלבושת, ליצור תלבושת חדשה, לייבא תלבושת מהאינטרנט או ממיקום על המחשב שלך. (הרחבה על אפשרויות העריכה הגרפית תוצג בספר בנפרד).

ההוראה – 

ההוראה קובעת את התלבושת של הדמות לפי בחירתנו. עבור כל דמות ההוראה מאפשרת בחירה של תלבושת מאוסף התלבושות של הדמות. לדוגמה לדמות עטלף יש שתי תלבושות הניתנות לבחירה:



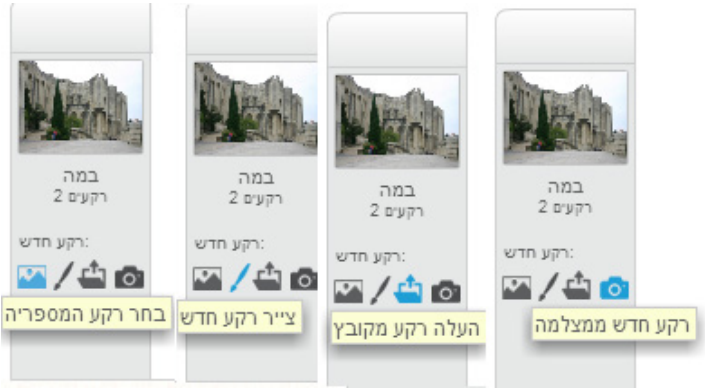
ההוראה – 

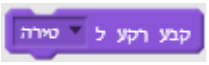
ההוראה 'התלבושת הבאה' מאפשרת מעבר בין תלבושות לפי סדר הגדרתן. בדוגמה שלנו עשינו שימוש בהוראה בתסריט והעטלף עבר באמצעות הוראה זו בין שתי התלבושות ובכך קבלנו אשליית תנועה של כנפי העטלף.

ר ק ע

בפרויקט חדש שאנו מתחילים, יש לבמה שלנו רקע, הרקע הוא המראה שאנו רואים מאחורי הדמויות – ברירת המחדל היא רקע לבן. אולם את הרקע ניתן לשנות ואפשר ליצור מספר רקעים שישמשו אותנו במצבים שונים של הפרויקט שלנו. במשחקי מחשב הרקע יכול להשתנות לפי רמות המשחק – אם נבנה פרויקט שמספר את סיפור כיפה אדומה נשנה את הרקע בהתאם למקום האירוע: היער, הבית של סבתא, ועוד.

אנחנו יכולים להשתמש ברקעים שעומדים לרשותנו בסביבת העבודה, ליצור רקעים או לייבא רקעים מקובץ (תמונה שהורדנו מרשת האינטרנט, או תמונה מהמצלמה).

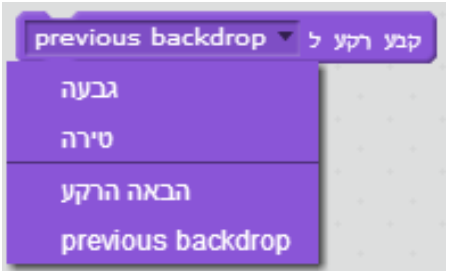


ההוראה – 

ההוראה מאפשרת להציג את אחד הרקעים שיש לנו בפרויקט, את הרקע הקודם או הרקע הבא.

שים לב! בדומה לתלבושות, השימוש בהוראה **'הרקע הבא'** מאפשר תנועה מחזורית בין רקעים שקיימים בפרויקט.

בפרויקט שלנו יש שני רקעים: גבעה; טירה, כך שההוראה מאפשרת בחירת אחד הרקעים הזמינים, ומאפשרת גם לנוע בין שני הרקעים במחזוריות שאנו בוחרים.



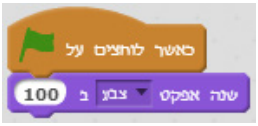
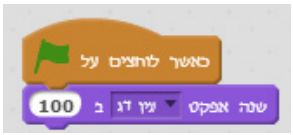
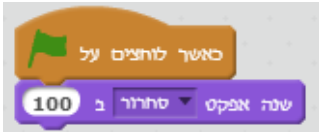
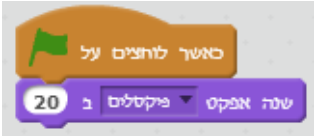
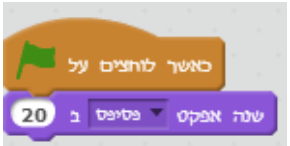
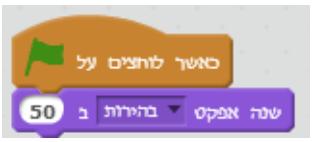
הוראות שינוי אפקט

ניתן לשלוט על מראה של דמות – שינוי באפקט של הדמות יוצר שינוי בדמות. השינוי נקבע לפי ערך שאנחנו קובעים. לא תמיד השינוי ידוע מראש ולכן כשאנחנו משתמשים בהוראה **'שנה אפקט'** עלינו לבחון את השינוי – לאחר שהשינוי מתאים לתסריט ולאירוע שאנו רוצים להבליט – אנחנו משלבים את הפקודה בתסריט.



ישנה קבוצה גדולה של שינויים שאנו יכולים לשנות בדמות באמצעות ההוראה **'שנה אפקט'**:

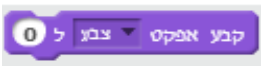
מוצגת דמות העטלף עם שינוי אפקט שונה וערך מסוים
כדי להמחיש את אופי השינוי של כל אפקט.

הדמות המקורית	הוראה	תוצאה
		
		
		
		
		
		

סיכום הוראות 'שנה אפקט'

ההוראה – 

משנה את האפקט המבוקש בערך המתווסף לערך הנוכחי של הדמות.

ההוראה – 

קובעת את ערך האפקט ללא תלות בערך קודם.

ההוראה – 

מנקה של כל האפקטים שנקבעו לדמות בתסריט, ומחזירה את ערכי ברירת המחדל של תמונת הדמות.

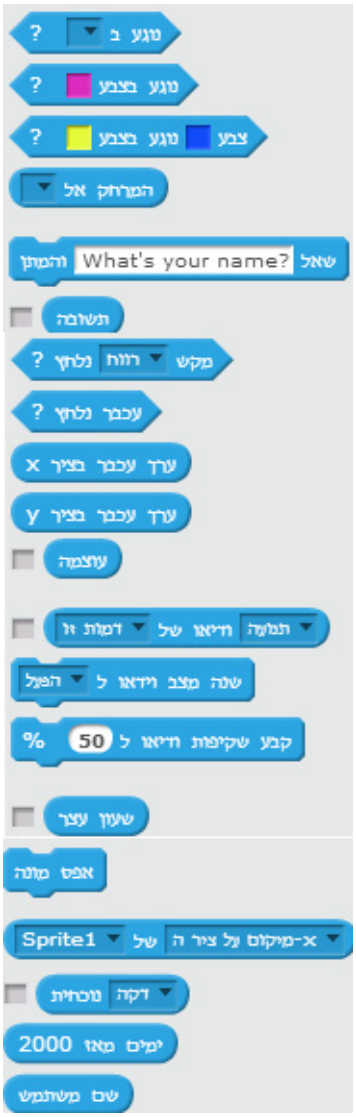
ההוראה – 

כל דמות מופיעה בגודל מסוים וניתן לשנות את גודל הדמות. ההוראה **'שנה גודל ב'** משנה את גודל הדמות לפי הערך בהוראה: אם הערך חיובי הדמות גדלה; אם הערך שלילי הדמות קטנה. התסריט הבא מקטין את הדמות ומגדיל אותה. השינוי הוא שינוי ביחס לגודל שקודם להוראה.

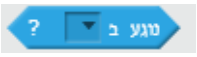


ההוראה – 

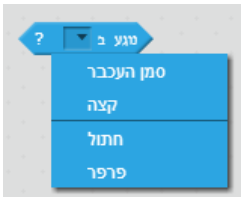
ההוראה **'קבע גודל ל'** משנה את גודל הדמות ללא קשר לגודל הנוכחי של הדמות.



קבוצת ההוראות חיישנים

החיישן – **'נוגע ב'** 

החיישן מחזיר ערך **אמת**/ נכון (TRUE) או **שקר**/לא נכון (FALSE) אם הדמות נוגעת/לא נוגעת ב: **סמן העכבר; בקצה הבמה**; או באחת מהדמויות הנוספות בפרויקט. לדוגמה עבור פרויקט עם הדמויות חתול, ברווז, פרפר: אם נשלב את החיישן בתסריט של דמות החתול נקבל את האפשרויות הבאות:



החיישן – **'נוגע ב'** 

החיישן מחזיר ערך **אמת**/ נכון (TRUE) או **שקר**/לא נכון (FALSE), אם הדמות נוגעת/לא נוגעת בצבע מסוים.

בחירת הצבע נעשית על ידי לחיצה על העכבר מעל הצבע, סמן העכבר הופך ל"יד" והעברת הסמן אל נקודה בבמה או לנקודה בדמות תבחר את הצבע הרצוי.

דוגמה לשימוש בהוראה **'נוגע בצבע'**: נניח שיש קבוצה של דמויות – עכברים שהחתול מעוניין לתפוס. ניתן לבחור צבע זהה לכל העכברים (צבע מלא או בהיקף הדמות) ואם אנו מקבלים ערך **אמת** משמעות החיישן **'נוגע בצבע'** שבחרנו היא שהחתול נוגע בעכבר.

דוגמה נוספת: משחק פקמן או כל משחק בו דמות נעה במסלולים. אנחנו רוצים לדעת האם הדמות חורגת מהמסלול. נבחר צבע מסוים לגבולות המסלול ונשתמש בהוראה **'נוגע בצבע'** שתחזיר ערך **אמת** אם הדמות המרכזית שלנו נוגעת בגבולות מסלול התנועה.



במשחק הפקמן נוכל להשתמש בחיישן נוגע בצבע כדי לדעת: האם הדמות נוגעת בנקודות המסלול, האם הדמות נוגעת בגבול המסלול, האם הדמות נוגעת באחת מהדמויות של המפלצות. בהתאם לתוצאות הבדיקה נקבע מה יתבצע.

החיישן – 

החיישן מבצע בדיקה יותר מורכבת של נגיעה בצבע ובודק האם צבע מסוים נוגע בצבע אחר.

דוגמה: אם חלקים של הדמות שלנו צבועים בצבעים שונים נוכל לבדוק האם חלק מסוים בדמות, בעל צבע ייחודי מסוים, נוגע בצבע אחר של דמות אחרת או בחלק מהרקע.



עבור דמות הברווז

באמצעות החיישן צבע נוגע בצבע נוכל לבדוק:

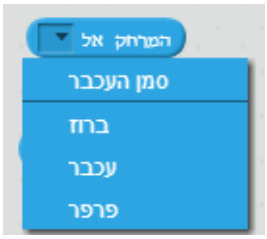
האם הברווז נוגע בצבע אחר באמצעות רגליו (האדומות) ---------- 

האם הברווז נוגע בצבע אחר באמצעות מקורו (הכתום) ---------- 

האם הברווז נוגע בצבע אחר באמצעות חלק אחר של גופו (צהוב)--- 

החיישן – 

הוראה זו מחשבת את המרחק בין הדמות לסמן העכבר או לדמויות אחרות. בפרויקט בו ישנן הדמויות חתול; ברווז; פרפר; ועכבר, בתסריט ששייך לחתול נקבל את האפשרויות:

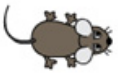


דוגמה: התסריט הקצר הבא מציג באמצעות החיישן '**המרחק אל**' וההוראה '**אמור**' את המרחק בין החתול לדמות לעכבר.



הרצת התסריט המחשב את המרחק בין החתול לעכבר.

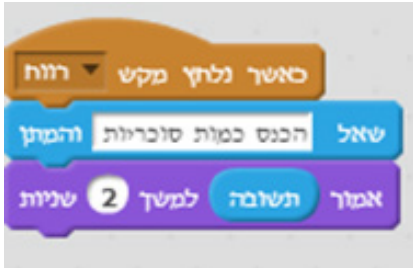




החיישן - 

חיישן זה חשוב ביותר ומאפשר לקלוט מידע ממשתמש בתסריט באמצעות המקלדת. המשתמש בוחר את הטקסט בהוראה והמידע שמתקבל מוכנס ל'משתנה' בשם '**תשובה**' - 

דוגמה:



התוצאה של התסריט: החתול מכריז את ההודעה שכתבנו בהוראה



המשחק מכניס את הערך שבו הוא מעוניין בשורת הקלט בתחתית הבמה



הערך שאנו מקלידים נכנס ל- תשובה בדוגמה זו הוקלד 76, וההוראה האחרונה אכן מציגה את הערך הזה המאוחסן ב- תשובה



דוגמה:



הסבר על תסריט הדוגמה: השתמשנו בהוראה מקבוצת 'אירועים' **'כאשר לוחצים על דמות זו'**. כלומר התסריט יחל לעבוד לאחר לחיצה עם העכבר על החתול. החיישן 'שאל' מציג את ההודעה למשתמש, וממתין לקלט. (משפט התנאי **'אם'** כולל מרכיב מקבוצת ההוראות 'מפעילים' ועל קבוצה זו שמאפשרת כתיבת ביטויים יורחב בהמשך). באופן מעשי, המפעיל שנבחר בודק האם המשתמש הקיש את המילה "מיאו", אם כן, מתבצעת הוראה מהקבוצה 'צליל' שמשמיעה את הצליל מיאו.

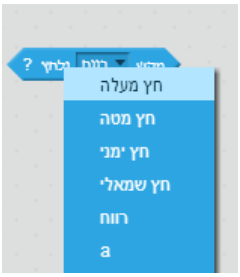
הבמה בתסריט זה



כתבנו בשורת הקלט **מיאו**. כדי לשמוע את הצליל החמוד הזה נסה את התסריט והקש מיאו.

החיישן – 

הוראה זו בודקת האם התרחש אירוע של לחיצה על מקש כלשהו. החיישן משמש לבדיקת לחיצה על כל אחד מהמקשים וזאת על ידי לחיצה על החץ הקטן בהוראה ובחירת המקש:



חשוב להבחין בין חיישן זה להוראה  . ההוראה **'כאשר נלחץ מקש'** שייכת לקבוצת ההוראות 'אירועים' שמשמעותם המתנה להתחלת תסריט לאירוע שהתרחש. ההוראה **'מקש נלחץ?'** אינה הוראה אלא חיישן שמשולב בהוראת תנאי או בהוראת לולאה, ויכול להיבדק מחזורית ואינו ממתין בביצוע.

דוגמה: הדוגמה הבאה תסביר את המשמעות של בדיקה מחזורית של האירוע 'מקש נלחץ?



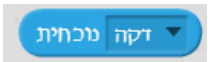
התסריט יחל בביצוע כאשר לוחצים על מקש רווח, זוהי הוראת המתנה לאירוע מקבוצת ההוראות אירועים.

בתסריט לולאה **'לעולמים'** שבמרכזה בדיקה בכל מחזור בלולאה האם התרחשו האירועים: מקש **'חץ ימני נלחץ?'**, או מקש **'חץ שמאלי נלחץ?'** בהתאם לאירוע שהתרחש מתבצע קוד שכלול במשפט התנאי. באופן מעשי התסריט הזה מבצע עבור כל דמות תנועה ימינה ושמאלה על המסך.

החיישן –  החיישן מבצע בדיקה האם העכבר נלחץ ומחזיר ערך **אמת** (TRUE) אם העכבר נלחץ, וערך **שקר** (FALSE) אם אחרת.

החיישנים –  

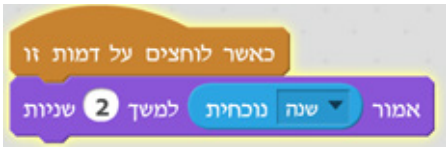
חיישנים אלו בודקים היכן ממוקם העכבר ומחזירים את ערך ה- X של מיקום העכבר או ערך ה- Y של מיקום העכבר.

חיישן – העברת מידע על תאריך וזמן עכשווי: 

האפשרויות בחיישן זה



בתסריט הבא החתול יאמר את השנה:



החתול אומר:



חיישן – שעון העצר 

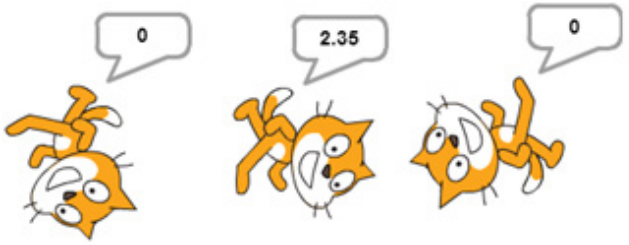
בסביבת סקרטץ' ניתן לעשות שימוש בשעון עצר – ניתן לאפס את השעון ולהציג את הזמן שחלף.

עצירת שעון העצר ואיפוסו – 

התסריט הבא מפעיל את שעון העצר ומפסיק אותו, החתול מציג את הזמן שחלף.



נביט באמירות של החתול לאורך ביצוע התסריט, הסדר משמאל לימין:



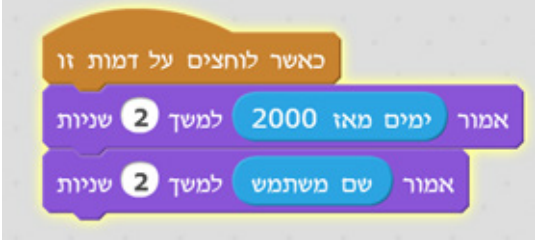
החיישן – ימים מאז 2000 

החיישן מחזיר כמה ימים חלפו מאז 2000/1/1

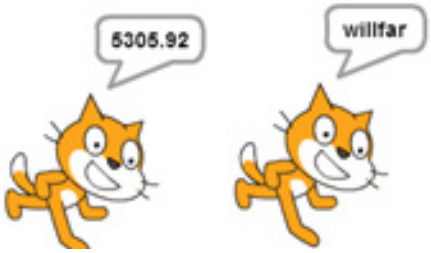
החיישן – שם משתמש 

החיישן מחזיר את שם המשתמש שרשום באתר של סביבת סקרטץ'.

תסריט דוגמה:



החתול יאמר



קבוצת ההוראות נתונים

נתונים הם מרכיב בסיסי בכל שפת תכנות. השם הנפוץ יותר למרכיב זה בשפת תכנות הוא '**משתנים**'.

המשתנים מאפשרים לנו לשמור מידע על הפרויקט שאנו כותבים, על הדמויות, על הרקע, וכל מידע חשוב שברצוננו לשמור כדי לבנות משחק או כל יישום כללי אחר מעניין.

אנו קוראים לנתונים **משתנים** כי הערך של הנתונים הללו יכול להשתנות במהלך התסריט. אם לדוגמה יש לנו משתנה שיכיל במשחק את מספר הנקודות שצברנו, הרי ברור שמספר זה ישתנה במהלך המשחק. או דוגמה אחרת – משתנה שיכיל את "העוצמה" של הדמות, גם ערך זה יכול להשתנות, לקטון או לגדול.

בקבוצת חיישנים, במידה מסוימת, עשינו שימוש במרכיב מהקבוצה שיש לו תכונות של משתנה – המרכיב 

כשאנחנו מבצעים את ההוראה 

וקולטים ערך מהמשתמש, ערך זה "נכנס" לתשובה ועתה אנחנו יכולים להשתמש בו להמשך ביצוע התסריט. ניתן לשלב ערך זה באחד המפעילים כפי שנראה בהמשך, להציג אותו על הבמה וכדומה.

גם בקבוצת ההוראות '**תנועה**' ישנם משתנים של המערכת בהם ניתן לעשות שימוש, לדוגמה:







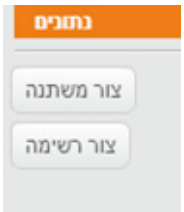
אלו משתנים העוקבים אחר מיקום ה- X ומיקום ה- Y של דמות וכן אחר הכיוון של הדמות. משתנים אלו מתעדכנים באופן אוטומטי לפי שינוי מיקום הדמות וכיוונה. סביבת סקרטץ' גם מאפשרת לנו לשנות את המיקום והכיוון בצורה ישירה באמצעות ההוראות:



'משתנים', אם כך, די דומים ל: תשובה, מיקום, כיוון, שהם בעצם משתנים שנוצרים על ידי סביבת סקרטץ'. אולם ב'משתנים' ניתן לבצע יותר פעולות באמצעות הוראות מיוחדות שעומדות לרשותנו ברגע שאנו יוצרים משתנה.

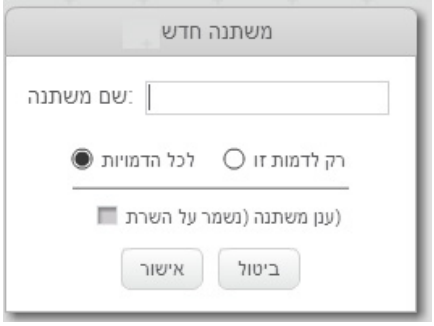
יצירת משתנה

בתפריט קבוצת נתונים יש לנו שתי אפשרויות:



נוכל ליצור משתנים ורשימות, בשלב ראשון נתמקד ביצירה ושימוש במשתנים, בהמשך יוסבר על 'רשימה'.

נבחר את האפשרות **צור משתנה** ונקבל את תיבת השיחה של יצירת משתנה



עתה נבצע שני צעדים:

1. ניתן שם למשתנה
2. נבחר האם המשתנה שייך לדמות מסוימת שפעילה ברגע זה או לכלל הדמויות בפרויקט

לא נבחר את המשתנה, 'ענן משתנה' ונשאיר את התיבה הזו לא מסומנת.

ניצור משתנה שתפקידו יהיה לשמור את מספר הפעמים שהחתול אמר "מיאו"

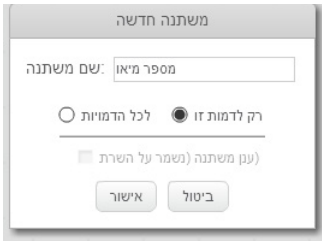
שם משתנה

א. ניתן לבחור שם בעברית או באנגלית ועניין זה כמובן נתון להעדפה אישית

ב. ניתן לבחור כל שם, אולם חשוב שהשם יהיה משמעותי ומרמז על תפקיד המשתנה בתסריט או מה המידע שהמשתנה שומר על הדמות לה הוא שייך.

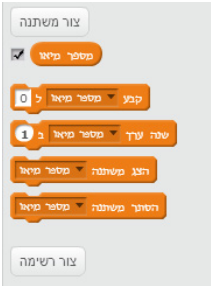
בדוגמה שלנו, המשתנה שומר מידע על מספר הפעמים שהחתול אמר "מיאו" לכן נוכל לקרוא למשתנה 'מספר מיאו'.

כעת נציג את השלבים בביצוע יצירת משתנה ומהי התוצאה של יצירת משתנה בסקרטץ', במיוחד המשתנה הראשון שיצירתו יוצרת את התפריט המלא של קבוצת משתנים.



נבחר שם משתנה – 'מספר מיאו', ונקבע שהמשתנה שייך רק לדמות החתול, לחיצה על לחצן **אישור** והמשתנה יוגדר וכל תפריט משתנים יופיע זמין לשימוש.

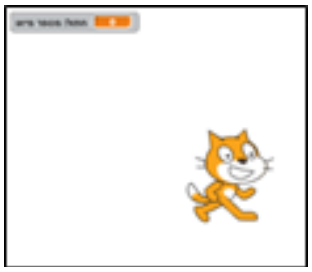
תפריט קבוצת ההוראות נתונים



בתפריט ניתן כמובן ליצור עוד משתנים והכפתור הזה זמין, כמו כן הכפתור של יצירת רשימה זמין ועתה קבלנו קבוצת פקודות להן יוצג הסבר.

סימון המשתנה מוסתר/נראה - 

האליפסה הכתובה עם שם המשתנה מייצגת לנו את המשתנה, ישנה אפשרות לראות את ערך המשתנה לאורך ביצוע התסריטים על הבמה או להסתיר את הערך. אם נסמן √ במשבצת הסימון נוכל לעקוב ולראות כיצד הערך במשתנה התצוגה מופיע על הבמה.



ברמה המעשית, הצגת ערכי המשתנים על הבמה אינה מומלצת, חלק משטח הבמה מבוזבז לטובת הצגה של המשתנים וכאשר מדובר בפרויקט לא קטן ייתכן מצב בו כל הבמה תתמלא בתצוגת משתנים. מומלץ להשתמש במצב ערכי משתנים מוצגים בתהליכי בדיקה של התסריטים, אם התסריט והשינוי במשתנים אכן תקין.

מתי יש צורך של הפרויקט להציג ערכי משתנים?

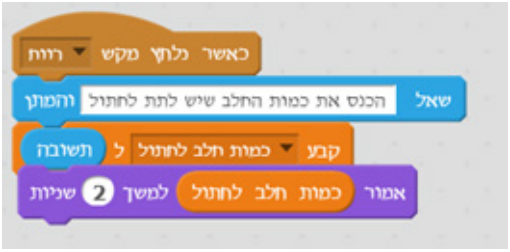
התשובה: כאשר המשתנה מציג ערך שעל המשתמש בתכנית לעקוב אחריו או שערך זה עשוי לעניין את המשתמש, במקרה זה נציג את הערך והשתנות הערך בעת הרצת התסריטים. לדוגמה: משתנה שיכיל כמה פסילות המשתמש קיבל כאזהרה לפני סיום משתמש, מספר הנקודות שמשתמש במשחק צבר וכדומה.

ההוראה – 

הוראה זו קובעת ערך למשתנה, ללא קשר לערכו הקודם, הערך הקודם נמחק. הערך שאנו מכניסים יכול להיות ערך קבוע, או ביטוי מתמטי מורכב הכולל משתנים נוספים, פעולות מתמטיות, סוגריים וכדומה.

דוגמה: כאשר אנו קולטים ערך באמצעות הפעולה **שאל** , הערך נכנס למשתנה תשובה משתנה זה שייך לכלל הפרויקט, אם נבצע שוב הוראת שאל באמצעות **שאל** הערך הראשון שקלטנו יימחק ויידרס על ידי הערך החדש.

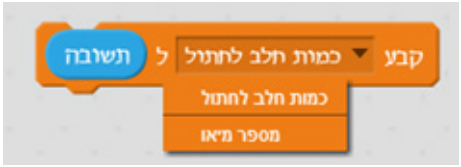
הפתרון הוא להעביר את הערך למשתנה שאנו מגדירים, וכך נוכל לקלוט ערך חדש וכמובן נוכל להשתמש בכלל הפעולות שאפשרי רק למשתנים שאנו מגדירים ושאינו אפשרי למשתני מערכת.



אנחנו קולטים בהוראה הראשונה ערך מספרי, הערך הזה מוכנס באופן אוטומטי ל- תשובה ,ההוראה הבאה שאנו משתמשים היא: **קבע**, הוראה זו קובעת את הערך של המשתנה 'כמות חלב לחתול' כערך שימצא ב- תשובה וההוראה האחרונה **'אמור'** ממחישה את העברת הערך למשתנה. אם נריץ את התסריט נוכל להבין היטב את העברת הערכים.

כיצד עובדים עם הוראות שונות כשיש כמה משתנים? **התשובה:** על ידי בחירת המשתנה הרצוי מתוך קבוצת המשתנים שהגדרנו. כאשר יש כמה משתנים ונניח שברצוננו להשתמש בהוראה **'קבע'** למשתנה מסוים, אנחנו פשוט לוחצים על החץ הקטן בהוראה, כמו שאנו מבצעים בהוראות רבות בהן יש מספר חלופות. בתהליך הזה אנחנו מקבלים את רשימת המשתנים הקיימת והזמינה עבור הדמות ומתוכם אנחנו בוחרים את הרצוי לנו.

האיור הבא מציג את השימוש בהוראה **'קבע'** ואת בחירת המשתנה באמצעות החץ הקטן, מתוך הרשימה הזמינה.



ההוראה – 

הוראה זו מאפשרת לבצע שינוי בערך של משתנה. השינוי תלוי בערך הקודם של המשתנה.

אם לדוגמה לפני ביצוע ההוראה ערך המשתנה היה 7, עתה לאחר ההוראה ערך המשתנה הוא 8, כי שינינו את ערך המשתנה בערך 1.

התסריט הבא מדגים את השפעת ההוראה **'שנה ערך'** על ערכו של משתנה.



נניח שהקלדנו את המספר 15, בעת ביצוע ההוראה **'שאל'**

ההוראה **'שנה ערך'** עדכנה את ערכו, ועתה הערך של המשתנה 'כמות חלב לחתול' הוא 8

ההוראה **'שנה ערך'** תשנה את ערכו שוב, ועתה הערך הוא 20.

ההוראות –

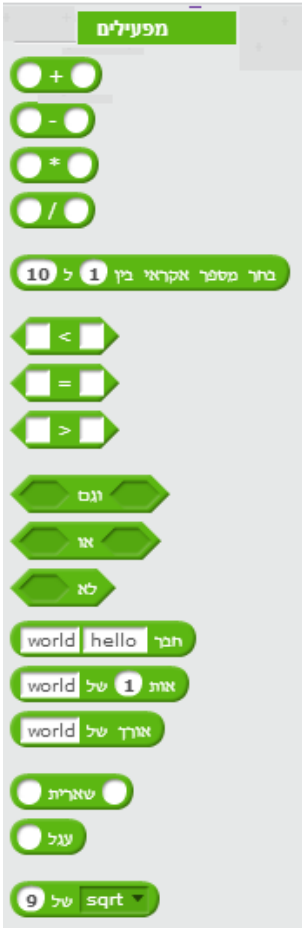


אנחנו יכולים לקבוע את מצב השקיפות של משתנה בעת הכתיבה של התסריטים על ידי סימון המשבצת המתאימה. אולם בעת פעולת התסריט נדרש לעיתים להסתיר את הערך של המשתנה או לחשוף את הערך של המשתנה. ההוראות 'הצג משתנה' ו- 'הסתר משתנה' מאפשרות לנו לשלוט על חשיפת ערך המשתנים והסתרתם לפי הצורך.

רשימות

נושא רשימות נכלל בתפריט 'נתונים' אולם נציג אותו יחד עם נושאים מתקדמים אחרים בפרק מיוחד.

קבוצת ההוראות מפעילים

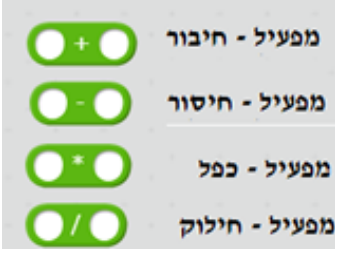


המפעילים מאפשרים לנו לבצע פעולות חשבוניות מורכבות ולהשתמש בתנאים לוגים פשוטים ובתנאים לוגיים מורכבים.

למרות שחלק מהמפעילים וההסברים למפעילים אלו הם מורכבים, ואולי תלמידים בגילאי בית ספר יסודי יתקשו להבין את מלוא המשמעויות, נשתדל להסבירם בצורה הפשוטה ביותר, ולתת דוגמאות כך שמי שלא הבין לעומק את הרעיונות עדיין יוכל להשתמש במפעילים המורכבים יותר.

קבוצת מפעילים – פעולות חשבוניות פשוטות

ארבעת המפעילים הבסיסים מאפשרים ביצוע פעולות חשבון בין ביטויים מתמטיים.



המפעילים האלו מאפשרים לנו לבצע את ארבע פעולות החשבון הבסיסיות.

דוגמאות:



שילוב מפעילים בהוראות

איך אנחנו משתמשים בביטויים שנוצרו באמצעות המפעילים? ניתן לשלב מפעילים בהוראות רבות, נציג דוגמאות עם הסברים:



אנחנו רואים כיצד שילבנו מפעילים בהוראות שונות.

ביטויים מורכבים במפעילים

במפעילים ניתן ליצור ביטויים מורכבים שכוללים משתנים ופעולות.

דוגמאות:



יצרנו ביטויים מורכבים הכוללים קבועים, משתנים ופעולות. שים לב בהוראה **'אמור'** בסוף התסריט, שילבנו מפעיל בתוך מפעיל וכך קיבלנו ביטוי מורכב משתי פעולות.

את השילוב עשינו בשלבים שמתוארים באיורים הבאים:

שלב א: ניצור את הביטוי: 

הביטוי כולל משתנה **מספר מיאו** וקבוע 7, הפעולה היא פעולת חיבור.

שלב ב: נבחר את המפעיל: 

שלב ג: נשלב את הביטוי שיצרנו בשלב א' במפעיל החדש בצד ימין: 

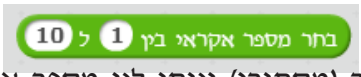
שלב ד: נשלים את הספרה 2 בביטוי בצד שמאל ונקבל את הביטוי המורכב הסופי: 

חשוב לשים לב!

א. סדר הפעולות מתבצע תמיד משמאל לימין.

ב. הביטו הפנימי 

מתפקד כמו 'סוגריים' בביטוי חשבוני ולכן ביטוי זה יחושב ראשון.

ג. המפעיל – 

זה מפעיל מעניין והוא תורם בתסריטים רבים ליצירת עניין רב. המפעיל מבצע חישוב (מסתורי) ונותן לנו מספר אקראי בתחום המופיע במפעיל, בהוראה שמופיעה בכותרת המספר האקראי היה בטווח 1-10.

מדוע צריך מספרים אקראיים?

בכדי לתמוך ביצירת אירועים שונים אקראיים. במשחק **יוגי והבננות** המופיע בספר, דמות של דובי חמוד מנסה לתפוס בננות הנופלות 'מהשמיים', הדובי נע לאורך הבמה ימינה ושמאלה ומנסה להכניס את אשכולות הבננות לתוך הסל שבידיו.

ברור שנרצה שהבננות יופיעו במיקומים שיפתיעו את המשחק ולא במיקום קבוע שחוזר על עצמו ולכן נשתמש במפעיל **'בחר מספר אקראי בין'** בכדי ליצור עניין במשחק. קטע מהתסריט של יוגי מופיע כאן וממחיש את הצורך במספרים אקראיים:



רואים שדמות הבננה מוצבת ב-X אקראי הנבחר בתחום (200-) ועד 200. כך הבננה הנופלת תיפול בכל מחזור בלולאה במיקום אקראי אחר.

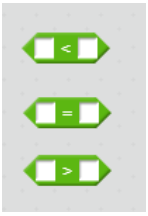
קבוצת מפעילים – יחסים לוגים

קבוצה זו כוללות שלושה מפעילים שמאפשרים לנו לבדוק יחס לוגי בין ביטויים:

יחס גדול מ >

יחס קטן מ <

יחס שוויון =



מפעילים אלו משולבים במשפטי לולאות ובמשפטי תנאי, ומאפשרים לנו לבצע תסריטים עם תהליכי בקרה וניתוב של התסריטים.

דוגמאות

בדיקה האם שני מספרים שווים: מפעיל יחס שוויון 

בתסריט יש שתי דמויות: ברווז; חתול. לכל דמות יש "עוצמה" שאנו קולטים בתחילת התסריט וברצוננו לבדוק האם העוצמות של שתי הדמויות שוות.

התסריט מבצע את המשימה עם:



גם המפעילים של יחס > ; יחס < ; מאפשרים לנו לבדוק ולהשוות בין ביטויים שונים. לדוגמה התסריט הבא מתבסס על תסריט הדוגמה הראשונה ובודק איזו עוצמה של דמות גדולה יותר.

הערה: ברמה העקרונית ניתן להסתפק ביחס > או ביחס < לבחינת יחסי גודל בין ביטויים, זאת על ידי שינוי בסדר של הביטויים במפעיל.

תסריט עוצמה של מי גדולה יותר



מפעילים – אופרטורים לוגיים



המפעיל הלוגי 'וגם'

לעיתים אנחנו רוצים לבדוק מצב של **'גם'** או מצב של **'או'**. (בהנחה שהמשפט אינו מספיק ברור נרחיב בהמשך). נניח, שברצוננו לבדוק **האם החתול נוגע בברווז** וגם **המרחק שלו מסמן העכבר קטן מ-30.** אנחנו לא יכולים לבצע זאת רק עם מפעיל יחס לוגי יחיד. נדרשים שני מפעילים משום שצריך לבצע שתי בדיקות. למצב זה בו נדרש לבצע שתי בדיקות לוגיות או יותר ושכולן יהיו **אמת** קוראים **'גם'** – מצב בו כל הבדיקות חייבות להיות נכונות – **אמת**. באנגלית או יותר נכון בשפה "מחשבית מקצועית" מצב זה נקרא 'אופרטור לוגי AND'. במידה ואחד התנאים אינו **אמת** (או שניהם), לא התבצעו ההוראות של החלק **'אז'** בהוראה **'אם'** אלא ההוראות של החלק **'ואם לא'**.

תסריט הדוגמה:



שלבים בכתיבת תנאי מורכב

א. כתיבת התנאי הראשון: 

ב. כתיבת התנאי השני: 

ג. בחירת מפעיל של תנאי AND (וגם) - 

ד. שילוב התנאים הבסיסיים שיצרנו בשלב א׳ ובשלב ב׳ בביטוי המורכב



ה. שילוב הביטוי המורכב בבלוק של ההוראה **'אם'**



המפעיל הלוגי 'או'

בביטויים לוגיים שילוב המפעיל **'או'** דומה לשילוב המפעיל 'וגם' בהבדל שבמפעיל הלוגי **'או'** התוצאה לבדיקה, האם הוא **אמת** או **שקר**, שונה – בביטוי לוגי עם **'או'** ישנה דרישה שרק אחד מהתנאים יהיה **אמת**, כלומר אם נכתוב את התנאי הבא: אם עוצמה> 30 **'או'** נוגע בחתול, התוצאה של התנאי המורכב תהיה אמת באחד משלושת המצבים הבאים:

א. עוצמה > 30 : **אמת**

ב. נוגע בחתול: **אמת**

ג. עוצמה > 30 ; נוגע בחתול: שני התנאים **אמת**

כלומר המצב היחידי בו נקבל תוצאה של **שקר** מהמפעיל היא כאשר שני התנאים הם **שקר**. כלומר משפט עם הרבה תנאים המחוברים ביניהם עם **'או'**, התוצאה **שקר** אם כל התנאים הם שקר, והתוצאה אמת אם **רק אחד** מהתנאים יהיה אמת.

המפעיל הלוגי 'לא'

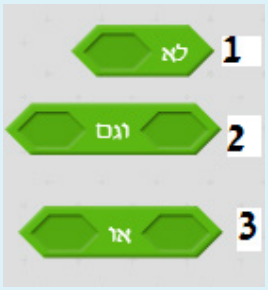
אם ברצוננו לדוגמה לבדוק את המצב שהברווז **'לא'** נוגע בחתול נשתמש במפעיל **'לא'**:



סיכום מפעילים לוגיים

שלוש המפעילים הלוגיים **'וגם'; 'או'** ; **'לא'** מאפשרים לנו ליצור תנאים מורכבים לבדיקה של מצבים מורכבים בתוך התסריטים שאנו כותבים.

חשוב שים לב! ניתן לחבר תנאים מורכבים ולשלבם בתנאים מורכבים אחרים וליצור מבנים מאוד מורכבים. זה לא תמיד מומלץ ורצוי לפרק את הבדיקות לבדיקות יותר מצומצמות, אולם חשוב לשים לב שיש סדר ביצוע של מפעילים לוגים לפי סדר העדיפות הבא:



קבוצת המפעילים לטיפול במילים

בכל שפת תכנות ישנם מפעילים ש"מטפלים" במילים. מילים בהגדרה מחשבית אינן כוללות רק אותיות אלא את כל המקשים הזמינים על המקלדת כולל: אותיות, ספרות, סימנים מיוחדים. בסקרטץ' ישנן שתי הוראות לטיפול במילים שבשפה המקצועית של אנשי המחשבים נקראות מחרוזות (String).

ההוראה – 

ההוראה הזו מחשבת אורך של מילה. בדוגמה הבאה נקלט משפט מהקלט באמצעות ההוראה '**שאל**'. נתייחס לכל המשפט כמילה ארוכה אחת; נכניס את המשפט למשתנה; נבדוק את אורך המילה, כלומר כמה סימנים-תווים יש במשפט.



הפעלת התסריט שלב אחר שלב:



ההוראה **'שאל'** הופעלה. הוכנס משפט ארוך שמכיל, כולל רווחים בין המילים, סה"כ 28 סימנים-תווים.



אנחנו רואים שבערך המשתנה המופיע בבמה יש את התוכן של המשפט שהוקלד והברווז אומר את המספר 28 שזה בדיוק מספר התווים הכלולים במשפט.

ההוראה – 

ההוראה הזו מאפשרת לגשת לתו-סימן מסוים בתוך מילה-מחרוזת. הערך שמכניסים בהוראה הוא מיקום התו שאליו רוצים לגשת בתוך המחרוזת. התסריט הבא מדגים קלט שנקלט שהוא ערך מספרי – אנו ניגשים לתו במילה לפי הערך שקיבלנו מהקלט.

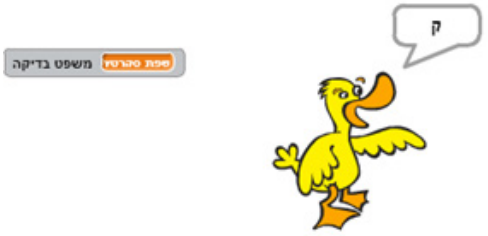


הפעלת התסריט:



הכנסנו את המשפט 'שפת סקרטץ'. הקלט המספרי הוא 6 – כלומר גישה לתו ה-6 במחרוזת עם ההוראה –





ואכן הברווז אומר את התו ה-6 במחרוזת שנקלטה: 'שפת סקרטץ', ותו זה הוא האות 'ק'.

המפעיל – 

מפעיל זה מאפשר לחבר בין שתי מילים; מחרוזות; או מחרוזת וערך והתוצאה היא מילה-מחרוזת שמורכבת משתי מילות המקור, או מחרוזת שהיא תוצאת חיבור בין מילה לערך מספרי.

קבוצת מפעילים – פעולות מתמטיות מתקדמות

קבוצת מפעילים זו משלימה את הקבוצה וכוללת 3 הוראות מיוחדות לביצוע פעולות מתמטיות מעבר ל-4 פעולות החשבון היסודיות.

המפעיל - 

המפעיל הזה מחשב את תוצאת השארית של חלוקת שני מספרים. פעולת השארית מחשבת כמה נשאר לאחר שמחלקים שני מספרים.

לדוגמה



נקבל: את הערך 1.9 מביצוע הוראת **'אמור'** הראשונה, שזו שארית החלוקה של 5.9 ב-2.
את הערך 1 מביצוע הוראת **'אמור'** השנייה, שזו שארית החלוקה של 9 ב-2.

המפעיל – 

המפעיל 'עגל' מבצע עיגול של ערכים שהוא מקבל. אם החלק העשרוני קטן מחצי הערך יעוגל לשלם הקרוב הקטן מערך המספר; אם הערך העשרוני של המספר גדול מ- 0.5 הערך יעוגל למעלה כלפי השלם הקרוב הגדול ביותר.

ולכן עבור התסריט הבא:

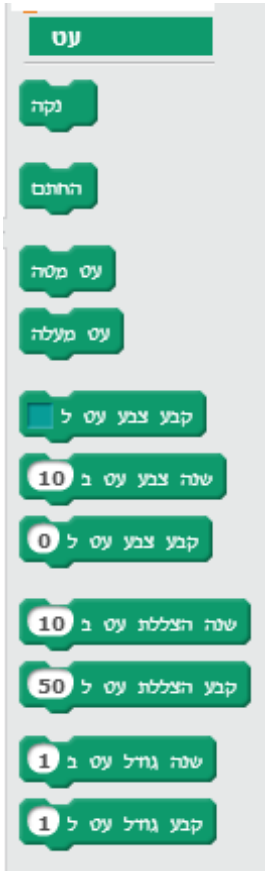


נקבל: 9 עבור המשפט הראשון
10 עבור המשפט השני והשלישי.

המפעיל – 

מפעיל זה מאפשר לבחור פונקציות חישוב שונות שחלקן הגדול נכלל במחשבונים מדעיים. הבחירה מקבוצת הפונקציות היא כמו תמיד בלחיצה על החץ הקטן בהוראה.





קבוצת ההוראות עט

משמעות ה'עט' בסביבת סקרטץ' היא שלכל דמות שנעה על הבמה יש את האפשרות לשרטט קו בעת שהיא נעה. במצב ברירת המחדל האפשרות הזו אינה פעילה ומצב העט של הדמות הוא 'עט מעלה'. כל עוד שזה מצב העט של הדמות התנועה של הדמות אינה מייצרת קו תנועה. אם מצב העט של הדמות משתנה למצב 'עט מטה' ניתן לראות שהדמות יוצרת קו בתנועתה (גם בהוראה '**קפוץ**' וגם בהוראה '**גלוש**').

קבוצת ההוראות עט, מטפלת בכל הקשור ל'עט' : הורדה והרמה של העט ; שינוי צבע העט ; שינוי עובי העט ; והוראות נוספות. מגוון הוראות זה מאפשר יצירת 'יצירות אומנות' מעניינות בשילוב מבני הבקרה 'לולאות', 'תנאים' וכלל הוראות סביבת סקרטץ'.

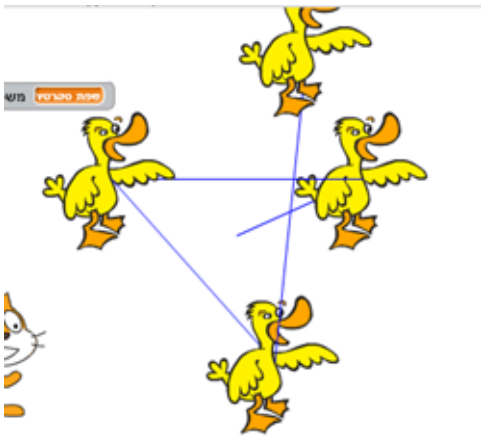
ההוראה – 

ההוראה מנקה את כל הבמה מכל הקווים שצוירו על ידי הדמויות.

ההוראה – 

ההוראה משאירה 'חותמת' על הבמה בדמות הדמות המקורית. הוראה זו פעילה כאשר מצב העט הוא 'עט מטה' או 'עט מעלה'.

דוגמה

תסריט	תמונת מסך שתתקבל
	

הברווז קפץ ממיקום למיקום וההוראה '**החתם**' השאירה את דמות הברווז בנקודות בהם הוא ביקר.

קבוצת הוראות שקובעות ושמשנות את צבע העט



ההוראה – 

הוראה זו מאפשרת לבחור צבע. לחיצה עם העכבר על ריבוע הצבע בהוראה משנה את הזמן של העכבר ל- 'יד' - , עם היד ניתן לגשת לכל צבע על המסך, לחיצה על העכבר מבצעת שינוי ובחירת צבע.

ההוראה – 

הוראה זה דומה להוראות של **'שנה ערך במשתנה'**, כלומר השינוי הוא ביחס לצבע הקודם. לכל צבע יש ערך מספרי והוספה לערך זה משנה את הצבע. התסריט הבא מדגים שימוש מעניין וכייפי של ההוראה **'שנה צבע עט'**.

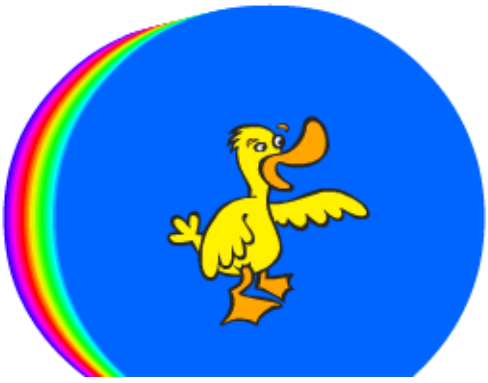


התסריט צייר קו דק ובצבעים משתנים. עם ההוראה הבאה התוצר של התסריט יהיה מעניין יותר.

ההוראה – 

ההוראה דומה להוראה **'שנה צבע עט ב'** פרט לכך שההוראה הזו משנה את גודל העט.

התסריט הבא דומה לתסריט דוגמה קודם עם תוספת של שינוי גדול.

תסריט	תמונת מסך שתתקבל
	

על הבמה אנו רואים עגול גדול המשנה את צבעו בכל מחזור בלולאה.

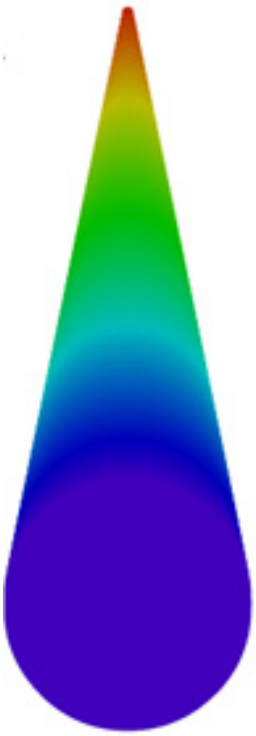
ההוראות הקובעות את גודל העט וצבעו.

ההוראה – 

הוראה זו קובעת את ערך הצבע לצבע מסוים לפי הערך בהוראה.

ההוראה – 

הוראה זו קובעת את גודל העט לפי הערך שאנו מכניסים בהוראה. עתה אנחנו יכולים לשלב בין ההוראות לקביעת ערך התחלתי של גודל העט לבין ההוראות לקביעת ערך התחלתי של צבע העט. בשילוב ההוראות לשינוי גודל וצבע העט נקבל ציור מרהיב.

תסריט	תמונת מסך שתתקבל
	

אנחנו רואים שהצבע מתחיל בצבע אדום – לצבע שמספרו 5 גודל העט ההתחלתי הוא 5, בכל מחזור בלולאה גודל העט משתנה וגם הצבע משתנה.

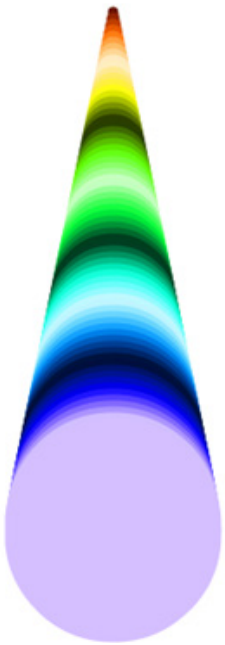
ההוראה – 

ההוראה קובעת את ההצללה של העט.

ההוראה - 

ההוראה משנה את ההצללה ביחס לערך הקודם.

נשנה עתה את התסריט מדוגמה קודמת, נשלב קביעת ערך הצללה בתחילת התסריט ושינוי דינמי בתוך הלולאה.

תסריט	תמונת מסך שתתקבל
	

סיכום הוראות קבוצת עט

קבוצת ההוראות **'עט'** מאפשרת יצירת ציורים יפים. ניתן לשלב הוראות תנועה וסיבוב ולתרגל ציור של צורות גיאומטריות. כמו כן משחקים שונים עושים שימוש בהוראה **'החתם'** ובהוראות נוספות ליצירת משחקים מעניינים בשילוב 'עט' פעיל.

נספח ג' מושגים מרכזיים בסביבת התכנות סקרטץ'

עבודה בסביבת שפת סקרטץ' כוללת שימוש בתפריטים שונים, בכלים מגוונים ליצירת דמות, ביצירת תלבושת חדשה ועוד. נספח זה כולל מושגים שונים שקשורים לשפת התכנות ולבלוקים הכלולים בה. חלק מהמושגים נסקרו בקצרה בנושא מושגים בכל פרק. המושגים שקשורים לעבודה בסביבת התכנות מתוארים כאן בלבד. נספח זה חשוב ביותר לתלמידים ולמורים בעבור עבודה איכותית ויעילה יותר בסביבת התכנות סקרטץ'.

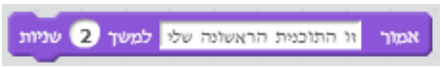
בלוק-לבנה

בלוק הוא הוראת מחשב, מפעיל, או חיישן שיש להם ייצוג ויזואלי בסביבת סקרטץ'. חלק מהבלוקים בסביבת סקרטץ' מייצגים הוראה/פקודת תכנות שמקבילה באופן מלא לפקודה שנכתבת בשפות אחרות, חלק אחר של הבלוקים בסביבת סקרטץ' שייכים לקבוצות שמרבית הבלוקים בהן הם מפעיל או חיישן. לעיתים נעשה שימוש במונח בלוק ולעיתים גם במונחים לבנה, הוראה ופקודה – כולם שמות נרדפים כשמתייחסים למרכיבים של תסריט אבל חשוב לשים לב שכאשר מדובר במפעיל או חיישן לא מדובר בדיוק בפקודת תכנות אלא, על פי רוב, לחלק מפקודת תכנות – לביטוי מתמטי; לתנאי לוגי. בלוק בסקרטץ' אינו כולל רק הוראה אלא מהווה מרכיב גרפי שמהווה חלק מסביבת תכנות ויזואלית.

דוגמה לבלוק בשפת סקרטץ'
ולתוצאה של הפעלת הבלוק עם לחיצה על ההוראה:



דוגמה: התכנית הראשונה שלי



הוראה כללית והוראה מפורטת

כאשר בוחרים הוראה באחת מקבוצת הבלוקים בסקרטץ' אנחנו בוחרים את הבלוק הכללי. הכנסת המידע הנדרש לבלוק בהתאמה לתפקיד של הבלוק בתסריט, הופכת את הבלוק מכללי לבלוק ייעודי ומפורט.

בלוק כללי –



בלוק מפורט בתסריט –



תסריט–תכנית מחשב

קבוצת בלוקים/הוראות שמאורגנת בסדר מסוים ונכתבת באמצעות שפת תכנות נקראת תכנית מחשב. בסביבת סקרטץ' אנחנו כותבים תסריטים וניתן להתייחס לכל תסריט כאל תכנית מחשב. התסריטים נכתבים באמצעות הוראות שפת סקרטץ' שמיוצגות על ידי בלוקים. שפת תכנות בה כותבים תכנית באמצעות בלוקים גרפיים שנבחרים על ידי המתכנת נקראת 'שפת תכנות ויזואלית'. פרויקט בסביבת סקרטץ' כולל את כל ההגדרות של הדמויות בתכנית וכיצד דמויות אלו נראות. כמו כן נכללים בפרויקט הצלילים, הרקעים וכל התסריטים ששייכים לכל הדמויות.

הרצת תסריט / תכנית מחשב

הרצת תכנית משמעותה ביצוע כל הוראה הכלולה בתכנית המחשב. בסביבת סקרטץ' אנחנו מריצים תסריטים שכוללים בלוקים.

מבנה סדרתי של בלוקים/הוראות

אוסף הוראות המאורגן בצורה סידרתית, כלומר הוראה עוקבת הוראה אחרת ומתבצעת בצורה זו – המחשב מבצע הוראה ועובר לבצע את ההוראה הבאה לפי הסדר. ברמה העקרונית זוהי ליבת פעילות המחשב בהרצת תכנית מחשב – שליפת הוראות מהזיכרון וביצוען לפי הסדר – רעיון בסיסי זה עוצב על ידי מדען המחשב פון ניומן בשנות ה-40 של המאה ה-20.

דוגמה: מבנה סדרתי

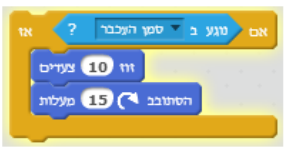


מבנה תנאי

מבנה תנאי משנה את הסדר הסדרתי של תכנית מתבצעת. כשמשתמשים בהוראת תנאי בתכנית מחשב, נבדק התנאי, אם התנאי מתקיים מתבצעת קבוצת הוראות מסוימת שחוברה על ידי המפתח וכותב התכנית; אם לא מתקיים התנאי תתבצע קבוצת הוראות אחרת (או לא יתבצע דבר). הבלוקים הבאים בסקרטץ' תומכים מימוש של מבנה תנאי. יש מבנה תנאי מצומצם שמטפל רק במצב שבו התנאי מתקיים, ויש מבנה תנאי מורחב שמטפל בשני המצבים: קבוצת הוראות למצב שהתנאי מתקיים, וקבוצת הוראות למצב שהתנאי לא מתקיים.

דוגמה: מבנה תנאי מצומצם ומבנה תנאי מורחב

מבנה תנאי מצומצם



מבנה תנאי מורחב



מבנה חוזר (לולאה)

מבנה חוזר הוא המרכיב האחרון שמשלים את המבנה הסדרתי ואת מבנה התנאי. שלושת המבנים הללו הם אבני הבסיס בארגון תכנית מחשב לפתרון בעיות פתירות (ישנן בעיות שהמחשב לא פותר). נוהגים לקרוא למבנה החוזר **לולאה** ובאנגלית LOOP. במבנה חוזר קבוצת הוראות מתבצעת מספר פעמים. מספר הפעמים שהלולאה מתבצעת וסיומה תלוי במטלה שהלולאה מבצעת בריצתה.

מבנה חוזר אינסופי – לולאה שלא מסתיימת

מבנה לא נפוץ בשימוש בתכניות מחשב – לולאה היא אינסופית והתכנית לא תסתיים לעולם. אולם בסקרטץ' מבנה זה נפוץ יותר במיוחד ביישומים של משחקים. הבלוק המממש מבנה זה:



מבנה מבוקר (חזרה לא מותנת)

מבנה זה נבחר כאשר אנו יודעים כמה פעמים הלולאה אמורה להתבצע. הבלוק המממש מבנה זה:



מבנה חוזר מותנה

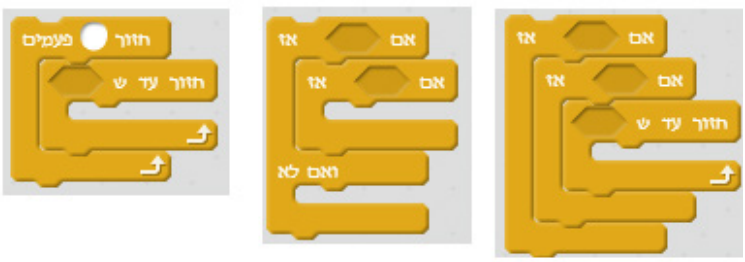
מבנה זה נבחר כאשר סיום הלולאה תלוי בתנאי מסוים והלולאה מתבצעת כל עוד שהתנאי מתקיים. הבלוק המממש מבנה זה:



קינון מבנים- nesting

כל המבנים שהוצגו כולל: סדרתי, תנאי, לולאה, ניתנים לשילוב ויצירת מעין מבנה מקונן. אין מגבלות בשילוב המבנים וכמובן שככל שאנחנו יוצרים מבנים משולבים ומקוננים, מורכבות התסריט גבוהה יותר. דוגמאות למבנה תנאי ומבנה לולאה מקוננים (ללא הוראות בתוך המבנים).

דוגמה: קינון מבני תכנות



עיבוד מקבילי

עיבוד מקבילי הוא רעיון חשוב ומרכזי במדעי המחשב ומשמעותו שמספר חישובים מתבצעים במקביל. בסביבת סקרטץ' עיבוד מקבילי מתבצע הלכה ולמעשה כאשר **בו זמנית** מבוצעים מספר בלוקים. לדוגמה אם יש כמה תסריטים המתחילים עם ההוראה '**כאשר נלחץ מקש רווח**', כולם יורצו במקביל עם לחיצה על מקש רווח. כמו כן בסביבת סקרטץ' רצים במקביל, תסריטים שונים ששייכים לאותה הדמות ותסריטים שונים ששייכים לדמויות שונות.

חשוב! סקרטץ' לא ממש מחשבת ומבצעת כמה בלוקים במקביל אולם המנגנון שתומך בתהליכים אלו בסקרטץ' יוצר מעין חיקוי לעיבוד מקבילי.

יישום

את המושג הזה אנחנו פוגשים רבות בחיי היום-יום. כל אדם מכיר את המושג בשמו 'אפליקציה' שמשמעותו: תכנית מחשב המבצעת משימה. כאשר מדובר בטלפונים חכמים וברשת האינטרנט המושג 'תכנית מחשב' הפך למושג 'אפליקציה'. כאן בספר לעיתים כאשר מתכוונים למושג 'פרויקט' משתמשים במושג 'יישום' כך שהמושגים: אפליקציה, תכנית, פרויקט ויישום הם מושגים נרדפים בספר זה.

דמות – Sprite

החתול שמופיע מיד עם תחילת העבודה בסביבת סקרטץ' על הבמה נקרא 'דמות' (באנגלית SPRITE). בסביבת סקרטץ' אנחנו כותבים תסריטים שמשויכים לדמויות שונות (וגם לרקע). ביישומי הדוגמה וביישומים שתכתוב יהיו כלולות לעיתים יותר דמויות מדמות אחת, וביישומים מורכבים תהיינה כלולות אפילו עשרות דמויות.

כתיבת תסריט בסביבת סקרטץ' זה תהליך של חיבור בלוקים ברצף, כמעט כמו בנייה בלגו. התסריט הבא מציג תסריט המניע את דמות החתול לאורך המסך ומשנה את גודל הדמות. בלוק ראשון מזיז את הדמות מספר צעדים, הבלוק השני משנה את גודל הדמות והבלוק השלישי יוצר המתנה של שנייה. שלושת הבלוקים נכללים בבלוק המאפשר ביצוע חוזר של 10 פעמים של שלושת הבלוקים.

דוגמה: קטע מתסריט המשלב מספר בלוקים

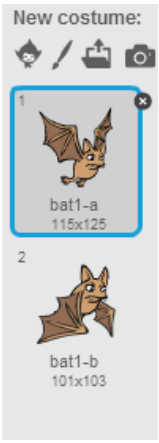


תלבושת – Costume

תלבושת מהווה מראה שונה גרפית של דמות, לכל דמות יש תלבושת אחת או יותר. במהלך ביצוע התסריט ניתן לשנות ולהחליף את מראה הדמות-תלבושת על-פי האירועים המתרחשים. לדוגמה אם כתבת פרויקט של SPACE INVADER (פלישה מהחלל) וחללית האויב נפגעת בקרב, יתכן שתרצה להציגה עולה באש או מתפרקת לחלקים, זה בדיוק המצב ליצור לחללית תלבושות שונות. תלבושות שונות מאפשרות ליצור אנימציה שנוצרת כתוצאה משינוי תלבושת בפרקי זמן קצרים. סביבת סקרטץ' 2.0 מספקת לחלק מהדמויות כמה תלבושות וניתן להשתמש בתלבושות אלו ללא המאמץ ליצור אותם.

כאשר אנחנו בוחרים את דמות העטלף אנחנו מקבלים שתי תלבושות לדמות העטלף שנקראת BAT1.

דוגמה: טיפול בתלבושות של דמות



תלבושת ראשונה נקראת BAT-1A והשנייה BAT-1B

התסריט הבא מבצע מעבר בין התלבושות שיוצר אנימציה של עטלף המנפנף בכנפיו.

דוגמה: תסריט חילוף תלבושות

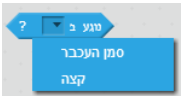


במה

הבמה זה האזור בו מתרחשת כל הפעילות של הפרויקט. באזור זה הדמויות ממוקמות ופועלות. לבמה יש גודל שניתן לשנות (להגדיל ולהקטין). הבמה ממופה עם מיקומים כמו במערכת קטרזית, או במילים פשוטות יותר לכל נקודה בבמה יש שני ערכים של מיקום אופקי ואנכי, ובשפת המתמטיקה לערך של המיקום האופקי אנו קוראים X ולערך של המיקום האנכי אנו קוראים Y.

מסגרת (קצה)

לבמה בסביבת סקרטץ' יש מסגרת, כלומר גבולות. בלוקים שונים בסקרטץ' יכולים להתייחס למסגרת כמו הבלוק ששייך לקבוצת החיישנים.



הבלוק מאפשר לבדוק האם דמות נוגעת בקצה הבמה (באחד מגבולות מסגרת הבמה). דוגמה נוספת להוראה המתייחסת למסגרת.



ערך בציר X של העכבר

בכל רגע נתון של העכבר על הבמה יש לו ערך X נוכחי שאנו יכולים להתייחס אליו ולהשתמש בו בתסריט.

ערך בציר Y של העכבר

בכל רגע נתון של העכבר על הבמה יש לו ערך Y נוכחי שאנו יכולים להתייחס אליו ולהשתמש בו בתסריט.

ערך בציר X של דמות

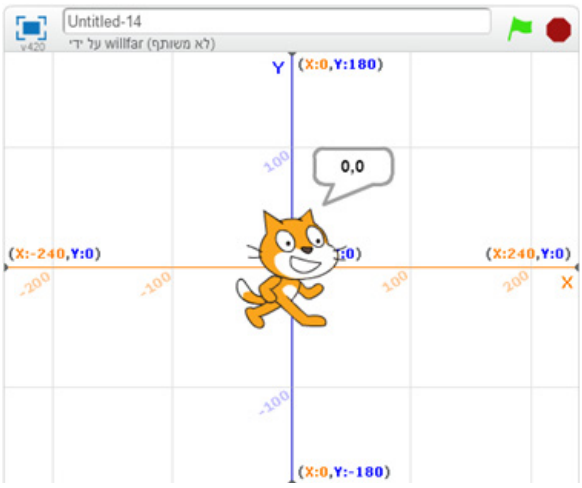
בכל רגע נתון לכל דמות על הבמה יש ערך X נוכחי שאנו יכולים להתייחס אליו ולהשתמש בו בתסריט. חשוב לציין שערך ה X- מתייחס למיקום של נקודת מרכז הדמות.

ערך בציר Y של דמות

בכל רגע נתון לכל דמות על הבמה יש ערך Y נוכחי שאנו יכולים להתייחס אליו ולהשתמש בו בתסריט. חשוב לציין שערך ה Y- מתייחס למיקום של נקודת מרכז הדמות.

דוגמה: במה ודמות החתול במרכזה

האיור מציג את הבמה ודמות החתול ממוקמת במיקום X=0 Y=0. אנחנו יכולים לראות שהמיקום הזה מתייחס למרכז הדמות. תסריט פשוט מורה לדמות לומר את מיקומה על הבמה.



כיוון

לכל דמות יש כיוון אליו הדמות מכוונת. במידה והיה מדובר רק בדמות חץ היה קל להבין את הרעיון של כיוון הדמות, אבל גם לדמות פרפר, דג ויונה יש כיוון וכאשר יש הוראה לדמות לזוז 

הדמות תנוע בכיוון הנוכחי שלה.

בחלון המידע על הדמות ניתן לראות את כיוונה בכל רגע נתון 

וכן במשתנה 'כיוון' תמיד יש את הכיוון הנוכחי של הדמות 

.

בלוקים, במיוחד בקבוצת הבלוקים תנועה, מספקים מידע על כיוון הדמות:

הבלוקים **'הסתובב ימינה'** 

או **'הסתובב שמאלה'** משנים את כיוון הדמות **יחסית לכיוונה** לפני ביצוע ההוראה, והבלוק **'פנה לכיוון'** 

משנה את כיוון הדמות באופן מוחלט ובלתי תלוי בכיוון הדמות לפני ביצוע הבלוק.

מסרים ותקשורת בין דמויות

דמויות בסביבת סקרטץ' "מדברות" ביניהן. הדמויות מעבירות מסרים. מסר הוא הודעה המורכבת מאוסף תווים (אותיות, ספרות וסימנים). המסרים משמשים להעברת מידע וכל דמות (וגם רקע), יכולה להשתמש בבלוקים: **'שדר מסר'**; **'כאשר התקבל מסר'**. דמות משדרת מסר לכל שאר הדמויות. כל דמות ששייך לה תסריט המתחיל באירוע **'כאשר התקבל מסר'** מבצעת תסריט בעקבות קבלת המסר. המסרים מאפשרים תיאום ובקרה של ריצת תסריטים בין

דמויות שונות, יצירת מצבים בהם דמויות מסוימות יחלו יחד לבצע תסריט וכדומה.

מספר אקראי

בלוק מפעיל שיוצר מספר אקראי בתחום שאנו בוחרים.

משתנה/רשימה

משתנה הוא מעין "קופסה" בזיכרון המחשב שניתן לשמור בה מידע. הערך במשתנה משנה את ערכו כתוצאה מביצוע בלוקים שמאפשרים לשנות את ערך המשתנה. במשתנה ניתן לאחסן מספרים וגם אוסף תווים (מחרוזת). בסביבת סקרטץ' בקבוצת ההוראות '**נתונים**' ניתן ליצור משתנים ולהשתמש בבלוקים נוספים לטיפול במידע הנמצא במשתנים. סקרטץ' מאפשרת ליצור משתנים שקשורים לדמות מסוימת (לדוגמה עוצמת הדמות), וגם להגדיר משתנה שמוכר על ידי כל הדמויות (לדוגמה מספר הנקודות שצבר משחק במשחק). לכל משתנה יש שם ייחודי שאנחנו נותנים לו בתהליך היצירה. ניתן גם ליצור רשימה שמהווה קבוצת משתנים בשם כולל (על "רשימה" יורחב בפרק ההעשרה למתקדמים).

שמות משתנים

לכל מרכיב בפרויקט סקרטץ' יש שם: לפרויקט יש שם; לדמות יש שם; לתלבושת יש שם; למשתנה יש שם; לרקע יש שם.

השמות מזהים את מרכיבי הפרויקט, הם מבדילים ומייחדים כל מרכיב. סביבת סקרטץ' נותנת באופן אוטומטי שם למרבית המרכיבים ברגע שהם נוצרים (מלבד משתנה או רשימה שהמשתמש יוצר ונותן להם שם).

חשוב לחשוב על השם שניתן וכמובן ניתן לשנות שמות שסביבת העבודה יצרה. יש חשיבות לתת לשם משמעות שקשורה לתפקיד של המרכיב. למשחק מרוץ מכוניות אפשר לקבל את השם שהסביבה יצרה כמו: UNTITLED-14 או לתת שם משמעותי כמו 'מרוץ מכוניות'. לשם משתנה שתפקידו לשמור כמה נקודות צבר שחקן אפשר לתת את השם X או את השם POINT, ברור איזה שם מצביע ומרמז מה תפקידו של המשתנה. אפשר לתת שמות של משתנים בעברית או באנגלית. מומלץ לא להשתמש בשתי שפות באותו פרויקט.

צלמיות מייצגות

סביבת סקרטץ' מאוד ידידותית. הצלמיות בסביבה מקלות על המשתמש לדעת את תפקידן. נציג את הצלמיות החשובות (שמוסברות בפרוטרוט בחלקים שונים של הספר):

 - מייצגת מקור לדמות חדשה מהספרייה של סקרטץ'

 - מייצגת מקור לרקע חדש מהספרייה של סקרטץ'

 - מייצגת מקור באמצעות שימוש בעורך הגרפי (דמות, תלבושת, רקע)

 - מייצגת מקור מקובץ מהמחשב (דמות, תלבושת, רקע, צליל)

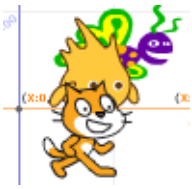
 - מייצגת מקור ממצלמה (דמות, תלבושת, רקע)

 - מייצגת מקור לצליל חדש באמצעות הקלטה ממיקרופון

 - מייצגת מקור לצליל חדש מהספרייה של סקרטץ'

שכבה

בפרויקט פשוט יש לנו 1-3 דמויות. בפרויקטים מורכבים ישנן יותר דמויות. כאשר הדמויות מוצגות על הבמה חלק מהדמויות נוגעות אחת ברעותה באופן שדמות או חלק מדמות מסתירה דמות אחרת. כשיוצרים פרויקט עם שלוש דמויות מגלים שהדמות בה מטפלים הופכת להיות הדמות שמסתירה את האחרות. במינוח מקצועי אנו אומרים שהדמויות מסודרות בשכבות, כל דמות ממוקמת בשכבה מסוימת וכל דמות שנמצאת בשכבה "מעל" דמות אחרת, מסתירה אותה.



בלוקים שונים מטפלים בנושא השכבות ויכולים לשנות במהלך תסריט את מיקום השכבה של דמות ביחס לדמויות אחרות.

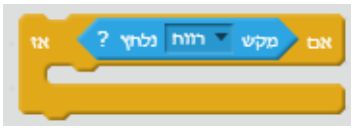
אנחנו רואים שהפרפר נמצא בשכבה התחתונה ביותר ולכן הוא מוסתר על ידי שתי הדמויות האחרות; גודו נמצא בשכבה השנייה; והחתול נמצא בשכבה העליונה ואף דמות לא מסתירה אותו.

החזרת ערך

רעיון מרכזי וחשוב טמון במושג 'החזרת ערך' – הוראות מקבוצות חיישנים, מפעילים ומקבוצות נוספות הן בחלקן אינן הוראות לביצוע פעולה אלא ביטויים לביצוע חישוב, ואנו משתמשים בחישוב להמשך ביצוע ההוראה.

לדוגמה החיישן 

בודק האם נלחץ מקש כלשהו (לפי בחירתנו), ביצוע הבדיקה מתבצע בחלקיק שנייה והתשובה לשאלה שלנו היא **אמת** או **שקר**, כלומר '**נלחץ מקש**' רווח או '**לא נלחץ**'. לתהליך של העברת תוצאת ביצוע החישוב שמשיב לשאלה שלנו אנו קוראים 'החזרת ערך' – הערך שאנחנו מקבלים מהחיישן משולב בהוראה / ביטוי, לדוגמה:



החזרת ערך מתבצעת בתהליך בדיקה של חיישן או מהפעלת מפעיל לדוגמה:



גם בדוגמה זו מתבצע תהליך "נסתר" שבו אנחנו מפעילים מפעיל מסוים "שיודע" ליצור מספר אקראי בתחום מבוקש, ובסיום הפעולה של המפעיל הוא "מחזיר" לנו את הערך ואנחנו משלבים את הערך בהוראה כלשהיא. לדוגמה:



בכל פרקי תכנית הלימודים נשתמש בביטוי 'החזרת ערך' כדי להציג את התהליך של הפעלת חיישן או מפעיל וקבלת ערך – מספרי, לוגי, מחרוזת – בו נשתמש להמשך התסריט.

בלוקים (לבנים) של המשתמש

כתיבת תסריט שניתן לו שם מיוחד ובכך נוצר בלוק חדש שמופיע בקבוצת ההוראות 

הרחבה למושג זה והיישום של הרעיון בפרק נושאים מתקדמים.

זימון / הפעלת בלוק משתמש

שילוב הוראה שנכתבה על ידי המשתמש נקרא '**זימון**' כי התהליך שמתרחש הוא הפסקת התסריט שמבוצע ומעבר אל תסריט אחר שנושא את הכותרת של הבלוק שזומן. עם סיום התסריט הכלול בהוראה של המשתמש, התסריט המקורי ממשיך מהוראה לאחר הוראת הזימון.

המחשת המושג **'זימון'**:

א. ניצור בלוק משתמש פשוט שנקרא לו זימון



ב. התסריט הראשי, כולל הוראה ש"מזמנת" את הבלוק



בסיום ביצוע הבלוק **'זימון'**, חוזרת השליטה לתסריט הראשי והוא ממשיך להתבצע הוראה לאחר הוראת הזימון (המחשת הזרימה של הביצוע).

איור: תהליך ביצוע זימון



פרמטרים וארגומנטים

תכונה חשובה נכללת במנגנון כתיבת בלוקים של המשתמש – בעת זימונו אפשר להעביר לבלוק שיצרנו ערכים של מספרים, מחרוזות, וערכים לוגים. באופן הזה הבלוק עובד שונה בכל פעם שהוא מזומן והתכונה הזו מאפשרת להגדיר בלוקים עם גמישות. לערכים שאנו מעבירים קוראים **ארגומנטים**, ולמשתנים בבלוק שהוגדר אנו קוראים **פרמטרים.**

הערכים (הארגומנטים) שאנחנו מעבירים לבלוק מאוחסנים במשתנים (הפרמטרים) שמוגדרים בתוך הבלוק. משתנים אלו מוכרים רק בעת הרצת הבלוק ואינם מוכרים מחוץ לבלוק!

דוגמה בסיסית להמחשת רעיון העברת ארגומנטים לבלוק אישי:

נגדיר בלוק שמקבל מחרוזת ומספר (כלומר שני פרמטרים) הבלוק מציג את המחרוזת מספר פעמים לפי ערך המספר.

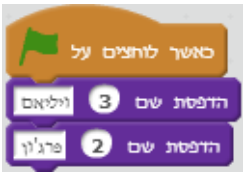
דוגמה: בלוק אישי – הדפסת שם



העובדה שהגדרנו את שני הפרמטרים הללו – משתני הבלוק – מאפשרת לנו לזמן הדפסת שם עם שני ערכים שונים בכל זימון.

תסריט עם שני זימונים והעברת ערכים (ארגומנטים) שונים לבלוק:

דוגמה: זימון בלוק



פרוצדורה / שגרה / רוטינה / שיטה

בסביבת סקרטץ' כאשר מוגדרים בלוקים חדשים על ידי כותב התוכנה הם מופיעים בקבוצת הוראות שנקראת **'לבנים'** – הרעיון של כתיבת ו"המצאת" הוראה חדשה שלאחר מכן ניתן להשתמש בה בתסריטים שונים הוא כלי רב עוצמה שבו אנחנו יכולים להרחיב את שפת התכנות בה אנו משתמשים; ליצור הוראה שכוללת קבוצת הוראות; ניתן לזמן את ההוראה מכל מקום בכל תסריטי הדמות וזאת עם היכולת להעביר ערכים (ארגומנטים) שונים בעת הביצוע. זה אכן כלי חשוב בהיבט שהוא מסייע לתכנת תכנות מודולארי ומאפשר חסכון בכתיבת קוד חוזר. הכלי הזה נקרא בשפות שונות ולאורך שנות התפתחות שפות תכנות בשמות שונים: פרוצדורה, שגרה, רוטינה, פעולה ובשנים האחרונות המילה **שיטה** הפכה להיות מושג נפוץ יותר מהמושגים האחרים.

חשוב להקפיד לא להשתמש במילה **שיטה** או כל מילה אחרת ולהישאר צמוד להגדרה יותר ארוכה ומסורבלת כמו: בלוק חדש שמוגדר על ידי המשתמש (או לבנה) והסיבה היא פדגוגית – להישאר עם המונח בלוק או לבנה שהוא מונח שקשור למהות סביבת סקרטץ' כשפה ויזואלית.

בלוק בהגדרת משתמש או **שיטה**; **בלוק** בהגדרת משתמש או **פרוצדורה**: המשמעות זהה ומתייחסת לרעיון מרכזי בשפות תכנות – האפשרות להרחיב את השפה ולתרום בכך לתוכנה שנכתבת באופן ברור יותר; שניתנת לתחזוקה; שמאפשרת ושתורמת לעבודת צוות; שחוסכת שורות קוד, זמן ושגיאות.

נספח ד' סביבת התכנות סקרטץ': תפריטים, תגיות, כלים

מבוא

סקרטץ' אינה רק שפת תכנות, זוהי סביבת למידה רב תחומית שדורשת מהילדים ואף מהמבוגרים שליטה במיומנויות המאה 21. כתיבת יישום בסביבת סקרטץ' מחייבת קביעת יעדים, תכנון, שימוש בכללים וכלים להשגת היעד. לעיתים קרובות נדרשת עבודה בניהול המשימה, בחלוקת המשימה לתתי משימות ובשיתוף קבוצה גדולה של תלמידים לביצוע תתי המשימות. ביצוע תתי המשימות דורש בשלב הבא את חיבורם וארגונם מחדש של מרכיבי המשימות לכלל המוצר הסופי.

בשלבי הביצוע של המשימה נדרשים התלמידים לחפש תמונות, לעצב דמויות בעורכים גרפים, להעביר את המידע בין חברי הצוות תוך שימוש בכלים לתקשורת כולל בדואר אלקטרוני, במחשוב ענן, ברשתות חברתיות ובכלי שיתוף רבים נוספים. כל המטלות הללו דורשות יישום של מיומנויות המאה 21. סביבת סקרטץ' מעודדת פיתוח ויישום מיומנויות אלו בצורה טבעית ולא מלאכותית, וללא דרישות מיוחדות מסביבת הלמידה – רק כמה מחשבים, חיבור אינטרנט, חבורת ילדים סקרניים ומורה מנחה ומוביל שמאפשר ונותן הזדמנויות ופחות מלמד ומעביר ידע. משימה בסקרטץ' כוללת לא רק תמונות ואיורים – ניתן לשלב בה צלילים שנבחרו מתוך סביבת סקרטץ' או צלילים שנוצרו על ידי משתתפי הפרויקט.

השילוב של כתיבת קוד בתהליכים משותפים; של שימוש ויצירת תמונות ודמויות גרפיות; של שילוב צלילים ויצירת צלילים, יוצר מרחב בו ילדים נדרשים ליצירתיות ונותנים דרור לדמיונם. מחנך שנכנס לכיתה לאחר חופשת קיץ ארוכה או סוף שבוע ארוך ומבקש מהתלמידים לספר לו על חופשתם מתחיל תהליך של יצירה ומתן מרחב לתלמידים לכתוב משהו אישי במקום הכתבה של טקסט, וכשנוסף על כך הוא מבקש מהתלמידים לעשות זאת באמצעות סביבת סקרטץ' המורה נותן דרור לתלמידים, דרור ליצירתיות ומקוריות. מורה למדעים שמבקש מתלמידיו לבנות הדמיה של התפרצות הר געש בסביבת סקרטץ' נותן הזדמנות

ללמידה אחרת, ללמידה משמעותית. בביצוע המטלה ליצירת סיפור בתמונות, צלילים ואנימציות על חופשת הקיץ, המחנך מאפשר לכל תלמיד להשתתף בביצוע המשימה, ומורה המדעים, במשימת הר געש, מאפשר לסגנונות למידה שונים לבוא לידי ביטוי ומתרגל למידת עמיתים פעילה. משימות אלו הן מתן הזדמנות לגעת באמנויות שונות: ביכולת לאייר, ביכולת לחבר מספר תמונות וליצור אנימציה, ביכולת ליצור רצף אירועים שממומשים בתמונות וצלילים לכלל סיפור או הדמיה ייחודיים. יכולות אלו מגוונות ונוגעות לכל התלמידים גם לאלו שמתקשים בכתיבה, שאינם עונים על שאלות בכיתה כי הם פוחדים לטעות – אין בסביבת סקרטץ' מושג של נכון או לא נכון אבל יש מרחב יצירתי בו כל תלמיד שישתתף בביצוע המטלה יעשה זאת לפי דרכו ויעשה שימוש באינטליגנציות אישיות בהם יש לו יתרון, והתוצאה תיתן לו תחושה של הצלחה ושל מסוגלות.

הנספח הזה מציג את סביבת העבודה ואת המסכים והתפריטים החשובים בהם אנו משתמשים בתהליכי תכנון ויישום פרויקטים.

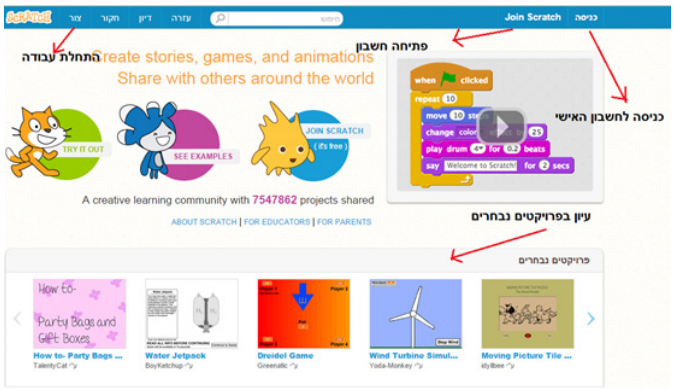
סביבת העבודה

תכנית הלימודים מבוססת על גרסת סקרטץ' 2.0, גרסה זו מאפשרת להשתמש בסביבה ישירות באמצעות דפדפן האינטרנט שעל מחשבך. אין חובה להתקין את סביבת העבודה. יתרון נוסף הוא שניתן לשמור את כל היישומים שנכתבים באתר בחשבון פרטי – מערכת של פתיחת החשבון וטיפול בקבצים אלו כולל שליפה ושמירה מהחשבון, נכללת בסביבת הפיתוח וכך היא מאפשרת לשתף את הפרויקטים שלך בקהילה גדולה (מעל 3,000,000 שותפו נכון להיום).

התחלת עבודה בסביבת סקרטץ' מתחילה בכניסה לאתר של סקרטץ' 2.0

בכתובת: HTTPS://SCRATCH.MIT.EDU

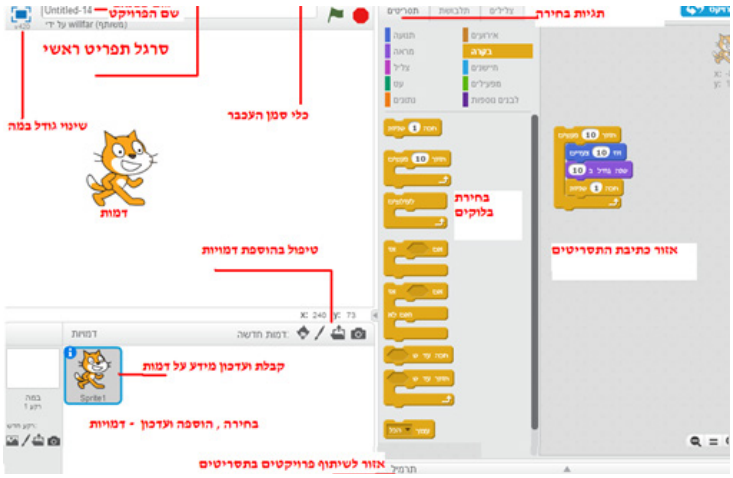
המסך כניסה למערכת שתקבל:



הערה: יתכן ומראה מסמך העמוד הראשון ישתנה. ישנם שינויים רבים ועדכונים באתר וכמובן יתכנו שינויים בעמוד הכניסה.

בחירה באפשרות **צור** תכניס אותך לממשק העריכה וההפעלה של פרויקטים בסביבת סקרטץ'.

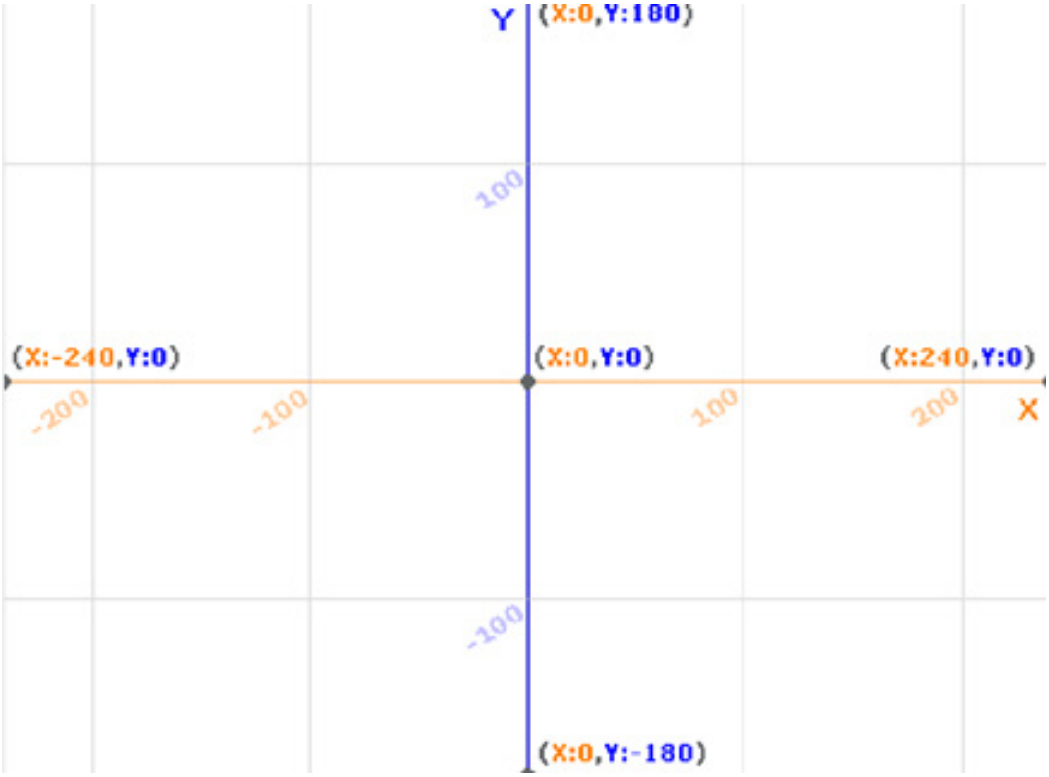
איור: ממשק עבודה ועריכת פרויקטים



הבמה

הבמה מהווה את האזור המרכזי בו אנחנו רואים את תוצאות הפעלת התסריטים. לבמה יש מיפוי נקודות שמוגדר באמצעות מערכת קרטזית, עם אורך הנמדד בפיקסלים (נקודות גרפיות).

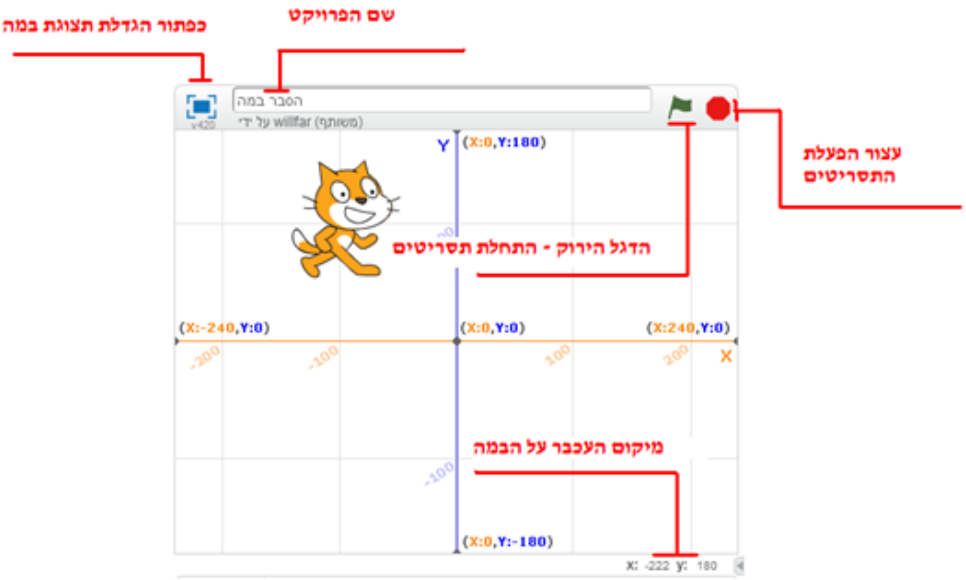
המערכת הקרטזית – מערכת מישור עם ציר X וציר Y מגדירה את המיקומים של הדמויות ומרכיבים נוספים שמופיעים על הבמה.

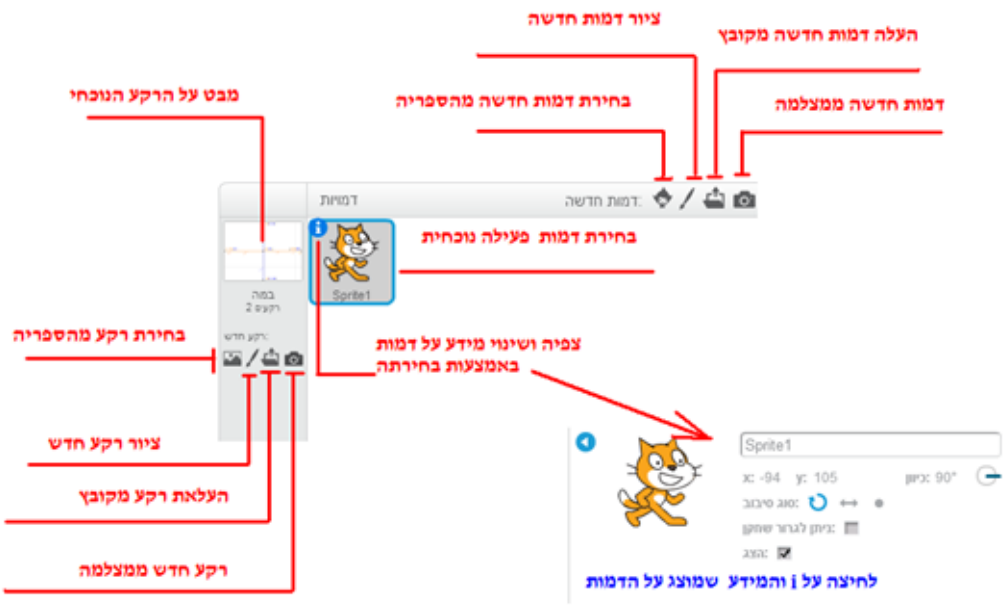


מיקומים על הבמה

נושאי המיקום של דמויות על הבמה ותנועה על הבמה באמצעות התייחסות למיקומי X ו- Y אינם משימה פשוטה, במיוחד לילדי בית ספר יסודי, אולם ישנם תהליכי למידה מקבילים מסייעים. הילדים המשתמשים בסביבה ובהוראות המתייחסות למיקומים ולומדים בדרך פעילה ובמרבית המקרים מהנה את מערכת הצירים.

אזור 'שליטה' על הבמה





בחירה בדמות

באזור יצירה וטיפול בדמויות יש לנו בצורה ממוזערת (THUMBNAILS) את כל הדמויות בפרויקט. הקלקה על איור ממוזער של דמות הופכת את הדמות לפעילה ואנחנו יכולים לטפל בדמות בכמה הקשרים:

- כתיבה ועריכת תסריטים של הדמות
- עריכה והוספת תלבושות לדמות
- עריכה והוספת צלילים לדמות
- שינוי שם הדמות
- קביעת סוג הסיבוב של הדמות
- הצגה/הסתרה של הדמות

בחירת הדמות (לחיצת עכבר על הדמות) יוצרת סביב הדמות מסגרת כחולה שמעידה שזו הדמות שבה ובמרכיביה אנו מטפלים.

איור: הדמות הפעילה



איור: לשוניות טיפול בדמות



בחירת דמות ולחיצה על הכפתור הימני בעכבר מציגה את התפריט הבא:

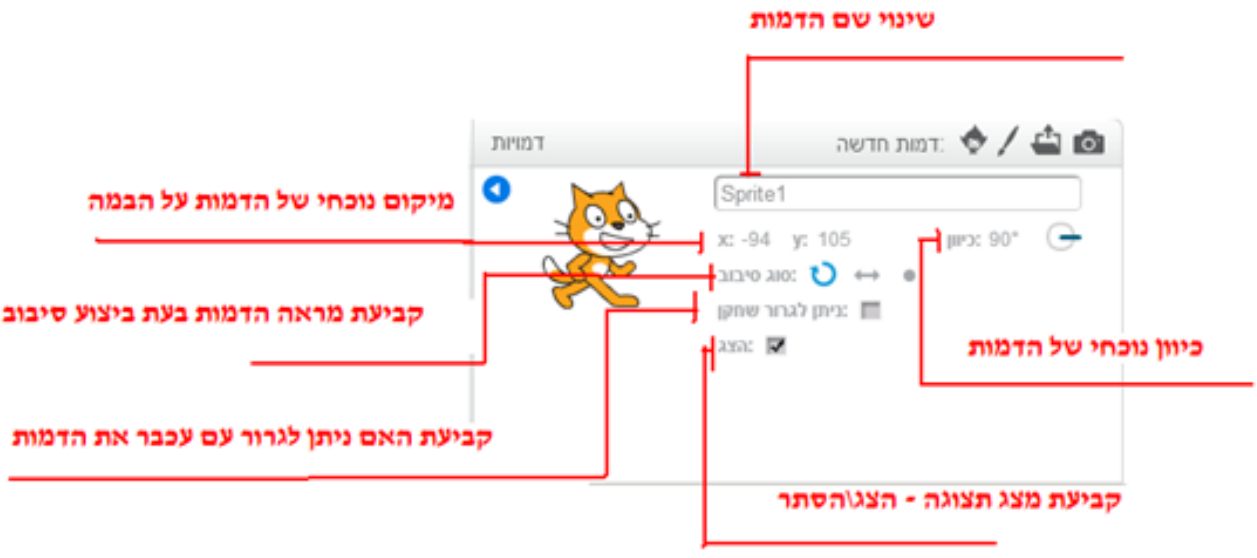


התפריט מאפשר:

- לעיין במידע על הדמות – INFO
- לשכפל את הדמות – קבלת דמות זהה נוספת עם שם אחר. אם לדמות המקור יש תסריטים גם הם יהיו אצל הדמות החדשה
- למחוק את הדמות ובכך למחוק את כל מה ששייך לדמות (תסריטים, תלבושות, צלילים, משתנים)
- לשמור לקובץ מקומי – שמירת הדמות כקובץ על המחשב
- הסתר – שינוי מצב הדמות ממצב 'הצג' למצב 'הסתר'

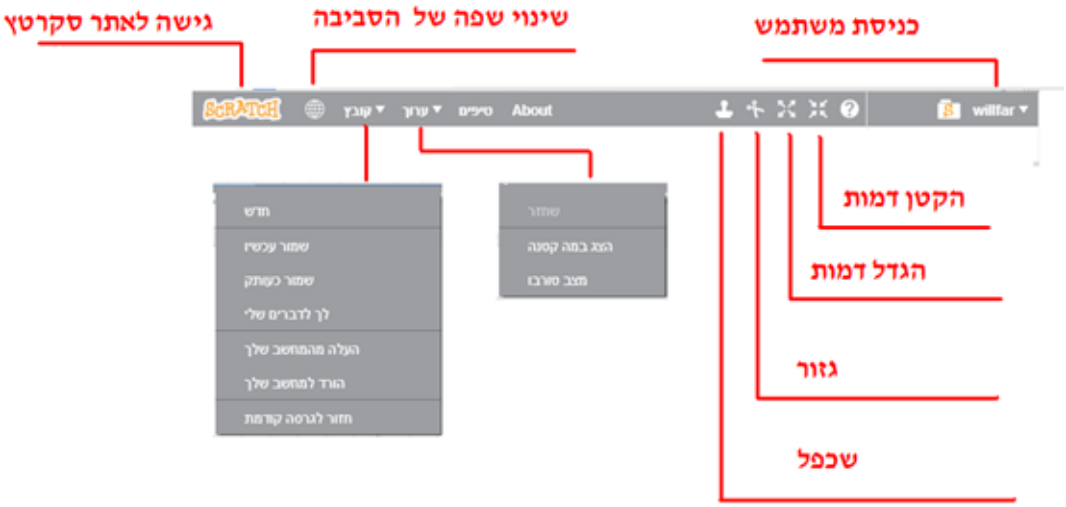
עיון במידע על הדמות – לחיצה על i

איור – אזור המידע על דמות



התפריט הראשי

איור תפריט ראשי



חשוב! מראה תפריט קובץ ותפריטים אחרים תלוי באם המשתמש מחובר לחשבון שלו, אינו מחובר או שאין לו חשבון

תפריט קובץ – אפשרויות

- חדש – פתיחת פרויקט חדש
- שמור עכשיו – שמירה מיידית של הפרויקט במקום האחסון הפרטי באתר
- שמור כעותק – שמירת עותק נוסף באתר
- לך לדברים שלי – גישה לפרויקטים בחשבון של המשתמש
- העלה למחשב שלך – טעינת פרויקט מהמחשב המקומי עליו המשתמש עובד
- הורד למחשב שלך – שמירת הפרויקט על המחשב המקומי של המשתמש
- חזור לגרסה קודמת – חזרה לגרסה הקודמת של הפרויקט

תפריט עריכה – אפשרויות

- שחזר – ביטול פעולות המחיקה האחרונות שבוצעו
- הצג במה קטנה – הצגת הבמה בצורה מוקטנת ומתן יותר מקום על המסך לתסריטים
- מצב טורבו – חלק מהבלוקים יפעלו יותר מהר במצב טורבו

העורך הגרפי

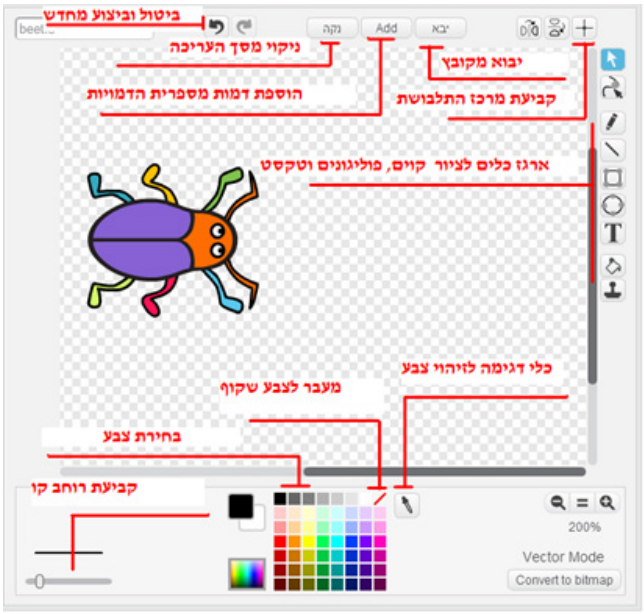
עריכה גרפית היא מטלה מרכזית בעיצוב יישומים אטרקטיביים. אנו נדרשים לערוך גרפית דמויות, תלבושות של דמויות ורקעים. ניתן לערוך כל מרכיב בסביבת העבודה (דמות, תלבושת, רקע) עם עורכים גרפים מתקדמים שאינם כלולים בסביבת העבודה כמו תוכנות עריכה גרפית שמשמשות גרפיקאים מקצועיים.

סביבת העבודה של סקרטץ' מספקת עורך גרפי שמאפשר גם לחסרי ניסיון בעיצוב גרפי לערוך את המרכיבים של הפרויקט שלהם.

מצב עריכה גרפית

חשוב! בעבודה במצב גרפי זכור לגשת לתפריט הראשי ולבחור באפשרות **ערוך** את 'שינוי מצב הבמה' ל- 'הצג במה קטנה' כך תוכל לראות את כל אפשרויות העורך הגרפי.

איור: העורך הגרפי



הסבר על אפשרויות מיוחדות בעורך הגרפי

קביעת מרכז התלבושת -

לכל תלבושת ניתן לקבוע מרכז (מעין "מרכז כובד גרפי"), סביב המרכז הזה התלבושת מבצעת את הוראות הסיבוב. במידה ולא יתבצע ולא ייקבע מרכז מתאים של התלבושת יתכן מצב בו הדמות תבצע סיבוב אולם הסיבוב יכלול לכאורה גם תזוזה של הדמות ממקומה, כאשר בפועל לא התבצעה הוראת זוז אלא רק הוראת סיבוב – קביעת מרכז התלבושת לא במיקום הנכון יוצרת תנועה שעלולה לפגוע בביצוע התסריט לפי המטרות של כותב התסריט.

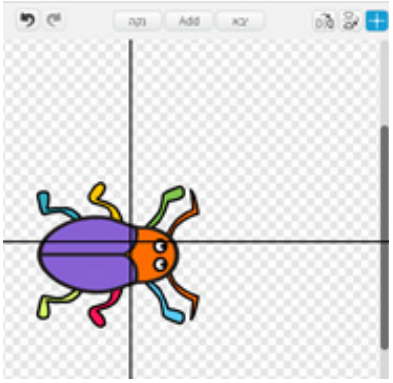
ביצוע קביעת מרכז התלבושת

א. יש לבחור את הצלמית של קביעת מרכז התלבושת – ממוקמת בקצה הימני עליון של מסך העריכה

ב. על המסך מופיעים שני קווים אופקי ואנכי שנחתכים בנקודה אותה אנו יכולים לשנות על ידי תנועת העכבר (שהסמן שלו השתנה לסימון של שני קווים אנכיים), כאשר מגיעים למרכז הרצוי מסיימים את התהליך.

האיור הבא מציג את בחירת מרכז התלבושת.

איור: קביעת מרכז התלבושת



כלי דגימה לזיהוי צבע –

הכלי מסייע לבחור צבע מתאים למטרות של המשתמש. לא תמיד אנחנו יכולים לבחור ממערכת הצבעים צבע שקולע למטרה שלנו, אבל אם אנחנו צריכים לבחור צבע שקיים במסך העריכה נוכל לבחור את הצבע עם הכלי לדגימת הצבע.

כלי מעבר לצבע שקוף –

כאשר שתי דמויות ממוקמות באותו מיקום או שהן קרובות זו לזו, אחת הדמויות מכסה חלק מהדמות שמתחת או שהדמויות גם מכסות חלק מהרקע של הבמה. אבל אם רוצים לראות כיצד הבמה נראית מאחורי הדמות יש להשתמש בעורך הגרפי כדי להפוך חלק מהדמות לשקוף.

לחץ בעורך הגרפי על הריבוע הקטן עם הקו האלכסוני האדום בכדי להפוך מרכיב בדמות לבלתי נראה ושקוף, עתה תוכל באמצעות בחירת הצבע השקוף והכלי של מילוי להפוך חלק מהתלבושת לשקופה, כך הרקע או דמות אחרת מתחת לא תוסתר.

התרמיל

התרמיל מהווה תכונה חדשה שכלולה בגרסת סקרטץ' 2.0. בתרמיל ניתן לשמור תסריטים של דמויות ולהשתמש בתסריטים אלו בפרויקטים אחרים. תהליך ההעברה של תסריט לתרמיל ושליפת תסריט מהתרמיל פשוט ומבוסס על בחר-גרור-שחרר.

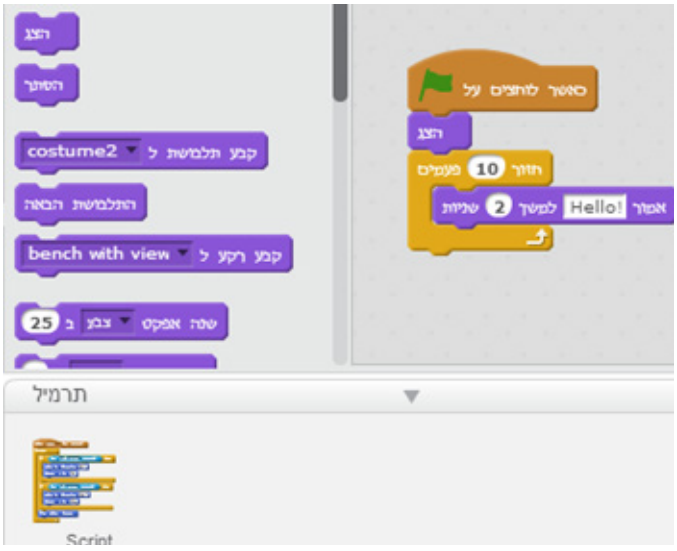
העברה לתרמיל:

1. בחר בדמות שלה שייך התסריט שברצונך לשמור בתרמיל.
2. כנס ללשונית תסריטים של הדמות ובחר את תסריט המטרה.
3. גרור את התסריט אל התרמיל ושחרר את התסריט – דמות ממוזערת של התסריט תופיע בתרמיל.

העברה מהתרמיל אל תסריטי דמות מסויימת:

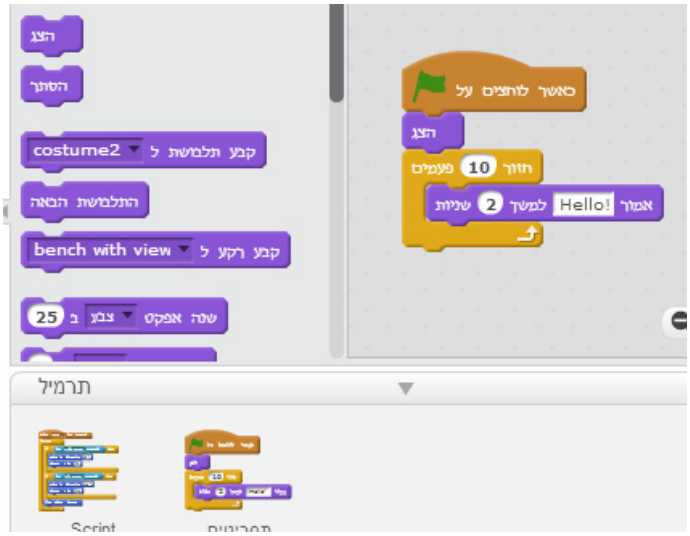
1. בחר את הדמות לה אתה מעוניין להעביר את התסריט.
2. כנס לאזור התסריטים של הדמות.
3. בחר מהתרמיל את התסריט הנדרש.
4. גרור ושחרר את התסריט אל אזור התסריטים של הדמות – התסריט הועבר.

איור: מצב התרמיל לפי העברת תסריט מדמות החתול



כעת אתה גורר את התסריט הנכלל בתסריטי הדמות 'חתול' אל התרמיל, לאחר פעולה זו מצב התרמיל:

איור: מצב תרמיל לאחר פעולת העברה



שים לב! העברת התסריט לתרמיל היא רק העברה של העתק התסריט, התסריט המקורי עדיין נשאר כלול בתסריטי הדמות.

איך יוצרים תסריט חדש

יצירת תסריט היא פשוטה ביותר (מההיבט הטכני). כל שעלינו לעשות הוא הצעדים הבאים:

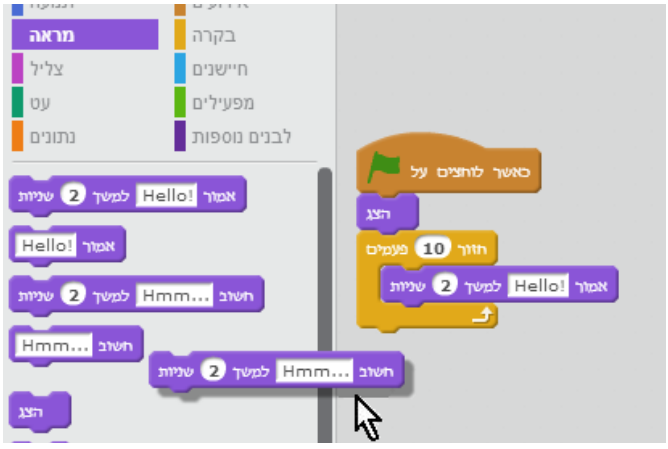
1. בחר את הדמות לה אתה מעוניין להוסיף תסריט חדש.
2. בחר את הלשונית 'תסריטים' והפוך אותה לפעילה.
3. בחר את קבוצת הבלוקים ממנה אתה מעוניין לבחור בלוק (הוראה).
4. גרור את הבלוק בו אתה מעוניין אל חלון התסריטים.

כל הוספת הוראה/בלוק דורשת את הצמדת ההוראה החדשה למיקום הנכון לפי בחירתכם.

השלבים לביצוע הוספת הוראה/בלוק לתסריט

1. בחר את התסריט בדמות לה הינך מעוניין להוסיף בלוק.
2. בחר את קבוצת הבלוקים ממנה ברצונך לבחור בלוק.
3. בחר את הבלוק שברצונך להוסיף.
4. גרור את הבלוק אל המיקום המדויק בתסריט ושחרר את הבלוק, ודא שהבלוק אכן נצמד לתסריט במיקום הנכון.

איור: בחירת בלוק והוספה לתסריט



העתקת תסריט מדמות אחת לדמויות אחרות

ניתן להעביר תסריט מדמות אחת לדמויות אחרות:

1. בחר את הדמות ממנה אתה מעוניין להעתיק תסריט.
2. היכנס ללשונית 'תסריטים' של הדמות
3. בחר את התסריט להעתקה, בחר-גרור-ושחרר מעל הדמות שאליה מעבירים את התסריט.

איור: העברת תסריט של דמות 'חתול' לדמות 'חיפושית'



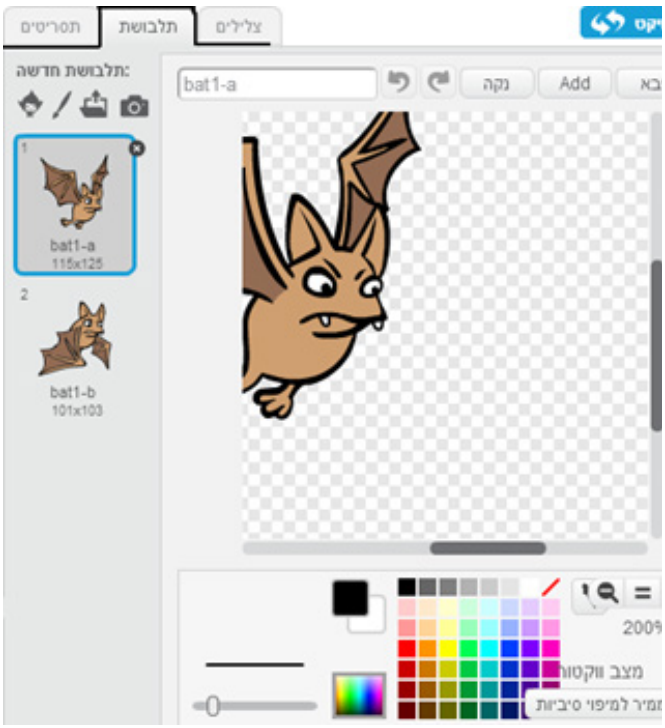
תלבושות - Costumes

לכל דמות בסביבת סקרטץ' יש את 'המראה' של הדמות. מראה זה נקרא תלבושת. דמות יכולה לכלול כמה תלבושות, כאשר ברגע מסוים הדמות "לובשת" רק תלבושת אחת. סביבת סקרטץ' כוללת בלוקים לשינוים בתלבושת של הדמות.

שינוי והוספת תלבושות לדמות

1. בחר את הדמות לה אתה רוצה להוסיף תלבושת או לערוך תלבושת.
2. בחר בלשונית 'תלבושת'.
3. ועתה נקבל את המסך הבא:

איור: מסך עריכת תלבושת



עבור דמות העטלף, אנו רואים שסביבת סקרטץ' מספקת שתי תלבושות לדמות.

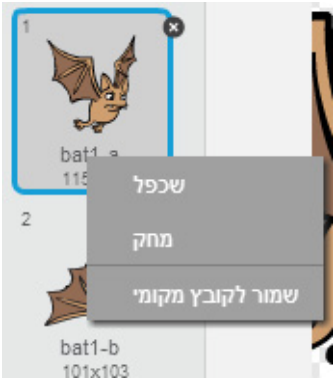
- תלבושת הנקראת - BAT1 - A
- תלבושת הנקראת - BAT1 - B

האפשרויות העומדות לרשותנו:

א. עריכה גרפית באמצעות ארגז הכלים של המסך הגרפי של הדמות וכך לשנות את הדמות

ב. העתקת אחת התלבושות BAT1 - A או BAT1 - B כדי לקבל תלבושת שלישית ורביעית וכך הלאה, ובשלב זה לערוך את התלבושת המועתקת.

איור: האפשרויות שעומדות לרשותנו בעת לחיצה על תלבושת באמצעות לחצן ימני של העכבר



ניתן לשכפל את התלבושת, למחוק את התלבושת, או לשמור לקובץ מקומי.

צלילים

איור: לשונית צלילים



לכל דמות ניתן לשייך צלילים. ניתן ליצור ולייבא צלילים ממקורות שונים כולל:

- הספרייה של סקרטץ' הכוללת עשרות צלילים ממוינים לפי קטגוריות
- להקליט צליל חדש (עם מיקרופון)
- להעלות צליל מקובץ (צליל שהורדת מרשת האינטרנט ושמרת על מחשבך או צליל שיצרת ושמרת)

לשונית צלילים כוללת תפריט עם האפשרויות: אפקטים; ערוך. תפריטים אלו כוללים אפשרויות רבות לעריכה של קובץ הקול לפני שילובו בתסריטי הדמות.

עריכת צלילים לא תילמד בספר זה. העורך הבסיסי של סביבת סקרטץ' מאפשר לבצע פעולות בסיסיות כמו העתקת חלק מהצליל והדבקתו ואפשרויות בסיסיות אחרות. יש ברשת כלי עריכה חינמיים לעריכה של קבצי קול לכל מי שמעוניין להיות מקצועי ולשלב ביישומים שלו בסקרטץ' צלילים מקוריים, ייחודיים ומותאמים בצורה מעולה לאירועים בתסריטים השונים.

יבוא קבצי צלילים

ניתן לייבא קבצי צלילים (לא קבצים דחוסים) בפורמטים הבאים:

WAV; חלק מהקבצים בפורמט MP3; קבצי AU (אך לא קבצים בדגימה של 24 ביט).

אם ישנם קבצי קול עם פורמט אחר, ניתן להשתמש בתוכנות עריכה דיגיטליות חינמיות כדוגמת: AUDACITY. התוכנה מאפשרת שינוי ועריכת הקובץ וגם שינוי פורמט הקובץ.

לשונית הרקע

סביבת סקרטץ' מאפשרת לשנות את הרקע של הבמה. השינוי והמעבר בין רקעים מתבצעים בבלוקים ייעודיים לכך. על המסך יש תפריט מיוחד שמאפשר בחירת רקעים נוספים ממקורות שונים:

- מספריית רקעים של סביבת סקרטץ'
- ציור רקע חדש על ידי המשתמש
- מקובץ מהמחשב
- רקע ממצלמה

איור: תפריט רקעים

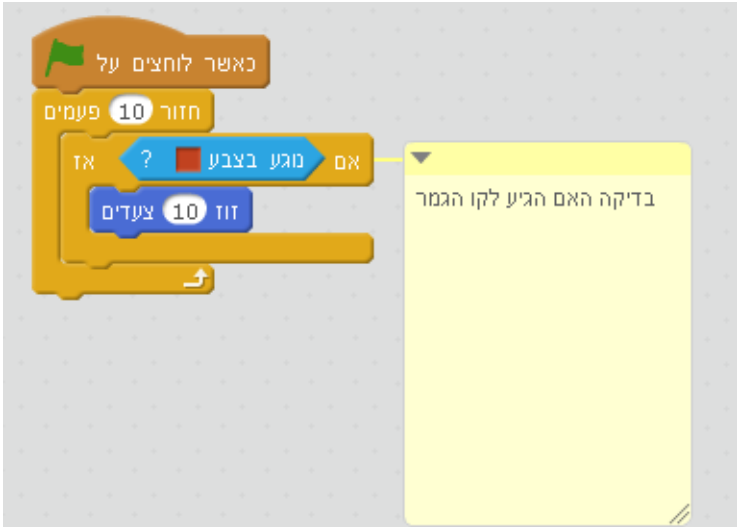


הערות ותיעוד

בעבודה על פרויקט גדול שכולל כמה דמויות ותסריטים שלעיתים קרובות נעשה בעבודת צוות, חשוב לתעד ולשלב הסברים בתסריטים. סביבת סקרטץ מאפשרת ביצוע של הוספת הערות :



לחיצה על לחצן ימני של העכבר מאפשרת הוספת הערה



במהלך תכנית הלימודים לא שולבו הערות מאחר ולכל תסריט שולבו דברי הסבר עבור כל מרכיב. אבל חשיבות תיעוד תסריטים גדולה מאוד ומהווה מרכיב מרכזי בהצלחה של הכתיבה, במיוחד בעבודת צוות.

נספח ה' ידע מתמטי במתמטיקה של תלמידי כיתות ד'; ה'

הנספח נועד לתת רקע למורים שאינם מורים למתמטיקה. התכנית מהווה בסיס להבנה מה הידע המתמטי של התלמידים בכיתות יסוד; כיצד ניתן להתבסס על ידע זה וגם כיצד ניתן בתכנית ב'מדעי המחשב' לתמוך בהבנה ויישום של ידע מתמטי וגיאומטרי. מאוד מומלץ, כאשר שוקלים לשלב יישום בתחומי מדע וטכנולוגיה, לעיין גם בתכנית הלימודים במדע וטכנולוגיה לבתי ספרי יסודי.

מקור תכנית לימודים חדשה במתמטיקה כל המגזרים תשס"ו 2006- משרד החינוך

HTTP://CMS.EDUCATION.GOV.IL/EDUCATIONCMS/UNITS/TOCHNIYOT_LIMUDIM/MATH_YESODI/PDF/

ידע במתמטיקה תחילת כיתה ד'

א. **המספרים הטבעיים בתחום הרבבה (10,000)**

1. המבנה העשרוני;
2. הגדלה והקטנה פי 10 ופי 100;
3. ערכי האותיות מ- א עד ת.

ב. **ישר המספרים (3 ש')**

ג. **פעולות החשבון בתחום הרבבה**

1. חיבור וחיסור במאונך;
2. שימוש בחוקי החילוף, הקיבוץ והפילוג;
3. שאלות חיבור וחיסור;
4. השלמה וביסוס של הנושא 'לוח הכפל' (כפל וחילוק);
5. כפל בעשרות שלמות ובמאות שלמות (בעל פה ובכתב);
6. סימני התחלקות ב-2, ב-5, וב-10;
7. כפל במאונך (במספרים חד-ספרתיים);
8. חילוק בתחום המאה, עם שארית (המחלק הוא חד-ספרתי);
9. סדר הפעולות והשימוש בסוגריים;
10. שאלות כפל וחילוק;
11. שאלות דו שלביות ושאלות נוספות;

ד. **שבר יסודי**

ה. **חקר נתונים**

ו. **מדידות וגיאומטריה**

1. זוויות, מאונכות, מקבילות, משולשים, מרובעים;
2. מידות ומדידות
   א. מדידות אורך;
   ב. מדידות משקל; היחידות: גרם, קילוגרם וטונה;
   ג. מדידות נפח: השוואה ומדידת נפחים של גופים;
   ד. מדידות זמן; יחידות זמן שונות: ימים, שעות, דקות, שניות.

ידע במתמטיקה תחילת כיתה ה'

א. **השבר הפשוט**

1. הכרת השבר הפשוט;
2. פעולות בשברים (היכרות ראשונה).

ב. **המספרים הטבעיים עד מיליון (10 ש')**

1. עקרונות המבנה העשרוני (חזרה);
2. שיטת האלף-בית.

ג. **ישר המספרים**

ד. **פעולות החשבון**

1. חיבור וחיסור (ללא הגבלה);
2. כפל במאונך (בגורם רב-ספרתי);
3. חילוק ארוך (המחלק הוא חד-ספרתי או עשרת שלמה);
4. אומדן ופיתוח תובנה מספרית;
5. שאלות כוללות (אינטגרטיביות);
6. סדר הפעולות והשימוש בסוגריים;
7. תכונות ה-0 וה-1;
8. מספרים ראשוניים ופריקים;
9. סימני התחלקות ב-3, ב-6 וב-9;
10. חזקות.

ה. **חקר נתונים וניתוח סיכויים**

ו. **צורות וגופים**

1. מצולעים, אלכסון;
2. ריבוע ומלבן;
3. תכונות של צלעות וזוויות במשולש;
4. תיבות.

ז. **סימטריה**

ח. **מדידות**

1. שטח; מ"ר, סמ"ר; נוסחאות שטח והיקף מלבן;
2. נפח תיבה, שטח פנים;