



## Scratch ךרד בשחמה יעדמב םיגשומו תונורקע

### Document Version:

!!Publisher's PDF, also known as Version of record

### Citation for published version:

Armoni, M & Ben-Ari, M 2010, ךרד בשחמה יעדמב םיגשומו תונורקע, Scratch. תארוהל הקלחמה עדמל ןמציו ןוכמ. <[https://stwww1.weizmann.ac.il/scratch/learning\\_book/](https://stwww1.weizmann.ac.il/scratch/learning_book/)> םיעדמה

*Total number of authors:*

2

### License:

CC BY-NC

### General rights

@ 2020 This manuscript version is made available under the above license via The Weizmann Institute of Science Open Access Collection is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognize and abide by the legal requirements associated with these rights.

### How does open access to this work benefit you?

Let us know @ [library@weizmann.ac.il](mailto:library@weizmann.ac.il)

### Take down policy

The Weizmann Institute of Science has made every reasonable effort to ensure that Weizmann Institute of Science content complies with copyright restrictions. If you believe that the public display of this file breaches copyright please contact [library@weizmann.ac.il](mailto:library@weizmann.ac.il) providing details, and we will remove access to the work immediately and investigate your claim.

(article begins on next page)

# רעיונות ומושגים במדעי המחשב

## דור SCRATCH

מיכל ארמוני  
מוטי בן-ארי



## כתיבה

ד"ר מיכל ארמוני  
פרופ' מוטי בן-ארי

## עריכה לשונית

נדן קלברמן

## עימוד ועריכה במחשב

אבי טל

## עיצוב

עמית תבור

## עיצוב עטיפה

ציפי עובדיה

## תודות

תודתנו נתונה לכל אלו שסייעו במלאכה ותרמו ממרצם ומכשרונם לתהליך היצירה.  
לד"ר אורני מרבאום-סלנט ולפאטמה קלותי-חלאק שערכו את המחקר על הוראת מושגים במדעי המחשב דרך Scratch.  
למורים שהשתתפו במחקר: מי-נועם ויינר מתיכון אמ"ת בנים ע"ש המר ברחובות, רות דאי מהתיכון ע"ש אהרון קציר ברחובות, נורין אבו סחאב מתיכון עירוני י"ב ביפו, עאמר קאיסי מביה"ס דה-לה-סאל בירושלים וואהיב שרבאטי מביה"ס סנט ג'ורג' בירושלים.  
לתלמידים שלמדו, יזמו, פיתחו וגם נהנו.  
לנדן קלברמן על העריכה הלשונית, לאבי טל על העיצוב והעריכה הגרפית ולציפי עובדיה על עיצוב העטיפה.

**Scratch is developed by the lifelong kindergarten  
Group at the MIT Media Lab.  
See: <http://scratch.mit.edu>**

אין לשכפל, להעתיק, לצלם, להקליט, לתרגם, לאחסן במאגר מידע, לשדר או לקלוט בכל דרך או אמצעי אלקטרוני, אופטי או מכני או אחר כל חלק, שהוא מהחומר שבספר זה.  
שימוש מסחרי מכל סוג שהוא בחומר הכלול בספר זה אסור בהחלט, אלא ברשות מפורשת בכתב מהמוציא לאור.



כל הזכויות שמורות למכון ויצמן למדע

נדפס יחידת הדפוס מכון ויצמן, תש"ע - 2010

# תוכן עניינים

5	הקדמה.....
9	פרק 1 - צעדים ראשונים.....
21	פרק 2 - ריבוי דמויות.....
29	פרק 3 - תסריטים קצרים וביצועים ארוכים.....
39	פרק 4 - תקשורת בין דמויות.....
47	פרק 5 - על רחבת הריקודים - בחזרה לביצוע חוזר.....
69	פרק 6 - משתנים - מנגנוני זיכרון.....
81	פרק 7 - זה תלוי - ביצוע מותנה.....
105	פרק 8 - מספרים.....
131	פרק 9 - רשימות - מנגנוני זיכרון מורכבים.....
155	פרק 10 - ביצוע מקבילי.....
177	פרק 11 - הרחבה והעמקה - במדעי המחשב ובסביבת Scratch.....
191	רעיונות ומושגים במדעי המחשב דרך Scratch - מבט אחורה.....



## הקדמה

בספר זה נערוך היכרות עם Scratch. בעזרת Scratch ניתן לייצר משחקים, סרטי אנימציה ועוד ועוד. המטרות של המשתמשים ב-Scratch מגוונות אף הן, וכך גם אופן השימוש: יש מי שיבנו משחק לשם בידור ויש שיבנו סימולציה של תופעה בפיזיקה או בכימיה כדי ללמוד וללמד. סביבת Scratch מספקת מרחב לאנשים עם כישורים שונים: לעיצוב משחק או סרט, לציור דמויות ולתכנות.

אשר למחברי ספר זה, מטרתם ממוקדת: לטפח את לימודי מדעי המחשב באמצעות עבודה עם Scratch. כמובן, אלו הלומדים בעזרת הספר יוכלו לנצל את הידע והמיומנות שנרכשו כדי להשתמש ב-Scratch בכל אופן אפשרי שיעלה בדעתם; אבל הדגש בספר הוא על הבנת רעיונות ומושגים חשובים של מדעי המחשב, ולא, למשל, על מתן הדרכה לבניית סרט או משחק.

### למה מדעי המחשב?

המחשב משמש ברוב תחומי החיים: רוב הסרטים שאנו רואים כוללים קטעים שנוצרו על ידי פעלולים ממוחשבים. כל חפץ מודרני שתרימו – נגן מוזיקה, טלפון סלולרי, מצלמה דיגיטלית – מכיל מחשב. רוב כספנו מגולם לא בצורת שטרות ומטבעות אלא כרישומים במחשבי הבנקים. רופא מפנה אותנו לבדיקות הנעשות על ידי מכשירים ממוחשבים. בכל נסיעה – במטוס, ברכבת ואפילו במכונית – אנו מפקידים את בטיחותנו בידי המחשבים.

העוסקים בבניית מחשבים, ובעיקר אלה העוסקים בפיתוח **תוכנה** למחשבים, אחראים לתרגם דרישות ורצונות שלנו לצורה המובנת על ידי מחשבים, שהם, כידוע, חרוצים וזריזים, אבל טיפשים למדי. פיתוח תוכנה הוא מקצוע מרתק ומאתגר, בגלל הצורך להתמודד עם ההבדלים בין יצורים כל כך שונים: בני אדם ומחשבים. אנו מקווים שהיכרות עם פיתוח תוכנה ב-Scratch תקרב אתכם למקצוע מדעי המחשב ואולי אף תגרום לכם לשקול להעמיק בלימוד המקצוע בעתיד.

### למה Scratch?

רובנו יודעים להשתמש במחשב לצורך גלישה באינטרנט (הורדת מוזיקה, דואר אלקטרוני, שיחות צ'אט, קניות), חיבור מסמכים ומשחקים. האם לא התעוררה בכם הסקרנות לגבי הדרך שבה כל הדברים הנפלאים האלה **בנויים**? ובכן, הם בנויים **כתוכניות מחשב** הכתובות בשפות מיוחדות, **שפות תכנות**, שאותן המחשב מסוגל להבין ולבצע. לצערנו, התוכניות השימושיות האלו מסובכות מאוד (ויכולות להכיל אפילו מיליוני משפטים בשפת תכנות), והשפות עצמן מיועדות למקצוענים, וקשה ללמוד אותן. כאן Scratch מגיעה לעזרתנו. השימוש ב-Scratch פשוט מאוד, וניתן להתחיל לבנות בה תוכניות מחשב, כלומר, **לתכנת** בעזרתה, כבר עם ההיכרות הראשונית. בנוסף ב-Scratch אפשר להשתמש בגרפיקה, באנימציה ובצלילים גם בלי להכיר את הפרטים הטכניים המסובכים הנחוצים להפקתם.

אל תתנו למראה הצבעוני של סביבת Scratch להטעות אתכם! אין מדובר במשחק מחשב. Scratch מאפשרת פיתוח תוכנה לכל דבר, והעבודה בה תספק לכם התנסות אמיתית הדומה מאוד לעבודתם

של עובדים מקצועיים במדעי המחשב. העבודה ב־Scratch מהנה ומשעשעת, והתוצרים יכולים להיות גם משחקים מהנים או סרטים משעשעים, אבל דרך העבודה ב־Scratch תתמודדו עם האתגר של פיתוח תוכניות: מה המחשב מסוגל לבצע ומה לא? איך אפשר לתרגם רצונות ודרישות מסוימים להוראות שאותן המחשב יכול לבצע? איך מזהים תקלות ומתגברים עליהן? אנו בטוחים שלימוד לעומק של Scratch יהיה חוויה מרתקת.

## מבנה הספר

אין לומדים לתכנת על ידי קריאת ספר אלא על ידי כתיבת תוכניות. הספר יכול רק להנחות ולספק הסברים על Scratch ועל שפת התכנות הגלומה בה. לכן ספר זה בנוי **כאוסף של משימות**. כל פרק מלמד מושג חדש, ולעיתים כמה מושגים הקשורים זה לזה, אבל המושג תמיד נדרש לפתרון של משימה מסוימת: ספירת תפוזים, אנימציה של דמויות רוקדות, בניית משחק מחשב וכו'. כל פרק מתחיל במשימה פשוטה, ולאחר שהמשימה הפשוטה נפתרת, הפרק ממשיך בשרשרת של הצגת משימות ופתירתן, כאשר כל משימה חדשה מתקבלת על ידי הרחבה או סיבוך הדרישות של משימה קודמת בשרשרת. כל שלב בשרשרת המשימות מחייב שימוש במושג או במושגים חדשים של מדעי המחשב, וכן בכלי חדש של Scratch או בשילוב חדש של כלים שכבר למדתם. תהליך הלימוד המומלץ הוא כזה:

- קראו את המשימה.
- חשבו כיצד אפשר לפתור אותה וכתבו את הפתרון במילים, בעברית.
- בדקו אם מוכרים לכם כל הכלים ב־Scratch הנחוצים לתרגום הפתרון מעברית לשפת התכנות, ונסו לבצע את התרגום.

- כל כלי חדש ב־Scratch מופיע במסגרת עם תיאור קצר. נסו להשתמש בכלים אלה בפתרונכם.

לכל שלב בפתרון של כל משימה כתבנו **הסבר מפורט מאוד**, וכמו כן ניתן להוריד פרויקטים של Scratch מאתר הספר שהם פתרונות מלאים של כל המשימות והתרגילים. אם נתקלתם בקושי כלשהו, קראו הסברים אלה, ואם גם זה אינו עוזר, עיינו בפתרון שלנו. גם אם פתרתם את המשימה ללא קושי, קראו בספר את ההסברים המתייחסים לפתרון. הסברים אלו מציפים ומאירים את מושגי מדעי המחשב שבאים לידי ביטוי בתהליך פתרון המשימה.

חלק מהמשימות מוגדרות כמשימות "רשות". לעיתים אלו משימות המערבות מושגים מתקדמים שאפשר לוותר על ההתמודדות עמם בשלבים הראשונים של הספר. ברוב המקרים אלו משימות העוסקות בנושאים שאינם מהותיים במדעי המחשב (כגון ציור דמויות או רקע) ושההתמודדות עמם אינה הכרחית להבנת המושגים והרעיונות החשובים של מדעי המחשב.

## מילה ללומדים

ככל שתנסו להתמודד לבד עם המשימות, לפני קריאת הפתרונות, כך תהיה הלמידה מוצלחת יותר. אבל אם ניסיתם לפתור משימה ונתקלתם בקושי כלשהו, **אל תהססו לקרוא את ההסברים ולעין בפתרונות**. ככל שתתקדמו בעבודה עם הספר, תגלו שאתם מתחילים להכיר את ה"ראש" של Scratch ולמידת המושגים תהיה קלה יותר!

דבר נוסף: **אל תתיאשו** כאשר מתעוררת בעיה. ללמוד לפתור משימות ולתרגם את הפתרונות לשפת תכנות הוא תהליך לא פשוט הדורש השקעה. התייחסו אל כל בעיה כאל אתגר שעליכם להתמודד איתו ולהתגבר עליו. גם בהקשר זה, ככל שתצברו ניסיון, יגבר הביטחון העצמי ביכולתכם להתמודד עם בעיות.

## מילה למורים

אין טעם וצורך לשאול אותנו: "האם מותר ל...?" התשובה היא תמיד "כן". אמנם אנו ממליצים ללמוד את המושגים ולפתור את המשימות על פי הסדר שבו הם מופיעים בספר, כי לפי דעתנו סדר זה יקל ככל האפשר על הלמידה. אולם אתם חופשיים לשנות את הסדר, לדלג על פרקים, להציג משימות אחרות או לשנות בכל דרך אחרת את תהליך הלמידה. בנוסף, ברור שכל תלמיד או תלמידה יתקדמו בקצב אישי. אין טעם לעצור תלמידים הרוצים לדהור קדימה, ולעכב אותם בבואם לחקור נושאים אחרים או לבנות פרויקטים אחרים.

הדבר היחיד שעליו אנו מבקשים לא לוותר הוא עמידה ביעדים שמציב הספר. כלומר, אנו מבקשים מכם לוודא שהתלמידים אכן לומדים את המושגים הבסיסיים והחשובים של מדעי המחשב. Scratch מפתה לעסוק במראה החיצוני של הפרויקטים: ציור דמויות, אפקטים גרפיים, צלילים. אנו לא טוענים שעיצוב הוא חסר חשיבות, אבל בשטח זה אין יתרון ל-Scratch על פני כל חבילת תוכנה אחרת. הגרפיקה ב-Scratch מגבירה את המוטיבציה ולעיתים אף מפשטת את תהליך התכנות, אבל עיסוק מרוכז מדי במראה החיצוני עלול להסיח את הדעת ממטרתנו: לקרב את התלמידים למדעי המחשב.





## פרק 1 צעדים ראשונים

בספר זה נערוך היכרות עם Scratch, שבעזרתה תוכלו ליצור אנימציות ממוחשבות (פשוטות בתחילה, אך גם יותר מורכבות, ככל שנעמיק את ההיכרות), וגם ללמוד כמה רעיונות ועקרונות חשובים ומעניינים של התחום הקרוי "מדעי המחשב".

### דוגמה 1: האנימציה הראשונה

את הפגישה הראשונה עם Scratch נעשה בעזרת אנימציה פשוטה מאוד שאותה כבר הכנו עבורכם.

### הפעלת סביבת Scratch

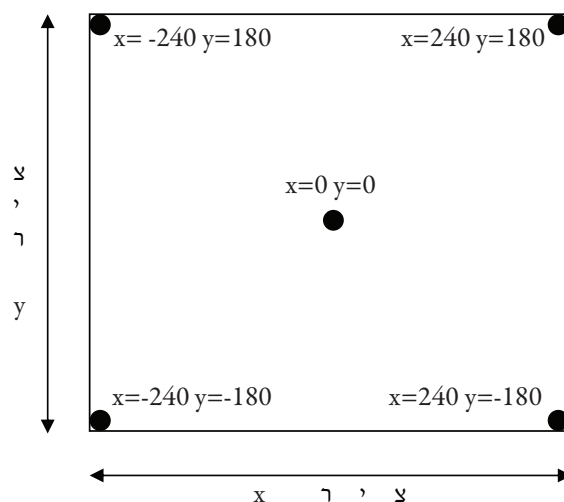
הפעילו את סביבת Scratch על-ידי לחיצה כפולה על צלמית החתול במשטח העבודה. נפתח החלון של Scratch. לחצו על הכפתור "File", המופיע בשורת הכפתורים העליונה, וברשימה שנפתחה בחרו בפריט "פתח...". כמו בכל יישום ב-Windows תופיע רשימה של תיקיות. בחרו בתיקייה המתאימה לפרק הראשון, ולחצו על הכפתור OK. בחרו ברשימה שמופיעה את הפרויקט move-10-steps, המכיל את הדוגמה הראשונה, ולחצו שוב על OK.

שם הקובץ: move-10-steps

### תיאור החלון של Scratch

מה מכיל כעת החלון של Scratch? בחלקו הימני של החלון מופיע ריבוע לבן גדול. זוהי **הבמה** שעליה נעות הדמויות השונות, אלו הדמויות שמשתתפות באנימציות שניצור. בדוגמה שלפנינו משתתפת דמות אחת בלבד, החתול של Scratch. בפינה השמאלית העליונה של הבמה מתוארים כיוונו של החתול ומיקומו על הבמה: הוא נמצא בדיוק במרכז הבמה, הנקודה  $(0, 0)$ , ופונה בכיוון של 90 מעלות, כלומר, ימינה.

ממדי הבמה הם 480 יחידות אורך ו-360 יחידות גובה. השרטוט הבא מראה את הנקודות המתאימות למרכז הבמה ולארבע פינותיה.



מתחת לבמה, בצד ימין, מתואר המיקום הנוכחי של מצביע העכבר. נסו להזיז את מצביע העכבר על הבמה ולנחש מה יהיו הערכים שיופיעו מתחת לבמה.

## הפעלת אנימציה

כדי להפעיל את האנימציה הצביעו בעזרת העכבר על הדגל הירוק המופיע מעל הבמה בצד ימין, ולחצו עליו. מה קרה? החתול שינה את מיקומו וזז מעט ימינה, בכיוון שאליו הוא פונה. גם מתארי המיקום שבפינת הבמה מצביעים על השינוי: על ציר ה־x החתול נמצא כעת במיקום 10, בעוד שלפני ההפעלה מיקומו על ציר ה־x היה 0. מיקומו על ציר ה־y לא השתנה, וכך גם כיוונו.

לחצו שוב על הדגל הירוק. מה קרה כעת? שוב שינה החתול את מיקומו וזז עוד 10 צעדים ימינה. לחיצות חוזרות על הדגל הירוק יגרמו שוב ושוב לאותה פעולה – תזוזה של 10 צעדים ימינה לאורך ציר ה־x. כך עד אשר החתול מגיע אל מעבר לקצה הבמה, ורק קצה דמותו נראה. במצב זה לחיצות נוספות על הדגל הירוק לא יגרמו לתזוזה נוספת.

## תסריטים

אם נביט בחלון האפור המוארך הנמצא לשמאלה של הבמה, נמצא שם את ההסבר לפעולתו של החתול. בחלון זה מתואר **תסריט** שלפיו החתול פועל. לכל דמות המשתתפת ב־Scratch ניתן לכתוב תסריטים המתארים את ההתנהגות שלה. הדמויות ב־Scratch הן חסרות רצון עצמאי ופועלות אך ורק לפי הוראות ברורות הכלולות בתסריטים שאנו מספקים להן. כצפוי, התסריט המתאר את פעולת החתול הוא קצר ופשוט:



על־פי התסריט, הפעולה שהחתול מבצע היא תזוזה של 10 צעדים, ופעולה זו מתבצעת בכל פעם שמתבצעת לחיצה על הדגל הירוק. התנהגות זאת מוסברת גם על־ידי **ההערה** המופיעה במסגרת ליד התסריט. ההערות הן חשובות, ומטרתן לעזור לנו להבין תסריטים חדשים כאשר הם מוצגים בפנינו. אך עלינו לזכור שההערות נועדו לקוראים, והתנהגות הדמויות נקבעת על־פי ההוראות הכלולות בתסריטים ולא על־פי ההערות.

### מושג חדש: הוראה

**הוראה** משמשת אותנו כדי להנחות דמויות חסרות רצון עצמאי. כדי שתוכל להיות מבוצעת על־ידי דמות חסרת רצון עצמאי, הוראה צריכה להיות **ברורה וחד־משמעית**.

### מושגים חדשים ב־Scratch: תסריט, הערות

**תסריט** הוא הכלי של Scratch אשר בעזרתו מנחים דמות כיצד לפעול. תסריט מורכב מהוראות שאותן הדמויות של Scratch יכולות להבין. חלק חשוב בתסריט הוא **ההערות** המשולבות בו. אמנם, הדמות המופעלת על־ידי התסריט אינה מתייחסת להערות הכלולות בו, אבל להערה יש תפקיד חשוב של **תיעוד** התסריטים כך שיהיו ברורים, לכל אלה שרוצים להשתמש בהם, לשפרם או לעדכן אותם.

## דוגמה 2: סדרת הוראות תנועה

ננסה ליצור לחתול התנהגות מורכבת מעט יותר, המכילה יותר מצעד תנועה אחד.

### משימה 1

יש ליצור אנימציה אשר כל הפעלה שלה (על־ידי לחיצה על הדגל הירוק) תגרום לחתול לזוז 100 צעדים ולהסתובב סיבוב של 90 מעלות נגד כיוון השעון.



שם הקובץ: `move-steps-turn-left`

### שינוי ערכים בהוראה

לצורך האנימציה החדשה נרצה קודם כול להגדיל את הקפיצה שמבצע החתול בכל פעם, מ־10 צעדים ל־100 צעדים. הוראת התזוזה היא למעשה **הוראה כללית** שאינה מחייבת מספר צעדים מסוים. על־ידי בחירת ערך לשילוב בהוראה, אנו הופכים את ההוראה הכללית ל**הוראה ממוקדת** הקובעת תזוזה בערך המסוים שבחרנו. למשל, נוכל לשנות את הערך 10, המשולב כעת בהוראה, לערך 100. לשם כך הצביעו עם העכבר על התסריט של החתול, על ההוראה **זוז 10 צעדים**, סמנו את המספר המופיע בחלונית הלבנה (על־ידי לחיצה עליו עם הכפתור השמאלי של העכבר), ושנו אותו מ־10 ל־100. היזו בעזרת העכבר את החתול שעל הבמה מקצה הבמה אל מיקום אחר, לחצו שוב על הדגל הירוק וראו מה קורה.

נסו לשלב בהוראה מספר שלילי, למשל -100. מה קורה כעת עם לחיצה על הדגל הירוק?



בחירת ערכים שליליים תגרום לתזוזה שמאלה, כלומר, בכיוון ההפוך מזה שאליו פונה החתול, בעוד שערכים חיוביים מביאים לתזוזה ימינה, כלומר, בכיוון שאליו פונה החתול. היווכחו בזאת על־ידי שילוב ערכים שונים בחלונית הלבנה ועל־ידי לחיצה על הדגל הירוק לאחר כל שינוי. עם כל הפעלה בחנו גם את מתארי המקום שבפינת הבמה.

שנו שוב את ההוראה כך שתזוזה על תזוזה ימינה של 100 צעדים, ונמשיך בעדכון האנימציה.

### מושג חדש: הוראה כללית והוראה ממוקדת

**הוראה כללית** הופכת ל**הוראה ממוקדת** על־ידי שילוב ערכים מתאימים בתוכה. הוראה כללית אינה מפורשת וחד־משמעית, ולכן לא ניתן להשתמש בה כדי להנחות ישות חסרת רצון עצמאי (כמו הדמויות של Scratch).

## הוראה חדשה ב-Scratch: תזוזה

ההוראה **זז** **צעדים** גורמת לדמות לשנות את מיקומה ולזוז כמספר הצעדים המשולב בחלונית שבהוראה, כאשר "צעד" הוא יחידה בסיסית על הבמה. אם המספר המשולב בהוראה הוא חיובי, הדמות תנוע בכיוון אליו היא פונה. אם המספר המשולב בהוראה הוא שלילי, הדמות תנוע בכיוון ההפוך לזה שאליו היא פונה.

## סיבוב של דמות

עתה נוסיף עוד מרכיב לפעולת החתול, של הסתובבות לכיוון אחר. הביטו בחלון השמאלי. יש שם מאגר עשיר של **לבנים**, אבני בניין לבניית התסריטים. כל לבנה מתאימה להוראה של Scratch. נבחר את הלבנה המתאימה להוראת סיבוב נגד כיוון השעון, ההוראה השלישית ברשימה שבחלון השמאלי. נצרף אותה לתסריט על-ידי כך שנגרור אותה לחלון האמצעי, אל התסריט הקיים בו, ונצמיד אותה מתחת ללבנה המתאימה להוראת התזוזה.

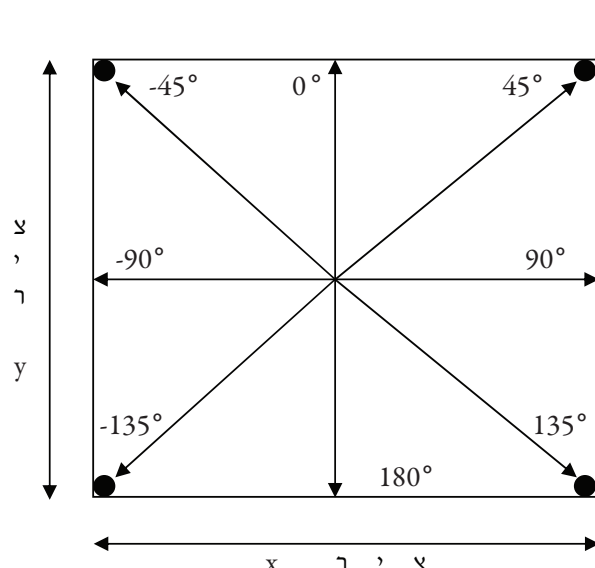
הגרירה מתבצעת על-ידי הצבעה עם העכבר על הלבנה המבוקשת ברשימה ועל-ידי לחיצה מתמשכת על הכפתור השמאלי של העכבר, בד בבד עם הזזתו. כאשר הלבנה מגיעה למקום המתאים, יש להפסיק את ההזזה ואת הלחיצה על הכפתור השמאלי.

שימו לב כי בעת גרירת הלבנה בחלון התסריטים, כאשר היא קרובה למקום שבו ניתן לשלב אותה, מופיע פס לבן המסמן מקום מתאים לשילוב. במקרה זה מופיע פס לבן מעל לבנת התזוזה, כאשר לבנת הסיבוב קרובה למקום זה; וכאשר לבנת הסיבוב קרובה לתחתית התסריט מופיע פס לבן גם מתחת ללבנת התזוזה. שתי אפשרויות השילוב נותנות תסריט חוקי. אנו נבחר כעת לשלב את הוראת הסיבוב כהוראה שנייה, מתחת להוראת התזוזה.

בדומה להוראת התזוזה, גם הוראת הסיבוב היא הוראה כללית. בלבנה המתאימה להוראת הסיבוב יש חלונית לבנה, ובה מופיע מספר המעלות המבוקש לסיבוב. נשנה את המספר המופיע בחלונית ל-90°, ובכך נהפוך את ההוראה להוראה ממוקדת, המורה על סיבוב של 90° מעלות נגד כיוון השעון.

## כיוונים על הבמה

לכיוונים ב-Scratch מתייחסים בדומה לכיווני שעון אנלוגי, עם מחוגים. התנועה יכולה להיות עם



כיוון השעון או נגדו. אך בעוד שבשעון הכיוונים מתוארים בעזרת שעות (12 כיוונים אפשריים) או דקות ושניות (60 כיוונים אפשריים), כאן הכיוונים מתוארים בעזרת מעלות (המסומנות ב-°), ויש 360 כיוונים אפשריים. הכיוונים העיקריים מבין 360 הכיוונים האפשריים מתוארים בשרטוט הבא. בחציו הימני של השרטוט הכיוונים הם חיוביים – מ-0° מעלות ועד 180° מעלות. בחציו השמאלי הם שליליים, מ-180° מעלות ועד 0° מעלות.



## דוגמה 3: התחלת ביצוע אנימציה ממצב קבוע

### משימה 2

יש ליצור אנימציה אשר כל הפעלה שלה (על-ידי לחיצה על הדגל הירוק) תגרום לחתול לזוז 100 צעדים ולהסתובב סיבוב של 90 מעלות נגד כיוון השעון. התנועה תתחיל תמיד כאשר החתול נמצא בנקודה (0, 0) ופונה ימינה.

שם הקובץ: `initialize-and-move`



### אתחול

שני התסריטים שהפעלנו עד כה גרמו לתזוזה של החתול. בתום ביצוע התסריט החתול נמצא במיקומו החדש. הפעלה חוזרת של התסריט מתבצעת ביחס למיקום החדש, ולכן הפעלות חוזרות ונשנות מביאות לתוצאות שונות. כעת ניצור תסריט יציב אשר כל הפעלה שלו, יהיה מיקומו של החתול בעת ההפעלה, אשר יהיה, יוצרת בדיוק אותו רצף תנועות. לשם כך נרצה קודם כול שבכל פעם שיופעל התסריט, יזוז החתול לנקודת התחלה קבועה. הסתכלו במאגר הלבנים שבחלון השמאלי. איזו מבין ההוראות המיוצגות בו נראית מתאימה לצורך שהגדרנו?

נוכל להשתמש לשם כך בהוראה **קפוץ אל-x: y**. בחרו בלבנה המתאימה להוראה וצרפו אותה

אל התסריט, כהוראה ראשונה בסדרת התנועה, מיד אחרי ההוראה **נאשר לוחצים על**. שימו לב שמספיק לגרור את הלבנה אל המקום הרצוי ולעזוב את העכבר, ומיד מתפנה להוראה החדשה מקום מתאים. בתסריט הזה נקודת ההתחלה של סדרת הפעולות תהיה מרכז הבמה – הנקודה (0, 0). ודאו שהערכים בחלונות ההוראה הם הערכים 0 ו0, ואם יש בכך צורך עדכנו אותם לערכים 0 ו0.

קביעת ערך התחלתי נקראת גם **אתחול**. בתסריט שאנו יוצרים כעת הוראת התנועה הראשונה מאתחלת את מיקום החתול.

#### הוראה חדשה ב-Scratch: קפיצה

ההוראה **קפוץ אל-x: y** גורמת לדמות לשנות את מיקומה ולעבור למיקום חדש, על-פי הערכים המשולבים בשתי החלונות. החלונות של x מתאימה למיקום בממד המאוזן, והחלונות של y מתאימה למיקום בממד האנכי.

בנוסף למיקום הקבוע נרצה שהתנועה תתחיל תמיד גם מכיוון קבוע. לכן, לאחר אתחול המיקום לערך ההתחלתי של מרכז הבמה, נאתחל גם את הכיוון. איזו הוראה מבין אלו העומדות לרשותנו תתאים? נוכל לבצע את האתחול בעזרת ההוראה הרביעית ברשימה **פנה לכיוון**.

צרפו את ההוראה לתסריט כהוראה שנייה בסדרת פעולות התנועה, מיד לאחר הוראת הקפיצה. הכיוון שבו נרצה להתחיל את הרצף התנועותי הוא ימין. ליד החלונות בלבנה של הוראה זו יש חץ קטן, המראה כי בהוראה זו ניתן לשלב ערך מתוך רשימה נתונה של ערכים. לחצו על החץ הקטן כך שתיפתח הרשימה ובחרו את הערך ימינה (90 מעלות). (הרשימה שנפתחת עם הלחיצה על החץ הקטן כוללת את ארבעת ערכי הכיוון העיקריים, אך אין בה כדי להגביל את אופן השימוש בהוראה. ניתן גם לבחור כל

כיוון אחר על ידי התעלמות מהחץ והחלפה ישירה של המספר הרשום בחלונית בכל מספר המתאים לכיוון כלשהו).

### הוראה חדשה ב־Scratch: פנייה

ההוראה **פנה לכיוון** גורמת לדמות לפנות אל הכיוון המופיע בחלונית של ההוראה.

התסריט שהתקבל מכיל סדרה של ארבע הוראות, המתבצעות זו אחרי זו, לפי הסדר, בכל פעם שמתבצעת לחיצה על הדגל הירוק. שמרו את התסריט שיצרתם בשם חדש.

לחצו כעת על הדגל הירוק, להפעלת התסריט החדש. החתול סיים את הרצף התנועתי שלו במיקום שהוא 100 צעדים ימינה למרכז הבמה, ובכיוון מעלה. הפעלה חוזרת תגרום שוב ושוב לאותה תנועה ולסיום באותו מיקום ובאותו כיוון. גם אם נזיז את החתול בעצמנו על-ידי גרירתו על הבמה בעזרת העכבר למקום אחר, אחרי הפעלת התסריט הוא יחזור לאותו מצב סיום – אותו מיקום ואותו כיוון.

### מושג חדש: אתחול

אתחול הוא קביעת ערך או מצב התחלתי. במקרים רבים בתחילתה של סדרת הוראות יש צורך בכמה הוראות אתחול, כדי שביצוע סדרת ההוראות יתחיל תמיד מאותו מצב.

## הוראות יחסיות ומוחלטות

ייתכן כי שמתם לב שארבע ההוראות הכלולות בתסריט הן משני סוגים שונים. הוראת התזוזה והוראת הסיבוב שייכות לסוג אחד, בעוד שהוראת הפנייה והוראת הקפיצה שייכות לסוג שני. האם תוכלו לשער מה ההבדל בין שני הסוגים?



הוראת התזוזה והוראת הסיבוב הן **הוראות יחסיות**. כלומר, תוצאתן תלויה לא רק בערך המשולב בהן אלא גם במצב שממנו הן מופעלות. תזוזה של 10 צעדים כאשר החתול נמצא במרכז הבמה תביא אותו למיקום אחד, בעוד שתזוזה של 10 צעדים כאשר החתול נמצא בפינה השמאלית התחתונה של הבמה תביא אותו למיקום אחר. בדומה, סיבוב של 90 מעלות נגד כיוון השעון כאשר החתול בכיוון ימין יגרום לו לפנות למעלה, בעוד שסיבוב של 90 מעלות נגד כיוון השעון כאשר החתול בכיוון שמאל יגרום לו לפנות למטה.

לעומתן, הוראת הפנייה והוראת הקפיצה הן **הוראות מוחלטות**. אלו תלויות רק בערכים המשולבים בהן. קפיצה לנקודה מסוימת תביא את החתול תמיד לאותה נקודה, ולא משנה כלל באיזו נקודה הוא נמצא לפני הקפיצה. בדומה, פנייה לכיוון מסוים תמיד לחתול לפנות לכיוון זה, ולא משנה לאיזה כיוון פנה קודם לכן.

### מושג חדש: הוראה יחסית והוראה מוחלטת

**הוראה מוחלטת** היא הוראה שתוצאת ביצועה תלויה רק בערכים המשולבים בה. לעומתה, ביצועה של **הוראה יחסית** תלוי גם בערכים המשולבים בה וגם במצב שבו היא מתבצעת.







## דוגמה 4: תנועה מתמשכת

### משימה 3

יש ליצור אנימציה אשר בה החתול מטייל מסביב לבמה, נגד כיוון השעון, החל בפינה השמאלית התחתונה.



שם הקובץ: `move-around-stage`

עד כה השתמשנו בהוראת תזוזה, הגורמת לקפיצה מיידית מנקודה לנקודה על פני הבמה. פתרון משימה 3 דורש להציג טיול, כלומר, נרצה לאפשר צפייה בדמות החתול בעודה זזה. לשם כך ניתן להשתמש בהוראת גלישה  גלוש  שניות ל- $x$ :   $y$ : 





גם הוראת הגלישה היא הוראה כללית. כדי להפוך הוראת גלישה מהוראה כללית להוראה ממוקדת, צריך לספק לה כמה ערכים – שני ערכים (בחלונות השנייה והשלישית) המציינים את המיקום שאליו אנו מעוניינים שהדמות תגיע (כמו בהוראת קפיצה), וערך נוסף (בחלונות הראשונה) המשפיע על קצב התזוזה כפי שתיראה על הבמה. ערך זה מציין את מספר השניות שייקח לדמות לעבור את המרחק שבין נקודת ההתחלה לנקודת היעד.

האם הוראת הגלישה היא הוראה יחסית או מוחלטת?



הוראת הגלישה היא הוראה יחסית. אמנם נקודת הסיום של הגלישה תלויה אך ורק בערכים המשולבים בהוראה ולא בנקודת ההתחלה, אך הדרך הנראית על הבמה תלויה גם בנקודה שבה נמצאת הדמות לפני תחילת ביצוע ההוראה.

### הוראה חדשה ב־Scratch: גלישה

ההוראה  גלוש  שניות ל- $x$ :   $y$ :  גורמת לדמות לזוז ממיקומה הנוכחי על הבמה אל המיקום שמתואר על-ידי הערכים המשולבים בחלונות של  $x$  ו- $y$ . משך זמן התזוזה של הדמות יהיה כמספר השניות המופיע בחלונות הראשונה.

## תיאור מילולי של התסריט החדש

בעזרת ההוראה שהכרנו כעת, הוראת הגלישה, והוראות אחרות שכבר הכרנו, נוכל כעת לכתוב תסריט אשר גורם לחתול לטייל מסביב לבמה, נגד כיוון השעון, החל בפינה השמאלית התחתונה.

מהי סדרת ההוראות הדרושה, שלפיה יפעל החתול? לפני שניצור את הסדרה בעזרת אבני הבניין הנתונות לנו, ההוראות של Scratch, נפרק למרכיביו את הרצף התנועתי של טיול מסביב לבמה. אחרי שנבין בדיוק מהם מרכיבי הרצף, יקל לנו להתאים הוראת תנועה של Scratch לכל שלב בו. ציירו את הרצף התנועתי על דף ונסו לפרק אותו למרכיביו.

נקבל את שלבי התנועה הבאים:

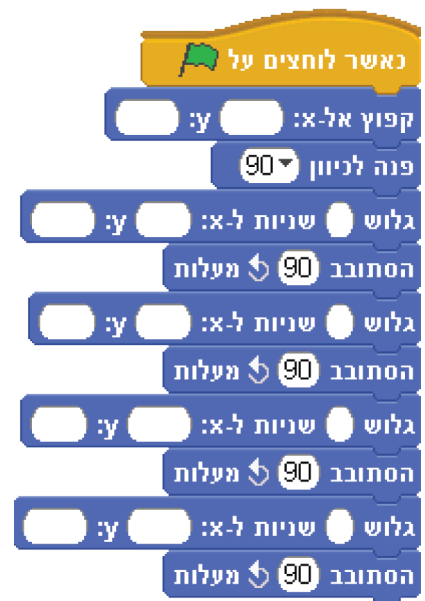
1. כאשר יש למקס את המגוון הנקודות ההתמחה הכיזויה, כאומה, לאגמח את המקום שבו אפינה השמאלית הגמגונה של הכמה.
2. אהפנה את המגוון ימינה.
3. אהנע את המגוון אל הפינה הימנית הגמגונה.
4. אסוב את המגוון לכיוון מלאה.
5. אהנע את המגוון אפינה הימנית העליונה.
6. אסוב את המגוון שמאלה.
7. אהנע את המגוון אפינה השמאלית העליונה.
8. אסוב את המגוון למטה.
9. אהנע את המגוון אפינה השמאלית הגמגונה.

#### מושג חדש: תיאור מילולי

בעזרת **תיאור מילולי** אנו מתארים את התנהגותה של דמות באופן חופשי יחסית, בלי להשתמש עדיין בהוראות מדויקות של Scratch. למרות שיש חופש רב בכתיבת תיאור מילולי, עדיין עליו להיות חד־משמעי וברור. תיאור מילולי מורכב מסדרה של **צעדים**.

### בניית התסריט החדש מהוראות של Scratch

את סדרת הפעולות הזאת נוכל לתרגם בקלות לתסריט של Scratch המורכב מרצף ההוראות הבא:



צרו את התסריט. תוכלו לעשות זאת על־ידי עדכון התסריט האחרון שיצרתם. את שתי ההוראות הראשונות בתסריט – הוראת הקפיצה והוראת הפנייה – ניתן להשאיר, אך צריך יהיה לעדכן את הערכים המשולבים בהוראת הקפיצה. מחקו את שתי ההוראות האחרונות בתסריט הקודם. ניתן לעשות זאת

על-ידי בחירת הלבנה של ההוראה השלישית וגרירתה בחזרה אל חלון ההוראות. יחד איתה נגררות גם ההוראות שמתחתיה – במקרה זה הוראה אחת. אחרי מחיקתן יש להוסיף עוד שבע הוראות, המתאימות לצעדים 3 עד 9 בתיאור המילולי. כעת לא נותר עוד אלא לשלב ערכים מתאימים בהוראות הקפיצה והגלישה, כדי להפוך אותן להוראות ממוקדות.

## שילוב ערכים בהוראות התנועה – המיקום מתייחס למרכז הדמות

שימו לב: מאחר שהמיקום שאנו מציינים בהוראות התנועה מתייחס תמיד למרכז של הדמות, ומשום שנרצה שבעת ביצוע האנימציה ייראה כל גופו של החתול ולא רק חלקו, יש לקבוע בזהירות את הערכים המתאימים לשילוב בהוראות התזוזה, כך שלא יתאימו למסגרת הבמה, אלא מעט יותר פנימה. כדי למצוא את הערכים המתאימים לשילוב בהוראות התנועה, הצביעו עם העכבר על מרכז של דמות החתול והזיזו אותה לכיוון קצה הבמה.

הזזת דמות מתבצעת בעזרת הכפתור השמאלי של העכבר, בדיוק באותו אופן בו הזזתם לבנה לצורך שילובה בתסריט. כשדמות החתול קרובה מאוד לקצה הבמה אך עדיין נראית בשלמותה, הסתכלו על הערכים המופיעים בתחתית הבמה ומתאימים למיקומו של מצביע העכבר. בכל מקרה, גם אם הערכים שתשלבו בהוראות התזוזה לא יהיו מדויקים, תוכלו לתקן זאת בהתאם לצורך.

## הפעלת התסריט

הפעילו את התסריט על-ידי לחיצה על הדגל הירוק ובדקו אם אכן התקבל הרצף התנועתי המבוקש. אם לא, בדקו כל הוראה בתסריט והשוו אותה לצעד המתאים לה בתיאור המילולי. לאחר שקיבלתם תסריט תקין, הקפידו לתעד אותו על-ידי עדכון ההערה להערה מתאימה עבורו. שמרו אותו בשם חדש.

**תרגיל 1:** כעת נסו ליצור בעצמכם תסריט המבצע רצף תנועתי שונה מעט: עם לחיצה על הדגל הירוק, יתחיל החתול לנוע מהפינה הימנית העליונה ויקיף את הבמה עם כיוון השעון. נתחו קודם כול את הרצף התנועתי הרצוי והבינו מהם מרכיביו. רק אחר כך תרגמו את סדרת התנועות שתכננתם לתסריט המשתמש באבני הבניין של Scratch. ההקפדה על תכנון לפני יצירת תסריט תלווה אותנו גם בהמשך, וחשיבותה תעלה ככל שהאנימציות שנרצה ליצור יהפכו מורכבות יותר ויותר.

בפרק זה ערכנו היכרות ראשונה עם Scratch ודרכה גם עם כמה מושגים בסיסיים במדעי המחשב, העוסקים בדרך שבה אנו יכולים לתת הוראות ברורות לדמויות ללא רצון עצמאי משלהן.

באנימציות המופעלות על-ידי Scratch משתתפות **דמויות**. הדמויות ב-Scratch נעות על **במה**, שהיא משטח המשתמש במערכת צירים גאומטרית לתיאור הנקודות שבו. מרכז הבמה הוא הנקודה  $(0, 0)$ . בכל רגע נתון לכל דמות ב-Scratch יש **מיקום**, המתואר על-ידי הנקודה שבה היא ממוקמת על הבמה, וכיוון, המתאר את הכיוון שאליו היא פונה.

דמויות ב-Scratch פועלות אך ורק על-פי רצף **הוראות** ברורות. רצף כזה קרוי **תסריט**. התסריטים ב-Scratch נכתבים בחלון האמצעי, ואבני הבניין שלהם הן הוראות הלקוחות מתוך מאגר נתון. עוד הכרנו ב-Scratch את **הדגל הירוק**, שיכול לשמש להפעלת תסריטים, אם בתסריט נאמר במפורש כי ביצעו צריך להתחיל בלחיצה על הדגל הירוק.

אם כך, ניתן לראות הוראה כדרך להפעיל דמות, שאינה פועלת על-פי רצון עצמאי משלה. ישנן הוראות שהן הוראות כלליות, אך ניתן להפכן להוראות ממוקדות על-ידי שילוב ערכים מתאימים בתוכן. כך למשל, **הוראה כללית** לתזוזה יכולה להפוך ל**הוראה ממוקדת** המורה על תזוזה של 10 צעדים או להוראה ממוקדת המורה על תזוזה של 100 צעדים.

ניתן לצרף כמה הוראות זו לזו, ובכך ליצור **סדרת הוראות**, המבוצעות **ביצוע סדרתי**: אחת אחרי האחרת, על-פי הסדר, מלמעלה למטה.

הוראות מסוימות הן **הוראות יחסיות**, שתוצאת ביצוען תלויה הן בערכים המשולבים בהן והן במצב שבו נמצאת הדמות טרם ביצוען. הוראות אחרות הן **הוראות מוחלטות**, שתוצאת ביצוען תלויה אך ורק בערכים המשולבים בהן.

לעיתים מתאים להשתמש בתחילתו של תסריט בהוראה מוחלטת אחת או יותר, כדי לבצע **אתחול** של מצבה של דמות, כלומר הבאתה למקום ולכיוון התחלתיים רצויים.

כאשר אנו פותרים משימה שעומדת לפנינו ועלינו לבנות אנימציה של Scratch העומדת בדרישות המשימה, השלב הראשון בפתרון יהיה כתיבת **תיאור מילולי** מפורט של התנהגותה של כל דמות, תיאור הבנוי **מצעדים**. לאחר שיש בידינו תיאור מפורט דיו, נוכל לתרגמו לתסריט של Scratch על-ידי תרגום כל צעד בתיאור להוראה או להוראות של Scratch. במקרים רבים החלק הקשה של הפתרון הוא למצוא תיאורים מילוליים העומדים בדרישות המשימה.

ההוראות שהכרנו עד כה, מתוך מאגר ההוראות העשיר של Scratch, היו בעיקר הוראות תנועה, שצבען כחול: **תזוזה**, **סיבוב במקום** (עם כיוון השעון או נגדו), **פנייה**, **קפיצה וגלישה**. כל אחת מהן יכולה להפוך להוראות ממוקדות שונות על-ידי שילוב ערכים מתאימים בתוכה. ניתן לשלב הוראות כאלו בכל דרך שנחפוץ לקבלת סדרת הוראות.

הכרנו גם הוראה אחת כתומה, שאינה הוראת תנועה. זו ההוראה המורה על **תחילת ביצוע עם לחיצה על הדגל הירוק**. מתחתיה ניתן לשלב הוראה אחת לביצוע או סדרת הוראות לביצוע באורך כלשהו. נרחיב על הוראה זו בפרק הבא.

**בניית תסריט** מתבצעת על-ידי בחירת לבנים שמתאימות להוראות הדרושות, גרירתן מהחלון השמאלי והנחתן במקום המתאים בחלון האמצעי.



## פרק 2 ריבוי דמויות

באנימציות שיצרנו בפרק הקודם השתתפה דמות אחת בלבד. אבל לעיתים מעניין ליצור אנימציות שבהן משתתפות כמה דמויות. ב־Scratch אפשר לעשות זאת משום שניתן לשתף בכל אנימציית Scratch כמה דמויות, כל דמות ותסריטים משלה. בפרק זה ניצור אנימציות כאלה.

### דוגמה: שתי דמויות ביחד

#### משימה 1

באנימציה יהיו שתי דמויות: החתול שכבר מוכר לנו, ודמות של כלב. שתי הדמויות יחצו את הבמה, **במקביל** (כלומר, באותו הזמן). החתול יחצה את הבמה משמאל לימין, ובאותו הזמן יחצה הכלב את הבמה מלמטה למעלה.



נפרק את האנימציה המתוארת למרכיביה ונתמודד עם כל אחד מהם לחוד. נתחיל עם החתול ונכין עבורו תסריט שלפיו החתול ינוע ממרכז הצלע השמאלית של הבמה, דרך מרכז הבמה, אל מרכז הצלע הימנית של הבמה.

#### יצירת תסריט חדש

בניגוד לדוגמאות שבפרק הקודם, את התסריט הזה ניצור מאפס, לא על בסיס דוגמה קיימת. הפעלה של Scratch מביאה אותנו למצב שבו כבר מוגדרת דמות אחת – דמותו של החתול (הקרויה כעת "דמות1"), אך החלון האמצעי, שבו נכתבים התסריטים, הוא ריק. ניצור בחלון זה תסריט המתאים לרצף התנועות שתוארו קודם.

מה מרכיבי הרצף הרצוי?



ראשית, צריך למקם את החתול בנקודה ההתחלתית המתאימה ובכיוון המתאים ולאחר מכן להניע אותו אל נקודת הסיום.

אילו הוראות של Scratch מתאימות ליצירת תסריט זה?



הוראת קפיצה, הוראת פנייה והוראת גלישה.

הכינו את הוראות התנועה המתאימות, עם הערכים המתאימים המשולבים בתוכן. הקפידו לשלב ערכים מתאימים כך שבנקודת ההתחלה ייראה כל גופו של החתול, ולא רק חלקו, וכך גם בנקודת הסיום.

## קבוצת הוראות הבקרה

כרגיל, להוראות התנועה יש להקדים את ההוראה המורה על תחילת ביצוע עם לחיצה על הדגל הירוק. בפרק הקודם כבר נכללה הוראה זו בדוגמה הראשונה שניתנה לכם. כעת תוכלו להוסיף אותה בעצמכם. מעל חלון ההוראות, שבו מופיע מאגר הלבנים, ישנו חלון ובו מראי מקום לקבוצות נוספות של הוראות, פרט לקבוצת הוראות התנועה הכחולות שאת הוראותיה שילבנו עד כה בתסריטים שיצרנו:



לחצו עם העכבר על כותרת קבוצת הבקרה (צבע כתום). הלבנים המופיעות כעת בחלון מתאימות להוראות ששייכות לקבוצת הבקרה. בחרו בלבנה המתאימה להוראה המוכרת, המורה על תחילת ביצוע עם לחיצה על הדגל הירוק. זוהי הלבנה הראשונה בקבוצת הבקרה.

סדר היצירה אינו משנה: תוכלו לבחור קודם בהוראת הבקרה ואחר כך לשלב מתחתיה את הוראות התנועה, או להכניס קודם לתסריט את הוראות התנועה ולאחר מכן להוסיף מעליהן את הוראת הבקרה. בכל מקרה, ההוראה המורה על תחילת ביצוע עם לחיצה על הדגל הירוק חייבת להיות ראשונה בתסריט המתקבל. על כך מרמזת גם צורתה של הלבנה המתאימה לה, צורה שאינה מאפשרת שילוב לבנה אחרת מעליה.

### הוראה חדשה של Scratch: תחילת ביצוע עם לחיצה על הדגל הירוק

ההוראה **נאשר לוחצים על** יכולה להופיע רק בתחילת תסריט. לחיצה עם העכבר על הדגל הירוק תגרום לביצוע של כל התסריטים שמתחילים בהוראה זו.

התסריט המתקבל הוא זה:



## הוספת דמות חדשה

כעת נרצה להוסיף דמות חדשה. החלון האפור שנמצא בפינה הימנית התחתונה הוא חלון הדמויות, שבו מפורטות כל הדמויות המשתתפות (בחלקו השמאלי מצוינת גם הבמה עצמה, אך לכך לא נתייחס בפרק זה). כעת נמצאת בו רק דמות החתול. נוסיף דמות חדשה על ידי כך שנלחץ על הכפתור האמצעי

מבין השלושה המופיעים מעל החלון **דמות חדשה**: בחירה בכפתור זה מאפשרת לבחור דמות מתוך מאגר הדמויות המסופק על ידי Scratch. מאגר זה מסודר בתיקיות שונות. נבחר בתיקייה הראשונה, תיקיית החיות (Animals), ונגלול את החלון עד שנגיע אל דמויות הכלבים.

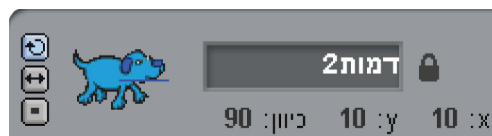
גלילת חלון מתבצעת על ידי הבאת מצביע העכבר אל הפס האנכי שבצידו הימני של החלון, ואז לחיצה רצופה על הכפתור השמאלי של העכבר בד בבד עם הזזת העכבר, עד שהחלון מגיע למקום הרצוי.

בחרו באחת מדמויות הכלבים (מאחר שאנו מייעדים לדמות הכלב תסריט של הליכה כדאי לבחור בדמות המתאימה לכך, דמות b או c).

### פעולה חדשה ב-Scratch: יצירת דמות חדשה

הכפתור האמצעי מעל החלון בפינה הימנית התחתונה מאפשר יצירת דמות חדשה, על ידי בחירתה מהמאגר של Scratch (או מכל תיקייה אחרת).

עם הבחירה בכלב נוספת גם הדמות שלו בחלון הדמויות (ולה השם "דמות2"), וכעת נוכל לכתוב תסריטים גם עבורו. כדי לעבור בין הדמויות ותסריטיהן, פשוט נבחר (בעזרת לחיצה על הכפתור השמאלי של העכבר) בדמות הרצויה בחלון הדמויות. כתוצאה מכך החלון האמצעי מתעדכן ומציג את התסריטים של הדמות שנבחרה. מובן שכעת בחירה בדמות הכלב תגרום להצגת חלון ריק, משום שטרם יצרנו תסריט עבור דמות זו. הדמות שתסריטיה מוצגים כעת בחלון הדמויות מוקפת תמיד במסגרת כחולה. כעת, כשכבר יש לנו יותר מדמות אחת, כדאי לתת להן שמות משמעותיים, כגון "חתול" ו"כלב", או כל שם אחר שתבחרו בו. במרכז החלק העליון של המסך יש חלון המתאר נתונים של הדמות המסומנת, ובין השאר גם את שמה.



נוכל לשנות את השם כרצוננו על ידי סימונו והחלפתו בשם אחר. כעת נוכל להכין עבור הכלב תסריט דומה לזה של החתול שלפיו הכלב ינוע ממרכז הצלע התחתונה של הבמה, דרך מרכז הבמה, אל מרכז הצלע העליונה של הבמה. תכננו וכתבו את התסריט המתאים.

הנה התסריט המתאים לכלב:



כאשר שני התסריטים – של הכלב ושל החתול – מוכנים, לחצו על הדגל הירוק. מה קרה? התסריטים הופעלו במקביל, ושתי הדמויות, הכלב והחתול, נעות במקביל ואגב כך הן נפגשות לרגע במרכז הבמה.



האנימציה שיצרנו כבר אינה מורכבת מתסריט אחד. היא מהווה פרויקט המכיל כמה מרכיבים. במקרה שלנו, שתי דמויות עם תסריט אחד לכל דמות.

הקפידו לתעד כל אחד מהתסריטים על ידי כתיבת הערה מתאימה. כדי ליצור הערה חדשה, הצביעו עם העכבר על שטח ריק כלשהו ברקע של התסריט ולחצו על הכפתור הימני של העכבר. ייפתח תפריט אשר הכניסה השלישית בו היא "הוסף הערה". לחצו עליה עם הכפתור השמאלי של העכבר. כעת ייפתח חלון צהוב שבו תוכלו לרשום את ההערה החדשה.

שמרו את הפרויקט בשם חדש, בתיקה ch2 המתאימה לפרק 2.

### מושג חדש: ביצוע מקבילי – דמויות הפועלות במקביל

כאשר אנו מנחים כמה דמויות כיצד לפעול, כל אחת על פי התנהגות משלה, מתקבל ביצוע מקבילי, שבו כל הדמויות פועלות באותו זמן. כל דמות פועלת על פי ההנחיות שקיבלה, בלי להמתין לדמות אחרת ובלי להיות תלויה בה.

**תרגיל 1 – תפוס את החתול:** בנו אנימציה שבה הן החתול והן הכלב מתחילים מהפינה השמאלית התחתונה וגולשים אל הפינה הימנית העליונה, כך שהכלב רודף אחרי החתול. כדי ליצור רושם של רדיפה, פצלו את תנועתו של הכלב לשתי הוראות גלישה: האחת מהפינה השמאלית התחתונה אל מרכז הבמה, והשנייה ממרכז הבמה אל הפינה הימנית העליונה. בראשונה יגלוש הכלב בקצב איטי, ובכך יאפשר לחתול להתרחק מעט. עם הגיעו למרכז הבמה יאיץ הכלב ויתפוס את החתול בפינה הימנית העליונה. את הקצב השונה תוכלו ליצור על ידי שינוי זמן הגלישה בכל הוראה. זמן גלישה ארוך יותר משמעו קצב איטי יותר.

שם הקובץ: dog-chases-cat

## הרחבה (רשות): תפריט פעולות העכבר

בדוגמה האחרונה התנהגות הכלב הייתה דומה מאוד להתנהגות החתול. למעשה, התסריטים היו זהים פרט לערכים המשולבים בהוראות השונות. כאשר לשתי דמויות התנהגות דומה או כאשר התנהגות של דמות מסוימת צריכה לחזור על עצמה שוב בהמשך אותו תסריט, נוח לשכפל קטעי תסריטים שלמים ולהעתיקם למקום הרצוי.

### שכפול


כדי לבצע פעולת שכפול נוכל להיעזר בתפריט הנמצא בחלקו העליון של המסך, מעל לבמה





. הכפתור השמאלי, עם ציור החותמת, משמש לשכפול. אם נלחץ עליו, מצביע העכבר ישנה את צורתו, ובמקום חץ, תהיה צורתו כשל חותמת. כעת נוכל לבחור קטע תסריט (בחירה בהוראה מסוימת תגרום לבחירת **כל קטע התסריט ממנה והלאה**). אם אנו מעוניינים בשכפול הקטע עבור אותה דמות, נוכל לגרור את הקטע שבחרנו אל המקום הרצוי בחלון התסריטים. אם אנו מעוניינים בשכפול הקטע עבור דמות אחרת נוכל לגרור את הקטע הנבחר אל הדמות הרצויה בחלון הדמויות. כעת נוכל לעבור אל חלון התסריטים של דמות זו ולראות כי התסריט ששוכפל מופיע שם.

באותו כפתור אפשר גם להשתמש כדי לשכפל דמות, על ידי בחירה בה על הבמה. תיווצר דמות חדשה עם שם חדש, אך עם אותם תסריטים.

#### פעולה חדשה ב־Scratch: שכפול קטעי תסריטים ושכפול דמויות

הכפתור השמאלי מעל לבמה  מאפשר שכפול תסריט או חלק מתסריט. יש ללחוץ על הכפתור, לבחור קטע מתסריט ולגרור אותו למקום אחר בחלון התסריטים או אל דמות אחרת. כפתור זה משמש גם לשכפול דמות (על כל תסריטיה) על ידי בחירתה על הבמה.

### מחיקה, הגדלת דמות והקטנתה

בתפריט זה ישנם כפתורי עריכה נוספים, המשמשים למחיקה של דמויות או של הוראות מתסריטים (כפתור המספריים), להקטנת דמויות (הכפתור הימני) ולהגדלתן (הכפתור השני מימין). שימו לב, אחרי ביצוע שכפול או מחיקה, מצביע העכבר חוזר לצורתו הרגילה. אם ברצוננו לבצע שכפול או מחיקה נוספים, יש ללחוץ שוב על הכפתור המתאים (חותמת או מספריים). לא כך הדבר לגבי הקטנה או הגדלה של דמויות. בחירה באחד מכפתורים אלה תגרום גם היא לשינוי צורתו של מצביע העכבר, לסימון המתאים להקטנה () או הגדלה (). המצביע יישאר בצורתו זו גם אחרי שנבצע כמה הקטנות או הגדלות. כדי שנוכל לחזור לעבודה רגילה, עלינו להחזירו למצבו הרגיל על ידי לחיצה עם העכבר במקום כלשהו שאין בו דמות.

**פעולות חדשות ב־Scratch:** הגדלת דמות והקטנתה, מחיקת דמות ומחיקת הוראות מתסריט שני הכפתורים הימניים מעל לבמה מאפשרים להגדיל או להקטין את ממדיה של דמות על הבמה. יש ללחוץ על אחד הכפתורים ואז ללחוץ (אפילו כמה פעמים) על הדמות שעל הבמה. כדי למחוק דמות, יש ללחוץ על הכפתור השני משמאל ואחר כך על הדמות. בעזרת כפתור המחיקה ניתן גם למחוק הוראות מתסריט על ידי בחירה בהן.

התנסו בכפתורים השונים, אך לא לפני שמירת הפרויקט, כדי שיהיה בידיכם עותק שמור שאינו מושפע מהשינויים שייגרמו כתוצאה מכך.

## הרחבה (רשות): שינוי מראה דמות

גם כאשר אנו משתמשים בדמויות הנתונות על ידי Scratch, יש לנו האפשרות לשנות את מראה של דמות נקרא ב־Scratch **תלבושת**.

### משימה 2

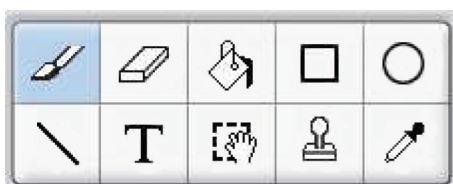
שם הקובץ: costumes

שנו את התלבושות של החתול והכלב.



בחרו בדמות החתול. מעל החלון האמצעי שבו מופיע התסריט של החתול, ישנן שלוש לשוניות (כפי שאכן קורה כעת), מוצגים בחלון זה התסריטים של הדמות. ניתן גם לבחור בלשונית אחרת. בחרו בלשונית התלבושות (על ידי לחיצה עליה עם העכבר). החלון הפך להיות חלון שבעזרתו ניתן לקבוע את המראה של הדמות.

בחרו בחלון זה בדמות החתול הראשונה ואז לחצו על הכפתור "ערוך". כתוצאה מכך ייפתח חלון "ערוך הצויר", הדומה לחלון של תוכנת הצייר (paint) ב־Windows. נדגים כעת שינוי אפשרי במראהו של החתול, ואחר כך תוכלו להתנסות בשינויים אחרים, בדמות החתול ובדמות הכלב. בחלקו השמאלי של החלון יש שתי שורות, עם חמישה כפתורים בכל אחת:



הכפתור האמצעי בשורה העליונה הוא כפתור המילוי, המראה פחית צבע נטויה וצבע נשפך ממנה. בחרו בו. כעת בחרו צבע מלוח הצבעים שבתחתית החלק השמאלי. הזיזו את העכבר אל דמות החתול שבחלקו הימני של החלון, ולחצו עם העכבר בנקודה כלשהי בחלק הכתום שבראש החתול. כתוצאה מהלחיצה משתנה הצבע הכתום לצבע שבו בחרתם. בחרו בצבע אחר, ולחצו על אפו השחור של החתול וראו כיצד צבעו משתנה לצבע שבו בחרתם. אם טעיתם או התחרטתם על בחירה שלקחתם, תוכלו להשתמש בכפתור "בטל" שבחלקו השמאלי העליון של החלון.

### פעולה חדשה ב־Scratch: עריכת תלבושת

בחלון התסריטים לחצו על הלשונית "תלבושות", ולאחר מכן לחצו על הכפתור "ערוך" עבור אחת מהתלבושות של הדמות. ייפתח חלון המאפשר עריכת התלבושת.

בפרק זה ראינו כי כמה דמויות יכולות לקחת חלק באנימציה של Scratch, כל אחת עם תסריטים משלה. לחיצה על הדגל הירוק תגרום להפעלה **מקבילית** של כל הדמויות.

פרט למאגר הוראות התנועה, שלבניהן צבועות בכחול, השתמשנו בפרק זה גם במאגר **הוראות הבקרה**, ההוראות עם הלבנים הכתומות, כדי לשלב את ההוראה שמתנה תחילת ביצוע תסריט עם לחיצה על הדגל הירוק.

יצרנו בפרק זה **תסריט חדש** – על ידי גרירת הלבנים המתאימות להוראות הרצויות אל חלון התסריטים, החלון האפור המוארך שבמרכז.

ליצירת **הערה חדשה** השתמשנו בלחיצה על הכפתור הימני של העכבר, בשטח ריק בחלון התסריטים, ובבחירת החלופה "הוסף הערה".

הוספנו **דמות חדשה** על ידי לחיצה על הכפתור האמצעי מבין שלושת הכפתורים שמעל חלון הדמויות. **למעבר בין דמויות** ותסריטיהן מספיק לבחור בדמות הרצויה בחלון הדמויות. **שינוי שם של דמות** מתבצע על ידי הקלדת שם חדש במקום השם המופיע בחלון שמתאר את פרטי הדמות ושנמצא במרכז החלק העליון של מסך ה-Scratch.

**תפריט פעולות העכבר**, הנמצא בחלקו העליון של מסך ה-Scratch, מאפשר על ידי בחירה בכפתור המתאים **להעתיק** חלקי תסריט או דמויות (כפתור שמאל), **למחוק** הוראות מתסריט או דמויות (כפתור שני משמאל), **להגדיל דמויות** (כפתור שני מימין) ו**להקטין דמויות** (כפתור ימני).

לדמות ב-Scratch יש **מראה (תלבושת)**. ניתן לערוך ולשנות את תלבושתה של דמות על ידי בחירה בלשונית התלבושות שמעל החלון המרכזי המוארך. עריכת תלבושת יכולה להתבצע בעזרת **עורך הציור**.

## פרק 3

### תסריטים קצרים וביצועים ארוכים

היכרותנו עם Scratch קצרה למדיי, ובכל זאת יצרנו בשני הפרקים הראשונים מגוון מעניין של אנימציות. הן כללו תנועות מסוגים שונים ואפילו שילוב של כמה דמויות. אבל את כל האנימציות שיצרנו עד כה ניתן לאפיין כך: כולן נבנו מתסריטים שכללו סדרה סופית של הוראות תנועה, סדרה שבוצעה הוראה אחרי הוראה, עם לחיצה על הדגל הירוק. אמנם הסדרות עצמן היו כאמור מגוונות למדיי – הן כללו הוראות מסוגים שונים, חלקן יחסיות וחלקן מוחלטות, והיו באורכים שונים – מתסריט פשוט שכלל סדרה בת הוראה אחת ועד תסריט שכלל סדרה של תשע הוראות. ובכל זאת, המבנה הכללי של כל התסריטים היה דומה מאוד, והאנימציה שהם יצרו הייתה בעלת מאפיינים דומים: סדרה סופית של תנועות שבוצעו זו אחר זו, תמיד באותו סדר. אורך צעדי סדרת התנועות המבוצעות היה תמיד כאורך סדרת הוראות התנועה שהנחתה אותה.

בפרק זה נרצה להעשיר עוד יותר את האנימציות שניצור. זאת נעשה על-ידי כך שנאפשר שילובים חדשים של הוראות בסיסיות – לא רק כסדרה סופית של הוראות המיועדות לביצוע בזו אחר זו, אלא גם בדרכים אחרות.

### דוגמה 1: ביצוע אינסופי

האנימציה הראשונה שניצור בפרק זה תהיה אנימציה אינסופית, שנמשכת עוד ועוד. כמובן, אין לנו אפשרות טכנית לכתוב תסריט המורכב מסדרת הוראות אינסופית. אך נלמד כיצד ליצור ביצוע אינסופי בעזרת סדרת הוראות סופית. כבסיס ניקח את האנימציה האחרונה שיצרנו בפרק הקודם, שבה חוצים החתול והכלב את הבמה במקביל, האחד משמאל לימין והאחר מלמטה למעלה.

#### משימה 1

שנו את האנימציה שנבנתה בפרק הקודם כך שייוצר רצף תנועותי שבו החתול והכלב חוצים את הבמה הלך וחזור, שוב ושוב. החתול משמאל לימין, מימין לשמאל, משמאל לימין וכך הלאה, והכלב מלמטה למעלה, מלמעלה למטה, מלמטה למעלה וכך הלאה.



שם הקובץ: `meet-forever`

בדומה למקרים קודמים, נפרק את המשימה העומדת בפנינו לחלקים ונתמודד עם כל אחד מהם לחוד. ראשית, ננתח את תנועתו של החתול.

כצעד ראשון, נרצה למקם את החתול בנקודה ובכיוון ההתחלתיים, בדומה למה שנעשה באנימציה האחרונה של פרק 2.

לאחר מכן נרצה לבנות סדרת הוראות היוצרת תנועה של הלך וחזור, הלך וחזור, כך שעם כל הגעה לקצה הבמה, ייסוב החתול ויפנה לכיוון ההפוך. את הרצף הארוך הזה נבנה מצעדים קטנים שיחזרו על עצמם שוב ושוב. בכל צעד יפסע החתול פסיעה קטנה, למשל, 10 צעדים. אחרי כל פסיעה כזו, יבדוק

החתול אם הגיע למסגרת הבמה, ואם אכן כך, יהפוך את כיוון תנועתו. על הפסיעה והבדיקה האלו יחזור החתול שוב ושוב.

נוכל לתאר זאת מילולית באופן הבא:

0. אם אחיזה על הדגל הימני:
1. קפץ אל נקודת ההגמלה  $(0, -200)$
2. פנה לכיוון ההגמלה  $(90)$
3. חזר שוב ושוב:
- 3.1. 10 שניות
- 3.2. אם הגענו למסגרת הבמה, הפוך כיוון

הידע שרכשנו עד כה מאפשר לנו לתרגם את צעדים 0, 1, 2 ו-3.1 להוראות של Scratch. כיצד נתרגם את צעדים 3 ו-3.2?

עבור צעד 3.2 יש הוראת תנועה מוכנה של Scratch: **בוגיעה במסגרת הפוך כיוון**.

### הוראה חדשה ב-Scratch: הפוך כיוון בעקבות נגיעה במסגרת הבמה

ההוראה **בוגיעה במסגרת הפוך כיוון** גורמת לדמות הנמצאת בתנועה להפוך את כיוון תנועתה כאשר היא נוגעת במסגרת של הבמה.

## ביצוע חוזר של הוראות

צעד 3 מתאר **ביצוע חוזר**, כלומר, רצף הוראות אשר מבוצעות יותר מפעם אחת. כדי לתרגם את צעד 3, דרושה לנו הוראה שאינה משפיעה על האנימציה באופן ישיר, אלא גורמת להוראות אחרות להתבצע שוב ושוב. המקום המתאים לחיפוש הוראה כזו הוא בקבוצת הוראות הבקרה (ההוראות הכתומות), שתפקידן לקבוע את האופן שבו מתבצעות הוראות אחרות בתסריט. נעבור לקבוצה הכתומה ונבחר מתוכה בהוראה



שימו לב כי הלבנה המתאימה להוראה זו שונה בצורתה מלבנים אחרות שבהן נתקלנו עד כה. הלבנים שבהן השתמשנו עד כה היו מלבניות ובהן שקעים ובליטות המאפשרים את שילובן זו מתחת לזו. יוצאת דופן היא הלבנה שמתנה ביצוע בלחיצה על הדגל הירוק. לבנה זו נועדה להיות תמיד ראשונה בתסריט, ולכן אין שקע בחלקה העליון. ללבנה החדשה יש שקע בחלקה העליון המאפשר לשלב אותה לאחר הוראות אחרות, אך אין בליטה בחלקה התחתון. בנוסף היא נראית כמין פה פעור לצד שמאל, ומשום כך היא יכולה למסגר ולעטוף הוראות אחרות, ההוראות שיבוצעו שוב ושוב.



כדי לראות כיצד היא משתלבת בתסריט, בחרו בלבנה וגררו אותה אל חלון התסריטים. בחרו בלבנה הכחולה המתאימה לצעד 3.1, וגררו אותה אל תוך הלבנה הכתומה. כעת בחרו בלבנה

הכחולה המתאימה לצעד 3.2, וגררו אותה אל מתחת להוראת התזוזה, בתוך הלבנה הכתומה. קיבלנו לבנה כתומה המתאימה להוראת החזרה שוב ושוב, העוטפת בתוכה שתי לבנים כחולות, המתאימות להוראת התזוזה ולהוראת היפוך הכיוון המותנה. המשמעות היא ששתי הוראות התנועה יבוצעו בזו אחר זו, שוב ושוב.



צרו את התסריט השלם, שהוא תרגום של סדרת הצעדים 0 עד 3.2. נסו להפעילו. שימו לב: הדרך היחידה לעצור את האנימציה היא ללחוץ על סימן העצור האדום, שמופיע ליד הדגל הירוק.

כפי שהזכרנו, ישנו הבדל חשוב נוסף בין צורתה של הלבנה המתאימה להוראת הביצוע האינסופי לצורתן של לבנים אחרות שבהן השתמשנו עד כה: חלקה התחתון הוא ישר, ולכן בתסריט Scratch לא ניתן לשלב הוראות נוספות **לאחר** הוראת הביצוע האינסופי. הסבירו את הסיבה לכך.



צרו עבור הכלב תסריט דומה. ההבדל היחיד בין שני התסריטים הוא בנקודת ההתחלה ובכיוון ההתחלתי. נקודת ההתחלה והכיוון ההתחלתי של הכלב צריכים להיות כפי שהיו במשימה האחרונה של הפרק הקודם: במרכז הצלע התחתונה של הבמה, בכיוון מעלה.

נסו ללחוץ על הדגל הירוק ולראות את שתי הדמויות נעות במקביל ברצף אינסופי. שמרו את האנימציה שיצרתם כפרויקט חדש.

**שימו לב:** בכל אחד משני התסריטים שיצרנו יש סדרת הוראות סופית. כל תסריט מורכב בסך הכול משש הוראות. אבל הביצוע שיוצר כל תסריט הוא אינסופי. זאת משום שאחת ההוראות הכלולות בתסריט היא הוראת חזרה אינסופית, במקרה זה – חזרה אינסופית על שתי הוראות תנועה.



#### מושג חדש: ביצוע אינסופי

בביצוע אינסופי סדרה באורך סופי של הוראות כלשהן מתבצעת שוב ושוב. לכן למרות שמספר ההוראות הוא סופי, הביצוע המתקבל הוא אינסופי ואינו מסתיים.

#### הוראה חדשה ב־Scratch: ביצוע אינסופי

ההוראה גורמת להוראות של Scratch המשולבות **בתוך** ההוראה, להתבצע שוב ושוב.





## תרגיל 1:

החתול חושש מהכלב ומחליט לקרוא לאחיו הגדול כדי שיפחיד את הכלב. הכלב מתייצב מולו ללא חשש, ואף נובח, אבל הכוחות שקולים, ומאזן ההפחדה בין השניים נמשך עוד ועוד, ללא הכרעה.

באנימציה שיש ליצור בתרגיל זה משתתפים הכלב, וחתול גדול יותר, כמו זה שניתן לייבא מתיקיית החיות (Animals) ונקרא שם cat4. בתחילת האנימציה מתייצב החתול הגדול במרכז הבמה ופונה לכיוון הכלב. הכלב קופץ סביבו שוב ושוב, פעם מתחת לחתול, פעם לשמאלו, פעם מעל החתול, ופעם לימינו וחוזר חלילה. בכל פעם הכלב פונה לכיוון החתול ונובח עליו. החתול מצידו פונה בכל פעם אל הכלב (הוא לא צריך אפילו ליילל, מראה פניו לבד כבר מפחיד דיו). הריקוד המוזר הזה נמשך עוד ועוד.

הכינו תיאורים מילוליים מתאימים להתנהגות הכלב ולהתנהגות החתול הגדול. תרגמו אותם לתסריטים של Scratch. לשם כך תרצו בוודאי להשתמש בהוראה של Scratch המנחה דמות לפנות לכיוונה של דמות אחרת. לבנה מתאימה להוראה זו נמצאת בקבוצת התנועה, **פנה לכיוונו של**. בחלונית של ההוראה ניתן לשלב כל אחת מהדמויות האחרות בפרויקט, או את מצביע העכבר. אלו כלולים ברשימה הנפתחת עם לחיצה על החץ הקטן שבחלונית.

שם הקובץ: big-cat

בדקו את הפרויקט המתקבל ותעדו אותו.

## דוגמה 2: ביצוע אינסופי משולב בבדיקה

באנימציה האחרונה שיצרנו חלפו הכלב והחתול זה על פני זה, בלי להתייחס זה לזה. כמובן, חלק חשוב המוסיף עניין לאנימציות הוא זה הנוגע לקשר בין הדמויות המשתתפות ולדרך שבה הן מגיבות זו לזו ננסה להוסיף גם ממד כזה לאנימציות שלנו.

### משימה 2

יש ליצור אנימציה שבה הכלב והחתול נעים ממרכז הבמה כלפי שולי הבמה. הכלב ינוע ממרכז הבמה כלפי מעלה, והחתול ינוע ממרכז הבמה ימינה. כל עוד הם קרובים מאוד זה לזה, נוגעים זה בזה, הם יתייחסו באופן כלשהו האחד לרעהו (למשל, על-ידי דיבור) וימשיכו בתנועתם. כשיהיו רחוקים מספיק זה מזה, הם יפסיקו לדבר ויעצרו במקומם.



שם הקובץ: say

כדי להדגיש את ההתייחסות ההדדית באנימציה, נרצה להאריך את משך הזמן שבו הם קרובים זה לזה. משום כך נאט את התקדמותם, כך שתתבצע בתזוזות של 5 צעדים במקום 10 צעדים.

אם כן, גם הפעם אנו מעוניינים בביצוע חוזר – חזרה שוב ושוב על תזוזה ודיבור – אך הפעם ישולב הביצוע החוזר בבדיקה מתמידה של הקירבה בין הדמויות. אם אכן מתקיימת קירבה, תימשך התנועה, וכל אחת מהדמויות תאמר משהו.



בפנינו את האפשרויות שמתוכן אפשר לבחור מה לשלב בתנאי (כלומר, איזו נגיעה יש לבדוק). במקרה זה האפשרות המתאימה היא בדיקת נגיעה בדמות הכלב, dog. ישנן גם אפשרויות אחרות המתאימות אולי לתרחישים אחרים.

#### מושג חדש: תנאי

תנאי מייצג שאלה שהתשובה עליה יכולה להיות "כן" או "לא", או מייצג מאורע שיכול להתקיים או לא להתקיים.

#### מושג חדש: ביצוע אינסופי משולב בבדיקה

בביצוע אינסופי המשולב בבדיקה נבדק תנאי שוב ושוב. בכל פעם שהתנאי מתקיים, מתבצעת סדרת הוראות סופית. לכן למרות שמספר ההוראות הוא סופי, ואף אם התנאי אינו מתקיים אף פעם, הביצוע המתקבל הוא אינסופי ואינו מסתיים.

#### הוראה חדשה ב-Scratch: ביצוע אינסופי משולב בבדיקה



ההוראה גורמת לתנאי המופיע בתוך החלונית להיבדק שוב ושוב. כל בדיקה שבה התנאי מתקיים גורמת להוראות של Scratch הנמצאות בתוך הוראת הבקרה להתבצע.



#### תנאי חדש ב-Scratch: תנאי נגיעה

התנאי  מייצג שאלות שהתשובות עליהן עשויות להיות:



"כן" – אם הדמות שבתסריטה משולב התנאי נוגעת בעצם המופיע בחלונית שבתנאי;


"לא" – אם הדמות אינה נוגעת בעצם זה.

### דמות "מדברת"

כיצד נתרגם את צעד 3.2? בקבוצת המראה, הקבוצה הסגולה, נמצא שתי הוראות המאפשרות לדמות "לדבר", כלומר, להראות בלון דיבור (כמו הבלונים המוכרים לנו מקומיקס), ובו המלל שאותו היא "אומרת". הראשונה מבין שתי ההוראות היא . נוכל לשלב בה את המלל הרצוי ולקבוע למשך כמה זמן יישאר בלון הדיבור על הבמה. ההוראה השנייה היא , הגורמת להופעת בלון דיבור שמכיל את המלל הרצוי, ונשאר על הבמה עד סיום ביצוע האנימציה.

## הוראות חדשות ב־Scratch: דיבור

ההוראה **אמר**  **למשך**  **שניות** גורמת להופעת בלון דיבור מעל לדמות, ובתוכו ייראה תוכן החלונית הראשונה. הבלון יופיע למשך מספר השניות המשולב בחלונית השנייה.

ההוראה **אמר**  גורמת להופעת בלון דיבור מעל לדמות, ובתוכו ייראה תוכן החלונית הראשונה. הבלון יישאר עד סיום ביצוע האנימציה.

השתמשו בלבנה המתאימה להוראת דיבור מוגבלת בזמן, כדי ליצור הוראה המורה לחתול לומר "הצילו" למשך חצי שנייה.

כעת השלימו את יצירת קטע התסריט המתאים לצעד 3, כולל התנאי וצעדים 3.1 ו־3.2 המשולבים בו, והשלימו את התסריט כולו, כולל הוראות המתאימות לצעדים 0, 1, ו־2.

הכינו תסריט מקביל עבור הכלב, שבו הוא נע ממרכז הבמה לכיוון מעלה, וכל עוד הוא נוגע בחתול, הוא אומר "יאמי" למשך חצי שנייה.

הפעילו את האנימציה החדשה על ידי לחיצה על הדגל הירוק. תעדו ושמרו את האנימציה החדשה בפרויקט חדש, בלי לפגוע בפרויקט המתאים לדוגמה 1.

## דוגמה 3: ריבוי תסריטים

כבר יצרנו כמה אנימציות שבהן השתתפו שתי דמויות, כל אחת והתנהגות שלה. בכל אחת מהאנימציות שיצרנו, ההתנהגות של כל דמות נקבעה על פי תסריט אחד שנכתב עבורה, כל דמות והתסריט שלה. אולם ב־Scratch ניתן לכתוב לכל דמות יותר מתסריט אחד. דבר זה נוח כאשר אנו רוצים להשפיע על התנהגותה של דמות בכמה דרכים. למשל, כעת נרצה ליצור אנימציה שמשלבת את שתי האנימציות הקודמות.

### משימה 3

יש ליצור אנימציה שבה החתול והכלב ינועו ברצף תנועתי אינסופי – האחד מאמצע הצלע השמאלית של הבמה ימינה, והאחר מאמצע הצלע התחתונה של הבמה מעלה, ובכל פעם שהם נפגשים, יאמר החתול "הצילו", והכלב יאמר "יאמי". **שם הקובץ:** `meet-forever-and-say`



כנקודת התחלה, נוכל לנצל עבור כל דמות את התסריט מהדוגמה הראשונה, שיוצר את התנועה האינסופית מעלה ומטה (עבור הכלב) וימינה ושמאלה (עבור החתול). בנוסף ניצור לכל דמות תסריט שבו היא בודקת שוב ושוב אם נתקלה בדמות השנייה. גם שני התסריטים החדשים ייכתבו כך שביצועם מתחיל עם לחיצה על הדגל הירוק. לכן לחיצה על הדגל הירוק תפעיל **במקביל** ארבעה תסריטים: שני התסריטים הישנים, שני התסריטים החדשים, ובסך הכול שני תסריטים עבור החתול, ושניים עבור הכלב.

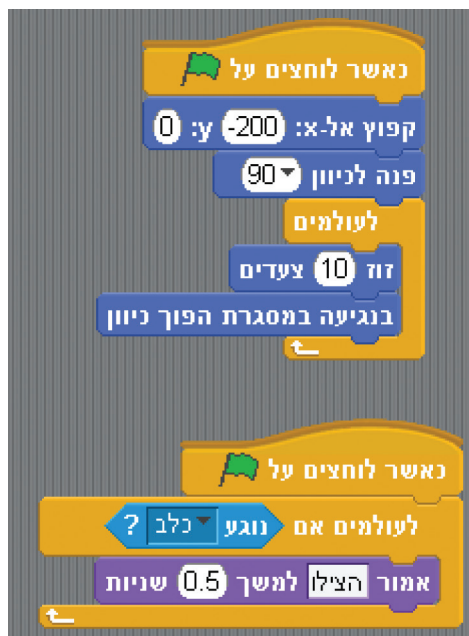
כאמור, עבור כל דמות נוכל להשתמש בתסריט כפי שהוא מופיע בדוגמה הראשונה. לשם כך פתחו את הפרויקט המתאים לדוגמה הראשונה ושמרו אותו בשם חדש. כעת הפרויקט החדש כבר מכיל את שני התסריטים הישנים, תסריט עבור כל דמות, ונותר רק להוסיף את שני התסריטים החדשים.

## התסריט האחראי על הדיבור

נפנה לכתובת התסריט השני עבור החתול (את התסריט המקביל עבור הכלב תוכלו להשלים אחר כך בעצמכם). תסריט כזה צריך לתאר בדיקה חוזרת ונשנית (האם נתקלתי בכלב?), ובהתאם לתוצאת הבדיקה אמירת "הצילו". כלומר, התיאור המילולי המתאים לתסריט הרצוי הוא זה:

0. אם אני צה על הדגל היחיד:
1. אלולאים, אם אלה נעל כאלה
- 1.1. אמור "הצילו" אם צי שנייה

למעשה, זה בדיוק התסריט מהדוגמה השנייה, אך ללא הוראת התזוזה (הוראה 3.1 מהתסריט המקורי). את הוראת התזוזה השמטנו, משום שהתסריט הראשון הוא זה שמטפל בתזוזה, ואילו התסריט השני מטפל רק בדיבור בהתאם לקירבה לכלב.



בחלון התסריטים של החתול, שבו נמצא כבר התסריט המתאים לתנועה האינסופית ימינה ושמאלה, הוסיפו את התסריט השלם המטפל באמירת "הצילו" עם ההיתקלות בכלב. תוכלו לעשות זאת פשוט על-ידי גרירת הלבנים החדשות אל מקום פנוי בחלון התסריטים. בחלון התסריטים של הכלב, הוסיפו את התסריט המקביל עבור הכלב. כעת לכל דמות, מופיעים בחלון התסריטים שלה שני תסריטים, למשל כך:

לחצו על הדגל הירוק וצפו באנימציה המתקבלת. תעדו ושמרו אותה בפרויקט חדש, בלי לפגוע בפרויקט המתאים לדוגמה 2.

### מושג חדש: ביצוע מקבילי – תסריטים הפועלים במקביל

כאשר דמות מונחית כיצד לפעול על-ידי יותר מתסריט אחד, מתקבל ביצוע מקבילי, שבו כל התסריטים מתבצעים באותו זמן.

### תרגיל 2:

באנימציה שיש ליצור בתרגיל זה משתתפים בחורה, שמש ומשקפי שמש. תוכלו לייבא דמויות מתאימות מתיקיית האנשים (People), מתיקיית הדמיון (Fantasy) ומתיקיית החפצים (Things). השמש נעה, אבל הבחורה מצוידת במשקפי שמש קסומים, ולכן היא יכולה להרשות לעצמה לשבת כל העת במקומה, ללא תזוזה. משקפי השמש עומדים

דרוכים ומוכנים. בכל פעם שהם חשים בשמש הם ממהרים להגן על עיניה של הבחורה, וחוזרים למקומם.

הנה תיאור מפורט יותר: הבחורה יושבת במקום קבוע על הבמה, למשל הנקודה (0, -150). השמש נעה כל הזמן מעלה ומטה, מתחתית הבמה אל גבולה העליון, ושוב למטה וכך הלאה. מסלולה מתחיל מהנקודה (-130, -50) לכיוון מעלה, ואז עובר שוב ושוב בין גבולה העליון של הבמה לגבולה התחתון. המשקפיים עומדים מוכנים לימינה של הבחורה, במסלולה של השמש (למשל, בנקודה (-50, 50)). בכל פעם שהמשקפיים חשים בשמש, הם ממהרים לכסות את עיניה של הבחורה, ואז חוזרים למקומם, לעמדת השמירה שלהם.

הכינו תיאורים מילוליים מתאימים להתנהגותם של הבחורה, השמש והמשקפיים. תרגמו אותם לתסריטים של Scratch. בדקו את הפרויקט המתקבל ותעדו אותו.

שם הקובץ: sunglasses

## הרחבה (רשות): שינוי רקע

גם בפרק זה נראה שימוש בכלים שסביבת Scratch נותנת לנו, כדי ליצור אנימציות צבעוניות ומעניינות.

### משימה 4

באנימציה האחרונה של החתול והכלב, שנו את הרקע של הבמה.

שם הקובץ: background



הסתכלו בחלון הדמויות, שנמצא בפינה הימנית התחתונה. מופיעות בו דמויות החתול והכלב, ובחלקו השמאלי מופיע ריבוע המתאים לבמה עצמה. לחצו על הריבוע. כעת החלון האפור והמוארך שבמרכז מתייחס אל הבמה, ובעזרתו אפשר לשלוט על מאפיינים שונים של הבמה, ובין השאר גם על הרקע שלה.

לחצו בחלון זה על הלשונית "רקעים" (הלשונית האמצעית). כעת באפשרותכם לצייר רקע, לייבא רקע קיים או להשתמש במצלמה לצילום תמונה שתשמש כרקע. אם נבחר בשימוש ברקע קיים, יפתח חלון דפדוף שבעזרתו נוכל לחפש בין הרקעים הקיימים במערכת. בחרו, למשל, בתיקייה המכילה רקעים של טבע (Nature), ומתוכה בחרו ברקע כלשהו, למשל – רקע היער (forest). אשרו את הבחירה. כעת לבמה יש רקע של יער, והוא מופיע גם בחלון הדמויות, בחלק המתאים לבמה עצמה. החלון האפור המוארך שבמרכז מונה את כל הרקעים המוגדרים, ולכן הוא מכיל כעת הן את הרקע הישן, הלבן, והן את הרקע החדש, רקע היער. תוכלו למחוק את הרקע הלבן על ידי לחיצה על הכפתור העגול עם סימן האיקס המופיע ברשימת הרקעים ליד הרקע הלבן.

### פעולה חדשה ב-Scratch: שינוי רקע

עבור דמות הבמה לחצו בחלון התסריטים על הלשונית "רקעים". ניתן לצייר רקע חדש, לייבא מקובץ או להיעזר לשם כך במצלמה על ידי לחיצה על אחד משלושת הכפתורים המופיעים בחלקו העליון של החלון.

שמרו את האנימציה והפעילו אותה על ידי לחיצה על הדגל הירוק.

## סיכום

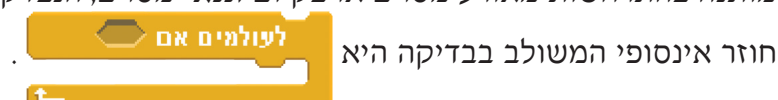
בפרק זה ראינו כיצד ניתן ליצור אנימציות שאורכן שונה, ואפילו שונה משמעותית, מאורך רצף ההוראות שמתאר אותן.

הכרנו כמה הוראות מתוך קבוצת **הוראות הבקרה**, ההוראות שמתארות כיצד ומתי יש לבצע הוראות אחרות בתסריט. הלבנים המתאימות להן מקובצות בקבוצת הבקרה, הקבוצה הכתומה.

**ביצוע חוזר** מאפשר לבצע הוראה או סדרת הוראות שוב ושוב, ברצף אינסופי. ההוראה המתאימה לביצוע



**ביצוע חוזר אינסופי משולב בבדיקה** מאפשר לבצע הוראה או סדרת הוראות שוב ושוב, אך ביצוען מותנה בהתרחשות מאורע מסוים או בקיום **תנאי** מסוים, הנבדק שוב ושוב. ההוראה המתאימה לביצוע



חוזר אינסופי המשולב בבדיקה היא עוד ראינו בפרק זה, כי התנהגותה של דמות ניתנת לתיאור משולב על-ידי **ריבוי תסריטים** העובדים במקביל. חלון התסריטים של דמות יכול להכיל יותר מתסריט אחד.

בנוסף, דרך האנימציות שנכללו בפרק הכרנו רכיבים נוספים של Scratch:

הוראת התנועה **בנגיעה במסגרת היפוך כיוון** מקשרת מאורע עם הוראת תנועה – מאורע **נגיעה במסגרת** עם הוראת **היפוך כיוון**. ההוראה מאפשרת לזהות מאורע של נגיעה במסגרת ובעקבותיו להפוך את כיוון התנועה.

הוראת הדיבור **אמור** **למשך** **שניות** גורמת **להצגת בלון מלל** עם התוכן המבוקש, למשך הזמן המבוקש. הוראת הדיבור **אמור** גורמת **להצגת בלון מלל** עם התוכן המבוקש, עד סיום ביצוע האנימציה. ניתן למצוא את הלבנים המתאימות בקבוצת **המראה הסגולה**.

ל**זיהוי מאורע נגיעה** מתאים התנאי **נוגע** **?**, שבדק בעזרת חיישני הדמות אם היא נוגעת בדמות אחרת, במסגרת או במצביע העכבר. ניתן למצוא את הלבנה המתאימה בקבוצת **החיישנים**, הקבוצה הכחולה-בהירה. שילובו של תנאי זה בהוראות בקרה מסוימות (כמו הוראת ביצוע אינסופי משולב בבדיקה, שהכרנו בפרק זה) מאפשר להתנות ביצוע הוראות בתסריט בהתרחשות מאורע נגיעה. ב-Scratch יש **ללבנת תנאי** צורה מלבנית עם פינות קטומות.

**שינוי רקע** של הבמה ניתן לבצע על-ידי בחירה בבמה בחלון הדמויות ואז בחירה בלשונית הרקעים בחלון התסריטים. ניתן לצייר רקע, לצלם או לייבא רקע מתוך מבחר נתון.

## פרק 4

### תקשורת בין דמויות

בפרק הקודם יצרנו שתי אנימציות שבהן הייתה התייחסות הדדית כלשהי בין הדמויות: היתקלות של הכלב והחתול יצרה תגובה – דיבור – הן של הכלב והן של החתול. בפרק זה נרחיב את הקשרים בין הדמויות, ואת האופן בו הן מגיבות זו לזו ולמאורעות שמתרחשים. לשם כך נלמד הוראות נוספות של Scratch, שיאפשרו לנו לכתוב תסריטים מורכבים יותר וליצור אנימציות עשירות ומעניינות יותר.

### דוגמה: בעיטת פתיחה

#### משימה 1

בנו אנימציה שתתאר בעיטת פתיחה במשחק כדורגל. השחקן הבועט יהיה החתול המוכר לנו, הפעם בתפקיד הכדורגלן האגדי פלה. פלה ימתין להוראת הבעיטה של השופט, וכשזו תגיע, יבעט בכדור.



שם הקובץ: cat-kicks

### הגדרת הדמויות

מי הן הדמויות המשתתפות?



כמובן, דמות השופט ודמות הכדורגלן. אך פרט להן אנו זקוקים לדמות נוספת. במשחק משתתפת ישות נוספת, הלוא היא הכדור, שממתין בעמדת הפתיחה, ונבעט על ידי פלה. כדי שהאנימציה שניצור תכלול גם את הכדור ואת תנועתו בעקבות הבעיטה, יש להוסיף גם עבורו דמות.

אם כן, בשלב ראשון, לאחר שהבהרנו מי הן הדמויות שצריכות להשתתף באנימציה, נוכל להגדיר דמות Scratch עבור כל אחת מהן.

פתחו פרויקט חדש של Scratch. כתמיד, דמות החתול כבר כלולה בו. שנו את שמה של דמות החתול לשם פלה (או, אם תעדיפו, לשמו של כדורגלן אחר). לאחר מכן הוסיפו דמות חדשה עבור השופט, על פי ההנחיות להוספת דמות חדשה שלמדתם בפרק 2. בין הדמויות של Scratch, בתיקיית האנשים (People), תוכלו למצוא דמות מתאימה (למשל, זו הנקראת referee2). שנו את שמה של הדמות החדשה (למשל, לשם "שופט"). לבסוף, הוסיפו דמות חדשה עבור הכדור. גם במקרה זה, תוכלו להשתמש בדמות מוכנה של Scratch, דמות הכדור הנמצאת בתיקיית החפצים (Things) ונקראת soccer1. שנו את שמה של הדמות החדשה לשם מתאים (למשל, "כדור").



## אתחול מיקומן וכיוון של הדמויות

את המשימה העומדת לפנינו נחלק לשני חלקים, או לשתי תת־משימות. בחלק הראשון תתייצב כל אחת מהדמויות במקומה ההתחלתי. חלקה השני של המשימה הוא ההתרחשות המרכזית של האנימציה: בעיטת הכדור. כדי לפתור את החלק הראשון, צריך לכתוב עבור כל אחת משלוש הדמויות תסריט, אשר עם לחיצה על הדגל הירוק מביא את הדמות אל נקודת ההתחלה שלה ואל הכיוון ההתחלתי שלה.

**תרגיל 1:** תכננו וכתבו עבור כל אחת מהדמויות תסריט אשר מאתחל את מיקומה וכיוונה עם הלחיצה על הדגל הירוק. הכדורגלן יתייצב בפינה הימנית העליונה כשהוא פונה שמאלה, הכדור ימוקם מעט לשמאלו (כך שאינם נוגעים זה בזה) ויופנה שמאלה, והשופט יתייצב בפינה השמאלית התחתונה כשהוא פונה ימינה. בדקו את התסריטים שכתבתם על ידי לחיצה על הדגל הירוק, תעדו אותם בעזרת הערה מתאימה ושמרו את הפרויקט שהתקבל כפרויקט חדש.

## תחילת ההתרחשות: שידור מסר

על פי התיאור שניתן קודם, השופט הוא שיוזם את ההתרחשות. הוא נותן הוראת בעיטה, וכל הדמויות מגיבות להוראה זו. לכן נרחיב כעת את התסריט הראשוני שיצרנו עבור השופט בהתאם לתיאור זה. כיצד ייתן השופט הוראת בעיטה? כדי שהצופים באנימציה יוכלו לראות שניתנה הוראת בעיטה, כדאי ליצור לה ביטוי חיצוני: למשל, להורות לשופט לדבר ולומר "בעט".

אבל להוראת הדיבור יש רק השפעה חיצונית. כלומר, הצופים באנימציה יוכלו לראות שניתנה הוראת בעיטה, אבל לשאר הדמויות המשתתפות באנימציה אין דרך לזהות שדמות "דיברה" ולהגיב בהתאם. לשם כך נוסיף לתסריט השופט הוראה נוספת, שגורמת לו **לשדר מסר** לכל הדמויות המשתתפות. כפי שנראה מיד, קבלת מסר ששודר היא מאורע שדמות יכולה לזהות, ולכן היא יכולה גם להגיב אליו.

הוראת שידור מסר היא הוראת בקרה, כלומר, ניתן למצוא אותה בקבוצת ההוראות הכתומות. זו ההוראה **שידר מסר**. הלבנה המתאימה לה ניתנת לשילוב כהוראה רגילה בתסריט, וניתן לקבוע את המסר עצמו על ידי לחיצה על החץ המופיע ליד החלונית שבתוך הלבנה, בחירה באפשרות "חדש..." והקלדת טקסט בחלון שנפתח בעקבות כך.

### הוראה חדשה ב־Scratch: שידור מסר



ההוראה **שידר מסר** גורמת לדמות לשדר לדמויות האחרות את המסר המופיע בחלונית.

השלימו את התסריט של השופט על ידי שילוב הוראת דיבור מתאימה (סגולה) והוראת שידור מסר (כתומה). הקפידו לעדכן את ההערה בתסריט השופט.

## המשך ההתרחשות: הכדורגלן קולט את ההנחיה לבעוט

כתגובה להוראת הבעיטה של השופט המתבטאת במסר ששודר על ידו לכל הדמויות, צריכה דמות הכדורגלן לבצע בעיטה. לפנינו, אם כך, שתי בעיות: ראשית, כיצד תזהה דמות הכדורגלן שהתקבל מסר? ושנית, מה משמעות הרצף התנועתי "ביצוע בעיטה"? כיצד ניתן לפרקו להוראות של Scratch?

נתחיל מהבעיה הראשונה. בעצם, אנו מעוניינים לתאר סדרת הוראות (שהיא התגובה לבעיטה) ולהבהיר לדמות הכדורגלן שסדרת הוראות זו צריכה להתבצע כאשר מתרחש מאורע מסוים – קבלת מסר. למעשה, כל התסריטים שכתבו עד כה ב־Scratch היו גם הם תסריטים בעלי מאפיין דומה: בכולם תוארה סדרת הוראות שהייתה צריכה להתבצע עם התרחשות מאורע מסוים – לחיצה על הדגל הירוק.


ואכן, ההוראה שבה נשתמש כעת דומה במהותה להוראת הבקרה . גם ההוראה בה נשתמש כעת היא הוראת בקרה, הוראה כתומה. זוהי ההוראה .

מצאו את הלבנה המתאימה להוראה בקבוצת הבקרה. שימו לב כי גם צורתה של הלבנה דומה לצורת הלבנה שמתנה ביצוע בלחיצה על הדגל הירוק. צורתה המעוגלת אינה מאפשרת שילוב הוראות מעליה. לכן בדיוק כמו ההוראה שמתנה ביצוע בלחיצה על הדגל הירוק, גם ההוראה שמתנה ביצוע בקבלת מסר, חייבת להיות הוראה ראשונה בתסריט.

אם כך, לדמות הכדורגלן יהיו שני תסריטים. האחד הוא התסריט שכבר כתבנו, שמתבצע עם לחיצה על הדגל הירוק וממקם את דמות הכדורגלן במקומה ההתחלתי ובכיוון ההתחלתי. השני הוא תסריט הבעיטה שיתבצע עם קבלת מסר הבעיטה.

גררו את הלבנה המתאימה להוראת הבקרה שמתנה ביצוע בקבלת מסר אל חלון התסריטים של הכדורגלן. שלבו בחלונות של ההוראה את המסר המתאים, הלוא הוא אותו מסר שמשדר השופט. לאחר שהוספנו לאחד התסריטים בפרויקט הוראת שידור מסר, המסר החדש כבר שמור בפרויקט, ובכל הוראה שמתייחסת למסרים, יופיע המסר החדש בתפריט בחירת המסרים.

#### הוראה חדשה ב־Scratch: קבלת מסר

ההוראה  יכולה להופיע רק בתחילת תסריט. התסריט יתחיל להתבצע מיד עם קבלת המסר ששודר על ידי דמות אחרת.

#### מושג חדש: תקשורת בעזרת מסרים

דמויות יכולות לבצע ביניהן תקשורת בעזרת מסרים. דמות יכולה לשלוח מסר לכל הדמויות, ודמויות אחרות יכולות להתנות ביצוע בקבלת המסר.

בצורה כזו מנגנון המסרים יכול לשמש לצורך **תיאום** בין דמויות, כדי לוודא שהוראות מסוימות אצל דמות אחת יתבצעו רק לאחר שדמות אחרת ביצעה הוראות מסוימות.

## המשך ההתרחשות: הכדורגלן "בועט"

נפנה כעת לבעיה השנייה. נשים לב כי למעשה, פעולת הבעיטה מתחלקת לשתי פעולות עוקבות: הכדורגלן בועט, והכדור נבעט, כלומר, משנה את מיקומו. שינוי מיקום הכדור הוא באחריותה של דמות הכדור, ולא באחריות דמות הכדורגלן. זאת בגלל שדמות יכולה להניע את עצמה על ידי ביצוע הוראת תנועה, אך אינה יכולה לגרום לדמות אחרת לנוע. דמות הכדורגלן צריכה רק לבצע את הבעיטה עצמה: התקרבות אל הכדור עד נגיעה בו. אם כך, זוהי בעצם הוראת תזוזה, שתגרום לחתול לזוז מספר צעדים לכיוון הכדור (הכיוון שבו החתול כבר נמצא) וכך לגעת בכדור.

צרו הוראת תזוזה מתאימה על ידי שימוש בלבנה כחולה המתאימה להוראת התזוזה. את הוראת התזוזה שלבו בתסריט החדש. עליכם לשלב בהוראה מספר צעדים רצוי לתזוזה. הערך המשולב צריך להיקבע על פי מיקומו ההתחלתי של החתול ומיקומו ההתחלתי של הכדור. זכרו כי מיקומה של דמות מתייחס למיקום נקודת המרכז שלה, בעוד שכדי שתתרחש נגיעה, מספיק שהקצה השמאלי של הכדורגלן והחלק הימני של הכדור יהיו באותה נקודה. תוכלו לנסות ערכים שונים ולשנותם בהתאם לצורך עד שתגיעו לערך המתאים.

כדי להדגיש את הקשר בין הדמויות ולהעשיר את התנהגותן, כדאי להרחיב את תגובת הכדורגלן: פרט לבעיטה עצמה, ניתן להוסיף לו גם דיבור, המראה כי "שמע" את הוראת הבעיטה של השופט והוא מתכוון לבעוט.

הוסיפו לתסריט השני של הכדורגלן הוראת דיבור מתאימה, כהוראה שנייה לפני הוראת התזוזה. זכרו להוסיף הערה לתסריט החדש. שמרו את הפרויקט שיצרתם עד כה.

## סיום ההתרחשות: הכדור נבעט

כעת תורו של השחקן השלישי באנימציה למלא את תפקידו. הכדור צריך להיבעט, כלומר, לשנות את מיקומו. כמובן, תנועה זו צריכה לבוא כתגובה למאורעות: ראשית, בדומה לכדורגלן, הכדור צריך להמתין לקבלת מסר בעיטה מהשופט. אבל גם לאחר שמסר כזה התקבל, הכדור אינו יכול להתחיל בתנועתו. עליו לחכות שהכדורגלן יבעט, כלומר, שיגיע עד אליו וייגע בו.

נוכל לתאר זאת מילולית באופן הבא:

0. אם קבלה מסר בעיטה:

1. האמן עד שהכדורגלן נעץ בקר

2. נעץ שמאלה, אל שולי הקמה

לכן גם עבור הכדור יש ליצור תסריט שני, אשר ההוראה הראשונה בו תהיה הוראת בקרה כתומה, שמתנה ביצוע בקבלת מסר.

את צעד 2 ניתן לתרגם להוראת תנועה כחולה. מאחר שאנו מעוניינים כי האנימציה תראה את תנועתו של הכדור, אין כאן מקום להוראת תזוזה או להוראת קפיצה, אלא להוראת גלישה.

כיצד נתרגם את צעד 1? דרושה לנו כאן הוראה המעכבת את המשך הביצוע עד התרחשות מאורע מסוים, במקרה זה, מאורע נגיעה. לכן המקום המתאים לחפש הוראה כזאת הוא בקבוצת הוראות

הבקרה, הקבוצה הכתומה. שם נוכל למצוא את ההוראה **חנה עד ש...**. זוהי הוראת המתנה מותנית: היא מורה על המתנה, והמשכה של המתנה זאת תלוי בהתרחשות מאורע או בקיום תנאי. כל עוד המאורע אינו מתרחש, או כל עוד התנאי אינו מתקיים, ההמתנה נמשכת. עם התרחשות המאורע או עם קיום התנאי, ההמתנה מסתיימת, והתסריט ממשיך אל ההוראה הבאה.

לכן בהוראת ההמתנה המתאימה לצעד 1, יש לשלב תנאי. ואכן, הלבנה המתאימה להוראה זו, מכילה חלונית המתאימה בצורתה ללבנת תנאי, חלונית שפינותיה קטומות. על פי התיאור המילולי, ברור כי התנאי המתאים לשילוב הוא תנאי הנגיעה, המוכר לנו מדוגמאות קודמות, תנאי שנוכל למצוא בקבוצת החיישנים הכחולה-בהירה.

### מושג חדש: שליטה על ביצוע ותיאום באמצעות המתנה מותנית

המתנה מותנית מאפשרת לעכב ביצוע הוראות עד אשר יתקיים תנאי מסוים. גם זו דרך לתאם ביצוע בין דמויות שונות.

### הוראה חדשה ב-Scratch: המתנה מותנית

ההוראה **חנה עד ש...** גורמת לתסריט המתבצע לעצור ולהמתין עד שיתקיים התנאי המופיע בחלונית. כשהתנאי מתקיים, ביצוע התסריט ממשיך מההוראה שאחרי הוראת ההמתנה המותנית.

צרו עבור הכדור את התסריט השני, המורכב משלוש הוראות של Scratch שמתאימות לצעדים 0 עד 2: ההוראה שמתנה ביצוע בקבלת מסר, הוראת המתנה מותנית והוראת גלישה. הפכו את שלוש ההוראות הכלליות להוראות ממוקדות על ידי שילוב ערכים מתאימים בתוכן. תוכלו לנסות ולבדוק את הערכים ששילבתם ולשנותם בהתאם לצורך, עד שתגיעו לתוצאה המבוקשת. הוסיפו הערה מתאימה לתסריט. שמרו את האנימציה שיצרתם ונסו להפעילה. באנימציה החדשה לחיצה על הדגל הירוק יוצרת את רצף ההתרחשויות הבא: הדמויות תופסות את מקומן על המגרש, השופט מורה בדיבור על בעיטה, הכדורגלן אומר דבר מה המעיד כי שמע והוא מתכוון לבעוט, הכדורגלן בועט, והכדור נבעט שמאלה.

**תרגיל 2:** הרחיבו את האנימציה שיצרנו. החתול לא יהיה הכדורגלן היחיד על המגרש. גם הכלב יצטרף למשחק עם כדור משלו, וגם הוא יגיב להוראת הבעיטה של השופט. הכלב יבעט בחלקו התחתון של הבמה. שימו לב: עליכם להגדיר שתי דמויות נוספות (לכלב ולכדור שבו הוא בועט), כל אחת עם שם מתאים לה. התסריטים עבור הדמויות החדשות יהיו דומים מאוד לתסריטים של הדמויות הקיימות. ההבדלים היחידים בין תסריטי הכלב ותסריטי החתול יהיו בערכים המשולבים בהוראות התנועה. התסריטים של שני הכדורים יהיו שונים גם הם בערכים המשולבים בהוראות התנועה. בנוסף יגיב הכדור החדש לבעיטת הכלב ולא לבעיטת החתול.

שם הקובץ: cat-and-dog-kick

### תרגיל 3:

באנימציה שבתרגיל 2 הכדור שנבעט על ידי הכלב מגיע עד לשופט. הרחיבו את האנימציה כך שהכדור הזה יגרום לנפילתו של השופט. שימו לב: יש לעדכן רק את התנהגותה של דמות השופט. את הנפילה ניתן להציג על ידי הוראת שינוי כיוון מתאימה, ועליה להתרחש כתגובה להגעת הכדור.

שם הקובץ: `cat-and-dog-kick-referee-falls`

### תרגיל 4:

מטרת תרגיל זה היא ליצור אנימציה שבה מתחרים שני כלבים על עצם אחת. התחרות היא תחרות מסודרת: שני הכלבים עומדים, כל אחד במקומו, וממתינים שהחתול יזניק אותם. אז הם רצים לעבר העצם, והראשון שמגיע אל העצם לוקח אותה איתו.

א. הכינו אנימציה שבה משתתפים שני הכלבים, החתול והעצם. בתחילת ביצוע האנימציה העצם נמצאת בחלקה התחתון של הבמה, באמצע, בנקודה  $(0, -160)$ ; החתול נמצא בחלקה העליון של הבמה, גם כן באמצע, בנקודה  $(0, 120)$ ; שני הכלבים נמצאים בחלקה השמאלי ובחלקה הימני של הבמה, בנקודות  $(-180, 50)$  ו- $(180, 50)$ , ופונים לכיוון העצם. החתול מזניק את הכלבים לדרכם, ואז הם יוצאים לכיוון העצם. הראשון שמגיע אל העצם פונה בכיוון מיקומו ההתחלתי וממשיך בתנועתו, והעצם איתו. הכלב השני ממשיך בתנועתו ללא שינוי כיוון.

השתמשו במסרים כדי לשלוט על ביצוע האנימציה: מסר אחד שבעזרתו החתול מזניק את הכלבים, מסר שני שבעזרתו מודיע כלב אחד לעצם כי עליה לנוע איתו, ומסר שלישי שבעזרתו מודיע הכלב האחר לעצם כי עליה לנוע איתו.

תכננו את האנימציה כך שמהירותם של הכלבים תהיה שונה, כדי להבטיח כי אחד מהם יגיע אל העצם לפני הכלב האחר.

שם הקובץ: `get-bone1`

ב. שנו את הפתרון שהכנתם עבור סעיף א' והשתמשו בשני מסרים במקום בשלושה: מסר אחד שבעזרתו החתול מזניק את הכלבים, ומסר שני שבעזרתו מודיע כל אחד מהכלבים לעצם כי עליה לנוע איתו.

שם הקובץ: `get-bone2`

ג. האם המשימה שמוצגת בתרגיל זה מחייבת שימוש בשני מסרים או די במסר אחד שבעזרתו החתול מזניק את הכלבים? הסבירו את תשובתכם. אם תשובתכם היא חיובית, הראו כיצד ניתן לפתור את המשימה על ידי שימוש במסר אחד בלבד, והסבירו מדוע לא ניתן לבצע דבר דומה לצורך פתרון משימה 1.

שם הקובץ: `get-bone3`

### תרגיל 5:

מטרת תרגיל זה היא ליצור אנימציה המתארת סוג מסוים של מירוץ שליחים. משתתפים ארבעה כלבים, כל אחד מוצב בפינה של הבמה. אחד הכלבים מתחיל להקיף את הבמה עם כיוון השעון, וכאשר הוא בחצי הדרך אל הכלב הבא, יוצא הכלב הבא לדרכו ומתחיל גם הוא להקיף את הבמה עם כיוון השעון. כך הדבר ממשיך עד שכל ארבעת הכלבים מקיפים את הבמה עם כיוון השעון, ואז הם ממשיכים בהקפה זו שוב ושוב.

א. הכינו גרסה ראשונית שבה כל הכלבים מקיפים את הבמה שוב ושוב במקביל, עם כיוון השעון, וכל כלב מתחיל מפניתו עם הלחיצה על הדגל הירוק.

שם הקובץ: `relay-race1`

ב. הרחיבו את הפתרון שיצרתם בסעיף א', כך שהכלבים יתחילו את תנועתם באופן מדורג, כפי שתואר בפתיחת התרגיל. עשו זאת על ידי הוספת **שתיים או שלוש** הוראות לכל דמות.

שם הקובץ: relay-race2

ג. באנימציה שיצרתם בסעיף ב' מהירותו של כל כלב משתנה במהלך תנועתו. כתוצאה מכך המרחק בין הכלבים אינו קבוע לאורך ביצוע האנימציה. שנו את האנימציה כך שמהירותו של כל כלב היא קבועה, והמרחק בין הכלבים נשאר אף הוא קבוע.

שם הקובץ: relay-race3

## סיכום

בפרק זה ראינו כיצד ניתן ליצור אנימציות שההתרחשות המתוארת בהן מבוססת על תקשורת בין הדמויות המשתתפות. תקשורת כזאת יכולה להתבצע על ידי שליחת **מסרים**: דמות יכולה לשלוח מסר לשאר הדמויות, ודמויות אחרות יכולות להתייחס בתסריטים שלהם למסר שצריך להתקבל.

ההוראות של Scratch שמאפשרות שידור מסרים וקבלתן שייכות לקבוצת **הוראות הבקרה**, ההוראות הכתומות.

**שידור מסר** משמעותו שליחת הודעה מבוקשת אל כל הדמויות. ההוראה המתאימה לשידור מסר היא



**ביצוע מותנה בקבלת מסר** מאפשר להתנות ביצוע סדרת הוראות בקבלת מסר מסוים. ההוראה שמתנה ביצוע בקבלת מסר היא תמיד הוראה ראשונה בתסריט. שאר הוראות התסריט יבוצעו רק עם קבלת



המסר. ההוראה המתאימה היא **נשמתיקבל מסר**. זוהי **הוראת המתנה מותנית**, המשהה את המשך הביצוע עד שמתרחש מאורע מסוים או מתקיים תנאי מסוים. ההוראה המתאימה היא **חנה עד ש...**, ובה יש לשלב תנאי.



## פרק 5

### על רחבת הריקודים – בחזרה לביצוע חוזר

בפרק 3 ראינו כי בעזרת הוראות של ביצוע חוזר ניתן ליצור אנימציות ארוכות מאוד – אפילו אינסופיות – על ידי סדרת הוראות סופית ואפילו קצרה מאוד. הכרנו בפרק 3 שתי הוראות של ביצוע חוזר, שתיהן אינסופיות. האחת של ביצוע חוזר אינסופי והשנייה של ביצוע חוזר אינסופי משולב בבדיקה. בפרק זה נכיר הוראות נוספות של ביצוע חוזר, המייצרות גם הן חזרה על סדרת הוראות, אך חזרה שאינה אינסופית אלא מוגבלת באורכה.

### דוגמה 1: ריקוד בסיסי – ביצוע חוזר מוגבל מראש

#### משימה 1

הציגו ריקוד המבוצע על ידי רקדנית. הרקדנית תנוע לרוחב הבמה בכיוון אחד, אחר כך בכיוון ההפוך ותעצור.



פתחו פרויקט חדש וייבאו דמות חדשה, דמות של רקדנית. דמות כזו תוכלו למצוא בתיקיית האנשים (People). בין הדמויות בתיקייה זו יש כמה דמויות של בחורה בשם קאסי (cassy), כמה מהן מציגות אותה רוקדת. לצורך האנימציה שניצור כעת, המתאימה ביותר מבין הדמויות האלו היא הדמות שבה קאסי מניפה את שתי ידיה כפופות – האחת למעלה והאחרת למטה (אם תעדיפו, תוכלו לתת לה שם חדש).

כאמור, בסך הכול נרצה שהרקדנית תנוע לרוחב הבמה בכיוון אחד ואחר כך בכיוון ההפוך.

כבר ברור שהרקדנית תצטרך לשנות את כיוון תנועתה, כלומר, להסתובב 180 מעלות, תוך כדי הריקוד. כאשר דמויות ב-Scratch מסתובבות, הסיבוב הוא מלא וכך עלולה הרקדנית למצוא את עצמה עומדת על ראשה. הסתכלו בחלון המתאר את מצבה של הדמות, מעל חלון התסריטים שלה. לשמאלה של הדמות יש שלושה כפתורים קטנים, זה מתחת לזה. אם הכפתור העליון מסומן בכחול, סיבוב של הדמות יהיה סיבוב מלא. אם הכפתור האמצעי מסומן בכחול, הדמות יכולה להסתובב ימינה ושמאלה בלבד. כאשר הכפתור התחתון מסומן בכחול, הדמות לא יכולה כלל להסתובב, ולכן בעקבות הוראת סיבוב ישתנה רק כיוון התנועה אך לא מצג הדמות. לכן, על ידי לחיצה על הכפתור האמצעי נמנע מהרקדנית לעמוד על ראשה בעקבות שינוי כיוון תנועתה.

כיצד נוכל להורות לרקדנית לבצע את הריקוד?

אפשרות אחת היא להשתמש בשתי הוראות תזוזה, הראשונה כדי להניעה משמאל לימין והשנייה כדי להניעה מימין לשמאל. אבל הוראות כאלו יגרמו רק לקפיצה מיידית מנקודת ההתחלה לנקודת הסיום ולא ייצרו רושם של ריקוד.



## ריקוד בגלישה

אפשרות אחרת היא להשתמש בהוראת גלישה.

### תרגיל 1:

נסו ליישם את ההצעה השנייה. כתבו תסריט שעם לחיצה על הדגל הירוק ממקם את הרקדנית בנקודת ההתחלה (100, -100) ובכיוון ההתחלתי 90 (כלומר, ימינה). התסריט יכלול שתי הוראות גלישה: האחת כדי להניע את הרקדנית 100 צעדים ימינה והשנייה כדי להניעה 100 צעדים חזרה בכיוון ההפוך (זיכרו לשנות את כיוון תנועתה של הרקדנית ב-180 מעלות בין הוראת הגלישה הראשונה לשנייה). נסו להפעיל את האנימציה שיצרתם והתרשמו מהתוצאה.

שם הקובץ: dancer-glides

## ריקוד בתזוזות קטנות

האנימציה שהתקבלה בתרגיל 1 היא נחמדה, אך התנועה הנוצרת אינה נראית ממש כתנועת ריקוד. ננסה לפעול בשיטה שונה: במקום להניע את הרקדנית בבת אחת 100 צעדים נניע אותה בתזוזות קטנות, למשל, עשר תזוזות של עשרה צעדים בכל פעם, ובין תזוזה לתזוזה ניתן לה להמתין זמן קצר מאוד, למשל, חמישית השנייה. בגלל שזמן ההמתנה הוא כה קצר הוא יורגש רק בקושי, אך הרושם שיווצר יהיה של צעדי ריקוד.

### המתנה

האם יש בידינו הכלים לכתוב תסריט המתאים לשיטה שתיארנו כעת? אנו כבר מכירים היטב את הוראת התזוזה, אך עדיין לא השתמשנו בהוראת המתנה כמו זו הדרושה לנו כאן. בפרק 4 הכרנו הוראת המתנה מותנית, כזו המורה על המתנה עד התרחשות מאורע מסוים, אך כעת דרושה לנו הוראה פשוטה ממנה, המורה על המתנה לפרק זמן מסוים.

המקום שכדאי לחפש בו הוראת המתנה הוא קבוצת הבקרה הכתומה (שם, כזכור, נמצאת גם הוראת ההמתנה המותנית), והלבנה המתאימה לה היא הרביעית בקבוצה: **חנה שניות**. זו הוראה כללית, וכדי להפכה להוראה ממוקדת, יש לשלב בחלונית את מספר השניות הרצוי להמתנה.

### הוראה חדשה ב-Scratch: המתנה

ההוראה **חנה שניות** גורמת לתסריט המתבצע לעצור ולהמתין למשך מספר השניות המופיע בחלונית. לאחר מכן ביצוע התסריט ממשיך מההוראה שאחרי הוראת ההמתנה.

נראה כי כעת כבר יש לנו כל שדרוש כדי לכתוב תסריט מתאים לשיטה שתיארנו. אפשר לכתוב רצף של עשר הוראות תזוזה של עשרה צעדים, כשאחרי כל הוראת תזוזה משולבת הוראת המתנה של חמישית השנייה. בסך הכול, רצף של עשרים הוראות. לאחר שינוי כיוונה של הרקדנית, צריך לחזור ולשלב את הרצף כולו כדי להניעה בחזרה.

אלא שמעבר לכך שבניית תסריט כזה היא מייגעת למדיי, זו בוודאי לא שיטה יעילה ליצירת שרשרת תזוזות ארוכה מאוד, למשל של 100 תזוזות או יותר.

## ביצוע חוזר מוגבל מראש

מה שאנו זקוקים לו כעת הוא דרך לציין כי הוראה, או סדרת הוראות (במקרה שלנו סדרה של שתי הוראות, הוראת תזוזה והוראת המתנה), תבוצע מספר פעמים (במקרה שלנו עשר פעמים). התיאור המילולי המתאים הן לתנועה הלך והן לתנועה חזור הוא זה:


1.  $10 \times 360^{\circ}$  ספירה;

1.1  $10 \times 360^{\circ}$  ספירה

1.2  $0.2 \times 360^{\circ}$  ספירה

דרושה לנו הוראה של ביצוע החוזר על עצמו מספר נתון של פעמים. את הוראות הביצוע החוזר שהכרנו בפרק 3 מצאנו בקבוצה הכתומה, קבוצת הבקרה, ובה נמצא גם את הוראת הביצוע החוזר החדשה, הוראה של ביצוע חוזר מוגבל מראש.



הלבנה המתאימה לה היא . זו הוראה כללית, וכדי להפכה להוראה ממוקדת, יש לשלב בחלונית מספר, הלוא הוא מספר הפעמים הרצוי לביצוע חוזר.

לבנה זו מזכירה בצורתה את הלבנים של הוראות הביצוע החוזר מפרק 3. גם היא נועדה לשילוב לבנים נוספות בתוכה, הלבנים המתאימות להוראות שיבוצעו שוב ושוב במסגרת הביצוע החוזר. אבל יש הבדל צורני חשוב בינה לבין הלבנים שפגשנו בפרק 3, הלבנים שמתאימות להוראת הביצוע החוזר האינסופי



ולהוראת הביצוע החוזר האינסופי המשולב בבדיקה



מהו ההבדל הצורני?



תחתיתן של הלבנים המוכרות לנו, המתאימות לביצוע חוזר אינסופי, היא ישרה ואינה מאפשרת שילוב לבנים נוספות אחריהן בתסריט. כלומר, כפי שכבר הבחנו בפרק 3, הוראת ביצוע חוזר אינסופי היא תמיד הוראה אחרונה בתסריט. זאת משום שאין טעם לנסות ולשלב הוראה נוספת אחרי הוראת ביצוע חוזר אינסופי. הרי הביצוע החוזר האינסופי אינו צפוי להסתיים.

אבל כאשר מדובר בביצוע חוזר שהוא מוגבל מראש במספר החזרות שהוא יוצר, אין מניעה לשלבו בתוך תסריט, ולא בהכרח בסוף התסריט. לכן תחתית הלבנה המתאימה להוראת הביצוע החוזר החדשה אינה ישרה ומאפשרת שילוב לבנים נוספות אחריה.

צרו את קטע התסריט המתאים לתיאור המילולי שנתנו ושלבו אותו פעמיים בתסריט שיצרתם בתרגיל 1, במקום הוראת הגלישה הראשונה ובמקום הוראת הגלישה השנייה. כעת מכיל התסריט שני קטעי ביצוע חוזר, או כפי שנוהגים לעיתים לקרוא להם – שתי **לולאות**. כאשר ביצוע התסריט מגיע לקטע של ביצוע חוזר, ללולאה, הוא "מסתובב" בו שוב ושוב, ובכל סיבוב מבצע את שתי ההוראות הכלולות בו. מספר הסיבובים הוא המספר שנקבע מראש בהוראה לביצוע החוזר.

### מושג חדש: ביצוע חוזר מוגבל מראש

בביצוע חוזר מוגבל מראש, סדרה באורך סופי של הוראות כלשהן מתבצעת מספר ידוע מראש של פעמים. הביצוע המתקבל הוא סופי.

### הוראה חדשה ב־Scratch: ביצוע חוזר מוגבל מראש



ההוראה גורמת לסדרת ההוראות של Scratch המשולבת בתוך ההוראה, להתבצע כמה פעמים. מספר הפעמים שהסדרה תבוצע הוא כמספר המופיע בחלונית של ההוראה.

כתבו הערות תיעוד, שמרו את הפרויקט בשם חדש ונסו להפעילו. **שם הקובץ: dancer-moves-loop**

בהמשך הפרק נרחיב את הדוגמה ההתחלתית. בכל ההרחבות נתבסס על אותה גישה ליצירת אנימציה של ריקוד שבה השתמשנו בדוגמה 1 (כלומר, פירוק תנועת הריקוד לתזוזות קטנות והמתנה ביניהן), ונשתמש באותו ריקוד בסיסי. בהרחבת הרשות שבסוף הפרק נראה כיצד אפשר לשנות גם את הריקוד הבסיסי, כך שיתקבל ריקוד דינמי יותר.

**תרגיל 2 – כבר לא לבד על הרחבה:** הוסיפו לאנימציה שיצרתם כעת רקדן הרוקד במקביל לרקדנית, ותנועתו דומה לזו של הרקדנית אך הפוכה לה. הוא יתחיל בנקודה (100, 100) כשהוא פונה שמאלה (90-) ומשם ינוע 100 צעדים, ואז ינוע 100 צעדים בכיוון ההפוך. דמות מתאימה לרקדן ניתן למצוא שוב בתיקיית האנשים (People), למשל, דמותו של ג'יי (jay). הוסיפו הערה מתאימה לתסריטו של הרקדן, שמרו את הפרויקט בשם חדש ונסו להפעילו.

**שם הקובץ: two-dancers-move**

## דוגמה 2: עד שניפגש – ביצוע חוזר מותנה

בתרגיל 2 הוספתם דמות שנייה לאנימציה, רקדן שרוקד במקביל לרקדנית. אך למעשה, אין בין שתי הדמויות כל קשר. הן חולפות זו על פני זו, והמפגש אינו משפיע באופן כלשהו על תנועתן. כעת ננסה ליצור קשר בין הרצפים התנועתיים של שתי הדמויות.

### משימה 2

שני הרקדנים ינועו זה לקראת זה. בעקבות המפגש ביניהם יהפכו שניהם את כיוון תנועתם ויתחילו לנוע בכיוון ההפוך.



שם הקובץ: `two-dancers-move-until-touch`

אם כך, כל אחת מהדמויות תנוע כמקודם בתזוזות של עשרה צעדים ותבצע המתנה קצרה לאחר כל תזוזה, אך הפעם לא ייקבע היפוך התנועה על פי מספר התזוזות וההמתנות שבוצעו, אלא יימשך עד הפגישה. כאשר תזוזה פגישה, תהפוך כל דמות את כיוון תנועתה ותנוע בכיוון ההפוך.

### התניית ביצוע חוזר בהתרחשות מאורע

גם הפעם דרושה לנו הוראה לביצוע חוזר, אך הפעם לא נרצה להצהיר מראש על מספר החזרות או מספר הסיבובים, כפי שעשינו בדוגמה 1. במקום זאת נרצה לתלות את סיום הביצוע החוזר בהתרחשות מאורע, מאורע הפגישה בין הדמויות. למשל, הנה תיאור מילולי המתאים לתנועת הרקדנית עד המפגש:

1. הִצֵּד עַד הַפְּגִישָׁה עִם הַרְקָדָן:

1.1 10 שֵׁס 10 דְּיִס

1.2 הַמָּגֵן 0.2 שְׁנִיט

צעד 1 מתאר **ביצוע חוזר מותנה**. סיום הביצוע החוזר הזה אינו תלוי במספר הסיבובים שבוצעו, אלא בהתרחשות מאורע. קיום המאורע נבדק בתחילת כל סיבוב, כאשר הביצוע מגיע לצעד 1. אם המאורע אינו מתקיים, מבצעים את הצעדים 1.1 ו-1.2 וחוזרים שוב לצעד 1, לתחילתו של סיבוב נוסף. אם בתחילת סיבוב מתגלה כי המאורע מתקיים, הביצוע החוזר מסתיים בלי שבוצעו בסיבוב זה ההוראות הכלולות בתוך הביצוע החוזר.



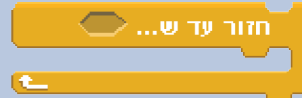
ההוראה המתאימה לצעד כזה היא ההוראה  , וגם את הלבנה המתאימה לה נמצא בקבוצה הכתומה, קבוצת הבקרה.

גם צורתה של לבנה זו מאפשרת את שילובה בתוך תסריט, ולא דווקא בסופו. שימו לב גם לצורתה של החלונית המשולבת בלבנת הביצוע החוזר המותנה. זוהי חלונית תנאי שפינותיה קטומות, כמו אלו ששולבו בהוראת הביצוע החוזר האינסופי המשולב בבדיקה (פרק 3) ובהוראת ההמתנה המותנית (פרק 4). יש לשלב בה לבנת תנאי, במקרה זה תנאי הנגיעה שבו השתמשנו בשני הפרקים הקודמים. כזכור, לבנת התנאי המתאימה נמצאת בקבוצת החיישנים הכחולה-בהירה.

### מושג חדש: ביצוע חוזר מותנה

בביצוע חוזר מותנה, סדרה באורך סופי של הוראות כלשהן מתבצעת שוב ושוב עד אשר מתקיים תנאי מסוים.

### הוראה חדשה ב-Scratch: ביצוע חוזר מותנה



ההוראה גורמת לסדרת ההוראות של Scratch המשולבת בתוך ההוראה, להתבצע שוב ושוב, עד אשר מתקיים התנאי המופיע בחלונת של ההוראה.

צרו עבור שתי הדמויות קטעי תסריט המתאימים לתנועה עד המפגש, ושלבו אותם בתסריטיהם של הרקדן והרקדנית במקום הלולאה הראשונה. עדכנו את תיעוד הפרויקט ושמרו אותו בשם חדש. נסו להפעילו.

### אז מה באמת קורה כאן?

את ביצועם של התסריטים שהתקבלו ניתן לתאר כך: כאשר ביצוע התסריט מגיע להוראת הביצוע החוזר המותנה, מתרחשת סדרת צעדים המשולבת בבדיקות חוזרות ונשנות. ראשית, נבדק קיום מאורע הפגישה. מאחר שעדיין לא התקיימה פגישה, מבוצעות הוראות התזוזה וההמתנה המשולבות בהוראת הביצוע החוזר המותנה. לאחר מכן נבדק שוב קיום מאורע הפגישה, ומאחר שעדיין אין הוא מתקיים, מבוצעות שוב הוראות התזוזה וההמתנה. כך הלאה, עד שבאחת הבדיקות מזוהה המפגש. התרחשות מאורע הפגישה מביאה לדילוג על הוראות התזוזה וההמתנה המשולבות בהוראת הביצוע החוזר המותנה, וביצוע התסריט ממשיך אחרי הלולאה, בהוראת היפוך הכיוון.

**תרגיל 3:** האם ייתכן כי ההוראות המשולבות בתוך הוראת הביצוע החוזר המותנה לא יבוצעו בכלל, אף לא פעם אחת? נסו לשנות את האנימציה כך שהמיקום ההתחלתי של שתי הדמויות יהיה זהה, למשל, הנקודה (0, 100). הפעילו את האנימציה החדשה והסבירו את המתרחש.

**תרגיל 4:** האם יש לשנות גם את הלולאה השנייה בתסריטיהם של הרקדנים? מה יקרה אם נחליף גם את ההוראה השנייה של הביצוע החוזר המוגבל מראש בהוראת ביצוע חוזר המותנה בפגישה? נסו והסבירו את התוצאה.

אם כך, בתרגיל 4 למדנו לקח חשוב בנוגע לביצועים חוזרים מותנים. עלינו לוודא שהתנאי הכלול בהוראת הביצוע החוזר אכן יתקיים בסופו של דבר, כי אחרת ההוראה יוצרת ביצוע חוזר אינסופי, ואם כלולות בתסריט הוראות נוספות – הן לא יבוצעו לעולם. חשש כזה לא הטריד אותנו בפרק 3, שם היו הביצועים החוזרים אמורים להיות אינסופיים, וגם לא בהקשר של הוראת הביצוע החוזר שלמדנו בתחילת פרק 5, שבה הוגבל מראש מספר הפעמים לביצוע חוזר.

## דוגמה 3: נרקוד עוד ועוד – ביצוע חוזר בתוך ביצוע חוזר


### משימה 3



הריקוד של הרקדנים ימשיך עוד ועוד, כרצף תנועתי אינסופי (או עד שביצוע האנימציה ייעצר בעקבות לחיצה של הצופה על הכפתור האדום).

במילים אחרות, נרצה לחזור שוב ושוב על אותו קטע ריקוד שיצרנו בדוגמה 2. כיצד נוכל לעשות זאת? למעשה, כבר למדנו בפרק 3 דרך ליצור ביצוע חוזר אינסופי. ההוראה שנראית מתאימה לצרכינו היא

לעולים

ההוראה . כזכור, בתוך הוראת הביצוע החוזר האינסופי אפשר לשלב כל סדרה של הוראות המיועדת לביצוע חוזר אינסופי. אם כן, מדוע שלא נשלב בה את סדרת הוראות הריקוד הכלולה בתסריטים שיצרנו בדוגמה 2?

אבל סדרת הוראות הריקוד כוללת בעצמה הוראה לביצוע חוזר, אפילו שתיים. האם אפשר שהוראה לביצוע חוזר (במקרה זה, ביצוע חוזר אינסופי) תשלב בתוכה הוראות לביצוע חוזר (במקרה זה, ביצוע חוזר מותנה וביצוע חוזר מוגבל מראש)?

כן, אין שום דבר שמונע זאת, אך צריך לעשות זאת בזהירות ולוודא אגב כך שהחזרות שמתבצעות בעקבות השילוב הן בדיוק אלו שאנו מעוניינים בהן.

אם כך, נבחן שוב את התסריטים שבדוגמה 2, למשל את תסריטה של הרקדנית. איזה קטע מהתסריט

לעולים

אנו רוצים לשלב בתוך הוראת הביצוע החוזר האינסופי, בתוך ההוראה  ?

ההוראה הראשונה בתסריט היא זו שמתנה את תחילת ביצועו בלחיצה על הדגל הירוק. האם יש לשלב אותה בתוך הוראת הביצוע החוזר האינסופי?



ההוראה שמתנה את תחילת ביצוע התסריט בלחיצה על הדגל הירוק, צריכה להישאר ההוראה הראשונה בתסריט, עוד לפני הוראת הביצוע החוזר האינסופי. כזכור, צורתה של הלבנה מגנה עלינו מפני ניסיון שילוב שכזה, ואינה מאפשרת לנו לשלבה בתסריט כהוראה שאינה ראשונה.

לאחר מכן כלולות בתסריט שתי הוראות האתחול: קביעת המיקום ההתחלתי והכיוון ההתחלתי. האם יש לשלב אותן בתוך הוראת הביצוע החוזר האינסופי?



נראה שלא. כפי ששמו מעיד, אתחול אמור להתבצע בתחילת התהליך. אין לנו סיבה לחזור ולאתחל שוב ושוב את מצבן של הדמויות. הרי האתחול הוא הכנה לריקוד ולא הריקוד עצמו.

הריקוד עצמו, זה שאנו מעוניינים לחזור עליו שוב ושוב, מתואר על ידי קטע התסריט המתחיל בלולאה הראשונה, בהוראת הביצוע החוזר המותנה, ונמשך עד סוף התסריט.

אם כך, נראה שכל שעלינו לעשות הוא לשלב את קטע התסריט הזה בהוראה לביצוע חוזר אינסופי,

## לעולמים

פשוט על ידי עטיפתו בלבנה . אם תבחרו בלבנה ותגדרו אותה אל התסריט, כך שראשה יהיה בין הוראת האתחול השנייה לבין הוראת הביצוע החוזר המותנה, פיה יקיף את כל קטע התסריט שמנקודה זו והלאה, כך:



**תרגיל 5:** בצעו את השינוי המוצע בשני התסריטים, בתסריט של הרקדן ובתסריט של הרקדנית, ובדקו את השפעתו על ידי לחיצה על הדגל הירוק. נסו להסביר את התוצאה.

שם הקובץ: [dancers-disappear](#)

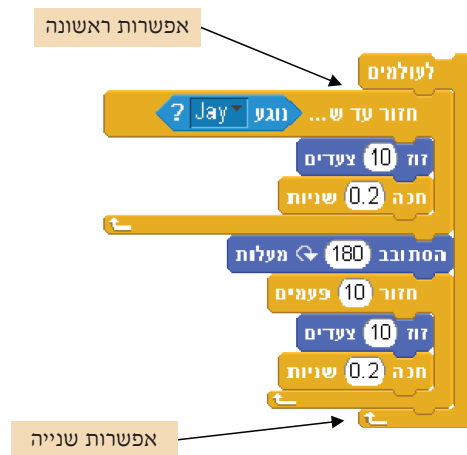
## תקלה וניתוח התקלה

השינוי לא יצר את התוצאה המקווה. כדי להבין את הסיבה, אפשר לעקוב אחר ביצוע התסריט, הוראה אחרי הוראה:

- כשהריקוד מתחיל, כל אחת מהדמויות פונה לכיוון ההתחלתי שלה, הרקדן פונה שמאלה והרקדנית פונה ימינה.
- אז הן מבצעות את קטע הריקוד הראשון, שבו כל אחת נעה על פי הכיוון שלה, אגב שילוב תזוזות והמתנות, עד המפגש.
- בעקבות המפגש כל אחת מהדמויות מסתובבת לכיוון ההפוך, והן מבצעות את קטע הריקוד השני, המורכב מעשר תזוזות ועשר המתנות.
- במצב הזה הביצוע החוזר מפנה אותן שוב אל הלולאה הראשונה, של הוראת הביצוע החוזר המותנה. כל אחת מהדמויות מתחילה לנוע, ואחרי כל מקטע של תזוזה והמתנה נבדק מצבן זו ביחס לזו – האם כבר נפגשו? אבל כיוון תנועתן נשאר כפי שהיה בקטע הריקוד השני. התסריט אינו כולל שום הוראה לשינוי כיוון לאחר קטע הריקוד השני, ולכן שתי הדמויות נעות כעת כשהן מתרחקות זו מזו, והתופעה המתרחשת דומה לזו שהודגמה בתרגיל 4: מאורע הפגישה לא יתקיים אף פעם, והתנועה שנגרמת היא אמנם אינסופית, אך לא זו שתכננו. הדמויות ממשיכות להתרחק זו מזו עד שהן מתחילות לגלוש מקצה הבמה, ושם הן נשארות, בלי שיחזרו ויתקרבו שוב זו אל זו.

## ניסיון לתיקון התקלה: שינוי כיוון

כיצד נפתור זאת? עלינו להכניס הוראת תנועה נוספת, שתגרום להיפוך הכיוון לפני שהרקדנים חוזרים לביצוע קטע הריקוד הראשון. ברור לנו מהי ההוראה המתאימה. זוהי ההוראה שבה השתמשנו בין שני קטעי הריקוד: **הסתובב 180 מעלות**. היכן נשלב אותה?



יש שתי אפשרויות לשילוב. כפי שראינו, הבעיה נוצרת לפני שהרקדנים חוזרים לביצוע קטע הריקוד הראשון. לכן אפשר לחשוב על שילוב הוראת היפוך הכיוון לפני קטע הריקוד הראשון, כהוראה ראשונה בתוך הוראת הביצוע החוזר האינסופי, או אחרי קטע הריקוד השני, כהוראה אחרונה בתוך הוראת הביצוע החוזר האינסופי:

לכאורה, לא נראה שיש משמעות למיקום של הוראת ההיפוך, כלומר אם תתבצע אחרי הקטע השני או לפני הקטע הראשון, שהרי אלו חוזרים שוב ושוב במעגל. הדבר מזכיר את המשחק שבו לפעמים משתעשעים ילדים קטנים: אם נחזור

שוב ושוב על המילה "רסתי" יישמע הדבר כאילו חזרנו שוב ושוב על המילה "תירס", בגלל החזרה המעגלית. החזרה שוב ושוב גורמת לכך שההברה "רס" נשמעת כאילו מבטאים אותה אחרי ההברה "תי" למרות שהיא מבוטאת לפנייה.

אבל בעצם יש הבדל, והוא נעוץ בתחילת הביצוע החוזר. ראינו כי הבעיה נוצרת לפני שהרקדנים חוזרים לביצוע קטע הריקוד הראשון. בפעם הראשונה שהם עומדים לבצע אותו, הבעיה לא קיימת, כי האתחול שביצענו דאג לכך שיימצאו בכיוון המתאים. מה יקרה אם נהפוך את כיוונם מיד עם תחילת הריקוד? כלומר, אם נפעל לפי האפשרות הראשונה ונשלב את הוראת היפוך הכיוון לפני קטע הריקוד הראשון, כהוראה ראשונה בתוך הוראת הביצוע החוזר האינסופי? נסו זאת.

## התקלה חזרה: המיקום של הוראת הסיבוב הוא חשוב

שוב נוצרת התופעה שהתרחשה בתרגיל 4. היפוך הכיוון קלקל את האתחול, וכבר בקטע הריקוד הראשון הרקדנים מתרחקים זה מזה, ומאורע הפגישה אינו יכול להתרחש.

לכן נבחר באפשרות השנייה. הוראת היפוך הכיוון לא תבוצע בכל פעם לפני קטע הריקוד הראשון, אלא אחרי קטע הריקוד השני. כלומר, נשלבה כהוראה אחרונה בתוך הוראת הביצוע החוזר האינסופי, לאחר הביצוע החוזר המוגבל מראש.

הוסיפו את הוראת היפוך הכיוון במקום המוצע ונסו להפעיל את האנימציה שנוצרה.

בדוגמה זו ראינו כי ניתן לשלב הוראות לביצוע חוזר זו בתוך זו. אך צריך לעשות זאת בזהירות, בעזרת תכנון מוקפד מראש ובדיקה מחושבת של התוצאה.

עדכנו את תיעוד הפרויקט ושמרו אותו בשם חדש.

שם הקובץ: **dance-indefinitely**



## מושג חדש: קינון של ביצועים חוזרים

בתוך הוראה לביצוע חוזר מסוג כלשהו ניתן לשלב כל הוראה, בפרט הוראה לביצוע חוזר. בכך מתקבל קינון של ביצועים חוזרים.

## שילובים אפשריים של ביצוע חוזר

**תרגיל 6:** האם כל השילובים של הוראות לביצוע חוזר זו בתוך זו, הם אפשריים? למשל, האם נוכל לשלב הוראה לביצוע חוזר אינסופי בתוך הוראה אחרת לביצוע חוזר?

ובכן, ישנם הרבה שילובים אפשריים, אך לא כולם מועילים. אין טעם לשלב הוראה לביצוע חוזר אינסופי (פשוט או משולב בבדיקה) בתוך הוראה כלשהי לביצוע חוזר (מוגבל מראש, מותנה או אינסופי). כפי שכבר ראינו, גם צורת הלבנים המתאימות לביצוע חוזר אינסופי אינה מאפשרת את שילובן בתוך הוראה אחרת. כאמור, שילוב כזה אין בו היגיון, כי הביצוע החוזר האינסופי אינו צפוי להסתיים, ואף פעם לא יגיע תורן של ההוראות הבאות בתור להתבצע.

שימו לב: כך הדבר גם אם נרצה לשלב הוראה לביצוע חוזר אינסופי כהוראה אחרונה בתוך הוראה אחרת לביצוע חוזר, כפי שמוצג באופן סכמטי בשרטוט הבא:



גם במקרה זה אין היגיון בשילוב. אמנם, ההוראה לביצוע חוזר אינסופי היא אחרונה, אבל ישנן הוראות המיועדות להתבצע אחריה. מדובר בהוראות שמופיעות לפנייה בתוך ההוראה לביצוע חוזר, ושלפיכך הן מיועדות להתבצע בסיבוב הבא בלולאה. אם ההוראה האחרונה בתוך הביצוע החוזר היא הוראה לביצוע חוזר אינסופי, לא יסתיים ביצועה לעולם, וכבר ביצעו של הסיבוב השני לא יוכל להתחיל.

## דוגמה 4: כוראוגרף נולד – הצופים שולטים על הריקוד

באנימציות שיצרנו עד כה, רצף התנועות נקבע מראש. לצופים באנימציה לא הייתה אף פעם השפעה על הרצף התנועתי שיתרחש. הם יכלו רק לצפות בתוצאה, כצופים בסרט קולנוע. אמנם, צפייה בסרטים היא מהנה, אבל כפי שבוודאי חוויתם, כאשר שיחקתם במשחקי מחשב שונים, שימוש במחשב יכול לספק גם חוויה יותר פעילה, שבה הצופים יכולים גם להשפיע על הדמויות ופעולותיהן.

### משימה 4



שנו את האנימציה כך שהצופים יוכלו להשפיע על תנועות הרקדנים. הרקדנים ינועו שוב ושוב, באופן דומה למה שכבר ראינו, אך את כיוון תנועתם יהפכו לא בעקבות מאורע של פגישה, או כעבור מספר ידוע מראש של צעדים, אלא בעקבות הוראה של הצופה.

שם הקובץ: `user-controls-dance`

הרקדנית תתחיל כמו קודם את תנועתה ימינה ותמשיך בה עד שהצופה יורה לה לפנות לכיוון ההפוך ולהתחיל לנוע שמאלה. זאת יעשה על ידי לחיצה על מקש החץ השמאלי. כאשר היא נעה שמאלה, יוכל לסמן לה לשוב ולנוע ימינה על ידי לחיצה על מקש החץ הימני.

כעת התיאור המילולי המתאים לקטע הריקוד הראשון של הרקדנית הוא התיאור הבא:

1. כ-83 שניות מקד החץ השמאלי:

1.1 10 sds 83 ד"ס

1.2 האמן 0.2 שני

ואילו התיאור המילולי שמתאים לקטע הריקוד השני של הרקדנית הוא התיאור הבא:

1. כ-83 שניות מקד החץ הימני:

1.1 10 sds 83 ד"ס

1.2 האמן 0.2 שני

אם כך את שתי הלולאות שבתסריט, הן זו של הביצוע החוזר המוגבל מראש והן זו של הביצוע החוזר המותנה, נרצה להחליף בלולאות אחרות. ההוראות המשולבות בהן יהיו זהות לאלו ששולבו בלולאות הקודמות, אך המסגרת, כלומר, ההוראה הממוקדת שמקיפה אותן, תהיה שונה. גם הפעם דרוש לנו ביצוע חוזר מותנה, אך הפעם המאורע שייבדק לא יהיה מאורע נגיעה, אלא מאורע לחיצה על מקש מהמקלדת.

### תגובה ללחיצה על מקש

אם כן, נשתמש בהוראה הרגילה לביצוע חוזר מותנה, אך נשלב בה תנאי חדש. את התנאי המתאים נמצא שוב בקבוצת החיישנים, הקבוצה הכחולה-בהירה. צורת הלבנה היא כשל לבנת תנאי, וזו הלבנה, כרגיל, לחיצה על החץ הקטן שבצידה הימני של החלונת המשולבת בלבנה, תגרום להופעת רשימה. במקרה זה תכלול הרשימה את כל מקשי המקלדת ונוכל לבחור במקש הרצוי לנו.

## תנאי חדש ב־Scratch: לחיצה על מקש

התנאי **מקש** **נלחץ?** מייצג שאלה שהתשובה עליה היא "כן" אם נלחץ המקש שמופיע בחלונית, ו"לא" אם הוא לא נלחץ.

שנו את תסריטה של הרקדנית כך שיתאים לתיאורים המילוליים שנתנו ויכלול שתי הוראות לביצוע חוזר מותנה. האחת תלויה במאורע לחיצה על מקש חץ שמאלי, והשנייה תלויה במאורע לחיצה על מקש חץ ימני.

## תנועת הרקדן

את התנהגותו של הרקדן נרצה לשנות באופן דומה, אך עבורו יש להקדיש שני מקשים אחרים. נבחר במקש החץ מעלה ובמקש החץ מטה. כדי שהחצים ייצרו השפעה הגיונית, נשנה את תנועתו של הרקדן. במקום לנוע בקטע הריקוד הראשון שמאלה, הוא ינוע בו למטה, עד שהצופה ילחץ על המקש של חץ מעלה; בקטע הריקוד השני, במקום לנוע ימינה, ינוע למעלה, עד שהצופה ילחץ על המקש של חץ מטה. לכן ראשית יש לשנות את האתחול של הרקדן כך שכיוונו ההתחלתי יהיה כלפי מטה (180). לאחר מכן נשנה את ההוראות לביצוע החוזר כך שיתאימו לתיאורים המילוליים הבאים:

1. **הִסָּרֵט מִמָּקַד הַחֵץ הַמַּלְא:**

1.1 **10 סֵג** **10 דִּיֵּס**

1.2 **הֲמָג 0.2 שְׁנִי**

1. **הִסָּרֵט מִמָּקַד הַחֵץ הַמַּטָּה:**

1.1 **10 סֵג** **10 דִּיֵּס**

1.2 **הֲמָג 0.2 שְׁנִי**

שנו את התסריט של הרקדן כפי שהוצע כאן. הוסיפו הערות מתאימות בכל התסריטים, שמרו את הפרויקט בשם חדש ונסו להפעילו.

## תרגיל 7:

בתרגיל זה יש ליצור כמה אנימציות ובהן פיל, פרה וסוס נעים לרוחב הבמה.

א. הכינו גרסה ראשונית שבה שלוש החיות מתייצבות עם הלחיצה על הדגל הירוק בצדה השמאלי של הבמה, זו מעל זו, ואז נעות במקביל אל צדה הימני של הבמה. תנועתה של כל חיה תהיה מורכבת מ־60 תזוזות קטנות של 5 צעדים. **שם הקובץ: fixed-run**

ב. הרחיבו את האנימציה שיצרתם בסעיף א' לאנימציה חדשה, שבה בעקבות הלחיצה על הדגל הירוק כל חיה מתייצבת, כמקודם, בצדה השמאלי של הבמה ומתחילה לנוע בתזוזות קטנות אל צדה הימני של הבמה, אך נעצרת כאשר נלחץ מקש המתאים עבורה. לחיצה על מקש החץ השמאלי תגרום לעצירת הסוס, לחיצה על מקש החץ הימני תגרום לעצירת הפרה, ולחיצה על מקש חץ למעלה תגרום לעצירת הפיל. **שם הקובץ: play-run1**

ג. צרו אנימציה אחרת, שבה עם הלחיצה על הדגל הירוק כל חיה מתייצבת בצדה השמאלי של הבמה, ורק בעקבות לחיצה על המקש המתאים לה, היא מתחילה לנוע בתזוזות קטנות אל צדה הימני של הבמה וממשיכה בתנועתה כל עוד המקש המתאים לחוץ. הפסקת הלחיצה על המקש תגרום לעצירת החיה, אך לחיצות נוספות יגרמו לתזוזות נוספות, שנמשכות כל עוד המקש המתאים נשאר לחוץ.  
**שם הקובץ:** `play-run2`

ד. צרו אנימציה חדשה, שדומה לאנימציה שיצרתם בסעיף ב'. גם באנימציה החדשה עם הלחיצה על הדגל הירוק כל חיה מתייצבת בצדה השמאלי של הבמה, מתחילה לנוע בתזוזות קטנות אל צדה הימני של הבמה ונעצרת כאשר נלחץ המקש המתאים עבורה. אבל באנימציה החדשה התנועה תתחדש כאשר המקש המתאים יילחץ שוב, וכך זה ימשיך: לחיצה נוספת תגרום לעצירה, ולחיצה נוספת תגרום לחידוש תנועתה, וכך הלאה. ייתכן שתצטרכו לשלב הוראות המתנה, כדי שיהיה לצופים זמן מספיק ללחוץ על מקש ולשחררו.

**שם הקובץ:** `play-run3`

## הרחבה (רשות): שינוי תלבושת תוך כדי אנימציה

הריקוד הבסיסי שיצרנו בפרק הזה חסר מאפיין יסודי של ריקוד. אמנם נראית בו תנועה, ותנועה זו מזכירה תנועת ריקוד, בגלל ההמתנות הקצרות שהכנסנו בין תזוזה לתזוזה, אבל הדמות הנעה נראית תמיד באותו מצב: גופה, ובפרט ידיה ורגליה, מוצבים תמיד באותו האופן. תנועת ריקוד אמينة יותר הייתה מתקבלת לו יכולנו "להניע" את איברי גופה של הדמות הרוקדת. זאת תהיה המשימה שבמרכז הסעיף הזה.

### משימה 5



האנימציה שיצרנו בדוגמה 3 תשלב תנועת ריקוד אמينة יותר, המראה גם שינוי במצב הגוף, ולא רק תזוזה.  
**שם הקובץ:** `dancers-change-costumes`

אשליית תנועה כזאת ניתן להשיג על ידי החלפה מתמדת בין שני מצבים אפשריים של הגוף. למעשה, כך מייצרים תנועה באנימציות קלסיות: למשל, כדי להראות תנועת הליכה, מחליפים במהירות בין שני ציורים או יותר, המראים כל אחד מצב גוף מעט שונה בשעת הליכה. ההחלפה המהירה, כאשר היא משולבת עם תזוזה על פני השטח, יוצרת אשליה של רצף תנועת.

עקרון כזה אפשר ליישם ב־Scratch על ידי שימוש בתלבושות שונות לאותה דמות. כפי שראינו בפרק 2, סביבת Scratch מאפשרת להחליף את תלבושתה של דמות על ידי ייבוא של תלבושת חדשה מתוך המאגר הקיים או על ידי עריכת התלבושת הקיימת. למעשה, ב־Scratch אפשר שיהיו לדמות כמה וכמה תלבושות, ואפילו אפשר להחליף ביניהן תוך כדי ביצוע תסריט. ההחלפה הזאת תאפשר לנו לדמות שינוי של מצב הגוף.

נדגים את השינוי עבור התנהגותה של הרקדנית בפרויקט שפותח עבור דוגמה 3. בחלון האפור המוארך נבחר בלשונית התלבושות. כרגע מופיעה עבור הרקדנית תלבושת אחת בלבד, והיא מוקפת במסגרת כחולה, המציינת כי היא התלבושת הפעילה. מעל התלבושת כתובה השורה "תלבושת חדשה", ולידה

### תלבושת חדשה: צייר יבא מצלמה

שלושה כפתורים, המייצגים שלוש דרכים לייצר תלבושת חדשה: "צייר", לציור תלבושת בעזרת עורך הציור, "יבא" לשימוש בתלבושת קיימת במאגר, ו"מצלמה" לצילום תלבושת חדשה. נוסף לרקדנית עוד תלבושת על ידי בחירה בכפתור "יבא", ובתיקייה המוכרת People נחפש דמות של אותה רקדנית אך במצב ריקוד שונה מעט, למשל, הדמות הקרויה ברשימה



cassy-dancing-1. לאחר הבחירה נלחץ על הכפתור OK. כעת יש בחלון התלבושות שתי תלבושות. התלבושת האחרונה שבחרנו מוקפת במסגרת כחולה, המציינת כי היא התלבושת הפעילה. ואכן, היא גם התלבושת שנראית כעת על הבמה. בחירה בתלבושת הראשונה תהפוך אותה לתלבושת הפעילה, כפי שניתן לראות על ידי צבע המסגרת והשינוי על הבמה.

### פעולה חדשה ב-Scratch: הוספת תלבושת לדמות

ניתן להוסיף לדמות תלבושות. לשם כך יש קודם כל לבחור בלשונית "תלבושות" שבחלון התסריטים. ניתן להוסיף תלבושות לדמות על ידי ייבוא מקובץ, ציור בעזרת עורך הציור או קליטת תמונה ממצלמה דיגיטלית.

כעת כאשר יש לרקדנית שתי תלבושות, נרצה להחליף ביניהן תוך כדי הריקוד. הדרך היחידה להורות לדמות לבצע פעולה כלשהי, כולל החלפת תלבושת, היא בעזרת הוראות המשולבות בתסריטים של הדמות. אם כן נבחר שוב בלשונית התסריטים, כי עלינו למצוא הוראה מתאימה להחלפת תלבושת, ולמצוא את המקום המתאים לה בתסריט של הרקדנית.

### הוראות לשינוי תלבושות

המקום הטבעי לחפש הוראות העוסקות בתלבושת הוא בקבוצת המראה, הקבוצה הסגולה. שתי ההוראות הראשונות בקבוצה עוסקות בשינוי תלבושת. ההוראה הראשונה **שנה לדמות** מאפשרת לקבוע תלבושת מסוימת שנרצה כי תחליף את התלבושת הנוכחית. בלבנה המתאימה להוראה זו יש חץ עם חלונית, וכדי להפוך את ההוראה להוראה ממוקדת, יש ללחוץ על החץ ולבחור אחת התלבושות שברשימה, רשימה שמכילה את כל התלבושות המופיעות בחלון התלבושות של הדמות.

**שימו לב:** למרות שההוראה היא "שנה לדמות", מה שמשתנה הוא התלבושת של הדמות ולא הדמות עצמה.



### הוראה חדשה ב-Scratch: שינוי לתלבושת מסוימת

ההוראה **שנה לדמות** גורמת לדמות להחליף את תלבושתה לתלבושת שמספרה הסידורי מופיע בחלונית.

ההוראה השנייה אומרת בפשטות **התלבושת הבאה**. הכוונה היא שבעקבות ביצוע ההוראה הדמות תשנה את תלבושתה לתלבושת הבאה המופיעה בחלון התלבושות. למשל, אם בחלון התלבושות של הדמות יש ארבע תלבושות, והתלבושת הפעילה היא השנייה מביניהן, ביצוע ההוראה **התלבושת הבאה** יגרום לדמות להחליף את תלבושתה לתלבושת השלישית. ביצוע נוסף של ההוראה **התלבושת הבאה** יגרום לדמות להחליף את תלבושתה לתלבושת הרביעית. ביצוע נוסף של ההוראה **התלבושת הבאה** יגרום לדמות להחליף את תלבושתה לתלבושת הראשונה, וכך הלאה.

### הוראה חדשה ב-Scratch: שינוי לתלבושת הבאה

ההוראה **התלבושת הבאה** גורמת לדמות להחליף את תלבושתה לתלבושת המופיעה ברשימת התלבושות של הדמות לאחר התלבושת הנוכחית (רשימת התלבושות נמצאת בחלון התלבושות של הדמות).

לגבי כל אחת משתי ההוראות לשינוי תלבושת, חשבו – האם היא הוראה מוחלטת או יחסית?



ההוראה הראשונה **שנה לדמות** היא הוראה מוחלטת. תהיה התלבושת הנוכחית של הדמות אשר תהיה, בעקבות ביצוע ההוראה תמיד תהיה תלבושתה הפעילה זו הכלולה בהוראה הממוקדת. ההוראה השנייה **התלבושת הבאה** היא הוראה יחסית. תלבושתה הפעילה של הדמות לאחר ביצוע ההוראה תלויה בתלבושת אשר הייתה פעילה לפני ביצוע ההוראה.

## שימוש בהחלפת תלבושות ליצירת אשליה של תנועה

כדי לגרום לדמות הרקדנית להחליף שוב ושוב בין שתי התלבושות שברשימה שלה, נוכל להשתמש בהוראה **התלבושת הבאה**. היכן נשלב אותה? תנועת הריקוד של הרקדנית כבר מפורקת למקטעים מקטעים. כל מקטע כולל תזוזה קטנה והמתנה. נוכל להוסיף לכל מקטע גם שינוי תלבושת. לכן נוסיף את ההוראה בין הוראת התזוזה להוראת ההמתנה, הן בלולאה הראשונה והן בלולאה השנייה.

כעת כאשר יש לדמות כמה תלבושות, והיא מחליפה ביניהן במהלך ביצוע האנימציה, חשוב להתייחס גם לתלבושת כאשר מאתחלים את מצבה של הדמות בתחילת התסריט. עוד בפרק 1 ראינו כי ללא אתחול, מצבה של דמות בתחילת ביצוע אנימציה, עם הלחיצה על הדגל הירוק, אינו קבוע, והוא תלוי במצבה בסיום ביצוע האנימציה בפעם האחרונה. כך גם לגבי התלבושת. אם לא נדאג לאתחל את תלבושתה של הדמות, ייתכן כי בהפעלות חוזרות של האנימציה תתחיל הדמות בכל פעם בתלבושת אחרת, היא התלבושת הפעילה שלה כאשר נעצרה ההפעלה הקודמת של האנימציה בעקבות הלחיצה על הכפתור האדום.

לצורך האתחול נוכל להשתמש בהוראת שינוי התלבושת המוחלטת **שנה לדמות**. בחרו בלבנה המתאימה להוראה ושלבו אותה בתסריט של הרקדנית כהוראה ראשונה, מיד אחרי הוראת הבקרה שמתנה תחילת ביצוע בלחיצה על הדגל הירוק. הפכו את ההוראה להוראה ממוקדת על ידי

לחיצה על החץ שמימין לחלונית ובחירה באחת התלבושות, למשל, בתלבושת איתה עבדנו עד כה:



נסו להפעיל את האנימציה כדי לראות את תנועת הריקוד החדשה.

**תרגיל 8:** בתרגיל זה נרצה לערוך שינוי מקביל עבור הרקדן. לרקדן המשתתף באנימציות שיצרנו בפרק זה אין תלבושת נוספת במאגר התלבושות. לכן לצורך השינוי נבחר בשתי תלבושות של רקדן אחר. בחלון התלבושות של דמות הרקדן יפאו שתי תלבושות חדשות מהתיקיה People, למשל, של dans ואחר כך של dan1, ומחקו את התלבושת שהשתמשתם בה עד כה. כעת שנו את התסריט של הרקדן בדומה לשינויים שנערכו עבור הרקדנית.

הוסיפו הערות מתאימות ושמרו את הפרויקט בשם חדש.

## הרחבה (רשות): הוספת צליל רקע

סביבת Scratch מאפשרת גם הוספת צלילי רקע.

### משימה 6

שם הקובץ: `dance-with-music`

לריקוד שנבנה עבור דוגמה 3 נוסיף גם ליווי מוזיקלי.



בדומה לציור הרקע של האנימציה (כפי שהוצג בהרחבה לפרק 3), צלילי הרקע אינם משויכים לאחת הדמויות המשתתפות בפרויקט, אלא לבמה. לכן כדי להוסיף לאנימציה צלילי רקע, יש לבחור בחלון הדמויות בריבוע המייצג את הבמה.

כעת בחרו בלשונית הצלילים בחלון התסריטים, החלון האפור המוארך שבמרכז. בדומה ללשונית התלבושות, כאן מופיעה רשימה של צלילים שיכולים להשתתף באנימציה. כדי להוסיף צליל חדש לרשימה, אפשר לייבא אותו מתוך הצלילים שמספקת סביבת Scratch או להקליט צליל חדש. רק אחרי שצליל מופיע ברשימה, אפשר לבחור אותו כצלילי רקע.

בחרו באפשרות של ייבוא צליל ואז בחרו למשל בתיקיה Music Loops. מבין הצלילים הכלולים בתיקיה זו נבחר למשל בצליל הגיטרה GuitarChords2. עם סימונו, עוד לפני הלחיצה על המקש OK לאישור הבחירה, יישמע הצליל המתאים כדי שתוכלו להתרשם ולהחליט בהתאם לכך על בחירתכם.

כעת הצליל החדש מופיע ברשימת הצלילים המוכרים. מתחת לשמו כתוב אורכו (במקרה זה שבע שניות), ובעזרת המקשים הצמודים אליו ניתן לשמוע אותו, לעצור את ההשמעה, או למחוק אותו מרשימת הצלילים המוכרים.

## פעולה חדשה ב־Scratch: בחירת צלילים

כדי להוסיף צליל יש לבחור בלשונית "צלילים" בחלון התסריטים. ניתן לייבא קטעי צליל מקבצים בפורמטים שונים כגון mp3 ו־wav. ניתן גם להקליט קטעי צליל ממיקרופון המחשב למחשב.

## קביעת צליל רקע של הבמה

לאחר שהצליל מופיע ברשימת הצלילים המוכרים, ניתן לבחור אותו כצליל רקע. מאחר שהבמה אחראית לצלילי הרקע, כדי לבחור צליל כצליל רקע, צריך להורות לבמה לנגנו. כלומר, יש להכין גם לבמה תסריט ובו הוראה להשמעת צליל.

אם כך חזרו שוב ללשונית התסריטים, כאשר בחלון הדמויות מסומנת הבמה. עבור הבמה, תכניו של חלון מאגר ההוראות המכיל לבנים שניתן לשלבן בתסריט, הם מעט שונים מאלו שאנו רגילים לראות. כך למשל, בחירה בקבוצה הכחולה, קבוצת הוראות התנועה, מובילה לחלון ריק, שאינו מכיל אף לבנה. זאת משום שאין הבמה יכולה לזוז, ולכן אין משמעות לשילוב הוראות תנועה בתסריטים שלה. גם בקבוצות אחרות ייתכנו הבדלים. למשל, הקבוצה הכחולה־בהירה, קבוצת החיישנים, אינה כוללת את לבני התנאי הבודקים נגיעה. הבמה תמיד נוגעת בכל שטחה, ובפרט בכל הדמויות, במסגרת ובמצביע העכבר, ולכן אין טעם לבדוק נגיעה ביחס אליה.

אנו נבחר בקבוצה הסגולה־זרודה, קבוצת הוראות הצלילים. בין ההוראות הכלולות בה שתי ההוראות הראשונות, המורות על ניגון צליל, נראות מתאימות לצרכינו. בכל אחת מהן משולבת חלונית עם חץ קטן בימינה. לחיצה על החץ תגרום לפתיחת רשימה ובה כל הצלילים המוכרים עבור הבמה. ההבדל בין שתי ההוראות הוא באופן שבו נשלב את השמעת הצליל שנבחר בביצוע התסריט. ההוראה הראשונה **נגן צליל** תגרום לתחילת השמעת הצליל ואז מעבר מיידי להמשך ביצוע התסריט, גם אם השמעת הצליל טרם הסתיימה. ההוראה השנייה **נגן צליל עד לסיום** תגרום להשמעת הצליל, ורק אחרי שהשמעתו מסתיימת ימשיך ביצוע התסריט מההוראה הבאה.

## הוראות חדשות ב־Scratch: נגינת צליל

ההוראה **נגן צליל** גורמת לדמות (או לבמה) לנגן את קטע הצליל שמשולב בחלונית. מיד לאחר שהצליל מתחיל להתנגן, ביצוע התסריט ממשיך מההוראה הבאה.

ההוראה **נגן צליל עד לסיום** אף היא גורמת לדמות (או לבמה) לנגן את קטע הצליל שמשולב בחלונית. אבל ביצוע התסריט ממשיך רק לאחר שהשמעת קטע הצליל מסתיימת.

בחרו בלבנה המתאימה להוראה השנייה וגררו אותה אל חלון התסריטים של הבמה. צרפו לפנייה את ההוראה שמתנה תחילת ביצוע עם לחיצה על הדגל הירוק.

כעת לחיצה על הדגל הירוק אכן תגרום להשמעת הצליל שבחרנו במקביל לריקוד הרקדן והרקדנית.



אבל קשה לקרוא לו צליל רקע. השמעתו אורכת זמן קצר, אותן שבע שניות, ולאחר מכן חוזר השקט. כיצד נהפוך אותו לצליל רקע של ממש?

עלינו לגרום לבמה לנגנו שוב ושוב, כל עוד נמשך הריקוד. מאחר שהריקוד, הן של הרקדן והן של הרקדנית, הוא אינסופי, כלומר, נשלט על ידי הוראות לביצוע חוזר אינסופי, גם השמעת הצליל צריכה לחזור שוב ושוב עד אינסוף. לכן יש לשנות את תסריטה של הבמה כך שההוראה להשמעת הצליל עד סיומו תשולב בתוך הוראה לביצוע חוזר אינסופי.

שנו את התסריט ונסו להפעילו. הוסיפו הערה מתאימה ושמרו את הפרויקט בשם חדש.

## הרחבה (רשות): שינוי צליל רקע תוך כדי אנימציה בהתאם למתרחש בה

צליל הרקע אינו חייב להיות קבוע לאורך כל ביצוע האנימציה. הבמה יכולה לשנות את צליל הרקע. למשל, בתוך ההוראה לביצוע החוזר האינסופי שנכללה בהרחבה הקודמת, אפשר לכלול הוראה נוספת לניגון צליל. כתוצאה מכך ישתנה צליל הרקע שוב ושוב: יושמע הצליל הראשון, אחר כך השני, שוב הראשון ואז השני, וחוזר חלילה. שינוי צליל הרקע יכול להיות מושפע גם מהמתרחש באנימציה עצמה.

### משימה 7



המוזיקה המלווה את הרקדנים תשתנה בהתאם לכיוון הריקוד.

שם הקובץ: `dance-change-music`

נחזור לדוגמה 3, שבה הרקדן והרקדנית חוזרים שוב ושוב על שני קטעי הריקוד הקבועים, כלומר, מתקרבים זה לזה, ואחר כך מתרחקים זה מזה, וחוזר חלילה. נוכל להשמיע צליל רקע אחד כאשר הם בתנועה זה לכיוון זה, וצליל רקע אחר כאשר הם מתרחקים זה מזה. כיצד נעשה זאת?

פעולותיה של הבמה, פעולות שינוי הצליל, צריכות עכשיו להיות תלויות במצבן של הדמויות האחרות. זה מחייב תקשורת בין הבמה לדמויות האחרות. כפי שראינו בפרק הקודם, דרך אפשרית לתקשורת בין הדמויות היא על ידי העברת מסרים.

## בקשה לשינוי צליל הרקע על ידי שידור מסר

כאשר הדמויות מתקרבות זו לזו, הן ישדרו מסר מתאים. הבמה תקלוט את המסר ובהתאם לכך תנגן צליל אחד. כאשר הדמויות מתרחקות זו מזו, הן ישדרו מסר אחר. הבמה תקלוט אותו וכתוצאה מכך תנגן צליל אחר.

למעשה, אין צורך ששתי הדמויות, גם הרקדן וגם הרקדנית, ישדרו את המסרים. בגלל האופן שבו כתובים התסריטים שלהם, תנועותיהם כבר מתואמות. מספיק שאחד מהם יודיע על התקרבות או התרחקות. נבחר באופן שרירותי את הרקדנית כמשדרת המסרים. עקב כך צריך לשנות בדוגמה 3, לצורך פתרון משימה 7, גם את תסריטה של הרקדנית וגם לכתוב הוראות פעולה עבור הבמה.

השינוי בתסריטה של הרקדנית יהיה פשוט: בכל פעם שהיא עומדת להתחיל בקטע ריקוד חדש, עליה לשדר מסר מתאים. לכן נוסיף הוראת שידור מסר לפני כל אחת מהלולאות בתסריטה של הרקדנית.

כזכור מפרק 4, הוראת שידור מסר כלולה בקבוצת הוראות הבקרה. בחרו בלבנה המתאימה להוראה. גררו אותה מיד לפני הוראת הביצוע החוזר המותנה. בחלונית שבלבנה בחרו במסר חדש והקלידו מסר מתאים (למשל, "מתקרבים"). בחלון מאגר הוראות הבקרה בחרו שוב בלבנה המתאימה להוראת שידור מסר וגררו אותה מיד לפני הוראת הביצוע החוזר המוגבל מראש. בחלונית שבלבנה בחרו במסר חדש והקלידו מסר מתאים (למשל, "מתרחקים").

## שינוי צליל הרקע כתגובה לקבלת מסר

נעבור כעת לתפקידה של הבמה. כאשר הבמה מקבלת מסר "מתקרבים", עליה לנגן צליל אחד. כאשר היא מקבלת מסר "מתרחקים", עליה לנגן צליל שני. ההוראה המתאימה לזיהוי קבלת מסר היא ההוראה

נשמע קבל מסר

שהכרנו בפרק 4. כפי שהוסבר בפרק 4, הוראת ביצוע המותנה בקבלת מסר היא תמיד הוראה ראשונה בתסריט. לכן בדומה למה שנעשה בפרק 4 עבור הכדורגלן והכדור, גם בדוגמה זו יש להכין עבור הבמה שני תסריטים שיעבדו במקביל. שניהם תסריטים דומים מבחינת ההוראות הכלולות בהם, אך נבדלים זה מזה בערכים המשולבים בהוראות.

ודאו כי ברשימת הצלילים המוכרים לבמה מופיעים שני צלילים (למשל, GuitarChords2 ו-GuitarChords1, הנמצאים שניהם בתיקייה Music Loops). תוכלו לבדוק זאת אם תבחרו בלשונית הצלילים. כעת הכינו בחלון התסריטים שני תסריטים דומים (תוכלו להכין קודם אחד מהם ולאחר מכן להיעזר בתפריט פעולות העכבר, שהוצג בהרחבה בפרק 2, כדי לשכפל). כל תסריט יתחיל בהוראה שמתנה ביצוע בקבלת מסר ואחריה הוראת ניגון צליל עד סיומו. דאגו לשלב בכל תסריט את המסר המתאים לו ואת הצליל המתאים לו.

נסו להפעיל את האנימציה. האם התוצאה שמתקבלת היא התוצאה המתוכננת?

## ניתוח תקלה ופתרונה

התוצאה נשמעת צורמת. זאת משום שישנה חפיפה חלקית בין הצלילים. לפני שמסתיימת נגינתו של הצליל הראשון, כבר מתחילה נגינתו של הצליל הבא. מדוע זה כך? האם אין הבחירה בהוראה **נגן צליל** עד לסיום אמורה להביא לכך שנגינתו של צליל תסתיים לפני שמתבצעת ההוראה הבאה?

לא בדיוק. ההוראה **נגן צליל** עד לסיום גורמת להשמעת הצליל עד סיומו לפני ביצוע ההוראה הבאה **באותו תסריט**. במקרה שלנו, ההוראה להשמעת הצליל השני נמצאת בתסריט אחר! שני התסריטים עובדים במקביל ואינם תלויים זה בזה. הרקדנית משדרת מסר התקרבות, ובמקביל לתנועת הרקדנים זה אל זה מנוגן הצליל הראשון, אך עוד לפני שהוא מגיע אל סיומו, הרקדנים כבר נפגשים, מסר ההתרחקות משודר והשמעת הצליל השני מתחילה.

כדי לפתור זאת, אפשר בכל אחד מהתסריטים לשלב לפני ההוראה להשמעת הצליל הוראה המשתקה את כל הצלילים. לבנה המתאימה להוראה זו היא הלבנה השלישית בקבוצת הצלילים הסגולה-ורודה

הפסק את כל הצלילים

## הוראה חדשה ב־Scratch: הפסקת כל הצלילים

ההוראה **הפסק את כל הצלילים** גורמת להפסקת השמעתם של כל הצלילים מכל התסריטים של כל הדמויות.

שנו את תסריטיה של הבמה בהתאם. עדכנו את תיעוד הפרויקט, שמרו אותו בשם חדש ונסו להפעילו.

### תרגיל 9:

בתרגיל זה יש ליצור כמה אנימציות שבהן מנגנים שלושה כלי נגינה: תוף, גיטרה וחצוצרה. הפעם הצלילים לא יהיו צלילי רקע, אלא צלילים של הדמויות, ולכן יש להגדיר כל צליל אצל הדמות המתאימה לו, בלשונית הצלילים שלה. ניתן להשתמש בפרויקט costumes-instruments בו מוגדרות תלבושות עבור שלושת הכלים.

א. הכינו אנימציה שבה שלושת הכלים מתחילים לנגן עם הלחיצה על הדגל הירוק. כל כלי ינגן צליל משלו 5 פעמים.

שם הקובץ: **fixed-tune**

ב. הרחיבו את האנימציה שיצרתם בסעיף א' לאנימציה חדשה, שבה בעקבות הלחיצה על הדגל הירוק כל כלי מתחיל, במקודם, לנגן את הצליל שלו, אך מפסיק את השמעת הצליל כאשר נלחץ מקש המתאים עבורו. לחיצה על מקש החץ השמאלי תגרום להפסקת השמעת צליל התוף, לחיצה על מקש החץ הימני תגרום להפסקת השמעת צליל החצוצרה, ולחיצה על מקש חץ למעלה תגרום להפסקת השמעת צליל הגיטרה. שימו לב: ייתכן שבעת הפעלת האנימציה ייאלצו הצופים ללחוץ על מקש לחיצה ארוכה לפני שתורגש השפעתו. מדוע כך הדבר?

שם הקובץ: **play-tune1**

ג. צרו אנימציה אחרת שבה השמעות הצלילים של שלושת כלי הנגינה נשלטות על ידי הצופים באופן הבא: עם הלחיצה על הדגל הירוק לא נשמע דבר, ורק בעקבות לחיצה על מקש מתאים לו, כל כלי נגינה מתחיל להשמיע את הצליל שלו, והוא ממשיך בהשמעת הצליל כל עוד המקש המתאים לחוץ. הפסקת הלחיצה על המקש תגרום לעצירת השמעת הצליל, אך לחיצות נוספות יגרמו להשמעות נוספות, שנמשכות כל עוד המקש המתאים נשאר לחוץ.

שם הקובץ: **play-tune2**

ד. צרו אנימציה חדשה שדומה לאנימציה שיצרתם בסעיף ב'. גם באנימציה החדשה כל כלי נגינה מתחיל להשמיע את הצליל שלו עם הלחיצה על הדגל הירוק ומפסיק את השמעת הצליל כאשר המקש המתאים עבורו נלחץ. אבל באנימציה החדשה תתחדש השמעת הצליל כאשר יילחץ שוב המקש המתאים, וכך זה ימשיך: לחיצה נוספת תגרום לעצירת ההשמעה, ולחיצה נוספת תגרום לחידוש ההשמעה, וכך הלאה. ייתכן שתצטרכו לשלב הוראות המתנה, כדי שיהיה לצופים זמן מספיק ללחוץ על מקש ולשחררו.

שם הקובץ: **play-tune3**

## סיכום

בפרק זה הכרנו שני סוגי ביצועים חוזרים שאינם אינסופיים: **ביצוע חוזר מוגבל מראש**, המאפשר לחזור על ביצוע סדרת הוראות מספר נתון של פעמים, ו**ביצוע חוזר מותנה**, המאפשר לחזור על ביצוע סדרת הוראות עד אשר מתרחש מאורע מסוים או מתקיים תנאי מסוים.

בעת השימוש בביצוע מותנה, יש לשים לב כי המאורע הכלול בו אכן יתרחש בשלב כלשהו, כדי שלא יתקבל ביצוע אינסופי.

ניתן לשלב סוגי ביצועים חוזרים אלה בתוך אלה, אך לא ניתן לשלב הוראה לביצוע חוזר אינסופי בתוך הוראה כלשהי לביצוע חוזר.

ההוראות המתאימות של Scratch שייכות לקבוצת הוראות הבקרה, ההוראות הכתומות. ההוראה המתאימה לביצוע חוזר מוגבל מראש היא **חזור פעמים**. ההוראה המתאימה לביצוע חוזר מותנה היא



הכרנו בפרק זה הוראת בקרה נוספת. זוהי **הוראת המתנה**, המשהה את המשך הביצוע למשך מספר שניות קצוב. ההוראה המתאימה היא **חכה שניות**.

בהוראה של ביצוע חוזר מותנה יש לשלב תנאי (מאורע). מאגר התנאים של Scratch מאפשר להתנות ביצוע גם ב**לחיצת מקשי מקלדת**, ובכך מתאפשרת תקשורת דו-כיוונית עם הצופים בכל הנוגע להשפעה שלהם על המתרחש באנימציה. הלבנה המתאימה לתנאי לחיצת מקש נמצאת בקבוצת החיישנים, הכחולה-בהירה.

לדמות יכולות להיות כמה תלבושות, כאשר שינוי חוזר של תלבושת המשולב עם הוראות תזוזה יכול ליצור אשליה אמינה של תנועה. **הוספת תלבושות** לדמות נעשית דרך לשונית התלבושות שמעל החלון המרכזי המוארך. **שינוי תלבושת תוך כדי ביצוע** נעשה בעזרת הוראות הכלולות בקבוצת המראה, הקבוצה הסגולה. יש הוראה מוחלטת לשינוי תלבושת מסוימת **שנה לדמות**, והוראה יחסית **התלבושת הבאה**, המשנה לתלבושת הבאה לפי סדר התלבושות ברשימת התלבושות של הדמות.

לאנימציה ניתן לצרף **צלילי רקע**. צלילי רקע הם באחריות הבמה. על ידי בחירה בלשונית הצלילים של הבמה ניתן ליצור רשימת צלילים המוכרים לבמה, וזאת באמצעות השימוש בצלילים קיימים או הקלטת צלילים חדשים. **ניגון צלילים** נשלט על ידי הוראות הכלולות ב**קבוצת הצלילים**, הקבוצה הסגולה-זורודה. ההוראות שהכרנו בקבוצה זו הן **נגן צליל**, **נגן צליל עד לסיום**, ו**הפסק את כל הצלילים**.



## פרק 6

### משתנים – מנגנוני זיכרון

בפרק זה ניצור משחק מחשב פשוט ובסיסי. דרכו נכיר מנגנון שימושי של Scratch שמאפשר לנו לזכור ולעדכן ערכים, כגון מספר נקודות צבורות לשחקן, ולהציגם לצופים או לשחקנים.

### דוגמה 1: הדרקון המשתנה – שינוי גודל דמות

כרגיל, נתחיל מדוגמה פשוטה. את הדוגמה הזאת נרחיב בהדרגה, דרך הדוגמאות הבאות, עד שנגיע למשחק הפשוט שבכוונתנו ליצור. בדוגמה הראשונה עדיין אין לצופים תפקיד פעיל, והיא רק מציגה בפניהם דמות המשנה את גודלה.

### משימה 1: שינוי גודלה של דמות

בפרויקט תהיה דמות אחת. לחיצה על הדגל הירוק תגרום לדמות להקטין את עצמה מגודלה הרגיל למחצית מגודלה הרגיל.



שם הקובץ: change-to-half-size

אנו מציעים לבחור בדמות הדרקון הנמצאת בתיקיית הדמיון (Fantasy).

אם כך, נראה כי התסריט שעלינו ליצור עבור הדרקון הוא פשוט מאוד וניתן לתארו כך:

0. אם לחיצה על הדגל היחיד:
1. הקטן גודלך למחצית מן גודלך הנוכחי

### אתחול הגודל

בכל פעם שאנו מתכננים תסריט, כדאי לעצור ולחשוב אם נדרש בו שלב אתחול. אם כך, מה לגבי התסריט שאנו מתכננים כעת – האם נדרש בו שלב אתחול?

מאחר שאנו מעוניינים לראות איך הדרקון משנה את גודלו, עלינו לוודא כי הוא מתחיל את התסריט כשהוא בגודלו הרגיל. אמנם, בפעם הראשונה שהתסריט יופעל, יהיה הדרקון בגודלו הרגיל, אבל אחרי ההפעלה הראשונה יקטן גודלו למחצית, ואז בהפעלות נוספות לא נבחין בשינוי גודל. לכן נדרש כאן אתחול. אם כך, את התסריט שעלינו ליצור נתאר כך:

0. אם לחיצה על הדגל היחיד:
1. אגמל גודלך לגודל הנוכחי
2. הקטן גודלך למחצית מן גודלך הנוכחי

ב-Scratch יש לכל דמות גודל, וישנן הוראות שדרכן ניתן להשפיע על גודלה של דמות. הוראות המתייחסות לגודל דמות ניתן למצוא בקבוצת המראה, הקבוצה הסגולה. יש שתי הוראות המשפיעות על גודל דמות: ההוראה הראשונה, **שנה גודל ב**, היא הוראה יחסית, הגורמת לשינוי הגודל בערך מסוים ביחס לגודל הנוכחי. ההוראה השנייה, **קבע גודל ל-**, היא הוראה מוחלטת, הגורמת לשינוי הגודל לערך מסוים, ללא חשיבות לגודל הנוכחי.

שימו לב כי למרות שההוראה היא מוחלטת, הערך המשולב בה מבוטא באחוזים, אך אלו הם אחוזים מגודלה המלא של הדמות ולא מגודלה הנוכחי של הדמות.

### הוראות חדשות ב־Scratch: שינוי גודל וקביעת גודל

ההוראה **שנה גודל ב**  גורמת לדמות לשנות את גודלה במספר היחידות המופיע בחלונית.

ההוראה **קבע גודל ל-** % גורמת לדמות לשנות את גודלה לאחוז מסוים של הגודל המקורי, כפי שמופיע בחלונית.

לכן, את ההוראה 1 נוכל לייצג ב־Scratch בעזרת ההוראה המוחלטת **קבע גודל ל-** 100%, אשר גורמת לדרקון להיות בגודלו הרגיל.

כיצד נייצג ב־Scratch את ההוראה 2, הגורמת לדרקון להקטין את גודלו למחצית מגודלו הרגיל?



נוכל להשתמש לשם כך בהוראה המוחלטת **קבע גודל ל-** 50%.

### כדאי לחכות קצת...

צרו תסריט עבור הדרקון, שמתאים לתיאור שניתן כאן, ונסו להפעילו. מה קורה?

מאחר שההקטנה מתבצעת מיד עם הלחיצה על הדגל הירוק, הצופים כלל אינם יכולים לשים לב לכך שהתבצעה הקטנה – הם רואים את הדמות ישר בגודל המוקטן. כיצד נוכל לפתור זאת? נוכל להוסיף הוראת המתנה לפני הוראת ההקטנה, למשל, המתנה של שתי שניות, כך:

0. **עם לחיצה על הדגל הירוק:**

1. **אגמא גודלך לגודל הנייל**

2. **המתן שני שניות**

3. **הקטן גודלך לאחוזי מקורך הנייל**


עדכנו את התסריט בהתאם ונסו להפעילו.

## הצגת גודל הדרקון על הבמה

כעת נרצה כי גודלו של הדרקון יוצג על הבמה, כערך מספרי. כלומר, מיד עם הלחיצה על הדגל הירוק יוצג על הבמה גודלו ההתחלתי של הדרקון, ולאחר שישנה את גודלו, יוצג על הבמה הגודל החדש.

סביבת Scratch מאפשרת לנו לבצע זאת בצורה פשוטה. בחלון הוראות קבוצת המראה כלולה לבנה המתאימה לגודל הדמות הפעילה: **גודל**. ליד הלבנה מופיע ריבוע לבן קטן. לחיצה עם העכבר על הריבוע תגרום להופעת סימן "וי" בתוכו, **גודל**. לחיצה נוספת גורמת למחיקת הסימן. כאשר הלבנה המייצגת את הגודל מסומנת באופן זה, הגודל הנוכחי מופיע על המסך.

ודאו כי לבנת הגודל של הדרקון מסומנת ונסו כעת להפעיל את התסריט על ידי לחיצה על הדגל הירוק.

כל שינוי בגודל הדרקון יגרום לשינוי בערך המופיע על הבמה. בהרחבה הראשונה בפרק 2 ראינו כי ניתן לשנות גודלה של דמות על ידי שימוש בכפתורי ההגדלה וההקטנה בתפריט שמעל הבמה, . נסו להשתמש בכפתורים אלה וראו את השפעת השינויים על הערך המוצג.

תעדו את הפרויקט על ידי הערות מתאימות ושמרו אותו בשם מתאים.

## דוגמה 2: הדרקון שמשתנה לפי ערך משתנה – הגדרת משתנים

את הדוגמה הפשוטה שיצרנו נרצה להרחיב מעט, כך שהצופים יהיו פעילים מעט יותר. ניתן לצופים להשפיע על שינוי הגודל. אמנם כפי שראינו, הצופים יכולים לשנות את גודל הדרקון גם בעזרת הכפתורים שבתפריט שמעל הבמה, אבל כעת נאפשר להם לשנות את הגודל ישירות על ידי מתן ערך מספרי.

## משימה 2: הצופים קובעים את גודל הדרקון

הצופים יוכלו לתת ערך מספרי שיקבע את גודלו של הדרקון.

שם הקובץ: `change-to-smaller-size`



למעשה, מה שהיינו רוצים הוא שהערך שישולב בהוראת שינוי הגודל יהיה ערך שיוכנס על ידי הצופים. אבל אין לנו אפשרות לתת לצופים לשנות באופן ישיר הוראה בתסריט של Scratch. נצטרך לעשות זאת בדרך עקיפה, שנתארה באופן ציורי כך: הצופים יקבעו ערך, וערך זה יישמר בקופסה. בהוראת קביעת הגודל משולב תמיד ערך. הפעם ייקבע ערך זה בעזרת הצצה לאותה קופסה. כלומר, הערך שנמצא בה יהיה הערך המשולב בהוראת קביעת הגודל, וכך שינוי הגודל יתבצע בהתאם לערך שבחרו הצופים.

כדי ליישם זאת, עלינו ללמוד כמה דברים: כיצד ליצור קופסה שכזאת, כיצד לשלב תוכן של קופסה בתוך הוראה, וכיצד לאפשר לצופים להשפיע על תוכנה של קופסה.

לקופסאות כאלו קוראים **משתנים**. לכל קופסה יש שם, ואנו יכולים לשמור בה ערך, לשנותו (משום כך השם "משתנה") ולבדוק מה הערך ששמור בה (להציץ לתוכה).



**שימו לב:** כל קופסה יכולה להכיל ערך אחד בלבד. הכנסת ערך לקופסה בהכרח גורמת למחיקת הערך הקודם שהיה בה.



### מושג חדש: משתנה

משתנה הוא שטח זיכרון. ניתן לראות בו מעין קופסה, שיכולה לשמור בתוכה ערך אחד בלבד. אפשר **להכניס** ערך לקופסה, **להציץ** לתוכה כדי לראות מה הערך השמור בה, **להשתמש** בערך ששמור בה או **לשנות** את ערכה (כלומר, להחליף את הערך ששמור בה בערך אחר).

## בניית קופסה – יצירת משתנה

ההוראות לעבודה עם משתנים כלולות בקבוצת המשתנים, הקבוצה האדומה-כתומה. בחרו בקבוצת המשתנים. היא נראית קטנה למדי: כלולים בה רק שני פריטים: **צור משתנה** ו- **צור רשימה**. אלה שונים בצורתם מהלבנים שראינו עד כה. ראשית, הם חסרי צבע. בנוסף, על פי צורתם לא ברור כיצד ניתן לשלבם בתוך תסריט. ואמנם, אלו אינם לבנים שמתאימות להוראות שנועדו לשילוב בתוך תסריטים. אלו הוראות יצירה שמשפיעות על מאגר הלבנים.

לחצו על הוראת היצירה **צור משתנה**. כתוצאה מהלחיצה נפתח חלון שבו אנו יכולים לתת שם למשתנה החדש, לקופסה החדשה. בחרו שם כרצונכם, למשל s. עוד אתם מתבקשים לסמן בחלון אם המשתנה החדש יהיה רק לשימושה של דמות הדרקון או גם לשימושן של דמויות אחרות. כרגע הדרקון הוא הדמות היחידה בפרויקט, אך כשנרחיב את המשחק נסיף לו גם דמויות חדשות ונרצה לאפשר גם להן גישה לקופסה. לכן נסמן כי המשתנה הוא לכל הדמויות, ונבחר OK.

שימו לב מה קרה כעת בחלון ההוראות, בקבוצה האדומה-כתומה. נוספו לבנים בצבע אדום-כתום המתייחסות למשתנה החדש. הראשונה מהן היא פשוט **לבנה המייצגת את המשתנה**, בדומה ללבנת הגודל שבה השתמשנו קודם. סימון הריבוע הלבן הקטן ליד הלבנה המייצגת את המשתנה יגרום להופעת תצוגה של ערך המשתנה על הבמה.

הדמיון בין לבנת הגודל ללבנה המייצגת את המשתנה החדש, והאופן שבו סימון או אי-סימון משפיע על התצוגות של הבמה אינו מקרי. למעשה, גם הגודל הוא מעין משתנה. גם הוא קופסה, אך קופסה מיוחדת שבה נשמר תמיד גודלה של הדמות. זהו משתנה שמוגדר אוטומטית על ידי Scratch, וההוראות המתייחסות אליו נמצאות בקבוצה הסגולה, קבוצת המראה, ולא בקבוצת המשתנים.

### פעולה חדשה ב־Scratch: יצירת משתנה

לחיצה על הוראת היצירה חסרת הצבע **צור משתנה** שבקבוצת המשתנים תגרום להופעת חלון שבו ניתן לקבוע שם של משתנה, וכן לקבוע אם הוא שייך רק לדמות זו או לכל הדמויות. לאחר יצירת המשתנה תופיע במאגר ההוראות של קבוצת המשתנים לבנה המסומנת בשם המשתנה, היא **הלבנה המייצגת את המשתנה**.

אם זהו המשתנה הראשון בפרויקט, בעקבות יצירתו תופיע במאגר ההוראות של קבוצת המשתנים גם רשימה של לבנים המתאימות להוראות שבעזרתן ניתן להשפיע על ערכו של המשתנה ועל הצגתו על הבמה.

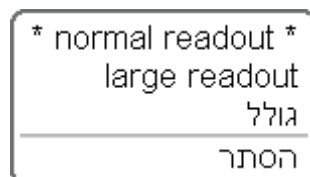
### פעולה חדשה ב־Scratch: הצגת ערכו של משתנה

ניתן להציג על הבמה את הערך של כל משתנה ב־Scratch (כולל משתנים המוגדרים מראש על ידי Scratch, כמו גודל דמות). יש ללחוץ על הריבוע הצמוד ללבנה המייצגת של המשתנה, במאגר ההוראות של קבוצת המשתנים. כתוצאה מכך יופיע סימן "וי" בתוך הריבוע, ועל הבמה תופיע חלונית עם ערכו של המשתנה.

## מילוי הקופסה – קליטת ערכים מהצופים לתוך משתנה

בעזרת הצגת הערך על הבמה הצופים יכולים לראות את תוכן הקופסה, אך כיצד יוכלו להשפיע על תוכנה? נשנה מעט את אופן תצוגת המשתנה על הבמה, כך שיינתן לצופים גם גולל (Slider), שבעזרתו יוכלו לשנות את הערך.

לשם כך הצביעו עם העכבר על התצוגה של ערך המשתנה החדש שעל הבמה ולחצו עם הכפתור הימני של העכבר. נפתחת רשימה אשר כוללת אפשרויות שונות לתצוגה. הפריט השלישי ברשימה הוא גולל.



בחרו בפריט. שימו לב כיצד השתנתה תצוגת המשתנה על הבמה: מתחתיה נוסף גולל, והזזתו בעזרת העכבר גורמת לשינוי הערך בתצוגה בהתאם לתזוזה.

**שימו לב:** כפי שכבר נאמר, גודלה של הקופסה מאפשר שמירת ערך אחד בלבד. בכל פעם שהצופים ישתמשו בגולל כדי לשנות את ערך המשתנה, **יחליף** הערך החדש את הערך הקודם, והערך הקודם יימחק.



## כלי חדש ב-Scratch: גולל

לאחר לחיצה עם הכפתור הימני של העכבר על תצוגה של משתנה בבמה, יופיע חלון. בעקבות בחירה בפריט "גולל", יופיע פס גלילה מתחת לתצוגת ערך המשתנה. בעזרת פס הגלילה ניתן לשנות את ערכו של המשתנה, על ידי גלילת הנקודה שעל הפס שמאלה או ימינה.

## הצצה לתוך הקופסה – שימוש בערכו של משתנה

כעת נרצה כי גודלו של הדרקון ייקבע על פי הערך שבתוך הקופסה, ערך המשתנה החדש כפי שנקבע על ידי הצופים בעזרת הגולל. ניתן לתאר את התהליך שבו אנו מעוניינים כך:

0. *אם אחיזה על הדגל היחיד:*

1. *אגמל גודלך לגודל מלא*

2. *האמן שלי שני*

3. *הקטן גודלך לפי הערך השמור במשתנה החדש*

כיצד נעשה זאת? התסריט המתאים צריך להיות דומה מאוד לתסריט האחרון שיצרנו בדוגמה 1, אך הוראת קביעת הגודל השנייה צריכה להיות שונה. בדוגמה 1 שולב בה הערך 50, וכעת צריך להיות משולב בה הערך השמור בתוך המשתנה.

עד כה, כאשר שילבנו ערך בתוך הוראה של Scratch, עשינו זאת בדרך כלל על ידי בחירת ערך מתוך רשימה (בהוראות שבהן ליד החלופית מופיע חץ קטן, כגון **פנה לכיוון**) או על ידי הקלדת ערך. אבל היו גם יוצאים מהכלל. היו מקרים שבהם בתוך חלופית של לבנה אחת שילבנו לבנה אחרת. למשל, כאשר שילבנו בהוראת ההמתנה המותנית **חנה עד ש...** לבנת תנאי **נוגע ?**. אם כך, הערך המשולב בחלופית של לבנה יכול להיות בעצמו לבנה, אך חשובה ההתאמה בצורת הלבנה המשולבת. למשל, החלופית בהוראת ההמתנה המותנית **חנה עד ש...** היא חלופית מלבנית עם פינות קטומות, וזו גם צורתה של לבנת תנאי כמו הלבנה **נוגע ?**.

מה צורתה של החלופית שבהוראת קביעת הגודל? זו חלופית המתאימה לערך מספרי. היא מלבנית ופינותיה מעט מעוגלות. ניתן לשלב בה כל לבנה שמתאימה לה בצורתה, כלומר, לבנת ערך שמתאימה אף היא לערך מספרי. אם נסתכל בחלון ההוראות בקבוצה האדומה-כתומה, נוכל לראות כי צורתה של לבנה שמייצגת משתנה, אף היא כזאת – מלבנית עם פינות מעוגלות. ובכן, גררו את הלבנה המייצגת את המשתנה החדש אל הוראת קביעת הגודל השנייה, ושלבנו אותה בחלופית במקום הערך המספרי 50 שמופיע בה.

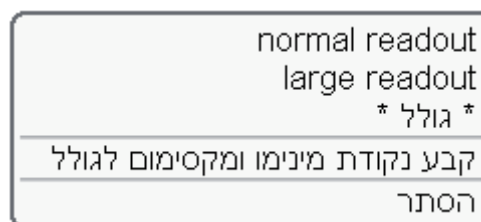
**שימו לב:** השימוש בערך המשתנה אינו משפיע על תוכנו. כל שביצענו הוא הצצה לתוך הקופסה, ואת הערך השמור בה שילבנו בהוראה. הקופסה ממשיכה לשמור את אותו ערך, עד אשר נכניס לתוכה ערך חדש. הדבר דומה לשימוש במספר טלפון הרשום על פתק. גם לאחר השימוש במספר הרשום, הוא עדיין נשאר על הפתק ואינו נמחק.



תעדו את השינוי שערכתם ושמרו את הפרויקט בשם חדש. נסו להפעילו על ידי לחיצה על הדגל הירוק. נסו זאת עוד כמה פעמים, כאשר אתם משנים בעזרת הגולל את הערך הרצוי לגודל.

## שינוי גבולותיו של גולל

בעזרת הגולל שהוספנו לתצוגת המשתנה שיצרנו הצופים יכולים להקטין את גודלו של הדרקון כרצונם, ואף להגדילו, אך אין הם יכולים להגדילו מעבר לגודלו המקורי. הסיבה היא שנקודת המקסימום של הגולל שהוספנו היא 100, לכן אפשר לקבוע את גודלו של הדרקון ל-100% לכל היותר מגודלו המקורי. אבל אפשר בקלות לשנות את גבולותיו של הגולל ובכך לאפשר גם להגדיל את הדרקון מעבר לגודלו המקורי. הצביעו עם העכבר על תצוגת המשתנה שיצרנו על הבמה, ולחצו על הכפתור הימני של העכבר. נפתחת רשימה אשר הפריט הרביעי בה הוא קבע נקודת מינימום ומקסימום לגולל.



בחרו בפריט זה ובחלון שנפתח הקלידו ערך מקסימום חדש, למשל 200. כעת ניתן לשנות את ערכו של המשתנה עד 200. נסו לשנות את ערכו למקסימום וראו מה קורה לאחר לחיצה על הדגל הירוק. תעדו את השינוי שערכתם ושמרו את הפרויקט בשם חדש.

שם הקובץ: [change-to-new-size](#)

### פעולה חדשה ב-Scratch: שינוי גבולות גולל

לאחר לחיצה עם הכפתור הימני של העכבר על תצוגה של משתנה עם גולל בבמה, יופיע חלון. ניתן לבחור בפריט "קבע נקודת מינימום ומקסימום לגולל" ולקבוע את הגבולות עבור הגולל.

בדוגמה זו אמנם נתנו לצופים אפשרות לשנות את גודלו של הדרקון, אך הם יכולים לבצע זאת רק על ידי שינוי ערכו של המשתנה שיצרנו בעזרת הגולל ולחיצה על הדגל הירוק. עבור כל שינוי נדרשת לחיצה נוספת על הדגל הירוק. כעת נרצה לשפר את המשחק שבנינו.

## משימה 3: שינוי חוזר של גודל הדרקון

הצופים ילחצו על הדגל הירוק פעם אחת בלבד, עם תחילת המשחק. לאחר מכן יוכלו הצופים לשנות את ערך המשתנה בעזרת הגולל כמה פעמים שירצו, וכל שינוי באמצעות הגולל ישפיע מידית על גודל הדרקון.



שם הקובץ: [change-size-repeatedly](#)

**תרגיל 1 - הדרקון שמשנתה ומשתנה ומשתנה:** תכננו את השינוי המבוקש, שנו את הפרויקט בהתאם

וודאו כי הוא אכן מבצע את הנדרש. תעדו ושמרו בשם חדש.

**הדרכה:** יש להשתמש בהוראת בקרה מתאימה.

### דוגמה 3: הוספת כפתורי משחק

כאשר אנו מאפשרים לצופים לשנות את גודלה של דמות שוב ושוב, סביר שבמהלך המשחק, מסיבה זו או אחרת, הם יהיו מעוניינים בהחזרת הדברים לקדמותם. אמנם, תמיד פתוחה בפניהם האפשרות להשתמש בגולל כדי להחזיר את הדמות לגודלה המקורי, אבל יהיה להם נוח יותר אם ניצור כפתור שיוקדש במיוחד למטרה זו.

### משימה 4: הוספת כפתור reset

יש להוסיף למשחק כפתור reset. לחיצה על הכפתור תגרום לביטול כל השינויים ולהחזרת הדמות לגודלה המקורי. כלומר, השפעתה של לחיצה על הכפתור תהיה כמו ההשפעה שמתקבלת כשהצופים משנים בעזרת הגולל את ערך המשתנה ל-100.



שם הקובץ: change-size-with-reset

למרות שכפתור הוא יצור דומם, כדי שיוכל לבצע את תפקידו, גם הוא צריך להיות מוגדר כדמות. למעשה, כבר ראינו בפרק 4 חפץ דומם, כדורגל, שמוגדר כדמות. שם הגדרנו את הכדור כדמות כדי שיוכל לזוז בעקבות בעיטה. במקרה זה הכפתור אינו צריך לנוע, אך בכל זאת מוטלת עליו מטלה, מטלה שתתורגם לתסריט של Scratch. את המטלה שמוטלת על כפתור ה-reset ניתן לתאר באופן הבא:

0. כאשר לוחצים עליך:

1. שנה את ערך המשתנה s ל-100.



כתוצאה משינוי ערך המשתנה על-ידי הכפתור, ישנה הדרקון את גודלו לגודלו המקורי בהתאם לתסריטו. דמות מתאימה לכפתור כבר מוכנה בתיקיית הפרויקטים של הספר. לחצו על הכפתור האמצעי מעל לחלון הדמויות כדי לייבא דמות חדשה מקובץ. נווטו לתיקיה עם הפרויקטים של פרק 6 ובחרו בדמות של כפתור "reset". מקמו אותה במקום מתאים על הבמה על ידי הזזתה בעזרת העכבר. כעת, כאשר יש לנו כפתור, נוכל לכתוב עבורו תסריט המתאים לתיאור שניתן קודם.


עבור צעד 0 יש לנו הוראה מתאימה בקבוצת הבקרה, הוראה שמתנה תחילת ביצוע תסריט של דמות בלחיצה על הדמות. זו ההוראה השלישית במאגר הלבנים בקבוצה הכתומה **נאשר לוחצים על דמות** (בלבנה יופיע תמיד שם הדמות אשר חלון התסריטים שלה הוא הפעיל). כמו שאנו כבר מכירים מלבנים אחרות של הוראות אשר מתנות תחילת ביצוע תסריט, גם צורתה של לבנה זו אינה מאפשרת לשלב לבנים אחרות מעליה, ולכן הוראה זו היא תמיד הוראה ראשונה בתסריט.

#### הוראה חדשה ב-Scratch: התניית תחילת ביצוע תסריט בלחיצה על דמות

ההוראה **נאשר לוחצים על דמות** יכולה להופיע רק בתחילת תסריט של דמות וכוללת את שמה של הדמות. לחיצה עם העכבר על תמונתה של הדמות על הבמה גורמת לתחילת ביצועו של התסריט.

## שינוי תוכן הקופסה – שינוי ערכו של משתנה

משמעותו של צעד 1 בתיאור שנתנו הוא שינוי ערכו של משתנה. בקבוצת המשתנים, הקבוצה האדומה כתומה, יש שתי הוראות המתאימות לשינוי ערכו של משתנה. ההוראה הראשונה  היא הוראה מוחלטת. ההוראה השנייה  היא הוראה יחסית.


ההוראה המתאימה לצעד 1 היא ההוראה המוחלטת הראשונה . בהוראה זו צריך לשלב שני ערכים. הערך הראשון הוא המשתנה שאת ערכו מעוניינים לקבוע. לחיצה על החץ הקטן שליד החלונית הראשונה תגרום להופעת רשימה ובה כל המשתנים שיצרנו עד כה. כעת, משום שיצרנו רק משתנה אחד, מופיע ברשימה זאת רק המשתנה s, וערכו כבר משולב כברירת מחדל בתוך הוראת קביעת הערך. הערך השני שיש לשלב בהוראת קביעת הערך הוא הערך שאליו רוצים לשנות את ערך המשתנה. במקרה זה נשלב את הערך 100.


התסריט המתקבל הוא התסריט הקצר הבא:



תעדו את השינוי שערכתם ושמרו את הפרויקט בשם חדש.

### הוראות חדשות ב־Scratch: קביעת ערך משתנה ושינוי ערך משתנה

ההוראה  היא הוראה מוחלטת שגורמת לקביעת ערכו של המשתנה ששמו מופיע בחלונית הראשונה לערך המופיע בחלונית השנייה.

ההוראה  היא הוראה יחסית שגורמת לשינוי ערכו של המשתנה ששמו מופיע בחלונית הראשונה בערך שמופיע בחלונית השנייה.

## העתקה ממשתנה למשתנה

אחרי שהוספנו למשחק כפתור אחד, כפתור reset המאפשר לבטל את כל השינויים ולחזור לגודל המקורי, נרצה להוסיף כפתורים נוספים. ייתכן שהצופים שינו את גודל הדמות לערך מסוים, וכעת הם מעוניינים לבדוק שינויים נוספים, אבל כך שיוכלו לחזור בקלות לאותו ערך מסוים, אם ירצו.

## משימה 5: שמירת ערך ושחזור

יש להוסיף שני כפתורים: "שמור" ו-"שחזר". לחיצה על הכפתור "שמור" תגרום לשמירת הגודל הנוכחי של הדרקון. לחיצה על הכפתור "שחזר" תשנה את גודל הדרקון לגודל שנשמר בפעם האחרונה שבה נלחץ הכפתור "שמור".



שם הקובץ: change-size-with-save-and-restore

גם עבור שני הכפתורים החדשים, "שמור" ו-"שחזר", יש כבר דמויות מוכנות. יבאו אותן אל הפרויקט, באותו אופן שבו פעלתם לגבי הכפתור "reset".

כעת עלינו להכין תסריטים מתאימים לכל אחד מהכפתורים החדשים. כאמור, התסריט המתאים לכפתור השמירה צריך לדאוג לשמירת ערך נוכחי בצד, ואילו התסריט המתאים לכפתור השחזור צריך להשתמש בערך השמור. כיצד נבצע את שמירת הערך והשימוש בו? נוכל לשמור גם את הערך הזה בקופסה, קופסה משלו. כלומר, עלינו ליצור משתנה נוסף, שאליו יתייחסו התסריטים החדשים.

צרו משתנה חדש, למשל בשם save. כתוצאה מפעולת היצירה מופיעה על הבמה תצוגה עבור המשתנה החדש.

האם תצוגת המשתנה save צריכה לכלול גולל?



לא. ערכו לא אמור להשתנות ישירות על ידי הצופים. ערכו ישתנה רק בעקבות לחיצה על כפתור השמירה.


על פי התיאור המילולי שנתנו קודם, הדמות המתאימה לכפתור השמירה צריכה לפעול על פי התיאור הבא:

0. כאשר לוחצים על:

1. שמור אל save-קופסה.

כבר ידוע לנו כיצד לכתוב ב־Scratch הוראה המתאימה לצעד 0. מה לגבי צעד 1?

בצעד 1 עלינו להכניס ערך לקופסה שבה שמור ערכו של משתנה השמירה, אך הפעם ערך זה צריך להתקבל על ידי הצצה לתוך קופסה אחרת. כבר השתמשנו בהוראה ששינתה את ערכו של משתנה, וכבר הייתה לנו הוראה שהשתמשה בערכו של משתנה. ההוראה שעלינו לכתוב כעת היא הוראה שמשתמשת בערכו של משתנה אחד כדי לשנות את ערכו של משתנה אחר.

נוכל גם הפעם להשתמש בהוראה המוחלטת  מקבוצת המשתנים האדומה-כתומה, אך הפעם בשתי החלוניות נשלב משתנה ולא ערך מספרי. בחלונית הראשונה נשלב את המשתנה שאת ערכו אנו רוצים לשנות, המשתנה save, ובחלונית השנייה נשלב את המשתנה שבערכו אנו רוצים להשתמש, המשתנה s.

צרו את התסריט המתאים לכפתור השמירה, ותעדו אותו.

כפתור השחזור צריך לפעול לפי התיאור הבא:

0. כאשר לוחצים על:

1. העקב אל save-קופסה.

**תרגיל 2:** כתבו תסריט המתאים לתיאור שניתן עבור כפתור השחזור.

נסו להפעיל את הפרויקט על ידי לחיצה על הדגל הירוק ועל ידי שימוש בכפתורים השונים ובגולל. שמרו את הפרויקט בשם חדש, אחרי שווידאתם כי כל התסריטים מתועדים.

### תרגיל 3:

בתרגיל זה נשנה את האנימציה שנבנתה בפרק 5, אנימצית הריקוד האינסופי של שני הרקדנים (משימה 3, הקובץ `dance-indefinitely`).

א. הוסיפו משתנה בשם `steps`, לשימושם של שני הרקדנים. זה יהיה משתנה עם תצוגת גולל, שערכו יוכל להיות בין 1 ל-10, ויוכל להשתנות על ידי הצופים. כעת שנו את כל הביצועים החוזרים בתסריטיהם של שני הרקדנים, כך שאלו יבוצעו מספר פעמים השווה לערכו של המשתנה `steps`.

שם הקובץ: `dance-steps1`

ב. החליפו את המשתנה החדש, `steps`, בשני משתנים – `Cassy-steps` ו-`Jay-steps`. כעת יוכלו הצופים לשלוט לחוד על מספר הביצועים החוזרים של כל אחד מהרקדנים.

שם הקובץ: `dance-steps2`

### תרגיל 4:

בתרגיל זה יש ליצור כמה אנימציות שבהן משתתף טיל.

א. הכינו אנימציה שבה עם הלחיצה על הדגל הירוק, הטיל מתמקם בפינה השמאלית התחתונה של הבמה ומתחיל לנוע אל הפינה הימנית העליונה (ייתכן שתצטרכו לנסות ערכים שונים עד שתמצאו את הערכים המתאימים ביותר עבור המיקום והכיוון ההתחלתיים של הטיל). התנועה תישלט על ידי ביצוע חוזר, וקצב ההתקדמות בלולאה, כלומר, מספר הצעדים שיתבצעו בכל סיבוב בלולאה, ייקבע על ידי משתנה מהירות, `speed`, שערכו ייקבע ל-1. דאגו להצגת ערכו של משתנה המהירות על הבמה.

שם הקובץ: `rocket1`

ב. הרחיבו את האנימציה שיצרתם בסעיף א' כך שהצופים יוכלו לשלוט על "מהירות" הטיל, והערכים האפשריים עבור המשתנה `speed` ינועו בין 10 ל-10. מה קורה כשערך המשתנה הוא שלילי? הסבירו.

שם הקובץ: `rocket2`

ג. למעשה, האנימציה שיצרתם בסעיף ב' אינה הגיונית. טילים אינם מאטים ומשנים כיוונם. מצד שני, מהירותם גדלה ככל שעולה כמות הדלק שנשרף. שנו את האנימציה שיצרתם בסעיף א' כך שלמשתנה המהירות לא תהיה תצוגת גולל, אבל בכל סיבוב של הביצוע החוזר יגדל ב-1 ערכו של המשתנה `speed`.

שם הקובץ: `rocket3`

ד. ובעצם, לא רק שמהירותם של טילים גדלה עם תנועתם, אלא קצב גידול המהירות גדל אף הוא! הרחיבו את האנימציה שיצרתם בסעיף ג' והוסיפו לה משתנה חדש, `changeBy`, שקובע את מידת השינוי בערכו של המשתנה `speed`. בתחילה יהיה ערכו של המשתנה החדש 1, אך עם כל סיבוב של הביצוע החוזר יגדל ערכו ב-1.

שם הקובץ: `rocket4`

ה. נראה כי הטיל נע מהר מאוד. מהר כל כך עד שקשה למדוד כמה זמן אורכת דרכו מפינה לפינה. מטרת סעיף זה היא למדוד את זמן הטיסה מפינה לפינה עבור סעיף ג' וסעיף ד'. לשם כך שנו את האנימציות שיצרתם בשני הסעיפים, כך שבכל אחת מהן יבצע הטיל את הדרך מהפינה התחתונה השמאלית לפינה העליונה הימנית מספר גדול של פעמים, למשל, 20. הפעילו את האנימציות החדשות ומדדו את זמן הביצוע שלהן. כעת חלקו את הזמנים שנמדדו במספר הביצועים (20), כדי לקבל את הזמן שאורכת הדרך מהפינה התחתונה השמאלית לפינה העליונה הימנית, עבור כל אחד מהסעיפים.

שמות הקבצים: `rocket3a`, `rocket4a`



## סיכום

בפרק זה ערכנו היכרות עם מגננון שמאפשר לנו לזכור ערכים, לשנותם ולהשתמש בהם בעזרת **משתנים**. משתנה הוא מעין קופסה, שיכולה לשמור בתוכה ערך אחד בלבד. נוכל להכניס ערך לקופסה, להציץ לתוכה ולהשתמש בערך ששמור בה, או לשנות את ערכה (כלומר, להחליף את הערך ששמור בה בערך אחר).

**יצירת משתנים** מתבצעת ב־Scratch על ידי הוראת יצירה שנמצאת בקבוצת המשתנים האדומה־כתומה. זאת אינה הוראה שנועדה לשילוב בתוך תסריט, אלא הוראה שנועדה להיות מופעלת בתוך Scratch על ידי בחירה בה בעזרת העכבר. כאשר מפעילים את ההוראה, נוספות לקבוצת המשתנים לבנים המתאימות להוראות שנועדו לשילוב בתסריטים ומאפשרות לשנות ערכו של משתנה ולהשתמש בערכו. כמו כן נוספת **לבנה שמייצגת את המשתנה** החדש עצמו.

**שינוי ערכו של משתנה** נעשה בעזרת ההוראה המוחלטת או בעזרת ההוראה היחסית .

**שימוש בערכו של משתנה** נעשה על ידי שילוב הלבנה המייצגת אותו בתוך הוראות אחרות. בהוראה המוחלטת לקביעת ערך משתנה אפשר לשלב גם משתנה אחר ובכך לבצע **העתקה ממשתנה למשתנה**.

הלבנה המייצגת משתנה היא סוג של לבנת ערך. כאמור, היא מאפשרת שימוש בערכו של משתנה על ידי שילובה בהוראות אחרות. בנוסף היא משמשת גם ליצירת **תצוגה של משתנה**. כאשר החלונית הקטנה ליד הלבנה המייצגת משתנה מסומנת, ערכו של המשתנה מוצג על הבמה.

Scratch מאפשרת לצרף לתצוגה של משתנה **גולל**, ובכך לאפשר לצופים לשנות את ערכו על ידי הזזת הגולל שמוצג על הבמה. הגדרה כזאת מתבצעת על ידי הצבעה עם העכבר על תצוגת המשתנה, לחיצה על הכפתור הימני של העכבר ובחירה בפריט "גולל". לאחר שנוסף גולל לתצוגה של משתנה, ניתן להגדיר לגולל גבול מקסימלי וגבול מינימלי כרצוננו. **הגדרת גבולות לגולל** מתבצעת אף היא בעזרת העכבר על ידי הצבעה עם העכבר על תצוגת המשתנה, לחיצה על הכפתור הימני של העכבר ובחירה בפריט של קביעת נקודת מינימום ומקסימום לגולל.

הכרנו משתנה מיוחד שמוגדר באופן אוטומטי על ידי Scratch עבור כל דמות ושומר את **גודל הדמות**. הוראות המתייחסות למשתנה הגודל נמצאות בקבוצת המראה הסגולה. ערכו של משתנה הגודל של דמות משתנה באופן אוטומטי עם כל שינוי בגודל הדמות, אך בקבוצת המראה ישנן הוראות המאפשרות לשנות את גודלו באופן ישיר: הוראה מוחלטת , והוראה יחסית . גם למשתנה הגודל מוגדרת לבנה שמייצגת אותו. היא מאפשרת להציג את ערכו על הבמה ולשלב אותו בהוראות אחרות. לא ניתן להוסיף גולל לתצוגה של משתנה גודל.

## פרק 7

### זה תלוי – ביצוע מותנה

הכרנו עד כה כמה מנגנוני בקרה, שמאפשרים לשלוט על ביצוע הוראות בתסריט: ביצוע חוזר (אינסופי או סופי, מותנה או מוגבל מראש); ביצוע הוראות תסריט כתגובה למאורע (לחיצה על הדגל הירוק, קבלת מסר, לחיצה על מקש או לחיצה על דמות); עיכוב ביצוע הוראות, כלומר, המתנה (למשך זמן קבוע מראש או עד קיום מאורע). בפרק זה נכיר מנגנון בקרה נוסף השולט על ביצוע הוראות בתסריט. מנגנון זה מתנה ביצוע הוראות בקיום מאורע כלשהו.

### דוגמה 1: משחק בסיסי של הליכה במבוך

גם בפרק זה ניצור משחק מחשב, מתקדם מעט יותר מזה שיצרנו בפרק הקודם. הוא דומה לאחד ממשחקי המחשב הוותיקים: במשחק זה יידרשו השחקנים להשתמש במקשים בזריזות כדי לאפשר לדמותו של פקמן (Pac-Man) להסתובב במבוך בלי להיתקל בקירותיו. נבנה את המשחק בשלבים.

### משימה 1: הכנת הבסיס למשחק

יש לבנות את הבסיס לאנימציה: פקמן, המסוגל לפתוח ולסגור את פיו לסירוגין, והמבוך שבו ינוע פקמן.



שם הקובץ: `pacman-opens-and-closes-mouth`

כיצד ניצור את המבוך? נוכל ליצור אותו על יד שינוי הרקע של הבמה, ונוכל גם ליצור דמות שתהיה המבוך. שתי האפשרויות טובות, אך אנו נבחר באפשרות השנייה.

בוודאי נידרש במשחק לדמות עבור השחקן, פקמן. הוא זה שינוע במבוך וישנה את כיוונו בהתאם להוראות שיעבירו לו הצופים באמצעות לחיצה על המקשים, בעודו פותח וסוגר את פיו לסירוגין.

לכן בשלב הראשון יהיו במשחק שתי דמויות. פתחו את הפרויקט costumes בתיקייה המתאימה לפרק 7, ושם תמצאו כי שתי הדמויות המתאימות כבר מוגדרות.

אחרי שהדמויות מוכנות, נמשיך בהכנות למשחק. לדמות המבוך לא ניצור בינתיים שום תסריט. אכן, לא נראה כרגע כי על המבוך לפעול לפי תסריט כלשהו, אך בהמשך נראה כי גם לו יהיו תפקידים בניהול המשחק, ולכן גם תסריטים מתאימים.

כאמור, פקמן צריך לנוע במבוך כאשר פיו נפתח ונסגר חליפות (כדי שבהמשך יוכל "לאכול" דברים הנקרים בדרכו). למעשה, ניתן לראות זאת כאילו תפקידו של פקמן במשחק מורכב משתי מטלות: תנועה במבוך, בהתאם להוראות מהצופים, ופתיחה וסגירה של פיו חליפות. על פי הדרישות במשימה 1, אנו מתמקדים כעת במטלה השנייה, ובמטלה הראשונה נטפל בהמשך.

את פתיחת הפה וסגירתו נוכל ליצור בעזרת החלפת תלבושות. שימו לב כי לפקמן מוגדרות כבר שתי תלבושות (כפי שניתן לראות על ידי מעבר ללשונית התלבושות): באחת פיו פתוח, ובשנייה פיו סגור. לכן תוך כדי התנועה במבוך עליו להחליף כל הזמן בין שתי התלבושות שלו. נכתוב תסריט פשוט, אשר

יטפל גם באתחול המיקום והכיוון של הדמות וגם בהחלפה בין התלבושות<sup>1</sup>. בקווים כלליים ניתן לתאר את סדרת הפעולות הרצויה כך:

0. אם אחיזה על הדגל הידוק;

1. ה3 אגמח (של התיקום, הכיוון והגלבוש)

2. ה3 שוב ושוב החלפה גלבוש

**תרגיל 1:** הפכו את התיאור הכללי שניתן כאן לתיאור מפורט, אשר בו כל צעד מתאים להוראה אחת של Scratch. את האתחול (צעד 1 בתיאור הכללי) בצעו כך שבתחילת הביצוע פקמן יעמוד בנקודה (115, -115), כשהוא פונה ימינה ופיו פתוח. את החלפת התלבושת (צעד 2 בתיאור הכללי) בצעו על פי הנלמד בהרחבת הרשות הראשונה לפרק 5. שימו לב, כדי שהחלפת התלבושת תיראה בבירור במהלך האנימציה, יש לשלב המתנה קצרה (למשל, 0.2 שניות) לפני כל הוראת החלפה.

בנו את התסריט הראשון של פקמן על פי התיאור המפורט שתכנתם. ודאו את נכונותו, תעדו ושמרו אותו בשם חדש.

## משימה 2: תנועתו של פקמן – הבסיס

יש להרחיב את האנימציה כך שפקמן ינוע במבוך, ממיקומו ההתחלתי, ללא הפסקה.



שם הקובץ: `pacman-moves`

ובכן, לאחר שטיפלנו במטלה של החלפת התלבושות, אנו פונים למטלה האחרת – מטלת התנועה במבוך. למטלה הזאת נכין תסריט משלה.

מאחר שמטרת המשחק היא להניע את פקמן במבוך בלי שייתקע בקירות, הרי למעשה פקמן אמור לנוע כל הזמן, והצופים אמורים לכוון אותו בעזרת המקשים ובכך למנוע ממנו להתנגש בקירות. כאמור, בשלב זה עוד לא נטפל בקבלת ההוראות מהצופים, או בזיהוי התנגשות בקיר, ונתמקד בפעולה הבסיסית בתנועתו של פקמן, ובשאלה כיצד היא משתלבת בהחלפת התלבושות המתמדת.

פעולת התנועה הבסיסית מורכבת מביצוע חוזר של תזוזות קטנות, זו אחרי זו. הביצוע החוזר יהיה אינסופי, משום שבשלב זה עדיין אין אנו מטפלים בזיהוי התנגשות בקיר.

אם כן, נראה כי בשלב זה, כאשר עדיין איננו מתייחסים לזיהוי התנגשות בקיר ולהוראות מהצופים, התסריט הוא פשוט מאוד, ועליו לענות לתיאור הבא:

0. אם אחיזה על הדגל הידוק;

1. אלולאים

1.1. 2 515 32

1 בהרחבת הרשות הראשונה בפרק 5 עסקנו בהחלפת תלבושות. אם לא למדתם את הנושא הזה במהלך לימוד פרק 5, למדו אותו עכשיו, לפני המשך ההתקדמות בפרק 7.

התסריט שנכתב על פי התיאור הזה יעבוד במקביל לתסריט שכבר כתבתם, אשר אחראי על החלפת התלבושות.

## שכפול האתחול

מסיבה הנעוצה בעבודה המקבילית של שני התסריטים, ושאותה לא נפרט כעת, נוסיף גם לתסריט זה שלב אתחול, שיטפל באתחול המקום, הכיוון והתלבושת, בדיוק כמו שלב האתחול בתסריט השני. בפרק 10 נתעמק בנושא המקביליות ואז גם ניתן הסבר מלא להוספת שלב האתחול. לכן התיאור של התסריט בקווים כלליים יהיה זה:

0. אם אחיזה על הדגל היחיד:

1. אם אגחול (ש) החיקום, הכיוון והתלבושת

2. אלואמים

2.1 2 sis 2 23 דיס

נסו לכתוב תסריט מתאים לתיאור הזה ולהפעיל במקביל את שני התסריטים שבניתם (התסריט החדש והתסריט המטפל בהחלפת התלבושות) על ידי לחיצה על הדגל הירוק.

## משימה 3: זיהוי התנגשות

יש להרחיב את האנימציה כך שפקמן יעצור ויצעק מכאב כאשר הוא מתנגש בקיר.

שם הקובץ: `pacman-moves-and-hits-wall`



ובכן, עלינו להרחיב את המשחק כך שהתנגשות בקיר תעצור את פקמן. בתסריט שכתבנו קודם לטיפול בתנועתו של פקמן משולבת הוראת תזוזה. כעת נרצה שהוראה זו תתבצע רק אם לא הגיע פקמן לקיר. אם פקמן הגיע לקיר, לא נרצה שתתבצע תזוזה, אלא פעולה אחרת, של אמירת "איי!". כלומר, בתיאור הקודם נחליף את צעד 2.1 בצעדים הבאים:

2.1 אם הגעלם בקיר

2.1.1 אמור "איי!"

2.2 אמור

2.2.1 2 sis 2 23 דיס

בתיאור הזה נעשה שימוש במנגנון בקרה, כלומר, במנגנון השולט על ביצוע של הוראות. במקרה זה יש שתי הוראות שביצוען נשלט על ידי מנגנון הבקרה: הוראת הדיבור והוראת התזוזה. ראשית, נבדק קיומו של מאורע (התנגשות בקיר). אם המאורע מתקיים, תתבצע הוראת הדיבור; ואחרת, אם הוא אינו מתקיים, תתבצע הוראת התזוזה. לפיכך מנגנון הבקרה הוא מנגנון של **ביצוע מותנה**. הוא מתנה את ביצוע הוראת הדיבור בהתרחשות מאורע התנגשות ואת ביצוע הוראת התזוזה בהתרחשות מאורע של חוסר התנגשות (משמע, באי־התרחשות של מאורע ההתנגשות). בכל פעם שמגיעים לצעד 2.1, מובטח כי בעקבות בדיקת המאורע, יבוצע או צעד 2.1.1 או צעד 2.2.1, ולא שניהם.

מאחר שזהו מנגנון בקרה, מתאים לחפש עבורו הוראה מתאימה בקבוצת הבקרה הכתומה. ואכן,



בקבוצה זו נמצא לבנה המתאימה להוראה . נבחן את צורתה של הלבנה: בדומה לכמה הוראות בקרה אחרות שהכרנו (ביצוע חוזר מותנה, ביצוע אינסופי משולב בבדיקה, המתנה מותנית) גם בלבנה זו יש חלונית המתאימה לשילוב לבנת תנאי. התנאי מתאים למאורע שבהתרחשותו מותנה הביצוע. במקרה שלנו, זהו מאורע התנגשות בקיר.

שלא כמו לבנים המתאימות להוראות בקרה אחרות שהכרנו, ללבנה הזאת יש שני "פיות". בתוך פה אחד ישולבו הלבנים המתאימות להוראות שצריכות להתבצע אם המאורע מתרחש (במקרה שלנו, הוראת דיבור, המתאימה לצעד 2.1.1). בתוך הפה השני ישולבו הלבנים המתאימות להוראות שצריכות להתבצע אם התנאי אינו מתרחש (הוראת תזוזה, המתאימה לצעד 2.2.1).

גררו את הלבנה המתאימה אל חלון התסריטים ושולבו בתוכה את ההוראות המיועדות לביצוע בתוך ה"פיות" המתאימים. שלבו את הוראת הביצוע המותנה שהתקבלה, עם ההוראות שבתוכה, בתסריט האחראי על תנועתו של פקמן, במקום הוראת התזוזה שנמצאת בו כעת.

#### מושג חדש: ביצוע מותנה

ביצוע מותנה כולל תנאי ושתי סדרות סופיות של הוראות. אם התנאי מתקיים – מבוצעת סדרת ההוראות הראשונה, ואם אין הוא מתקיים – מתבצעת סדרת ההוראות השנייה.

#### הוראה חדשה ב־Scratch: ביצוע מותנה



בביצוע ההוראה נבדק התנאי המשולב בהוראה. אם הוא מתקיים, מתבצעות ההוראות המשולבות בתוך חלק ה"אם". אם התנאי אינו מתקיים, מתבצעות ההוראות המשולבות בתוך החלק "אם לא".

### תנאי ההתנגשות בקיר

עוד עלינו להשלים את התנאי המשולב בהוראה, המתאים למאורע התנגשות בקיר. כיצד נזהה התנגשות בקיר? החיישנים של Scratch יכולים לעזור לנו בכך. הדרך בתוך המבוך היא בצבע אחד, סגול, ואילו חלקי המבוך האחרים הם בצבע אחר, כחול. לכן בעצם מאורע התנגשות בקיר הוא גם מאורע של נגיעה בצבע הכחול. כבר הכרנו ב־Scratch תנאי נגיעה והשתמשנו בו בכמה וכמה דוגמאות. אבל תנאי הנגיעה נועד לבדוק נגיעה בעצמים (מסגרת הבמה, מצביע העכבר או דמויות אחרות).

יש תנאי אחר המתאים לצרכינו, **תנאי נגיעה בצבע** מסוים. בקבוצת החיישנים הכחולה-היריה נוכל למצוא לבנה שמתאימה לו, הלבנה **נוגע בצבע** ?

ללבנה זו יש חלונית רבועה עבור הצבע שאותו אנו מעוניינים לשלב בתנאי. כיצד נשלב צבע בחלונית? הצביעו עם העכבר על חלונית הצבע שבלבנה ולחצו על הכפתור השמאלי של העכבר. סימון מצביע העכבר שינה צורתו, והוא נראה כעת כמו טפטפת צבע. הזיזו את העכבר עד שהטפטפת תגיע לשטח שצבעו כצבע שאתם רוצים לשלב בהוראה. במקרה שלנו נזיז את העכבר עד שהטפטפת תגיע לאזור כחול במבוך. כעת לחצו שוב על הכפתור השמאלי של העכבר. החלונית שבתוך הלבנה שינתה צבעה לצבע שנבחר.

שלבנו את תנאי הנגיעה בצבע בתוך הוראת הביצוע המותנה.

### תנאי חדש ב־Scratch: נגיעה בצבע

התנאי **נוגע בצבע** ? מתקיים אם הדמות נוגעת בצבע המשולב בחלונית, ואינו מתקיים אם הדמות אינה נוגעת בצבע זה.

נסו להפעיל את שני התסריטים במקביל על ידי לחיצה על הדגל הירוק. תעדו את המשחק שהתקבל ושמרו אותו בשם חדש.

## משימה 4: הצופים שולטים על התנועות – שימוש במשתנה

נרחיב את המשחק כך שהצופים יוכלו לשלוט על כיוון התנועה של פקמן באמצעות מקשי החצים.



שם הקובץ: `user-control-of-pacman`


אם כך, השלב הבא הוא לאפשר לצופים לשלוט על תנועותיו של פקמן ולכוונו בתוך המבוך. כך יוכלו הצופים (אם יהיו זריזים מספיק) למנוע מפקמן להתנגש בקירות המבוך. הנחיות הצופים יועברו בעזרת המקלדת: אם הם לוחצים על מקש החץ מעלה, כוונתם היא לסובב את פקמן לכיוון מעלה; חץ מטה יסמן סיבוב לכיוון מטה. לחיצה על מקש החץ הפונה ימינה תסמן הנחיית סיבוב לכיוון ימין, ולחיצה על מקש החץ הפונה שמאלה תסמן הנחיית סיבוב לכיוון שמאל.

### האופן שבו לחיצה על מקש חץ שולטת על הביצוע


כיצד תתבצע התקשורת בין הצופים לבין פקמן? אנו רוצים כי בכל פעם שנלחץ מקש, ישנה פקמן את כיוונו בהתאם. בפרק 5 יצרנו אנימציה שבה הצופים השפיעו על תנועות הדמויות בעזרת לחיצה על מקשים. שם השתמשנו בהוראת בקרה מותנית שהייתה תלויה בתנאי לחיצה על מקש. הוראת הבקרה שהשתמשנו בה בפרק 5 הייתה ביצוע חוזר מותנה. האם נוכל גם עכשיו לשלב בתסריט התנועה של פקמן הוראת בקרה מותנית שתהיה תלויה בלחיצה על מקש ובכך לפתור את הבעיה?

התשובה היא חיובית. ניתן להשתמש בגישה זו, אך אנו נבחר לא לעשות זאת כעת, משתי סיבות: ראשית, השימוש בהוראת הבקרה המותנית יהיה מורכב למדי, ושנית, התסריט המתקבל יהיה ארוך

מאוד וקשה למעקב. באחד התרגילים שבהמשך הפרק תתבקשו לשנות את הפרויקט כך שיעבוד לפי גישה זו. בשלב ההוא כבר תכירו מספיק את הוראת הביצוע המותנה כדי להשתמש בה גם באופן המורכב הנדרש לשם כך. אבל בשלב זה נבחר בפתרון אחר.

בפתרון שלנו נשתמש בהוראת בקרה חדשה, שמתנה תחילת ביצוע תסריט בלחיצה על מקש. זוהי ההוראה  , אשר הלבנה המתאימה לה נמצאת בקבוצת הבקרה הכתומה. כמו הוראות אחרות שהכרנו, הוראה זו חייבת להיות ההוראה הראשונה בתסריט.

### הוראה חדשה ב-Scratch: התניית תחילת ביצוע תסריט בלחיצה על מקש

ההוראה  יכולה להופיע רק בתחילת תסריט. לחיצה על המקש המופיע בחלונית גורמת לתחילת ביצועו של התסריט.

אם כך, אפשרות אחת היא להכין לפקמן ארבעה תסריטים נוספים, אשר כל אחד מהם מתחיל בהוראת התניית ביצוע התסריט בלחיצה על מקש חץ, ובכל אחד מהם משולבת הוראת תנועה של פנייה לכיוון שמתאים למקש החץ. אפשרות זו היא נכונה, אך החיסרון שלה הוא שחלון התסריטים של פקמן הופך להיות עמוס למדי: הוא יכול שישה תסריטים וקשה יהיה להתמצא בו.

## הטלת אחריות התגובה למקשים על המבוך

ישנה גם אפשרות אחרת – להיעזר בדמות המבוך שכרגע לא מוטלת עליה שום מטלה, ולחלק את מטלות המשחק בין שתי הדמויות. על דמות המבוך נטיל את האחריות לקליטת הוראות הצופים, ועל פקמן טוטל, כמקודם, האחריות לתזוזה אגב זיהוי התנגשות ולהחלפת תלבושות. אם כך, דמותו של פקמן תמשיך לפעול כמקודם על פי שני תסריטים, האחד שאחראי על החלפת תלבושות והאחר שאחראי על התנועה. לדמות המבוך יהיו ארבעה תסריטים, כל אחד מתאים למקש חץ אחר, וכל אחד מתנה את תחילת ביצועו בלחיצה על המקש המתאים.

כיצד יתבצע הקשר בין הדמויות? הרי דמות המבוך אינה יכולה לשנות את כיוונה של דמות אחרת! בפרק 4 ראינו דרך אחת לתקשורת בין דמויות על ידי העברת מסרים. אבל כדי להשתמש בדרך זו נצטרך להשתמש בארבעה מסרים, אחד לכל מקש, ולכן פקמן שוב יזדקק לארבעה תסריטים נוספים, אחד לטיפול בכל מסר אפשרי מבין הארבעה. לכן אין פתרון זה מסייע לנו להוריד את עומס המשימות מדמותו של פקמן.

אבל נוכל לקשר בין המבוך לבין פקמן בעזרת מנגנון המשתנים שהכרנו בפרק הקודם. במקום להעביר ישירות מסרים בין דמות המבוך לפקמן, יוכל המבוך להעביר לפקמן הודעות דרך "תיבת דואר". בכל פעם שנלחץ מקש, יעדכן המבוך את הערך שבתיבת הדואר לכיוון המתאים למקש שנלחץ. בכל פעם שפקמן ירצה לעדכן את כיוונו, הוא יעשה זאת על פי הערך השמור בתיבת הדואר.

## תקשורת בין הדמויות בעזרת משתנה

אם כך, נזדקק למשתנה שיהיה בתפקיד תיבת הדואר. נגדיר משתנה בעזרת הוראת היצירה צור משתנה

**צור משתנה**

שאליו צריך פקמן לפנות (0, 90, 180 או 90-). נקרא לו בשם המתאים לתפקידו, למשל turn. מובן שנרצה לאפשר לכל הדמויות גישה למשתנה זה, משום שגם המבוך וגם פקמן ישתמשו בו. הפעם לא נסמן את הלבנה המייצגת של המשתנה, משום שלא נרצה להציג את ערכו על הבמה: תפקידו של המשתנה הוא פנימי לניהול המשחק, והצופים אינם צריכים אפילו לדעת על קיומו.

כאמור, האחריות לעדכון ערכו של המשתנה בהתאם להוראות הצופים היא על דמות המבוך. היא זקוקה לשם כך לארבעה תסריטים, כל אחד למקש אחר. למשל, התסריט שמטפל במקש החץ למעלה צריך לפעול על פי התיאור הבא:

0. אם אחיזה על מקש החץ מעלה:

1. עדכן את turn במעלה 0-1

**תרגיל 2:** הכינו לדמות המבוך ארבעה תסריטים, כל אחד לטיפול במקש חץ אחר, שיפעלו על פי תיאורים דומים לזה שניתן כאן עבור מקש החץ למעלה.

**תרגיל 3:** לאיזה ערך יש לאתחל את המשתנה החדש, turn? הוסיפו הוראת אתחול מתאימה לסדרת הוראות האתחול שבתסריטיו של פקמן (תסריט התנועה ותסריט החלפת התלבושות).

## פקמן משתמש במשתנה

אם כך, טיפלו בקליטת ההוראות מהצופים ובעדכון המשתנה בהתאם להן. כעת צריך לעדכן את התנהגותו של פקמן כך שישתמש בערכו של המשתנה כדי לפנות לכיוון המבוקש.

כרגע תנועתו של פקמן מתחילה בכיוון ימין, והיא מורכבת מתזוזות קטנות של שני צעדים. נוכל להחליט שאחרי כל תזוזה יעדכן פקמן את הכיוון שלו, אם יש בכך צורך, כלומר, אם ההוראה האחרונה שהתקבלה מהצופים אינה תואמת את הכיוון הנוכחי. מאחר שהתזוזות הן כה קטנות, הזמן בין תזוזה אחת לשנייה הוא קצר מאוד, ולכן לא יוכלו הצופים להרגיש בזמן שעבר מהרגע שנלחץ מקש ועד שפקמן שינה בהתאם את כיוון תנועתו.

לכן יש לעדכן את תסריטו של פקמן בהתאם לתיאור הבא:

0. אם אחיזה על הדגל היחיד:

1. עדכן את turn (שם המקום, הכיוון והגובה)

2. אלחץ

2.1 אם הגענו בקיר

2.1.1 אחור "איי!"

2.2 אחת

2.2.1 2 503 עדכן

2.2.2 פנה לכיוון turn



שימו לב: כעת באחד מחלקיה של הוראת הביצוע המותנה (החלק המתאים ל-"אחרת") צריכות להיות משולבות שתי הוראות. ואכן, אין מגבלה על מספר ההוראות המשולבות בכל אחד מחלקיה של הוראת ביצוע מותנה. בתוך כל אחד משני ה"פיות" של הלבנה המתאימה ניתן לשלב לבנים במספר כלשהו. עדכנו את התסריט בהתאם, תעדו אותו ושמרו את הפרויקט בשם חדש. כעת נסו להפעיל את האנימציה ולשחק במשחק.

## קביעת קצב המשחק

בוודאי שמתם לב כי אין זה פשוט לשחק במשחק. פקמן נע במבוך במהירות רבה מאוד ונדרשת זריזות רבה כדי לכוונו היטב. נוכל להקל מעט על הצופה אם נאט מעט את תנועתו של פקמן. זאת נוכל לעשות על ידי שילוב הוראת המתנה קצרצרה לאחר כל תזוזה. כלומר, בתיאור שניתן למעלה יש להוסיף צעד נוסף לאחר הוראת הפנייה, כך:

2.2.3 האמן 0.01 שניות

עדכנו את התסריט בהתאם ונסו לשחק בו שוב. תוכלו לנסות ערכי המתנה שונים אשר מאפשרים שליטה סבירה בפקמן, אך עדיין גורמים לו לנוע מהר מספיק ליצירת משחק מעניין. תעדו ושמרו את הפרויקט.

## דוגמה 2: פקמן אינו מתייאש – המשחק מתחדש

למעשה, המשחק שיצרנו מסתיים בעקבות התנגשות. כאשר פקמן מתנגש בקיר, הוא נשאר במקומו וממשיך להחליף תלבושות ולומר "איי!". ניתן אמנם להזיז את פקמן בעזרת העכבר אל נקודת ההתחלה, אך נסו זאת ותראו שהוא ממשיך לנוע בהתאם להוראות הצופים, אך גם ממשיך לומר "איי!". כדי לחזור ולשחק במשחק כראוי, צריך ללחוץ שוב על הדגל הירוק.

## משימה 5: חידוש אוטומטי של המשחק

יש להרחיב את האנימציה כך שבעקבות התנגשות יאמר פקמן "איי!" למשך פרק זמן קצר, ואחר כך יתאפס המשחק: פקמן יחזור למצב ההתחלתי ויתחיל שוב לנוע בהתאם להוראות הצופים.



שם הקובץ: `restart-game`

נתבונן בתסריט האחראי על תנועתו של פקמן. לאחר האתחול התנועה כולה מתנהלת בתוך ביצוע חוזר אינסופי. פקמן חוזר ובודק שוב ושוב את תנאי ההתנגשות. אם כך כל שעלינו לעשות הוא לעדכן את המתבצע בעקבות התנגשות:

במשחק הנוכחי, כאשר פקמן מזהה התנגשות, הוא אומר "איי!" ואינו זז. לכן תנאי ההתנגשות ממשיך להתקיים, ופקמן נשאר במקומו. בגרסה החדשה, כאשר פקמן מזהה התנגשות, הוא יאמר "איי!", אך לפרק זמן קצוב, למשל שתי שניות, ואחר כך יזיז את עצמו שוב לנקודת ההתחלה ולמצב ההתחלתי. בגלל הביצוע החוזר, לאחר ההתמקמות במצב ההתחלתי, יבדוק שוב פקמן את תנאי ההתנגשות, יזהה שאין התנגשות וימשיך לנוע.

זה התיאור המתאים:

עם לחיצה על הדגל הירוק:

0. בצד אגמול (ש' המיקום, הכיוון, הגאנצט וחסגנה הכיוון)

1. אלולאמ

1.1 אס הגעשט בקי

1.1.1 אמוך "איי!" אלשך ש' שניג

1.1.2 בצד אגמול (ש' המיקום, הכיוון והגאנצט)

1.2 אמכ

1.2.1 2 513 דים

1.2.2 פנה לכיוון turn

1.2.3 האגן 0.01 שניג

עדכנו את התסריט האחראי על התנועה בהתאם לתיאור זה. שימו לב, יש להחליף את הוראת הדיבור בהוראת דיבור חדשה, התחומה בזמן. כמקודם, צעד האתחול טומן בתוכו מספר הוראות: האחת – לאתחול מיקום, השנייה – לאתחול כיוון, השלישית – לאתחול תלבושת, והרביעית – לאתחול משתנה הכיוון. לאחר השינוי תהיה גם בחלקה הראשון של הוראת הביצוע המותנה יותר מהוראה אחת. תעדו את התסריט המעודכן ושמרו את הפרויקט בשם חדש. כעת נסו להפעיל את האנימציה ולשחק במשחק.

### דוגמה 3: פקמן מוריק מבושה בעקבות התנגשות – עוד על ביצוע מותנה

בגרסה האחרונה שיצרנו, התנגשות גורמת לפקמן לעצור ולדבר. בגרסה שניצור כעת תגרום ההתנגשות לפקמן לשנות את צבעו לירוק.

### משימה 6: פקמן ישנה את צבעו לירוק

יש להרחיב את האנימציה כך שבעקבות התנגשות פקמן ישנה את צבעו לירוק.



לכאורה, השינוי נראה פשוט. למעשה, נראה כי כל שינוי באלגוריתם דורש מחשבה וזהירות, משום שהוא עלול לגרום לבעיות לא צפויות. במקרה שלנו, הבעיות שיעלו בעקבות השינוי יאפשרו לנו היכרות מעמיקה יותר עם ביצוע מותנה, היכרות שתידרש לצורך פתרון.

### ניסיון ראשון

ובכן, ברור כי קודם כול עלינו ליצור לפקמן תלבושת חדשה, אשר בה צבעו הוא ירוק. עברו ללשונית התלבושות של פקמן, ובתלבושת הראשונה לחצו על הכפתור "העתק". כתוצאה מכך נוספת לפקמן

תלבושת חדשה, שלישית, זהה לראשונה. תנו לה שם מתאים, ותוכלו לערוך אותה ולשנות את צבעה לירוק בעזרת עורך הציור.

כעת, כאשר לפקמן יש תלבושת מוכנה למקרה התנגשות, נראה כי כל שעלינו לעשות הוא לשנות מעט את רצף ההוראות המתבצע אם מזוהה התנגשות. את הוראות האתחול השונות יש להשאיר ללא שינוי, את הוראת הדיבור התחומה בזמן יש למחוק ובמקומה להכניס הוראה מוחלטת של שינוי תלבושת לתלבושת החדשה שיצרנו. מאחר שמיד אחרי ההוראה החדשה מתבצע אתחול מיקום, כיוון ותלבושת, כדאי להכניס אחרי ההוראה החדשה הוראת המתנה של שתי שניות כדי שהצופים יספיקו להבחין בתלבושת הירוקה המעידה על התנגשות.

**תרגיל 4:** נסו לבצע את השינוי והפעילו את המשחק. תארו מה קורה. נסו להסביר את התופעה.

שם הקובץ: `pacman-change-to-green-bug1`

## ניתוח התקלה

ובכן, השינוי שערכנו לא התחשב בכל ההיבטים של התנהגותו של פקמן. פרט לתסריט האחראי על התנועה, פקמן פועל גם על פי תסריט שאחראי על החלפת תלבושות, ושני התסריטים פועלים במקביל. השילוב בין תסריט התלבושות לתסריט המעודכן יוצר שתי בעיות:

ראשית, בתסריט התלבושות יש החלפה מעגלית אינסופית בין כל תלבושותיו של פקמן על ידי שימוש בהוראה היחסית **התלבושת הבאה**. כמובן, קודם היו רק שתי תלבושות והיינו מעוניינים שפקמן יחליף בין שתיהן. כעת יש ברשימת התלבושות של פקמן שלוש תלבושות. שימוש בהוראה **התלבושת הבאה** גורם לכך שהחלפת התלבושות המתבצעת תוך כדי תנועה כוללת גם את התלבושת החדשה, הירוקה. בנוסף, ההחלפה ממשיכה להתבצע גם כאשר מתרחשת התנגשות, ולכן גם תוך כדי ההמתנה של שתי השניות לפני שפקמן חוזר לנקודת ההתחלה, תלבושתו ממשיכה להתחלף ואינה נשארת ירוקה.

## ניסיון שני

הדרך לפתור את הבעיה היא לעדכן את תסריט החלפת התלבושות כך שלא יעבוד בצורה מעגלית. תסריט זה לא יאפשר החלפה לתלבושת הירוקה, וההוראה היחידה שתאפשר שינוי לתלבושת הירוקה תהיה הוראת השינוי בעקבות התנגשות שבתסריט התנועה של פקמן.

בתסריט האחראי על התלבושות יש למחוק את ההוראה **התלבושת הבאה**. במה נחליף אותה?

אם איננו יכולים להשתמש בהוראת ההחלפה המעגלית היחסית, עלינו להשתמש בהוראה המוחלטת של שינוי תלבושת. מאחר שאנו מעוניינים להחליף בין שתי התלבושות, נצטרך להשתמש בהוראה המוחלטת פעמיים. פעם בהוראה הממוקדת **שנה לדמות** `pac-man-closed` ופעם בהוראה הממוקדת **שנה לדמות** `pac-man-open`. כיצד נדע מתי להשתמש באיזו הוראה?

אם תלבושתו הנוכחית של פקמן היא התלבושת הראשונה, `pac-man-open`, עלינו לשנות לתלבושת השנייה, `pac-man-closed`; ואם תלבושתו הנוכחית של פקמן היא התלבושת השנייה, `pac-man-closed`, עלינו לשנות לתלבושת הראשונה, `pac-man-open`. בכך נבטיח שתסריט זה לא ישנה את התלבושת הירוקה וגם לא יגרום החלפה אל התלבושת הירוקה.

ננסה ליישם את הפתרון המוצע בעזרת הוראת הביצוע המותנה שהכרנו. נראה כי התרגום של הרעיון לתיאור מילולי של צעדי ביצוע הוא זה:

4.2 אם הגלגול היא הגלגול האחרונה

4.2.1 פנה אלגלגול pac-man-closed


4.3 אחר

4.3.1 פנה אלגלגול pac-man-open

## השוואת ערכים

כדי שנוכל לתרגם את השינוי המוצע ל־Scratch, עלינו ללמוד עוד כיצד לתאר תנאי המשווה את התלבושת הנוכחית לתלבושת הראשונה.

אין ב־Scratch תנאי שנוצר בדיוק למטרה זו. במקום זאת יש תנאי כללי הבודק שוויון בין שני דברים. אנו מעוניינים לדעת אם התלבושת הנוכחית היא הראשונה, ולכן בעצם אנו רוצים לבדוק אם מספרה של התלבושת הנוכחית שווה למספר 1. לשם כך נוכל להשתמש בתנאי השוויון הכללי ולשלב בו את הערכים הדרושים.


תנאי השוויון הכללי נראה כך . הוא נמצא בקבוצת המפעילים, הקבוצה הירוקה-בהירה. צורתו היא כצורתו של כל תנאי, כלומר, פינותיו קטומות. הוא מכיל שתי חלונות, שביניהן סימן שוויון. בחלונות ניתן לשלב ערכים כרצוננו, ובכך להפוך את התנאי הכללי לתנאי ממוקד, הבודק שוויון בין הערכים המשולבים.

במקרה שלנו נרצה לשלב בצד אחד של סימן השוויון את המספר 1, ובצד השני נרצה לשלב את מספרה הסידורי של התלבושת הנוכחית. בקבוצת המראה, הקבוצה הסגולה, יש לבנה המתאימה לצורך זה, הלבנה **תלבושת**, שהיא הלבנה המייצגת את מספרה הסידורי של התלבושת הנוכחית.

כזכור, עד כה כאשר שילבנו לבנה בתוך חלונית, הקפדנו תמיד על התאמה בין צורת החלונית לצורת הלבנה המשולבת בה. למשל, בתוך חלונית תנאי שפינותיה קטומות יכולנו לשלב רק לבנת תנאי, שגם פינותיה קטומות. בדומה, בתוך חלונית ההוראה לשינוי גודל שפינותיה מעוגלות, יכולנו לשלב רק לבנה שפינותיה מעוגלות (כמו לבנה מייצגת של משתנה).

החלונות שבתנאי השוויון הכללי צורתן מלבנית. פינותיהן אינן קטומות ואף לא מעוגלות. לכאורה, נראה שצורה זו אינה מאפשרת לנו לשלב בחלונות את הלבנה **תלבושת** שפינותיה מעוגלות. אך למעשה, צורה זו היא צורה מתירנית למדיי, ומותר לשלב בתוכה כל לבנה שהיא לבנת ערך, ולא לבנת הוראה. כלומר, כל לבנה שחלקה העליון והתחתון ישרים (אין בה שקע או בליטה), ניתנת לשילוב בתוך חלונית מלבנית, ואין זה משנה מה צורת פינותיה של לבנת הערך המשולבת. בפרט, נוכל לשלב בצדו האחד של תנאי השוויון הכללי את הלבנה **תלבושת**, ובצדו השני פשוט נכתוב את הערך 1 בחלונית.

### תנאי חדש ב־Scratch: תנאי השוויון הכללי

התנאי  מתקיים אם הערכים המופיעים בשתי החלונות שלו שווים זה לזה, לפי ערך מספרי. התנאי אינו מתקיים אם הערכים שונים זה מזה.

## לבנת ערך חדשה ב־Scratch: המספר הסידורי של תלבושת

לבנת הערך **תלבושת** היא הלבנה שמייצגת את מספרה הסידורי של התלבושת הנוכחית של הדמות (על פי מיקומה ברשימת התלבושות של הדמות).

**תרגיל 5:** בצעו את השינוי המוצע והפעילו את המשחק. תארו מה קורה. נסו להסביר את התופעה.

שם הקובץ: `pacman-change-to-green-bug2`

## ניתוח התקלה

פתרנו את הבעיה הראשונה מבין שתי הבעיות שנוצרו קודם. התסריט החדש אכן אינו גורם אף פעם להחלפה לתלבושת הירוקה. זאת משום שאף אחת משתי ההוראות המוחלטות לשינוי התלבושת שכלולות בתסריט, אינה מתייחסת לתלבושת הירוקה. לכן הפעולה הרגילה של החלפת התלבושות, תוך כדי תנועה, מתבצעת היטב.

לעומת זאת, הבעיה השנייה לא נפתרה. כאשר מתרחשת התנגשות בפרק הזמן של שתי השניות לפני חידוש המשחק, פקמן לא נשאר בתלבושתו הירוקה, אלא ממשיך להחליף תלבושות.

מדוע זה קורה? הפתרון שהצענו היה אמור גם להבטיח שתסריט התלבושות לא יבצע שום שינוי תלבושת כאשר תלבושתו של פקמן היא ירוקה. הרעיון היה נכון, אך הבעיה היא שהשינוי שערכנו לא תרגם את הרעיון שלנו באופן מדויק. ההבדל הוא קטן, אך משמעותי. אנחנו רצינו לבצע את הדבר הבא:

**אם** תלבושתו הנוכחית של פקמן היא התלבושת הראשונה שנה לתלבושת `pac-man-closed`,

**ואם** תלבושתו הנוכחית של פקמן היא התלבושת השנייה שנה לתלבושת `pac-man-open`.

אך התרגום שלנו הוא זה:

**אם** תלבושתו הנוכחית של פקמן היא התלבושת הראשונה שנה לתלבושת `pac-man-closed`,

**אחרת** שנה לתלבושת `pac-man-open`.

האם שני הניסוחים זהים? לא, בהחלט לא. ההבדל ביניהם נעוץ בחלק השני של המשפט.

**תרגיל 6:** מצאו תלבושת שמבחינה בין הניסוחים. כלומר, תארו מצב שבו תלבושתו של פקמן היא

כזאת, שעל פי הניסוח הראשון לא יתבצע השינוי לתלבושת `pac-man-open`, אך על פי

הניסוח השני יתבצע שינוי לתלבושת `pac-man-open`.

שני הניסוחים פועלים באופן שונה כאשר תלבושתו של פקמן ירוקה. על פי הניסוח הראשון: אם תלבושתו של פקמן היא ירוקה, היא אינה התלבושת הראשונה, ולכן לא מתבצע שינוי לתלבושת `pac-man-closed`; היא גם לא התלבושת השנייה, לכן לא מתבצע שינוי לתלבושת `pac-man-open`. לעומת זאת, על פי הניסוח השני: אם תלבושתו של פקמן היא ירוקה, היא אינה התלבושת הראשונה,

ולכן לא מתבצע שינוי לתלבושת pac-man-closed; אבל מאחר שהיא לא התלבושת הראשונה, תנאי ה"אחרת" מתקיים, ולכן מתבצע שינוי לתלבושת pac-man-open. מהרגע שהתלבושת הירוקה מוחלפת בתלבושת הראשונה, ההחלפה המעגלית בין שתי התלבושות (פה פתוח ופה סגור) יכולה להמשיך ולהתבצע.

## תיקון התקלה

כאמור, הרעיון שלנו היה נכון, אך לא תרגמנו אותו באופן מדויק להוראות של Scratch. כדי לבצע תרגום נכון, עלינו להשתמש בביצוע מותנה בדרך מורכבת מעט יותר.

התרגום שלנו צריך להיות כזה:

4.2 אם התלבושת היא התלבושת הכאש

4.2.1 שנה אל תלבושת pac-man-closed

4.3 אחרת

4.3.1 אם התלבושת היא התלבושת השנייה

4.3.1.1 שנה אל תלבושת pac-man-open

המבנה שהתקבל הוא מורכב למדי, והוא משלב שני רעיונות חדשים של ביצוע מותנה:

## קינון של ביצועים מותנים

ראשית, יש כאן שימוש בביצוע מותנה בתוך ביצוע מותנה: המילה "אם" מופיעה כאן פעמיים. בצעד 4.2 אנו בודקים אם התלבושת הנוכחית היא התלבושת הראשונה. אם התנאי מתקיים מתבצעת הוראת שינוי תלבושת, אבל אם הוא אינו מתקיים, בחלק של ה"אחרת", בצעד 4.3.1 מתבצעת בדיקה נוספת, של תנאי אחר. כעת אנו בודקים אם התלבושת הנוכחית היא התלבושת השנייה, ואם התנאי מתקיים, מתבצעת הוראה אחרת לשינוי תלבושת.

נוהגים לקרוא לשימוש כזה **קינון** של ביצוע מותנה. למעשה, כבר ראינו שימוש דומה, אך עבור ביצוע חוזר ולא עבור ביצוע מותנה. בפרק 5 השתמשנו בביצוע חוזר בתוך ביצוע חוזר. רעיון זה של קינון אמנם עלול להיראות מורכב ומבלבל, אך עלינו לזכור כי בתוך הוראת בקרה ניתן לשלב כל הוראה אחרת של Scratch (כל עוד ההוראה המשולבת אינה של ביצוע אינסופי). בפרט, בתוך הוראת בקרה ניתן לשלב הוראת בקרה אחרת. לכן בתוך הוראת ביצוע מותנה ניתן לשלב הוראת ביצוע מותנה. גם שילובים אחרים הם אפשריים. למשל, כבר בדוגמה הראשונה בפרק זה השתמשנו בביצוע מותנה בתוך ביצוע חוזר אינסופי.

### מושג חדש: קינון של ביצועים מותנים

בתוך הוראה לביצוע מותנה ניתן לשלב כל הוראה, בפרט הוראה לביצוע מותנה. בכך מתקבל קינון של ביצועים מותנים.

## ביצוע מותנה ללא חלופה

אבל שימו לב איזו הוראה של ביצוע מותנה משולבת בפנים, בתוך ההוראה הראשונה לביצוע מותנה: זוהי הוראה לביצוע מותנה שאין לה חלק של "אחרת". הוראה כזו מתנה ביצוע של הוראה או הוראות בקיום תנאי (במקרה שלנו, ביצוע הוראת שינוי תלבושת מותנה בכך שהתלבושת הנוכחית היא התלבושת השנייה), אך אינה נותנת שום הוראה לביצוע, אם התנאי אינו מתקיים. אם התנאי אינו מתקיים, דבר אינו מתבצע. זוהי הוראה לביצוע מותנה ללא חלופה.

על פי התיאור שלנו ייתכנו כמה אפשרויות:


אם התלבושת הנוכחית היא הראשונה, בדיקת התנאי הראשון עוברת בהצלחה, התנאי השני לא ייבדק ויתבצע שינוי התלבושת הראשון.

אם התלבושת הנוכחית היא השנייה, בדיקת התנאי הראשון מובילה לתשובה שלילית (התנאי אינו מתקיים), התנאי השני נבדק, הבדיקה עוברת בהצלחה ושינוי התלבושת השני מתבצע.

אם התלבושת הנוכחית היא השלישית (הירוקה), בדיקת התנאי הראשון מובילה לתשובה שלילית (התנאי אינו מתקיים), התנאי השני נבדק, הבדיקה מובילה לתשובה שלילית (התנאי אינו מתקיים), ודבר אינו מתבצע.

כיצד נתרגם תיאור מילולי כזה ל-Scratch? בקבוצת הבקרה יש הוראה המתאימה לביצוע מותנה ללא




חלופה. ההוראה היא . גם בה משולבת חלונית תנאי, אך לה יש "פה אחד", המאפשר לשלב רק הוראות שיתבצעו אם התנאי מתקיים.

### מושג חדש: ביצוע מותנה ללא חלופה

ביצוע מותנה ללא חלופה כולל תנאי וסדרה סופית של הוראות. אם התנאי מתקיים, מבוצעת סדרת ההוראות, ואם אין הוא מתקיים – הסדרה אינה מתבצעת.

### הוראה חדשה ב-Scratch: ביצוע מותנה ללא חלופה



בביצוע ההוראה  נבדק התנאי המשולב בהוראה. אם הוא מתקיים, מתבצעות ההוראות המשולבות בתוך ההוראה.

השתמשו בהוראת הביצוע המותנה ללא חלופה כדי לתקן את הפתרון האחרון וליישם את התרגום האחרון שהוצע. בדקו אם בעקבות השינוי החלפת התלבושות מתבצעת כראוי לאורך כל המשחק. תעדו את השינוי ושמרו את הפרויקט.

**תרגיל 7:**

שם הקובץ: `pacman-change-to-green-correct`

במהלך דוגמה 2 הזכרנו פתרון חלופי לטיפול בהוראות הצופים. בפתרון החלופי האחריות לטיפול היא של פקמן ולא של המבוך. לשם כך פקמן ישתמש בהוראת בקרה מותנית, התלויה באירוע של לחיצת מקש על ידי הצופים. בתרגיל זה עליכם לשנות את הפרויקט של דוגמה 2 בהתאם להצעה זו.

בפתרון החדש לא יהיו למבוך שום תסריטים, ולא יהיה שימוש במשתנה `turn`. במקום זאת, תיכלל בתסריט התנועה של פקמן הוראת ביצוע מותנה הבודקת לחיצה על מקש חץ, ובהתאם ישנה פקמן את כיוונו. השתמשו בקינון כדי לשלב את ארבע הוראות הביצוע המותנה הדרושות, אחת לכל חץ. האם בין ארבע הוראות הביצוע המותנה צריכות להיות הוראות של ביצוע מותנה ללא חלופה?

שם הקובץ: `user-control-nested-if`

## דוגמה 4: פקמן אוכל בננות – המשחק המלא

ולבסוף, נשלים את המשחק שיצרנו כך שתוך כדי ההליכה במבוך, נוסף להתחמקות מהתנגשויות, יוכלו הצופים להשקיע מאמץ נוסף ולכוון את פקמן כך ש"יאכל" דברים הנקרים בדרכו.

## משימה 7: פקמן אוכל בננות

יש להרחיב את המשחק כך שבמקומות שונים במבוך נמצאים אשכולות של בננות. כאשר פקמן מגיע אל אשכול בננות, הוא אוכל אותן. שם הקובץ: `pacman-eats-bananas`



אם כך, בגרסה שלנו נפזר במבוך בננות, כך שפקמן יוכל "באמת" לאכול אותן. במשחק אמיתי, אכילת הבננות יכולה להעניק לצופים נקודות, אך בגרסה שלנו לא נעסוק בזה (למעוניינים: ההיכרות שערכתם עד כה עם סביבת Scratch, והעקרונות שלמדתם דרכה, בהחלט מאפשרים לכם להרחיב את המשחק גם בכיוון זה, אם תרצו, אך לאחר לימוד הפרק הבא, זה יהיה קל יותר).

אנו נסתפק בהוספת שתי בננות. שתי הדמויות יהיו זהות בהתנהגותן, ולכן יהיה נוח ליצור דמות אחת ואחר כך לשכפל אותן. ניצור דמות חדשה, של בננה (מתיקית החפצים, `Things`). השתמשו בכפתור הקטנת הדמויות כדי להקטין את הדמות כך שתיכנס בנוחות לתוך שביל במבוך. מקמו את הבננה במקום כלשהו במבוך. רצוי שיהיה זה מקום רחוק מנקודת ההתחלה, כך שלא יהיה זה פשוט עבור הצופים לכוון את פקמן עד לבננה.

תפקידה של דמות הבננה הוא פשוט. בתחילת המשחק היא ניצבת במקומה, ושם היא נשארת עד שפקמן מגיע אליה. אז היא נאכלת. כשהמשחק מתחדש, עליה להימצא שוב במקומה.

מבחינת הצופים, כאשר הבננה נאכלת, היא פשוט נעלמת מהמבוך. אם כך, ראשית עלינו למצוא דרך לגרום לדמות להיעלם ולהופיע. אחר כך עלינו לדאוג שההיעלמות תתרחש כשפקמן נוגע בבננה, ושההופעה מחדש תתבצע כאשר המשחק מתחדש.



## העלמת אשכולות הבננות והופעתם מחדש, בתזמון הנכון

בקבוצת המראה הסגולה ישנן שתי הוראות פשוטות המורות לדמות להראות את עצמה ולהעלים את עצמה. אלו הן ההוראות **הצג** ו- **הסתר**. כיצד נדאג לכך שהוראות אלו יבוצעו בזמן הנכון?

נתחיל מההצגה. דמות הבננה צריכה להציג את עצמה כשהמשחק מתחיל וכשהוא מתחדש. את תחילת המשחק קל לדמות הבננה לזהות על ידי לחיצה על הדגל הירוק, אך כיצד תזהה שהמשחק מתחדש? פקמן הוא זה שגורם למשחק להתחדש, אחרי התנגשות. לאחר התנגשות הוא מביא את עצמו שוב למצב ההתחלתי. מאחר שהוא זה שיודע על כך, תפקידו הוא ליידע את דמות הבננה בכל פעם שזה מתרחש.

פקמן יוכל ליידע את דמות הבננה על ידי כך שיעביר לה מסר. ובכן נוסיף לתסריט התנועה של פקמן הוראה של שליחת מסר. פקמן יעביר את המסר (למשל "מאתחל") כאשר הוא גורם לחידוש המשחק, כלומר, אחרי רצף הוראות האתחול של המיקום, כיוון, משתנה הכיוון והתלבושת. הבננה תזדקק לתסריט שיבצע עם קבלת המסר, וההוראה הראשונה בתסריט זה תגרום לה להציג את עצמה, כמו בתיאור הבא:

0. **עס קבאל המסר "מאתחל"**

1. **הציג אל עצמך**

כדי לחסוך לבננה תסריט נוסף לצורך אתחול, שיתבצע עם הלחיצה על הדגל הירוק, יוכל פקמן לשלוח את מסר האתחול גם עם תחילת המשחק, אחרי רצף הוראות האתחול הראשונות.

מה לגבי ההיעלמות? למעשה, לאחר שהבננה הציגה את עצמה, לא נותר לה דבר לעשותו, עד שפקמן ייגע בה, ואז עליה להעלים את עצמה. כלומר, היא יכולה להמתין עד שפקמן ייגע בה. כשזה יקרה, היא תעלים את עצמה. אם כך, את תיאור פעולתה של הבננה נרחיב באופן הבא:

0. **עס קבאל המסר "מאתחל"**

1. **הציג אל עצמך**

2. **המתינ עד שפקמן נגע בך**

3. **העלמי אל עצמך**

הוסיפו בתסריט התנועה של פקמן את הוראות העברת המסר המתאימות. כתבו עבור הבננה תסריט מתאים לתיאור שניתן כאן. השתמשו בהוראת המתנה מותנית ובתנאי נגיעה. לאחר מכן שכפלו את דמותה של הבננה לקבלת דמות בננה שנייה עם אותו תסריט (שכפול דמות ניתן לביצוע על ידי הצבעה עם העכבר על הדמות בחלון הדמויות, לחיצה על הכפתור הימני של העכבר ובחירת הפריט "שכפל" ברשימה שהופיעה בעקבות הלחיצה). מקמו את הבננה השנייה במקום אחר במבוך.

נסו את המשחק המתקבל, תעדו אותו ושמרו את הפרויקט בשם חדש.

### תרגיל 9:

## הוראות חדשות ב-Scratch: הצגת דמות והסתרתה

ההוראה **הצג** גורמת לדמות להראות את עצמה על הבמה במיקומה הנוכחי.

ההוראה **הסתר** גורמת לדמות להפוך את עצמה לבלתי נראית על הבמה.

## דוגמה 5: מספרים אקראיים

לסימו של הפרק נציג דוגמה החוזרת אל מגרש הכדורגל. בדוגמה זו נשתמש בביצוע מותנה ובמספרים אקראיים כדי ליצור גרסה משעשעת של אנימצית משחק הכדורגל מפרק 4. בפרק 4 השתתפו במשחק הכדורגל הראשון שלוש דמויות: שופט, שנתן הוראה לבעיטת פתיחה, שחקן – פלה – שבעט בכדור בעקבות ההוראה, והכדור שנבעט. בגרסה הנוכחית נוותר על השופט ועל הוראת הבעיטה ובמקומו נצרף שוער.

### משימה 8:

יש לשנות את אנימצית משחק הכדורגל מפרק 4, כך שמיד עם תחילת האנימציה יבעט פלה בכדור וינסה להבקיע שער. אם הצליח, יישאר הכדור במקומו. אם השוער הצליח לעצור את הכדור, יוחזר הכדור לכיוונו של פלה. **שם הקובץ:** `cat-kicks-ball-randomly`



אנו מבינים שהמשחק צריך להתנהל כך שהבעיטה, כלומר תנועת הכדור, לא תהיה תמיד לאותו כיוון. בכל פעם תכונן הבעיטה לכיוון כלשהו, לא ידוע מראש.

### השחקן בועט

נתחיל בהתנהגותו של השחקן, פלה. מאחר שכעת אין הוא צריך להמתין להוראת הבעיטה של השוער, אין גם צורך בהמתנה למסר. מיד עם הלחיצה על הדגל הירוק הוא צריך להתייבב במקומו, כשהוא פונה לכיוון הכדור. אז יאמר דבר מה, בדומה למה שהיה באנימציה המקורית מפרק 4 (למשל, "קדימה"), וינוע לכיוון הכדור. אין הוא צריך לעשות דבר נוסף, משום שכזכור, חלקו בבעיטה מסתכם רק בתנועה לכיוון הכדור, והכדור הוא זה שצריך "להשלים" את הבעיטה בכך שיזוז לאחר שפלה נוגע בו. אם כך, החלק שנראה בעייתי – יצירת בעיטה לכיוון שאינו ידוע מראש – הוא בעצם באחריותו של הכדור, ולא של השחקן.

**תרגיל 10:** הפכו את התיאור הכללי שניתן כאן עבור התנהגותו של פלה לתיאור מילולי מפורט, ובנו תסריט שמתאים לו. מקמו את פלה בנקודת ההתחלה (0, 190) כשהוא פונה שמאלה. לצורך הבעיטה הניעו אותו 40 צעדים.

## השוער עומד בשער ומקווה לטוב

נעבור לשוער. השוער הזה לא יתאמץ יותר מדי לעצור את הכדור. הוא פשוט יעמוד במקומו ויקווה שהכדור יגיע אליו. לכן התנהגותו פשוטה מאוד: הוא מתייצב במקומו בתחילת המשחק. מעבר לזה אין הוא צריך לעשות דבר. אמנם, על פי דרישות המשחק אם השוער "מצליח" לעצור את הכדור עליו להחזירו לכיוונו של פלה, אבל בדיוק כמו בבעיטת הפתיחה, תזוזת הכדור בעקבות הבעיטה היא באחריותו של הכדור ולא באחריות הבעט.

**תרגיל 11:** הוסיפו את דמותו של השוער (תוכלו למצוא דמות מתאימה בתיקיית האנשים (People), למשל הדמות הראשונה בתיקייה הנקראת amon1). בנו תסריט מתאים עבור דמות השוער. השוער ימוקם בנקודת ההתחלה (0, -200) כשהוא פונה ימינה.

## התנהגות אקראית של הכדור

נותר לנו הטיפול בכדור. כפי שראינו, עיקר המשחק הוא באחריות הכדור. הוא זה שזז בעקבות בעיטתו של פלה, ואם הבעיטה נעצרת על ידי השוער, עליו לזוז בחזרה לכיוונו של פלה. לכן עליו לפעול לפי התיאור הכללי הבא:

0. אם אחיזה על הדגל היחיד:

1. אגמא אג מיקומך וכיווןך

2. האם עדיף שפלה נקע בקר

3. נעל לכיוון אקראי באלה

4. אם פגעה בשוער

4.1 הסגובה ונעל מלט לכיוון של פלה

הידע שרכשתם עד כה מאפשר לכם לפרט ולתרגם כל אחד מהצעדים שבתיאור כללי זה, פרט לצעד 3. כמו בפרק 4, את תנועתו של הכדור בצעד 3 נממש בעזרת הוראת גלישה. אילו ערכים נשלב בהוראה עבור נקודת הסיום של הגלישה, כדי לקבל בעיטה בכיוון כלשהו? ערך ה־x של נקודת היעד יהיה תמיד אותו ערך, והוא גבולו השמאלי של המגרש, השווה לערך ה־x של מיקומו של השוער. את כיווני הבעיטה השונים נשיג על ידי כך שבכל פעם שהוראת הגלישה תבוצע, יהיה ערך ה־y שישולב בה ערך כלשהו. למעשה, מה שהיינו רוצים שיקרה הוא כאילו להגריל באקראי נקודת y אפשרית, וערכה ישולב בהוראת הגלישה.

סביבת Scratch מאפשרת לנו לבצע זאת בעזרת הוראה שמגרילה מספר אקראי בתחום נתון ונותנת לנו את ערכו. בקבוצת המפעילים, הירוקה-בהירה, ישנה לבנה המתאימה להוראה זו: **בחר מספר אקראי בין ל**. כאשר לבנה זו מופעלת, היא נותנת לנו מספר אקראי בתחום המבוקש. הפעלות חוזרות של הלבנה יכולות לתת (וסביר שיתנו) ערכים שונים בתחום.

שימו לב לצורתה של לבנה זו: היא מלבנית עם פינות מעוגלות, כמו לבנה שמייצגת גודל, תלבושת או לבנה שמייצגת משתנה. זו לבנת ערך, וניתן לשלבה בכל מקום שבו נדרש ערך מספרי. לכן ניתן לשלבה כערך נקודת y בהוראת גלישה.

### מושג חדש: בחירת מספר אקראי

במקרים רבים עלינו לדמות הגרלה של ערך אחד מתוך קבוצת ערכים, כך שהסיכויים לבחירת כל אחד מהערכים הם שווים. לשם כך ניתן לבחור באופן אקראי מספר מתוך תחום המתאים לקבוצה. יש סבירות גבוהה שבחירות אקראיות חוזרות מתוך אותו תחום יתנו ערכים שונים.

### הוראה חדשה ב־Scratch: בחירת מספר אקראי

ההוראה **בחר מספר אקראי בין**  ל־  גורמת לבחירת מספר אקראי שנמצא בין שני המספרים המופיעים בחלונות של ההוראה. הלבנה המתאימה היא לבנת ערך, המייצגת את ערכו של המספר שנבחר.

**תרגיל 12:** צרו תיאור מפורט מהתיאור הכללי שניתן עבור הכדור. מיקומו ההתחלתי של הכדור יהיה  $(-20, 120)$ , והוא יפנה שמאלה. תנועת הגלישה שלו תאריך שנייה אחת. ערכי  $y$  אפשריים לנקודת יעד הגלישה יהיו בין  $-120$  ל־ $120$ . אם הכדור פגע בשוער, הוא ינוע בכיוון הפוך, אל פלה,  $150$  צעדים. בנו על פי התיאור המפורט שתכנתם תסריט לפעולתו של הכדור. נסו את המשחק המתקבל, תעדו אותו ושמרו בשם חדש.

## הרחבה (רשות): אפקטים של צבע ובהירות

נשפר את המשחק שיצרנו במשימה הקודמת.

### משימה 9:

יש להרחיב את האנימציה כך שאם פלה הצליח להבקיע גול, הוא יבהיר וישנה צבעו מרוב שמחה ויקפוץ. לעומתו, ישחיר השוער מבושה ויפול ארצה. אם פלה לא הצליח להבקיע, הוא ייסוב אחור כשהוא משחיר מבושה. השוער לעומתו יבהיר וישנה צבעו מאושר ויקפוץ לגובה.



שם הקובץ: `cat-kicks-ball-randomly-change-effect`

### ושוב האחריות היא של הכדור

בגרסה זו אנו מעשירים את פעילותם של פלה והשוער. אבל פעילות זו תלויה בפעולתו של הכדור. מאחר שחלק הארי של הבעיטה מבוצע בעצם על ידי דמות הכדור, הוא זה שיודע אם הגיע אל השוער או לא.

ובכן, גם במקרה זה יש צורך בתקשורת בין הדמויות, וכמו במקרים רבים נבצע את התקשורת בעזרת מסרים. אם הכדור הגיע אל השוער, אז חוץ מההסתובבות והתנועה לכיוונו של פלה ישלח הכדור מסר כגון "הכדור נעצר". אם לא הגיע אל השוער, ישלח הכדור מסר כגון "גול".

**תרגיל 13:** עדכנו את התסריט של הכדור בהתאם. שימו לב, פרט להוספת שתי הוראות של שליחת מסר, יש גם צורך בהחלפת הוראת הביצוע המותנה. מדוע?

## חלקם של פלה והשוער

כיצד נשנה את התנהגותם של פלה והשוער? לכל אחד מהם עלינו להוסיף שני תסריטים: תסריט אחד שיתבצע אם הגיע מהכדור מסר של הבקעת גול, ותסריט שני שיתבצע אם הכדור שלח מסר של עצירת הבעיטה. על השוער לפעול לפי התיאורים הבאים:

0. אם קבאל המסר "גול!"

1. השמר את מצב

2. פול ארצה

0. אם קבאל המסר "הכדור נעצר"

1. הבהר את מצב

2. שנה מצב

3. קפוף מלא

על פלה לפעול לפי התיאורים הבאים:

0. אם קבאל המסר "גול!"

1. הבהר את מצב

2. שנה מצב

3. קפוף מלא

0. אם קבאל המסר "הכדור נעצר"

1. השמר את מצב



2. סוב אחור

לתיאורים יש כמה צעדים משותפים. למשל, צעד הקפיצה למעלה נמצא גם בתסריט ההבקעה אצל פלה וגם בתסריט חסימת הבעיטה אצל השוער. כך גם לגבי צעדי ההבהרה ושינוי הצבע. בדומה, צעד ההשחרה נמצא גם בתסריט חסימת הבעיטה אצל פלה וגם בתסריט ההבקעה אצל השוער.

נתחיל מצעדי הנפילה, הסיבוב אחור והקפיצה, שהכלים לכתובתם כהוראות של Scratch כבר בידיכם. את הנפילה ארצה אפשר ליצור פשוט בעזרת שינוי כיוון לכיוון מטה. הסיבוב אחור של החתול יכול להתקבל על ידי פנייה לכיוון ימין. את הקפיצה אפשר ליצור בעזרת הוראת גלישה של שנייה אחת 100 צעדים לכיוון מעלה, כלומר, עם אותו ערך x כמו המיקום הנוכחי, וערך y הגדול ב-100 מערך y הנוכחי.

**תרגיל 14:** צרו עבור השוער ופלה את התסריטים המתאימים לתיאורים ושלבם בהם בינתיים את הוראות התנועה המתאימות לקפיצה למעלה, לסיבוב לאחור ולנפילה ארצה.

## שינוי אפקטים


את הוראות ההשחרה, שינוי הצבע וההבהרה נוכל לבצע על ידי שימוש באפקטים המוגדרים ב־Scratch. ישנן כמה הוראות המטפלות באפקטים גרפיים, ואלו נמצאות בקבוצת המראה הסגולה. לצרכינו מתאימה הוראה אחת, ההוראה  ל־0. זו הוראה מוחלטת. בקבוצה הסגולה יש גם הוראה יחסית  ב־1.


החלונית הראשונה בהוראת קביעת האפקט המוחלטת היא חלונית עם חץ, המאפשרת בחירת אפקט לקביעה מתוך רשימת אפקטים נתונה. בחלונית השנייה יש לשלב ערך מספרי, הקובע את מידת האפקט. למשל, עבור אפקט הבהירות, ככל שהמספר המשולב בחלונית השנייה קטן יותר, האפקט הופך את הדמות לכהה יותר. מספר גדול יותר יגרום לדמות להפוך לבהירה יותר. לכן מספרים חיוביים מבהירים ומספרים שליליים מכהים. שילוב המספר 0 בחלונית השנייה לא ייצור שום שינוי בבהירות הדמות. הערכים יכולים לנוע בין 100- ל־100 (ניתן לשלב גם ערכים גדולים יותר או קטנים יותר, אבל השפעתם לא תהיה גדולה יותר משל ערכי הקיצון 100 או 100-). עבור אפקטים אחרים הערכים המספריים משפיעים בצורה אחרת. נסו אפקטים שונים עם ערכים שונים והתרשמו מהשפעותיהם על הדמות.

את פעולות ההשחרה וההבהרה ניתן לבצע בעזרת אפקט הבהירות. את פעולת שינוי הצבע ניתן לבצע בעזרת אפקט הצבע.

**תרגיל 15:** השלימו עבור השוער ופלה את התסריטים המתאימים לתיאורים על ידי שילוב הוראות ההשחרה, הבהרה ושינוי צבע בעזרת שימוש באפקטים.

### הוראות חדשות ב־Scratch: קביעת אפקט ושינוי אפקט

ההוראה  ל־0 גורמת לקביעת האפקט המשולב בחלונית הראשונה על פי הערך המופיע בחלונית השנייה של ההוראה.

ההוראה  ב־1 גורמת לשינוי האפקט המשולב בחלונית הראשונה, בערך המופיע בחלונית השנייה של ההוראה.

**תרגיל 16 – חגבים עצבניים:** בכל הסעיפים בתרגיל זה הרגישו חופשיים להתנסות בערכים שונים עבור מיקום וכיוון התחלתיים, מספר הצעדים לתזוזה ותחום הבחירה האקראית.

א. צרו אנימציה בסיסית שבה משתתף חגב יחיד (תוכלו למצוא דמות מתאימה בתיקיית החיות, Animals). החגב יהיה חסר מנוחה למדיי וינוע בתזוזות עצבניות ימינה ושמאלה. לכן כדאי להגביל את תנועתו של החגב כך שיוכל לנוע רק ימינה ושמאלה, כך שלא יתהפך על ראשו עם שינוי כיוון תנועתו. כפי שהוזכר בפרק 5, תוכלו לעשות זאת על ידי בחירה בכפתור האמצעי מבין שלושת הכפתורים שנמצאים לשמאלה של הדמות, מעל חלון התסריטים שלה.

עם הלחיצה על הדגל הירוק יתייצב החגב במרכז הבמה כשהוא פונה ימינה, ויתחיל לנוע בתנועה אינסופית, ארבעה צעדים בכל פעם. אבל מדי ארבעה צעדים יחליט החגב אקראית אם לפנות לכיוון האחר או לא. לשם כך הוא יגריל מספר בין 1 ל-2 ויבדוק אם המספר שהוגרל שווה ל-1. אם הוא אכן שווה ל-1, הוא יפנה לכיוון ההפוך. כדי להקל על הצופים לעקוב אחר מיקומו של החגב ואחר ה"בחירות" שהוא עושה, הציגו על הבמה את מיקומו על ציר ה-x. תוכלו לעשות זאת על ידי סימון הריבוע הקטן שנמצא על יד הלבנה שמייצגת את מיקום הדמות על ציר ה-x, בקבוצת התנועה הכחולה.

שם הקובץ: grass1

ב. הוסיפו חגב נוסף שיתמקם כ-100 צעדים מימין למיקומו ההתחלתי של החגב הראשון, כשהוא פונה לכיוונו. החגב הנוסף לא יעשה דבר, אבל כאשר החגב הראשון פוגש תוך כדי תנועתו בחגב החדש, הוא יאמר "נעים לפגוש אותך".

שם הקובץ: grass2

ג. הוסיפו חגב שלישי כך שעם תחילת ביצוע האנימציה שני החגבים הנוספים ניצבים במרחקים שווים משני צדיו של החגב הראשון, כשהם פונים לכיוונו. כאשר החגב הראשון פוגש תוך כדי תנועתו באחד החגבים, הוא יאמר "נעים לפגוש אותך חגב2", או "נעים לפגוש אותך חגב3", בהתאם לחגב שפגש.

נסו להפעיל כמה פעמים את האנימציה שיצרתם. האם החגב הנע פוגש בשני החגבים האחרים במידה שווה?

שם הקובץ: grass3

ד. שנו את האנימציה שיצרתם בסעיף ג' כך שהחגב הנע לא יאמר דבר, ושני החגבים האחרים הם אלו שידברו כאשר הוא פוגש בהם. הם יציגו את עצמם ויאמרו כי נעים להם לפגוש בו.

שם הקובץ: grass4

ה. רק לשם הגיוון, שנו את האנימציה כך שבזמן האתחול ישנה כל אחד משני החגבים הנייחים את צבעו על ידי שימוש באפקט הצבע. השתמשו בערכים שונים עבור השינוי לכל אחד משני החגבים.

שם הקובץ: grass5

ו. כעת יוצבו סביב החגב הנע ארבעה חגבים נייחים שיפנו לכיוונו: אחד מימינו, אחד משמאלו, אחד מעליו ואחד מתחתיו. החגב שבמרכז יוכל לנוע בכל הכיוונים, ולאחר כל תזוזה הוא יבחר באופן אקראי באחד מהכיוונים על המעגל (כלומר, מ-180° עד 180°). כל אחד מארבעת החגבים הנייחים ידבר אם החגב הנע פוגש בו. ארבעת החגבים הנייחים ישנו את צבעם בזמן האתחול, כל אחד לצבע אחר.

שם הקובץ: grass6

## תרגיל 17 – פרחים חברותיים: בתרגיל זה משתתפים כמה פרחים מצבעים שונים. כל פרח שואף

להתקרב אל פרח אחר שצבעו זהה לשלו ולגעת בו.

שימו לב: הדמויות השונות באנימציה פועלות לפי תסריטים שדומים מאוד זה לזה. כדאי לכם להיעזר באופציית שכפול התסריטים שהוצגה בפרק 2 כדי ליצור את התסריטים עבור תרגיל זה.

א. השתמשו בפרויקט flowers-costumes כבסיס. בפרויקט זה מוגדרות שש דמויות של שישה פרחים: שניים אדומים, שניים צהובים ושניים סגולים. כמו כן מוגדר כפתור start. עם הלחיצה על הדגל הירוק ימקם כל פרח את עצמו בנקודה אקראית כלשהי על ציר ה־x (מ־200 ועד 200), כשהוא פונה ימינה. כאשר מתבצעת לחיצה על הכפתור start, כל פרח מתחיל לנוע בתזוזות קטנות, עד אשר הוא פוגש בפרח שצבעו זהה לשלו. כדי לזהות נגיעה בפרח בעל צבע זהה, תוכלו להשתמש בתנאי  שבקבוצת החיישנים, ולשלב בשתי החלוניות שלו את צבע הפרח המתאים. שימו לב: ייתכן שפרח יגיע למסגרת הבמה לפני שהצליח לפגוש בפרח מתאים. לכן יש לשלב הוראה לשינוי כיוון עם הנגיעה במסגרת הבמה.

שם הקובץ: flowers1

ב. שנו את האנימציה שיצרתם בסעיף א' כך שגם הכיוון ההתחלתי של כל פרח יהיה אקראי, ימינה או שמאלה.

שם הקובץ: flowers2

ג. צרו גרסה דו־ממדית של האנימציה שיצרתם בסעיף ב', שמאפשרת כי הכיוון ההתחלתי של כל פרח ייבחר אקראית מבין הערכים 180- עד 180.

שם הקובץ: flowers3

ד. שנו את האנימציה שיצרתם בסעיף ג' כך שגם ערך ה־y של מיקומו ההתחלתי של כל פרח ייבחר אקראית וינוע בין 150- ובין 150.

שם הקובץ: flowers4

ה. שנו את האנימציה שיצרתם בסעיף ד' כך שתנועתם של הפרחים תהיה אינסופית: בכל פעם שפרח פוגש בבן־זוגו, הוא יאמר "מצאתי אותך" וימתין מעט ולאחר מכן יקפוץ למיקום אקראי חדש, כשהוא פונה בכיוון אקראי, ויחזור לחפש את בן־זוגו.

שם הקובץ: flowers5



## סיכום

בפרק זה הכרנו מבנה בקרה חדש – **ביצוע מותנה** – המאפשר להתנות ביצוע הוראות מסוימות בקיום תנאי, ולהתנות ביצוע הוראות אחרות באי־קיום התנאי. הכרנו גם **ביצוע מותנה ללא חלופה**, שמתנה ביצוע הוראות מסוימות בקיום תנאי, ואם התנאי אינו מתקיים – דבר אינו מתבצע.

ניתן לשלב ביצועים מותנים, אלה בתוך אלה, ליצירת **קינון** של ביצועים מותנים.

ההוראות המתאימות של Scratch שייכות לקבוצת הוראות הבקרה, ההוראות הכתומות. ההוראה






המתאימה לביצוע מותנה היא . ההוראה המתאימה לביצוע חוזר מותנה ללא






חלופה היא .

בכל אחת מהוראות הביצוע המותנה יש לשלב תנאי. בפרק זה הכרנו שני תנאים נוספים ב־Scratch:

**תנאי נגיעה בצבע**  **ונוגע בצבע**  מקבוצת החיישנים הכחולה־בהירה, המאפשר לדמות לבדוק אם היא נוגעת בצבע מסוים.

**תנאי השוויון הכללי**  שבקבוצת המפעילים הירוקה־בהירה, המאפשר להשוות בין שני ערכים מספריים כלשהם. בחלוניות תנאי השוויון הכללי ניתן לשלב גם לבנת ערך, כמו לבנים המייצגות משתנים, לבנה המייצגת גודל או לבנה המייצגת תלבושת.

בשימושים רבים יש צורך בהגרלת **מספר אקראי**. ב־Scratch יש הוראה להגרלת מספר אקראי מתוך תחום נתון. זו ההוראה **בחר מספר אקראי בין**  שבקבוצת המפעילים הירוקה־בהירה. הלבנה המתאימה להוראה היא לבנת ערך, ואפשר לשלב אותה בכל הוראה עם חלונית מלבנית, עם פינות רגילות או מעוגלות.

דמות יכולה להסתיר את עצמה או לגרום לעצמה להופיע. ההוראות המתאימות הן **הסתר**  ו־**הצג**  מקבוצת המראה.

ב־Scratch ניתן ליצור **אפקטים גרפיים** מסוגים שונים. ההוראות המתאימות לכך הן ההוראה המוחלטת

**קבע אפקט**  וההוראה **החסית**  . בחלונית הראשונה ניתן לשלב

אפקט מתוך רשימת אפקטים נתונה, למשל צבע, בהירות ועוד. בחלונית השנייה משולב ערך מספרי הקובע את מידת האפקט עבור ההוראה המוחלטת, ואת מידת השינוי באפקט עבור ההוראה היחסית.

## פרק 8 מספרים

מספרים שולבו בפרויקטים שיצרנו החל מהדוגמה הראשונה. שילבנו מספרים בהוראות שונות, ובכך הפכנו אותן מהוראות כלליות להוראות ממוקדות: המספרים המשולבים בהוראות הביעו מספר צעדים לתזוזה, מספר זוויות לפנייה, מספר סיבובים בלולאה, מספר שניות להמתנה, מספר סידורי של תלבושת ועוד ועוד. כמו כן השתמשנו במשתנים שזכרו בתוכם ערכים מספריים. בפרק זה נעמיק את השימוש במספרים ונראה כיצד אפשר לשלבם בהוראות שונות באופנים נוספים.

### דוגמה 1: תפוזים לנסיך

בפרק זה ניצור אנימציה אינטראקטיבית שבה ניתן יהיה להשתמש, למשל, כמשחק המתרגל ילדים צעירים בחיבור של מספרים. באנימציה משתתף נסיך שמעוניין ב־12 תפוזים ומבקש אותם מהצופים. את התפוזים "מספקים" הצופים על ידי לחיצה על כפתורי המשחק, כפתור לכל כמות של תפוזים שניתן לספק במשחק: 2, 3, 4 או 5. כרגיל, נבנה את המשחק בשלבים. בשלב ראשון נתמקד בכפתורים השונים ובתפוזים. דמותו של הנסיך תצטרף מאוחר יותר.

### הגדרת הדמויות ובניית השלב הראשון של המשחק

#### משימה 1: הצגת תפוזים בהתאם לבחירה

צרו אנימציה המתנהלת באופן הבא:

על הבמה יהיו ארבעה כפתורים המסומנים במספרים 2, 3, 4 ו־5. לחיצה על כפתור מסוים תגרום לתפוזים שמספרם מתאים לכפתור, להופיע על הבמה. כך למשל, לחיצה על הכפתור 2 תגרום לשני תפוזים להופיע על הבמה, לחיצה נוספת על הכפתור 4 תגרום לארבעה תפוזים נוספים להופיע על הבמה וכן הלאה. התפוזים יופיעו על הבמה בערימה המורכבת משורות; כל לחיצה על כפתור תגרום למספר התפוזים המתאים להופיע בערימה בשורה חדשה.



שם הקובץ: `get-oranges`

מי יהיו הדמויות?

ברור כי יש ארבע דמויות לפחות, אחת לכל כפתור. אבל במשחק משתתפות ישויות נוספות – הלא הם התפוזים המופיעים על הבמה – אלא שמספרן אינו ידוע מראש, והוא תלוי בפעולות הצופים תוך כדי משחק. אם כך ברור שלא ניתן להגדיר דמות עבור כל תפוז שאמור להופיע על הבמה. אנו נראה מיד שמספיק לייצג את כולם על ידי דמות אחת של תפוז.

#### התפוז המשתכפל – דמות אחת ותמונות רבות

כיצד גורמים לכך שעל הבמה יופיעו תפוזים רבים, אשר מספרם תלוי בבחירות הצופים במהלך המשחק, כאשר יש דמות אחת בלבד של תפוז?

ב־Scratch דמות יכולה לשכפל את תמונתה על הבמה פעמים רבות על ידי שימוש בהוראת החתמה. הוראה זו "מחתימה" את תמונתה של הדמות במיקום שבו היא נמצאת ברגע ההחתמה, כך שגם אם לאחר מכן תשנה הדמות את מיקומה, העתק של תמונתה ימשיך להימצא במיקום שבו הייתה בעת החתמתה. ההוראה המתאימה היא ההוראה **חותמת**, והלבנה המתאימה לה נמצאת בקבוצת העט, הקבוצה הירוקה־כהה.

#### הוראה חדשה ב־Scratch: החתמה

ההוראה **חותמת** גורמת לדמות "להחתים" את עצמה על הבמה. כלומר, במיקום הנוכחי של הדמות מופיעה תמונה שלה, והיא נשארת שם גם אם הדמות זזה אחר כך למקום אחר.

התפוז הוא זה שישכפל את עצמו, ולכן הוראת ההחתמה תהיה משולבת בתסריטיו. אבל ההחתמה צריכה להתבצע בעקבות לחיצת הצופים על כפתורי המספרים, כלומר, בעקבות פעולה שמזוהה על ידי דמויות אחרות, דמויות הכפתורים. אם כך נדרשת כאן תקשורת בין דמויות: דמות של כפתור (2, 3, 4 או 5) תגיב ללחיצה ובעקבות כך תיצור קשר עם דמות התפוז כדי שתבצע החתמה. כמו בדוגמאות שנכללו בפרקים קודמים, את התקשורת בין הדמויות נבצע בעזרת מסרים.

אם כך נוכל כבר לתת תיאור כללי וחלקי של פעולת דמויות הכפתורים והתפוז. למשל, דמות כפתור המספר 3 צריכה לפעול על פי התיאור הבא:

0. אם לחיצה על דואג:

1. הודע לדואג הגפוס על הוספת 3 גפוסים חדשים

#### המסרים המועברים

דמות התפוז צריכה להגיב להודעות שהיא מקבלת. כאן יש לנו כמה אפשרויות:

אם אכן דמות הכפתור 3 תשלח בצעד 1 הודעה אחת שמשמעותה "יש 3 תפוזים חדשים", תצטרך דמות התפוז להגיב על ידי ביצוע סדרה של 3 החתמות. במקרה כזה תהיה לכל כפתור הודעה מיוחדת משלו (למשל, לחיצה על הכפתור 5 תגרום למשלוח הודעה "יש 5 תפוזים חדשים"), ולדמות התפוז נצטרך להכין תסריט מתאים לכל הודעה.

אפשרות אחרת היא להשתמש בהודעה אחת פשוטה יותר, שתתאים לכל הכפתורים. הודעה זו תדווח על תפוז חדש אחד. דמויות הכפתורים הן אלו שיצטרכו לדאוג למשלוח ההודעה במספר מתאים של פעמים. כך למשל, צעד 1 בתיאור שנתנו כעת עבור דמות הכפתור 3 הוא בעצם צעד מורכב שבו נשלחות שלוש הודעות. תגובתה של דמות התפוז להודעה זו תהיה פשוטה – ביצוע החתמה אחת, ולכן עליה לפעול על פי התיאור הבא:

0. אם קבלה הודעה על גפוס חדש:

1. קצץ המתנה

כפי שראינו, התיאורים שניתנו כאן הם כלליים מאוד. למשל, צעד 1 אצל דמות הכפתור 3 מייצג בעצם צעד מורכב של משלוח כמה הודעות. אבל התיאורים האלו הם גם חלקיים. הם אינם מטפלים כלל

בדרישה שכל סדרת תפוזים חדשה תמוקם בשורה חדשה, כל תפוז במקומו המתאים. כלומר, כל החתמה צריכה להתבצע במקום חדש: לחיצה על כפתור של מספר צריכה להביא לכך שהתפוז הבא יתווסף בשורה חדשה, וכל התפוזים הנוספים אחריו בעקבות אותה לחיצה יתווספו בהמשכה של אותה שורה.

### העברת מידע על מיקום ההחתמה באמצעות משתנים

דמות התפוז מקבלת הודעה זהה על כל תפוז חדש שצריך להתווסף, ולכן היא אינה יודעת אם מדובר בתפוז שצריך להתחיל שורה חדשה או בתפוז שצריך להופיע בהמשכה של שורה. לכן נוציא מאחוריותה את חישוב המקום הבא להחתמה. דמויות הכפתורים הן אלו שיחשבו את המיקום הבא לתפוז החדש, יעדכנו אותו עבור דמות התפוז, והיא תשתמש בערך שחושב.

אם כך, שוב נדרשת תקשורת בין הדמויות, בין אלו שמחשבות את המקום החדש לבין זו שמשתמשת בו. הפעם המנגנון המתאים לצורך התקשורת הוא זה של "תיבת הדואר" שהכרנו בפרק הקודם, כלומר, שימוש במשתנים.

נשתמש בשני משתנים לצורך התקשורת, משתנה בשם  $x$  לתיאור רכיב ה- $x$  של המיקום, ומשתנה בשם  $y$  לתיאור רכיב ה- $y$  של המיקום. דמות התפוז אינה צריכה לעדכן את המשתנים האלו, אלא רק להשתמש בהם. לכן את התיאור החלקי שנתנו קודם עבור התפוז נוכל להשלים כך:

0. אם קבאל הודעה אל גבוס חדש:

1. עבור לנקודה  $(x, y)$

2. בצע החממה

### התנהגות הכפתורים

כיצד יפעלו דמויות הכפתורים? למשל, את התיאור שנתנו קודם עבור דמות הכפתור 3 נוכל להשלים לתיאור הבא, שהוא עדיין תיאור כללי:

0. אם לחיצה על דמור:

1. עדכן את המשתנים  $x$  ו- $y$  כך שיצביעו לפני המקום הראשון בשורה חדשה

2. בצע 3 פעמים:

2.1 קדם את ערכו של המשתנה  $x$  במקום הבא בשורה

2.2. הודע אל גבוס חדש

שימו לב: צעד שליחת ההודעה בתיאור הקודם הפך עכשיו לצעד מורכב של ביצוע חוזר. בכל סיבוב בביצוע החוזר מתעדכן המקום בשורה, ונשלחת הודעה. צעד 1 בתיאור הנוכחי מתאר למעשה שתי פעולות: עדכון ערך המשתנה  $x$  ועדכון ערך המשתנה  $y$ .

**תרגיל 1:** חשבו: מדוע בצעד 1 ערכו של  $x$  נקבע כך שיצביע לפני המקום הראשון בשורה, ולא על המקום הראשון בשורה? (רמז: חשבו על הסדר של הצעדים 2.1 ו-2.2).

האם אפשר היה לפעול אחרת: לקבוע בצעד 1 את ערכו של  $x$  כך שיצביע על המקום הראשון בשורה? אם עניתם בחיוב, אילו שינויים נדרשים בעקבות שינוי זה?

## סידור שורות התפוזים

את שורות התפוזים נסדר באופן הבא: המרחק בין שורה לשורה יהיה 50 נקודות. המרחק בין תפוז לתפוז בשורה יהיה גם הוא 50 נקודות. התפוז הראשון בכל שורה ימוקם בנקודה 150-.


בתיאור פעולת הכפתור שניתנה כאן יש שתי פעולות קידום למקום הבא: ערכו של המשתנה  $x$  מקודם בצעד 2.1 למקום הבא בשורה, וערכו של המשתנה  $y$  מקודם בצעד 1 לשורה הבאה. לכן לצורך פעולות אלו מתאים להשתמש בהוראות יחסיות המוסיפות 50 לערכו של המשתנה המתאים. בצעד 1 יש גם פעולה של אתחול המשתנה  $x$  כך שיצביע לפני המקום הראשון בשורה. לשם כך מתאים להשתמש בהוראה מוחלטת המכניסה ערך למשתנה  $x$ .

**תרגיל 2:** מהו הערך שיש לשלב בהוראה המוחלטת לקביעת ערכו של המשתנה  $x$  (בצעד 1) כך שיצביע לפני המקום הראשון בשורה?


## בעיית תיאום בין התסריטים

שימו לב: התקשורת בין הדמויות מתבצעת בפרויקט זה בשני ערוצים מקבילים: דרך מסרים ודרך תיבות דואר. כדי שהתקשורת תעבוד נכון, עליה להתבצע בזהירות. בפרק 10 נתמקד בקשיים שנובעים מהעבודה במקביל של דמויות שונות (ולעתים גם של תסריטים שונים של אותה דמות) ונעמיק בהם. כאן נראה בעיה אחת שעלולה להתעורר עקב ריבוי הדמויות וריבוי ערוצי התקשורת:

בצעד 2 דמות הכפתור שולחת 3 הודעות, כל אחת מהן צריכה לגרום להופעת תפוז חדש אחד על הבמה. ייתכן שדמות הכפתור תשלח מסר של תפוז חדש, אך לפני שדמות התפוז תספיק לקבל את המסר ולנהוג לפיו, כלומר, להשתמש בערכי ה- $x$  וה- $y$  שהוכנו עבורה, דמות הכפתור כבר תתקדם הלאה בתסריטה ותקדם את ערכו של המשתנה  $x$  למקום הבא בשורה, לפני שדמות התפוז עשתה שימוש בערכו הנוכחי.

כדי למנוע זאת נדאג לעכב מעט את התקדמותה של דמות הכפתור. אחרי שליחת המסר היא תמתין לקבלתו על ידי דמות התפוז, ורק אחרי שהמסר התקבל וטופל, היא תמשיך הלאה. לשם כך ניתן להשתמש בהוראת שליחת מסר משולבת בהמתנה, **שדר מסר**  **וחנה**, הנמצאת בקבוצת הבקרה הכתומה.

### הוראה חדשה ב-Scratch: שידור מסר עם המתנה

ההוראה **שדר מסר**  **וחנה** גורמת לשידור המסר המופיע בחלונית. לאחר שידור המסר ביצוע התסריט נעצר, והוא מתחדש רק לאחר שהמסר מטופל על ידי דמות אחרת.

**תרגיל 3:** פתחו את הפרויקט costumes, שנבנה עבור פרק זה ובו מוגדרות דמויות התפוז והכפתורים. הגדירו משתנים מתאימים ל- $x$  ו- $y$  (אין צורך להציגם על הבמה) והכינו תסריטים מתאימים לתיאורים שניתנו כאן עבור דמות התפוז ודמות הכפתור 3. הכינו עבור הכפתורים האחרים תסריטים דומים לזה שהכנתם עבור הכפתור 3.

## ומה עם אתחולים?

עדיין לא השלמנו את הטיפול בהופעת התפוזים על הבמה. מה חסר?

כתמיד, עלינו לבדוק את הצורך באתחולים, שצריכים להתבצע עם ההפעלה, כלומר, עם הלחיצה על הדגל הירוק. ואכן, גם במקרה שלנו יש כמה אתחולים שצריכים להתבצע:

כפתורי המספרים אינם זזים במהלך המשחק, אבל התפוז זז עם כל פעולת החתמה. לכן בתחילת המשחק נרצה שיימחקו כל החותמות מההפעלה הקודמת, ושהתפוז ימקם את עצמו בנקודה התחלתית ובה ימתין לתחילת המשחק. נקודה זו תהיה הפינה הימנית העליונה של הבמה, הנקודה (200, 150). אם כך נצטרך להכין עבור התפוז תסריט אתחול שיפעל עם הלחיצה על הדגל הירוק, על פי התיאור הבא:

0. **אם לחיצה על הדגל הירוק:**

1. **נקה את כל מוטמוג'י על הבמה**

2. **עבור לנקודה (200, 150)**

בנוסף לאתחול המתייחס למופעי של התפוז, יש לבדוק גם את הצורך באתחול משתני המיקום,  $x$  ו- $y$ . אין צורך באתחול ערכו של  $x$  עם הלחיצה על הדגל הירוק, משום שבתסריטי דמויות הכפתורים מתבצע עם כל לחיצה אתחול מחודש של ערך המשתנה  $x$ , ואין שום התייחסות לערכו לפני שנלחץ אחד הכפתורים.

אבל שימו לב כי בתסריטי דמויות הכפתורים כל ההוראות המתייחסות לערכו של  $y$  הן הוראות יחסיות המקדמות את ערכו, ואין שום הוראה מוחלטת הקובעת לו ערך. משום כך ערכו של  $y$  חייב להיות מאותחל בתחילת ביצוע הפרויקט, כלומר, עם הלחיצה על הדגל הירוק, כך שיצביע על השורה הראשונה. מאחר שאתחול המשתנה  $y$  צריך להתבצע רק פעם אחת, עם הלחיצה על הדגל הירוק, ומאחר שלתפוז כבר יש תסריט אתחול משלו, נוסיף לתסריט זה גם את הטיפול באתחול המשתנה  $y$ , על פי התיאור הבא:

0. **אם לחיצה על הדגל הירוק:**

1. **נקה את כל מוטמוג'י על הבמה**

2. **אמח'א את ערך משנה  $y$  לשורה הראשונה (-200)**

3. **עבור לנקודה (200, 150)**

לפני שנוכל לבנות תסריט המתאים לתיאור זה, עוד עלינו ללמוד כיצד לתרגם להוראות של Scratch את צעד 1. הוראה מתאימה לצעד 1 נמצאת בקבוצת העט הירוקה-כהה, והיא ההוראה **נקח**.

### הוראה חדשה ב-Scratch: מחיקת חותמות

ההוראה **נקח** גורמת לכך שכל המופעים של כל הדמויות אשר נוצרו על ידי פעולות החתמה, נמחקים מהבמה.

**תרגיל 4:** הכינו תסריט אתחול עבור דמות התפוז על פי התיאור שניתן כאן. בדקו את הפרויקט שבניתם, תעדו ושמרו בשם חדש.

## מניית מספר התפוזים הכולל

על פי תיאור המשחק, כפי שניתן קודם לכן, בכל רגע ורגע במשחק יש לדעת כמה תפוזים נצברו, כדי שהנסיך (שעדיין לא נוצר) יוכל לדווח לצופים אם עדיין חסרים לו תפוזים. בתסריטים שבנינו עד כה נקודת המבט היא מקומית: דמות התפוז מתייחסת בכל פעם להופעה בודדת של תפוז חדש, ודמויות הכפתורים מתייחסות בכל פעם ללחיצה בודדת על הכפתור המתאים להן.

## משימה 2: שמור את מספר התפוזים

שמור והצג את מספר התפוזים הנצבר לאורך כל המשחק, לאחר הלחיצה על הדגל  
שם הקובץ: `store-count` הירוק.



לצורך פתרון משימה 2 נידרש למשתנה נוסף, שימנה את מספר התפוזים הנספרים. בכל פעם שיילחץ אחד מכפתורי המשחק, ובעקבות כך יתווספו תפוזים למסך, צריך יהיה גם לעדכן את ערכו של המשתנה החדש.

## האחריות על מניית התפוזים

אילו דמויות יהיו אחראיות לעדכון ערכו של המשתנה החדש? ישנן כמה אפשרויות:

אפשר לבצע עדכון נפרד על כל תפוז שנוסף, כלומר, להוסיף למשתנה בכל פעם 1. עדכון כזה יכול להתבצע על ידי דמות התפוז, כאשר מתקבל מסר של תפוז חדש, או על ידי דמות של כפתור בכל פעם שהדמות שולחת מסר של תפוז חדש.

אפשר לבצע את העדכון ברמה גבוהה יותר, של קבוצת תפוזים שנוספת ולא של תפוז בודד. על פי גישה זאת, בכל פעם שכפתור נלחץ, תעדכן דמותו את ערך המשתנה החדש במספר המתאים לה. כך למשל, דמות הכפתור 2 תוסיף 2 לערכו של המשתנה, ודמות הכפתור 4 תוסיף 4 לערכו של המשתנה. נבחר באפשרות האחרונה.

בנוסף להוראות העדכון תוך כדי המשחק, חשוב מאוד לאתחל את ערכו של המשתנה החדש. הוראות העדכון הן יחסיות, הן מגדילות את ערכו הנוכחי של המשתנה ב-2, 3, 4 או 5. אם לא נדאג לשים בהתחלה ערך נכון במשתנה, גם הערכים הבאים שלו יהיו שגויים. בתחילת המשחק לא נצברו עדיין תפוזים, ולכן יש לאתחל את המשתנה החדש ל-0 על ידי הוראה מוחלטת. את הוראת האתחול של המשתנה מתאים למקם בתסריט האתחול שבאחריותה של דמות התפוז.

**תרגיל 5:** שלבו בפרויקט את המשתנה החדש: הגדירו אותו (דאגו להציג את ערכו על הבמה, אך אין לאפשר לצופים לשנות את ערכו!), הוסיפו הוראת אתחול מתאימה לתסריט האתחול שנמצא אצל דמות התפוז והוסיפו לתסריטים של דמויות הכפתורים הוראות עדכון מתאימות. עדכנו את תיעוד הפרויקט ושמרו אותו.

## משתנה כצובר

למשתנה כזה הצובר ערכים נוהגים לקרוא **צובר**. הוא שימושי מאוד בהקשרים שונים (למשל, כדי לספור מספר נקודות צבורות לשחקנים במשחק). מאפייני השימוש בו בהקשרים השונים יהיו דומים תמיד: יש לאתחל את ערכו בתחילת הביצוע (בדרך כלל ל-0) על ידי הוראה מוחלטת ולעדכן את ערכו בערך מתאים על ידי הוראה יחסית.

### מושג חדש: צובר

צובר הוא משתנה שנועד לסכום ערכים מצטברים. תבנית השימוש בו מורכבת מאתחול לערך 0 ומביצוע חוזר של צבירת ערכים, כלומר, של הגדלת ערכו של המשתנה בערך הנצבר התורן.

סוג מיוחד של משתנה צובר הוא כזה שערכו מתעדכן תמיד ב-1, ולא בערכים אחרים. למשל, אם נרצה למנות את מספר הפעמים שבהן נלחץ כפתור מסוים במשחק, נוסיף למשתנה שהוגדר למטרה זו 1 בכל פעם שהכפתור נלחץ. משתנה כזה נקרא **מונה**, וגם הוא שימושי מאוד בהקשרים שונים. מאפייני השימוש בו דומים אף הם: אתחול ל-0 עם תחילת הביצוע ופעולות עדכון יחסיות שכל אחת מהן מגדילה את ערכו ב-1.

### מושג חדש: מונה

מונה הוא משתנה שנועד למנות ערכים. תבנית השימוש בו מורכבת מאתחול לערך 0 ומביצוע חוזר של ספירה, כלומר, של הגדלת ערכו של המשתנה ב-1.

## הנסיך מגיע: כמה תפוזים יש?

כעת נטפל בהוספת דמותו של הנסיך, כדי להשלים את הגדרת המשחק כפי שתואר בתחילת הפרק.

### משימה 3: הנסיך מבקש תפוזים

הנסיך מבקש מהצופים 12 תפוזים. בכל פעם שהצופים מספקים לו עוד תפוזים, הנסיך מודיע אם יש בידיו כמות התפוזים הדרושה, אם עדיין חסרים לו תפוזים, או שמא יש לו כבר יותר מדי תפוזים.



שם הקובץ: [prince-says-when-enough-oranges](#)

להתנהגותו של הנסיך שני מרכיבים, ואנו נצטרך להתאים תסריט לכל מרכיב. התסריט הראשון הוא תסריט פשוט שיופעל עם תחילת הביצוע ויתאים לתיאור הבא:

0. אם אחיזה על הדגל הידוק:

1. אמור "גן אי בקשה 12 גפוזים" לפניך 2 שניות

### הנסיך בודק: האם כבר יש מספיק תפוזים?

בתסריט השני צריך להשוות את מספר התפוזים הצבור למספר הדרוש, 12, ולפעול בכל פעם שמתווספים תפוזים חדשים, כלומר, בכל פעם שנלחץ כפתור משחק.

שוב נדרשת תקשורת בין דמויות: דמויות הכפתורים צריכות להודיע לדמות הנסיך על תוספת קבוצת



תפוזים. לצורך התקשורת הזאת נשתמש במסרים. נוסף לנסיך תסריט שיפעל עם קבלת מסר מתאים (למשל, "קבוצה חדשה"), על פי התיאור הבא:

0. אם קבאל הודעה על קבוצה גפופים חדשה:

1. אם מספר הגפופים הצבור קטן מ-12

1.1. אחור "אני צריך בקשה עוד גפופים" אלפך 2 שניג

2. אחור

2.1. אם מספר הגפופים הצבור גדול מ-12

2.1.1. אחור "יש לי יותר מדוי גפופים" אלפך 2 שניג



2.2. אחור

2.2.1. אחור "גודה על 12 הגפופים"


## השוואת מספרים


כיצד נתרגם את התיאורים האלו לתסריט Scratch? בידכם כל הכלים לביצוע תרגום כזה, פרט לתרגום התנאים שבתסריט השני, בצעדים 1 ו-2.1. בשני הצעדים האלו דרוש לנו תנאי המשווה בין שני ערכים, ובודק אם האחד מהם גדול מהאחר.

התנאים הדרושים מזכירים את תנאי השוויון הכללי שהכרנו בפרק הקודם, אך התנאי ההוא, כפי שמעיד שמו, בדק שוויון, וכאן אנו מעוניינים לבדוק אי־שוויון. במקרה הראשון (צעד 1) הבדיקה שלנו תצליח אם ערך אחד (מספר התפוזים הצבור) קטן מהערך האחר (12). במקרה השני (צעד 2.1) הבדיקה שלנו תצליח אם ערך אחד (מספר התפוזים הצבור) גדול מהערך האחר (12).

בדומה לתנאי השוויון הכללי, גם את התנאים הדרושים לנו כעת נוכל למצוא בקבוצת המפעילים, הירוקה־בהירה. בנוסף לתנאי השוויון, נמצאים בקבוצה זו גם התנאים  (תנאי אי־שוויון "קטן") ו-  (תנאי אי־שוויון "גדול"). הלבנים המתאימות להם, כמו לבנת תנאי השוויון, הן לבנים שפינותיהן קטומות, ומשולבות בהן חלונות מלבניים. בחלונות אלו נוכל לשלב ערך מספרי (12) ולבנה מייצגת של משתנה (המשתנה הצובר).

### תנאי חדש ב־Scratch: תנאי אי־שוויון

התנאי  מתקיים אם הערך המופיע בחלונת השמאלית קטן מהערך המופיע בחלונת הימנית.

התנאי  מתקיים אם הערך המופיע בחלונת השמאלית גדול מהערך המופיע בחלונת הימנית.

## תרגיל 6:

בתסריט המתאים לנסיך תוכלו להשתמש בשני תנאי האי־שוויון שהכרנו כעת, תנאי אי־שוויון "גדול" ותנאי אי־שוויון "קטן". האם יש הכרח להשתמש בשניהם בתסריט, או ניתן להסתפק בתנאי אחד מבין השניים, שישמש גם בהוראה שמתאימה לצעד 1 וגם בהוראה שמתאימה לצעד 2.1? אם לדעתכם ניתן להסתפק בתנאי אחד, הראו מה תשלבו בשתי החלונות בכל אחד משני השימושים.

## תרגיל 7:

השלימו את המשחק: הגדירו דמות עבור הנסיך (דמות מתאימה תוכלו לייבא מתיקיית האנשים). שלבו בתסריטי דמויות הכפתורים הוראת שליחת מסר של קבוצה חדשה וכתבו עבור הנסיך שני תסריטים לפי התיאורים שניתנו כאן. תעדו את התוספות, בדקו את הפרויקט ושמרו אותו.

## דוגמה 2: שינוי כללי המשחק – כפתור אחד, אבל מפתיע

נשנה מעט את המשחק שיצרנו.

### משימה 4: כמה תפוזים נקבל הפעם?

יש לשנות את האנימציה באופן הבא:

כמקודם, הנסיך מבקש מהצופים 12 תפוזים. אבל במקום ארבעה כפתורי מספרים, יהיה רק כפתור אחד, וכל לחיצה עליו תגרום להוספת תפוזים במספר אקראי מבין המספרים 2, 3, 4 או 5.

שם הקובץ: `random-number-of-oranges`



אם כן, גם בגרסה זו יבקש הנסיך 12 תפוזים, אבל כעת לא יבחרו הצופים בכל פעם כמה תפוזים יתנו – שניים, שלושה, ארבעה או חמישה – על ידי לחיצה על הכפתור המתאים. במקום ארבעת הכפתורים יהיה רק כפתור אחד, וכל לחיצה עליו תגרום להוספת תפוזים במספר כלשהו מבין המספרים 2, 3, 4 או 5. הצופים לא יוכלו לדעת מראש כמה תפוזים יתווספו בעקבות לחיצה על הכפתור. הבחירה במספר התפוזים להוספה תהיה בחירה אקראית.

המשמעות היא שלכל אחד מהמספרים סיכוי שווה להיבחר בכל לחיצה. הדבר דומה להטלת קובייה: בכל פעם שאנו מטילים קובייה שעל פאותיה המספרים 1 עד 6, אנו בעצם מבצעים בחירה אקראית (במילים אחרות, מגרילים) בין ששת המספרים, ולכל אחד מששת המספרים סיכוי שווה להיבחר. במקרה שלנו אין מדובר בקובייה, משום שאנו מגרילים בכל פעם בין ארבעה מספרים ולא בין שישה. אלה מכם שמשחקים במשחקי מבוכים ודרקונים ודאי נתקלו ב"קובייה" מתאימה להגרלה שלנו: זוהי פירמידה משולשת אשר לה ארבע פאות.

### התנהגות הכפתור היחיד

במשחק החדש יהיו כמקודם דמויות הנסיך והתפוז. אבל במקום ארבע דמויות של כפתורים, נזדקק רק לדמות אחת של כפתור שעליו לא יצוין מספר אלא תווית, כמו "עוד".

לארבעת הכפתורים שהיו לנו הייתה התנהגות דומה מאוד, ארבעת תסריטיהם היו כמעט זהים ונבדלו רק בערך המספרי ששולב בהם (2, 3, 4, או 5). גם לכפתור החדש תהיה התנהגות דומה מאוד, שתענה לתיאור הבא:

0. אם אחיזה על דמוגר:
1. עדכן את המשגנים  $x$  ו- $y$  כך שיציגו לפני המקום הראשון השורה חדשה
2. בחר אקראי מספר בין 2 ל-5
3. בצע את ההזזות האלו מספר פעמים השווה למספר שהגדל:
- 3.1 קדם את ערכו של המשגנה  $x$  במקום הבא השורה
- 3.2 הודע על גפוז חדש
4. הוסף למשגנה הצובר את המספר שהגדל
5. הודע על קבוצה חדשה

נוכל לכתוב ב־Scratch הוראה המתאימה לצעד 2 על ידי שימוש בהוראה **בחר מספר אקראי בין 0 ל-5** מקבוצת המפעילים הירוקה־בהירה, שנלמדה בדוגמה 5 בפרק 7.

עלינו לשמור את הערך שהוגרל, כדי שנוכל להשתמש בו בצעד 4. את שמירת הערך נוכל לבצע בעזרת משתנה. נגדיר משתנה חדש, עבור המספר המוגרל. בעזרת הוראה מוחלטת נציב למשתנה את הערך שהוגרל. ההוראה המתאימה לשמירה היא ההוראה המוכרת לנו מקבוצת המשתנים **קבע ל-0**, ובה נשלב את לבנת הערך הירוקה־בהירה המתאימה לבחירה האקראית.

**תרגיל 8:** הגדירו את הדמות החדשה, דמות כפתור ההוספה, וצרו עבורה תסריט בהתאם לתיאור שניתן כאן. לשם כך תצטרכו להגדיר גם משתנה חדש עבור הערך המוגרל (שערכו יוצג על הבמה), שניתן לקרוא לו, למשל, how many.

את תסריטה של הדמות החדשה תוכלו ליצור בקלות על ידי העתקת התסריט של אחת מדמויות הכפתורים הקודמות, טרם מחיקתן, והכנסת השינויים המתאימים. תעדו את השינויים בפרויקט ושמרו אותו בשם חדש.

האם בעקבות שינוי המשחק יש צורך להכניס שינויים בתסריטיהן של דמויות התפוז והנסיך?



גם כעת לדמות הכפתור אחריות רבה: היא מחשבת את המקום המתאים ל"הנחתו" של כל תפוז חדש, מודיעה על כל תפוז חדש ועל כל קבוצה חדשה של תפוזים ומעדכנת את המשתנה הצובר. לדמות הנסיך ולדמות התפוז אחריות פחותה ונקודת מבט צרה יותר: דמות התפוז מגיבה לתוספת תפוז בודד, ודמות הנסיך מגיבה להוספת קבוצת תפוזים ומתייחסת לערך המשתנה הצובר, שעודכן על ידי דמות הכפתור. לכן השינוי שערכנו כעת אינו משפיע על התנהגותם של הנסיך והתפוז. עם זאת, יש להוסיף אתחול למשתנה החדש וכדאי למקמו בתסריט האתחול שנמצא אצל דמות התפוז.

## הנסיך מתגמש: גם קצת יותר מ־12, זה בסדר

נכניס עוד שינוי קטן במשחק. זאת לאור גורם האקראיות שנוסף למשחק ושמקטין את ההסתברות שהנסיך יקבל 12 תפוזים בדיוק.

## משימה 5: "לא כל כך משנה" לנסיך כמה תפוזים יקבל

יש לשנות את האנימציה כך שהנסיך יבקש בין 12 ל־14 תפוזים. קבלת 12, 13, או 14 תפוזים תספק את הנסיך.

שם הקובץ: prince-says-how-many-oranges



**תרגיל 9:** מי הן הדמויות שאמורות להיות מושפעות מהשינוי שתואר?

נצטרך לשנות רק את תסריטיו של הנסיך, שיפעלו כעת על פי התיאורים הבאים:

0. אם אחיזה על הדגל הידוק:

1. אחור "גן אי בקשה בין 12 ל־14 גפוזים" אלמשך 2 שניות

0. אם קבאל הודעה על קבוצת גפוזים חדשה:

1. אם מספר הגפוזים הצבור קטן מ־12

1.1. אחור "אני צריך בקשה עוד גפוזים" אלמשך 2 שניות

2. אחור

2.1. אם מספר הגפוזים הצבור גדול מ־14

2.1.1. אחור "יש לי יותר מדוי גפוזים" אלמשך 2 שניות

2.2. אחור

2.2.1. אחור גודעה על הגפוזים שקיבל

## כיצד אומרים תודה באופן ברור ומנומס?


השינויים שיש לערוך הם מקומיים וקלים לביצוע. אך בתסריט השני נרצה שמשפט התודה שאומר הנסיך בצעד 3.1 יהיה משמעותי ויכלול כמקודם גם את מספר התפוזים שיתקבל.

בגרסתו הקודמת של המשחק קל היה ליצור משפט תודה משמעותי, משום שמשפט כזה נאמר תמיד בעקבות קבלה של 12 תפוזים בדיוק, וניתן היה לנסחו מראש כך שיכלול את המספר 12. בגרסה הנוכחית ייתכן שנצטרך לשלב במשפט את המספר 12, 13 או 14, ואין לנו דרך לדעת זאת מראש כאשר אנו כותבים את התסריט ואת המשפט המשולב בהוראת הדיבור.



ניתן, כמובן, לפתור זאת אם נבדוק בעזרת הוראות לביצוע מותנה מה מספר התפוזים המדויק שהתקבל, ונכין הוראת דיבור מתאימה לכל אפשרות. אלא שהתסריט המתקבל יהיה מסורבל למדי. בנוסף, גישה זו לא תהיה מעשית אם הטווח יגדל, כמו במקרה שבו יסכים הנסיך לקבל בין 12 ל־30 תפוזים.

נוכל לפתור זאת בדרך קלה יותר על ידי יצירת משפט מורכב שחלקו הוא סדרת מילים ידועה מראש, וחלקו הוא ערך שייוודע רק תוך כדי הביצוע.

## צירוף שתי סדרות סימנים לסדרה אחת

ב־Scratch ניתן ליצור משפט מורכב כזה בעזרת ההוראה  שבקבוצת המפעילים הירוקה־בהירה. הוראה זו מצמידה או מצרפת את שני החלקים הכלולים בחלוניות ויוצרת מהם משפט אחד. נבחן את צורתה של הלבנה המתאימה, וכך נבין את ההקשרים שבהם ניתן להשתמש בהוראה זו.

צורתן של החלוניות המשולבות בהוראה היא מלבנית, בדומה לחלוניות תנאי השוויון ותנאי האי־שוויון. לכן נוכל לכתוב בחלוניות ערכים או מילים, ונוכל גם לשלב בהן כל לבנה מלבנית – לבנה שאין בה בליטות או שקעים – בין שפינותיה מעוגלות ובין שקטומות. הערכים המשולבים מוצמדים זה לזה ליצירת רצף של סימנים.

כך למשל, אם נשתמש בהוראת ההצמדה כך: , התוצאה המתקבלת היא הרצף יוסי2. אם נשתמש בהוראת ההצמדה כך: , התוצאה המתקבלת תהיה הרצף 54, שהוא בעצם המספר 54. סדר הצירוף הוא משמאל לימין (כלומר, תוכן החלונית הימנית מוצמד אל תוכן החלונית השמאלית). אם אחת הסדרות מכילה אות עברית, תוצמד הסדרה הימנית משמאל לסדרה השמאלית. אם אין אות עברית באף אחת מהסדרות, תוצמד הסדרה הימנית מימין לסדרה השמאלית.

למעשה, היינו מעוניינים ליצור משפט המורכב משלושה חלקים: סדרת המילים "תודה על", מספר התפוזים והמילה "התפוזים". כך יתקבל, למשל, המשפט "תודה על 13 התפוזים", בדומה למשפט התודה שנאמר בגרסה הקודמת של המשחק. אבל להוראת ההצמדה שתי חלוניות בלבד, ואלה מאפשרות הצמדה של שני חלקים. לכן נתחיל ביצירת משפט בעל ניסוח מוצלח פחות, למשל, "תודה על 13", ונראה מיד כיצד ליצור את המשפט המלא.


כדי ליצור את המשפט החלקי, עלינו לשלב בין סדרת המילים "תודה על" לבין מספר, שערכו שמור במשתנה הצובר. ובכן, נשלב בחלונית השמאלית את המילים תודה על (עם רווח לאחר המילה "על" כדי שמספר התפוזים לא יהיה צמוד למילה), ובחלונית הימנית נשלב את הלבנה המייצגת של המשתנה הצובר את מספר התפוזים.

הכינו את הלבנה המתאימה להוראת ההצמדה בחלון התסריטים של הנסיך (מחוץ לשני תסריטיו). כדי לבדוק אם מילאתם את החלוניות בצורה נכונה, לחצו עליה. לצדה תופיע תוצאת פעולת ההצמדה.

כיצד נשלב את לבנת ההצמדה שהכנתם כעת בתסריטיו של הנסיך? צורתה של הלבנה היא מלבנית עם פינות מעוגלות. לכן ניתן לשלבה בכל חלונית מלבנית שפינותיה ישרות או מעוגלות, בפרט בתוך הוראת דיבור. נשנה בתסריטו השני של הנסיך את הוראת הדיבור המתאימה לצעד 3.1 על ידי כך שנשלב בחלונית את לבנת ההצמדה שיצרנו.

נסו לבצע זאת ובדקו את התוצאה.

### הוראה חדשה ב־Scratch: צירוף סדרות סימנים

ההוראה  גורמת לדמות לצרף את סדרת הסימנים שמופיעה בחלונית הימנית אל סדרת הסימנים שמופיעה בחלונית השמאלית. אם אחת הסדרות כוללת אות עברית, היא תיקרא מימין לשמאל, והסדרה המצורפת תיכתב מימין לשמאל.

## צירוף של כמה סדרות סימנים לסדרה אחת

נפנה כעת לשיפור ניסוח המשפט שאומר הנסיך.

כאמור, ברצוננו לשלב שלושה חלקים, ואילו הוראת ההצמדה מאפשרת לנו לחבר רק שניים. לכן נבצע זאת בהדרגה: ראשית נצמיד את החלקים "תודה לך" ומספר התפוזים. לאחר מכן נצמיד את המשפט שנוצר אל החלק השלישי, אל המילה "התפוזים". ניתן לעשות זאת משום שבחלוניות של לבנת ההצמדה ניתן לשלב כל לבנה מלבנית, בפרט לבנת הצמדה שכבר מצמידה שני חלקים לאחד.

הדבר דומה לביצוע פעולת סכום מספרים, אם מותר לנו לסכם בכל פעם רק שני מספרים. במקום לבצע  $(8+3)+2$  נבצע  $((8+3)+2)$ . אנו משתמשים בפעולת הסיכום פעמיים, כדי לסכם שלושה מספרים, ובהוראת ההצמדה פעמיים, כדי להצמיד שלושה חלקים. באופן דומה נוכל להצמיד גם מספר גדול יותר של חלקים על ידי שימוש במספר מתאים של הוראות הצמדה.

**תרגיל 10:** צרו את לבנת ההצמדה החדשה: גררו לבנת הצמדה ריקה מקבוצת המפעילים. בחלונית הראשונה (השמאלית) של הלבנה הריקה שלבו את לבנת ההצמדה שיצרתם קודם, ובחלונית השנייה כתבו את המילה "התפוזים". שלבו את הלבנה החדשה בהוראת הדיבור המתאימה לצעד 3.1. נסו להפעיל את הפרויקט המתקבל.

שם הקובץ: prince-says-with-join

האם ניתן ליצור את לבנת ההצמדה גם בדרך אחרת, בסדר אחר? הציעו דרך נוספת ובדקו גם אותה. תעזו את הפרויקט ושמרו אותו בשם חדש.

## הנסיך מתעייף: "לא צריך לדבר כל כך הרבה..."

הנסיך אמנם הגמיש את דרישותיו, אבל כעת הוא מחליט שהתשובות שהוא נותן לצופים הן מפורטות מדי.

### משימה 6: הצופים מנחשים אם יש יותר מדיי או פחות מדיי תפוזים

יש לשנות את האנימציה באופן הבא: בעקבות כל לחיצה המוסיפה תפוזים, יאמר הנסיך לצופים אחת מהשתיים: או יודה על התפוזים שקיבל או יאמר כי אין בידיו מספר התפוזים הרצוי לו.

שם הקובץ: prince-says-with-compound-condition



שימו לב: משמעות ההודעה הראשונה היא שהנסיך קיבל בין 12 ל-14 תפוזים. ההודעה השנייה היא דו-משמעית: ייתכן שיש פחות מ-12 תפוזים, וייתכן שיש כבר יותר מ-14. הצופים יצטרכו להבין זאת בעצמם לפי מספר התפוזים שנצבר.

## תנאים מורכבים

החלטתנו של הנסיך מאפשרת לנו לפשט את התסריט שלפיו הוא פועל. בגרסה הקודמת היה עלינו להכין משפט דיבור מתאים לכל אחת מהאפשרויות: פחות מ-12, יותר מ-14, בין 12 ל-14. בהתאם לכך בנינו ביצוע מותנה מקונן לפי המתווה הבא:

**אם מספר הגפוזים הצבור קטן מ-12**

...

**אחרת**

**אם מספר הגפוזים הצבור גדול מ-14**

...

**אחרת ...**

כעת עלינו להתעניין רק בשתי אפשרויות: בין 12 ל-14 או לא בין 12 ל-14. לכן מספיק לבנות ביצוע מותנה לא מקונן לפי המתווה הבא:

**אם מספר הגפוזים הצבור הוא בין 12 ל-14**

...

**אחרת ...**

לפיכך זהו התיאור שלפיו הנסיך צריך לפעול:

0. אם קבאל הודעה על קבוצת גפוזים חדשה:

1. אם מספר הגפוזים הצבור הוא בין 12 ל-14

1.1. אמור גודל על מספר הגפוזים שקיבל

2. אחרת


2.1. אמור "אין לי מספר הגפוזים הצבור" אלמשך 2 שניות

זהו תיאור קצר יותר וממוקד יותר. אבל כדי שנוכל ליישמו, עלינו ללמוד כיצד לתאר תנאי המתאים לצעד 1.

המשמעות של "בין 12 ל-14" היא בעצם: "גדול מ-11 וקטן מ-15". לכן התנאי הדרוש לנו הוא תנאי מורכב שבדוק קיום של שני תנאי אי-שוויון: מספר התפוזים הצבור גדול מ-11 וגם מספר התפוזים הצבור קטן מ-15. כדי לבדוק אם התנאי המורכב מתקיים, יש לבדוק כל אחד מתנאים המרכיבים אותו. למשל, אם מספר התפוזים הצבור הוא 13, התנאי המורכב מתקיים, כי 13 גדול מ-11, וגם 13 קטן מ-15. לעומת זאת התנאי אינו מתקיים עבור 9: אמנם תנאי האי-שוויון השני מתקיים, כי 9 קטן מ-15, אבל תנאי האי-שוויון הראשון אינו מתקיים, כי 9 אינו גדול מ-11. מאחר שרק אחד מתנאי האי-שוויון מתקיים, התנאי המורכב אינו מתקיים.

**תרגיל 11:** האם התנאי המורכב שתיארנו מתקיים עבור 17? הסבירו את תשובתכם.

**תרגיל 12:** האם קיים מספר שעבורו שני תנאי האי-שוויון אינם מתקיימים? אם עניתם בחיוב, הראו מספר כזה כדוגמה. אם עניתם בשלילה, הסבירו מדוע לא קיים מספר כזה.


ב־Scratch ניתן ליצור תנאי מורכב כזה בעזרת הפעולה "וגם" שמרכיבה שני תנאים ויוצרת מהם תנאי חדש. התנאי החדש מתקיים אך ורק כאשר שני התנאים המרכיבים אותו מתקיימים. פעולה זו שייכת גם היא לקבוצת המפעילים הירוקה־בהירה, והלבנה המתאימה לה היא . שימו לב לצורתה של הלבנה: זו לבנה מלבנית שפינותיה קטומות, כלומר, לבנה המייצגת תנאי. שתי החלונות המשולבות בה אף הן מלבניות עם פינות קטומות, ולכן יש לשלב בהן תנאים.

**תרגיל 13:** שנו את תסריטו השני של הנסיך כפי שתואר כאן. תעדו את הפרויקט ושמרו אותו בשם חדש.

#### מושג חדש: תנאי מורכב

תנאי מורכב הוא תנאי אשר מתקבל על ידי צירוף שני תנאים לכדי תנאי חדש. פעולות המשמשות להרכבת תנאים וליצירת תנאי מורכב הן, למשל, הפעולה "וגם" או הפעולה "או". תנאי מורכב שנוצר בעזרת הפעולה "וגם" מתקיים אך ורק אם שני התנאים שמרכיבים אותו מתקיימים. בכל מקרה אחר אין תנאי זה מתקיים. תנאי מורכב שנוצר בעזרת הפעולה "או" מתקיים אם שני התנאים שמרכיבים אותו מתקיימים, או אם כל תנאי שהוא מהשניים מתקיים. התנאי אינו מתקיים רק אם שני התנאים שמרכיבים אותו אינם מתקיימים.

#### תנאי חדש ב־Scratch: הרכבת תנאים על ידי "וגם"

התנאי  מתקיים אם התנאי המופיע בחלונית הראשונה מתקיים וגם התנאי המופיע בחלונית השנייה מתקיים.



## דוגמה 3: שורות מסודרות של תפוזים

בדוגמה האחרונה בפרק זה נבנה גרסה נוספת של משחק התפוזים. שוב ירצה הנסיך בדיוק 12 תפוזים, לא יותר ולא פחות. אבל בנוסף לכך הוא ירצה שהתפוזים הניתנים לו יופיעו על הבמה בצורה מסודרת יותר.

### משימה 7: סדר כבר את התפוזים שלך!

יש לשנות את האנימציה שנבנתה עבור משימה 4 כך שהתפוזים יסודרו על הבמה באופן אחר: כל שורה תכיל בדיוק ארבעה תפוזים, פרט לשורה האחרונה שיכולה להכיל פחות מארבעה תפוזים.

למשל, אם הנסיך קיבל בהתחלה 3 תפוזים, יופיעו שלושתם בשורה הראשונה. תוספת של שני תפוזים תביא לתוספת תפוז אחד בשורה הראשונה וליצירת שורה שנייה שבה יש תפוז אחד. בשלב זה תוספת של חמישה תפוזים תמלא את השורה השנייה, כך שתכיל ארבעה תפוזים ותכניס עוד שני תפוזים לשורה השלישית.



שם הקובץ: oranges-in-piles

## חישוב המיקום של התפוז הבא

הדרישה החדשה משפיעה בעיקר על קביעת המיקום המתאים לתפוז הבא שמוצג על הבמה, ומשימה זו היא באחריותה של דמות הכפתור. בגרסה הקודמת, אחרי שהכפתור נלחץ, עברה דמות הכפתור תמיד לשורה חדשה (כלומר, התבצע קידום ערך  $y$ ), ואחר כך הוצג כל תפוז בהמשך של אותה שורה (התבצע קידום ערך  $x$ ). כעת דמות הכפתור צריכה לקדם את ערך  $y$  רק כאשר מתמלאת שורה, וקידום זה יכול להתבצע גם באמצע הטיפול בקבוצת תפוזים ולאזן דווקא בתחילת הטיפול בקבוצה. כאשר מתבצע קידום של  $y$ , צריך לשנות גם את  $x$  ולעדכן את תחילת השורה.

התבוננו בתיאור המתאים לפעולת דמות הכפתור מדוגמה 2:

0. אם אחיצה על דמוג:

1. עדכן את המספרים  $x$  ו- $y$  כך שיהיו  $x$  ו- $y$  המקום הראשון בשורה חדשה

2. במחשבים מספרים בין 2 ל-5 ושאר אוגד המספר how many

3. בצד מספר פסגים השווה לערכו של how many:

3.1 קדם את ערכו של המספר  $x$  המקום הבא בשורה

3.2 הודע על גבול חדש

4. הוסף למספר הנוכחי את ערכו של how many

5. הודע על קבוצה חדשה

צעד 1 בתיאור זה שינה את ערכיהם של  $x$  ו- $y$  בעקבות הלחיצה על הכפתור, כך שקבוצת התפוזים החדשה תוצג בשורה חדשה. כעת צעד זה הוא מיותר. במקומו יש להכניס שינוי בצעד הביצוע החוזר, צעד 3, הבודק אם יש צורך לעבור שורה, כלומר, לעדכן בהתאם את ערכי  $x$  ו- $y$ .

לכן כעת דמות הכפתור צריכה לפעול על פי התיאור הבא:

0. אם לחיצה על דמוגר:
1. במחשבוני מספר בין 2 ל-5 ושמו הוא המשנה how many.
2. בצל מספר פלטים השווה לערכו של how many:
  - 2.1 אם השורה הנוכחית מלאה
    - 2.1.1 קדם אל y לשורה הבאה.
    - 2.1.2 שנה אל x כך ש'צביע' לפני המקום הראשון בשורה
  - 2.2 קדם אל ערכו של המשנה x המקום הבא בשורה
  - 2.3 הודע על גבול חדש
3. הוסף למשנה הצובר אל ערכו של how many
4. הודע על קבוצה חדשה

### האם התמלאה שורה?

כיצד תדע דמות הכפתור אם השורה הנוכחית מלאה (צעד 2.1)?

לשם כך נוכל להשתמש במשתנה נוסף, חדש, שבו נשמור את המקום בשורה של התפוז האחרון שהוצג על הבמה, ובהתאם נקרא לו last. למשל, אם על הבמה יש בשורה האחרונה שלושה תפוזים, בתוך המשתנה last יהיה הערך 3. אם על הבמה יש בשורה האחרונה ארבעה תפוזים (כלומר, היא מלאה), יהיה בתוך המשתנה last הערך 4. לכן התנאי שצריך להיבדק בצעד 2.1 הוא תנאי שוויון, המשווה את ערך המשתנה החדש למספר 4:

2.1 אם ערכו של last שווה ל-4

בכל פעם שדמות הכפתור "מעבירה" תפוז נוסף להצגה על הבמה, עליה לעדכן את ערך המשתנה החדש. אם התפוז הנוסף התחיל שורה חדשה, ערכו של המשתנה last אחרי העברת התפוז צריך להיות 1. אחרת, ערכו של last צריך לגדול ב-1.

**תרגיל 14:** שנו את תיאור הפעולה של דמות הכפתור כך שיכלול את הטיפול במשתנה החדש last, משתנה המיקום האחרון בשורה. שימו לב כי עדכון last צריך להיעשות בכל פעם ש"מועבר" תפוז, כלומר, בכל פעם שמעודכן ערך x. בנוסף, סוג העדכון צריך להיות דומה לזה של x: אם ערך x מקודם למקום הבא בשורה (צעד 2.2), גם ערכו של last צריך לגדול ב-1. אם ערך x משתנה בעקבות הצורך במעבר לשורה חדשה (צעד 2.1.2), גם ערכו של last צריך להשתנות כך שיעיד על הכנתה של שורה חדשה. לכן יש צורך בהוספת שני צעדים חדשים המעדכנים את ערכו של המשתנה החדש last, לאחר צעד 2.1.2 ולאחר צעד 2.2.

חשבו: מה הערך שצריך להיות משולב בצעד העדכון שתוסיפו לאחר צעד 2.1.2?

**תרגיל 15:** שנו את התסריט של דמות הכפתור על פי התיאור שהכנתם בתרגיל 14. זכרו להגדיר קודם כול את המשתנה החדש last.

## עדכון האתחולים

שינוי המשחק אינו אמור להשפיע על התנהגותו של הנסיך. אבל צריך לבדוק בזהירות את התנהגותה של דמות התפוז, משום שהיא אחראית על האתחולים. ראשית, הוספנו משתנה חדש שדמות הכפתור בודקת את ערכו מיד עם הלחיצה על הכפתור, כדי לברר אם השורה הנוכחית מלאה. לכן יש לדאוג לאתחל את ערכו של המשתנה החדש מיד בתחילת ביצוע המשחק, עם הלחיצה על הדגל הירוק.

**תרגיל 16:** לאיזה ערך צריך לאתחל את `last` עם הלחיצה על הדגל הירוק? עליכם לענות בזהירות ובהתחשב באופן שבו מתבצעים שינויים בערכו של `last` בתסריטה של דמות הכפתור. שלבו בתסריט האתחולים של דמות התפוז הוראת אתחול מתאימה.

חוץ מהוספת משתנה חדש, הכנסנו בתסריטה של דמות הכפתור שינויים נוספים הנוגעים למשתנים `x` ו-`y`. לכן עלינו לבדוק אם כתוצאה מהשינויים שהכנסנו, יש לשנות את האתחול של שני המשתנים האלה. נבחן קודם כול את הצורך בשינוי האתחול עבור `y`. בגרסה הקודמת, בתסריט האתחול של דמות התפוז, אותחל ערכו של `y` לערך 200-. בכל פעם שנלחץ כפתור ההוספה, הגדילה דמות הכפתור את ערך `y` ב-50 לפני שהתחילה להוסיף תפוזים. לכן מיקום השורה הראשונה היה 150-, וכל שורה חדשה מוקמה 50 נקודות מעל הקודמת.

כעת דמות הכפתור אינה מקדמת את ערך `y` מיד עם הלחיצה על הכפתור, וכאשר יש מקום בשורה הנוכחית, היא משתמשת בערך `y` הקיים. לכן כעת האתחול של `y` בתסריט האתחול של דמות התפוז צריך להתאים למיקום השורה הראשונה, וערך האתחול צריך להיות בהתאם 150- ולא 200-.

מה לגבי המשתנה `x`? בגרסת המשחק הקודמת לא כלל תסריט האתחול של התפוז אתחול עבור `x`, משום שמיד עם הלחיצה על כפתור ההוספה, קבעה דמות הכפתור את ערכו של `x` כך שיצביע לפני המקום הראשון בשורה (200-). כעת אין שורה כזאת בתסריטה של דמות הכפתור: אם התפוז הנוסף נכנס לשורה קיימת, דמות הכפתור מסתמכת על ערכו הנוכחי של המשתנה `x` ומקדמת אותו בהוראה יחסית. רק אם בעקבות ההוספה האחרונה התמלאה שורה, נעשה שימוש בהוראה מוחלטת כדי לעדכן את ערכו של `x` לתחילתה של שורה.

לכן כעת יש צורך באתחול של המשתנה `x`. לאיזה ערך צריך לאתחל אותו? כך שיצביע לפני תחילת השורה, כמו שנעשה קודם בתחילת תסריטה של דמות הכפתור. לכן יש לאתחל את `x` לערך 200-.

**תרגיל 17:** שנו את תסריט האתחול של דמות התפוז כך שיבוצע אתחול נכון של המשתנים `x` ו-`y`. בדקו את הפרויקט שהתקבל, תעדו אותו ושמרו אותו בשם חדש.

## ביצוע הגדלה מעגלית בעזרת חישוב שארית

כעת נשנה עוד קצת את הפרויקט שבנינו. האנימציה לא תשתנה, ולכן לא ירגישו הצופים בהבדל כלשהו. אך מסתבר שניתן לבצע חלק מפעולות הפרויקט בצורה שונה.

## משימה 8: הסביבון מסתובב – חישוב במעגל

יש לשנות את הפרויקט כך שסידור התפוזים בשורות ייעשה בעזרת חישוב שארית.

שם הקובץ: oranges-in-piles-with-mod



## חישוב מעגלי

אם נתבונן בתשומת לב בערכים שמקבל המשתנה last, נראה שתחום ערכיו מוגבל למדיי: הוא מאותחל לערך 0, אחר כך ערכו גדל ב-1 עם תוספת תפוז, אך כאשר ערכו מגיע ל-4, הוא חוזר ומתאפס שוב. לכן last מכיל לאורך כל המשחק רק את הערכים 0, 1, 2, 3 ו-4. אין זה מפתיע, משום שתפקיד המשתנה הוא לציין כמה מקומות כבר נתפסו בשורה האחרונה, ומאחר שמספר המקומות בשורה הוא מוגבל, גם מספר הערכים האפשריים למשתנה מוגבל. אין אפשרות לתפוס יותר מ-4 מקומות בשורה האחרונה.

למעשה, מספר הערכים האפשריים יכול להיות מוגבל אף יותר, ואין צורך שהמשתנה יקבל את הערך 4. כדי לראות זאת נתבונן מה מסמן לנו כל ערך אפשרי עבור last:

כאשר ב-last נמצא הערך 1 המשמעות היא כי התפוז האחרון שנוסף הוא הראשון בשורה, והתפוז הבא יהיה במקום השני. כאשר ב-last נמצא הערך 2 המשמעות היא כי התפוז האחרון שנוסף הוא השני בשורה, והתפוז הבא יהיה במקום השלישי. כך גם לגבי הערך 3.

מה משמעות הערכים 0 ו-4?

last יקבל את הערך 4 לאחר שנוסף תפוז במקום הרביעי והאחרון בשורה, והתפוז הבא יהיה במקום הראשון בשורה הבאה.

last יקבל את הערך 0 באחד משני מקרים: המקרה הראשון הוא בעת האתחול, כאשר לא הוצג עדיין אף תפוז, והתפוז הבא יהיה במקום הראשון. המקרה השני הוא באותו מצב שבו מתקבל הערך 4, משום שמיד אחרי הכנסת הערך 4, הוא מוחלף בערך 0. כאמור, זה קורה לאחר שנוסף תפוז במקום האחרון בשורה, המקום הרביעי, והתפוז הבא יהיה במקום הראשון בשורה הבאה.

אם כך גם הערך 4 וגם הערך 0 מסמנים כי התפוז הבא ייכנס במקום הראשון. לכן אפשר להסתפק בערכים 0, 1, 2 ו-3. המשמעות של ערך מסוים תהיה כי התפוז הבא יהיה במקום שגדול ב-1 מערך זה: בעקבות האתחול יהיה ב-last הערך 0, כי התפוז הבא ייכנס למקום הראשון. אחרי תוספת תפוז יתעדכן הערך מ-0 ל-1, ובכך יעיד כי התפוז הבא ייכנס למקום השני. אחרי תוספת של עוד תפוז, יתעדכן הערך מ-1 ל-2, ובכך יעיד כי התפוז הבא ייכנס למקום השלישי. בעקבות תוספת של עוד תפוז יתעדכן הערך מ-2 ל-3, ובכך יסמן כי התפוז הבא ייכנס למקום הרביעי. הצגת תפוז נוסף תגרום לעדכון הערך מ-3 ל-0 ובכך יסמן כי התפוז הבא ייכנס למקום הראשון בשורה הבאה. ככל שיתקדם המשחק, כך ימשיכו הערכים להשתנות בצורה מעגלית: 0, 1, 2, 3, 0, 1, 2, 3 וכן הלאה.

לכן כעת, עם הוספת תפוז, לא נרצה עוד להגדיל את ערכו של last ב-1. במקום זאת נרצה לבצע פעולה שתביא לשינוי מעגלי כמו זה שתואר כאן, שינוי השומר כל הזמן את ערכו של last בתחום המספרים 0 עד 3. עבור 0, 1 ו-2 משמעות השינוי המעגלי היא בדיוק כמו הגדלה ב-1, אך עבור 3, משמעות השינוי היא איפוס.

## שינוי התסריטים לצורך השימוש בחישוב מעגלי

נתבונן שוב בתיאור הנוכחי, בחלק שמטפל בחישוב המיקום של התפוז הבא:

2.1 אק ערכו  $el$  last שווה ל-4

2.1.1 קדמ אק  $y$  לשורה הבאה

2.1.2 שנה אק  $x$  כך שִׁיצביע אפני המקום הראשון בשורה

2.1.3 אפס אק  $last$

2.2 קדמ אק ערכו  $el$  המשגנה  $x$  המקום הבא בשורה

2.3 הגדל אק ערכו  $el$  last ב-1

ראשית, נשים לב שבתנאי המשמש בצעד 2.1 לא נשווה כעת את ערך המיקום האחרון בשורה (ערכו של  $last$ ) ל-4 אלא ל-0, משום שהערך 4 לא יהיה בין הערכים האפשריים עבור  $last$ . כמו כן, מיד לאחר ההשוואה ל-4, אופס ערכו של  $last$ , וכעת איפוס זה הוא כמובן מיותר.

כלומר, נרצה שדמות הכפתור תפעל על פי התיאור המילולי הבא:

2.1 אק ערכו  $el$  last שווה ל-0

2.1.1 קדמ אק  $y$  לשורה הבאה

2.1.2 שנה אק  $x$  כך שִׁיצביע אפני המקום הראשון בשורה

2.2 קדמ אק ערכו  $el$  המשגנה  $x$  המקום הבא בשורה


2.3 הגדל מעצאי ב-1 אק ערכו  $el$  last

עכשיו עוד עלינו לראות כיצד לבצע את פעולת ההגדלה המעגלית שבצעד 2.3. ניתן לבצע פעולה כזאת בעזרת חישוב שארית. ראשית נשים לב, כי תוצאת הפעולה של חישוב שארית אכן נותנת לנו תמיד ערך מתוך תחום ערכים מוגבל. כאשר מחלקים מספר ב-4, שארית החלוקה יכולה להיות 0 (אם המספר מתחלק בדיוק ב-4), 1, 2, או 3. באופן כללי, נוכל לקבל ערכים מתוך תחום מוגבל על ידי חישוב של שארית חלוקה בגודל התחום המוגבל (4 במקרה שלנו).

כדי לשמור את  $last$  בתחום הרצוי נגדיל את ערכו ב-1 כמקודם, אך מיד אחרי ההגדלה ב-1 נחשב את שארית החלוקה של הערך החדש ב-4. אם ערך  $last$  היה 1, בעקבות ההגדלה ב-1 יהיה ערכו 2, ולאחר חישוב שארית של חלוקה ב-4 יישאר ערכו 2. כך גם לגבי 0 ו-2; ביצוע של פעולת הגדלה ב-1, ולאחריה חישוב שארית חלוקה ב-4, ייתנו תוצאה זהה לביצוע הגדלה ב-1. במקרה של 3, בעקבות ההגדלה יהיה ערכו 4, ולאחר חישוב שארית החלוקה ב-4, נקבל 0, כנדרש.

אם כך, צעד 2.3 צריך להיתרגם לצעד המורכב הבא:

חשב את שארית החלוקה של  $(last + 1)$  ב-4 ושמור בתוך  $last$  את השארית.

ב־Scratch ניתן לחשב שארית בעזרת הפעולה  , שבקבוצת המפעילים הירוקה־בהירה. בחלונית הימנית יש לשלב את המספר המחולק (במקרה שלנו זהו ערכו של  $last$  ועוד 1). בחלונית השמאלית יש לשלב את המספר המחלק (במקרה שלנו 4). תוצאת הפעולה היא השארית המתקבלת מהחלוקה.

**שימו לב:** תוצאת החישוב היא מספר, ולכן הלבנה היא לבנת ערך, כפי שמעידה צורתה.



#### מושג חדש: הגדלה מעגלית

כאשר משתנה יכול לקבל ערך רק מתוך תחום מוגבל של ערכים, הגדלת ערכו צריכה להתבצע בצורה מיוחדת: אם ערכו של המשתנה הוא הגדול ביותר בתחום, המשמעות של הגדלת ערכו ב-1 היא לשנות את ערכו לערך הקטן ביותר בתחום. בכל מקרה אחר פעולת הגדלת הערך היא הפעולה הרגילה של הוספת הערך 1 לערך השמור במשתנה.

ניתן לבצע הגדלה מעגלית בעזרת חישוב שארית: מבצעים הגדלה רגילה, המוסיפה 1 לערך השמור במשתנה. לאחר מכן מחשבים את השארית המתקבלת מחלוקת הערך החדש בגודל תחום הערכים של משתנה. הערך המתקבל שווה לערכו של המשתנה לאחר ביצוע הגדלה מעגלית של הערך השמור בו.

#### פעולה חדשה ב-Scratch: שארית

הפעולה **שארית** מחשבת את השארית המתקבלת מחלוקת המספר המופיע בחלונת הימנית במספר המופיע בחלונת השמאלית.

**תרגיל 18:** השתמשו בפעולת השארית ושלבו בלבנה המתאימה לה את הערכים המתאימים. שלבו אותה בתסריט דמות הכפתור במקום המתאים לצעד 2.3.

כרגיל, עלינו לבדוק אם השינויים שערכנו מחייבים גם שינוי באתחולים של הפרויקט. תסריט האתחולים של דמות התפוז כולל כעת ארבע הוראות אתחול למשתנים. נבחן אם הן דרושות גם כעת.

את ערכו של המשתנה last עדיין צריך לאתחל ל-0. זה מסמן לדמות הכפתור שהתפוז הראשון צריך להיכנס במקום הראשון בשורה.

גם את ערכו של הצובר oranges יש לאתחל ל-0.

האתחול ל-y עדיין הכרחי, משום שהוראות העדכון של משתנה זה אצל דמות הכפתור הן כולן הוראות יחסיות, כמקודם. אבל צריך לוודא אם האתחול הוא לערך הנכון.

**תרגיל 19:** בפרויקט הקודם אותחל y ל-150. היזכרו בסיבה לכך וקבעו אם גם בפרויקט החדש יש לאתחל את y לאותו הערך. הסבירו את תשובתכם. אם עניתם בשלילה, לאיזה ערך יש לאתחל כעת את y, ומדוע?

בגרסה האחרונה הוספנו אתחול למשתנה  $x$ , אתחול שלא היה בגרסאות מוקדמות יותר. זאת משום שדמות הכפתור דאגה להחזיר את ערכו של  $x$  אל המקום שלפני תחילת שורה, רק אחרי שהתמלאה שורה וערכו של last הגיע ל-4.

**תרגיל 20:** האם כעת נדרשת פעולת אתחול של המשתנה  $x$ ? הסבירו בפירוט את תשובתכם. אם לדעתכם ניתן לוותר על פעולת האתחול של  $x$  בתסריט האתחול של התפוז, שנו אותו בהתאם ובדקו את הפרויקט שהתקבל.

תעדו את הפרויקט ושמרו אותו בשם חדש.

**תרגיל 21:** תרגיל זה הוא הרחבה של תרגיל 7 מפרק 5. לצורך פתרון התרגיל כדאי (אך אין הכרח) להשתמש בהוראה **המרחק אל** שבקבוצת החיישנים. הרעיון העומד מאחורי השימוש בהוראה וההוראה עצמה מוסברים בפרק 11, פרק הרשות שבסוף הספר.

א. באנימציה שיצרתם בסעיף ב' של תרגיל 7 מפרק 5, הרחיבו את התנהגותן של שלוש החיות, כך שלאחר שחיה עוצרת, היא בודקת מי משתי החיות האחרות קרובה יותר אליה, ואומרת משפט מתאים (למשל, "אני קרובה יותר אל הפרה"). הציגו על הבמה את המיקום על ציר ה- $x$  של כל אחת מהחיות.

שם הקובץ: play-run1

ב. הבעיה באנימציה שיצרתם בסעיף הקודם היא שיייתכן כי חיה מדווחת על המרחק בינה לבין החיות האחרות כאשר הן עדיין בתנועה. האנימציה שתיצרו בסעיף זה תספק מידע על המרחק של החיות זו מזו כאשר הן כבר אינן בתנועה. לצורך כך יהיה שימוש בשלושה משתנים: *מרחק-פיל-מפרה*, *מרחק-סוס-מפרה* ו*מרחק-סוס-מפיל*. ערכם של משתנים אלה יוצג על הבמה ויתעדכן עם עצירתן של החיות המתאימות, ובכך יוצג המידע הרצוי (ללא שימוש במשפטי דיבור).

לצורך פתרון הסעיף השתמשו בשלושה משתנים נוספים (שאין להציג את ערכם על הבמה): *פרה-עצרה*, *פיל-עצר* ו*סוס-עצר*. ערכו של כל אחד משלושת המשתנים יהיה 0 כאשר החיה המתאימה לו נמצאת בתנועה, ויהיה 1 לאחר עצירתה. חשבו: מדוע אין צורך גם במשתנים *מרחק-פרה-מפיל*, *מרחק-פרה-מסוס* ו*מרחק-פיל-מסוס*? כיצד יש לחלק בין שלוש הדמויות את האחריות לעדכון משתני המרחקים?

שם הקובץ: play-run2

ג. אם בפתרון הסעיף הקודם לא השתמשתם בתנאי מורכב, שנו את פתרונוכם כך שישתמש בתנאי מורכב.

שם הקובץ: play-run3

ד. **(מתקדם)** שנו את האנימציה האחרונה שיצרתם כך שבסיומה תאמר אחת החיות מהו המרחק בין זוג החיות הקרובות ביותר זו אל זו. על איזו משלוש החיות הטלתם את האחריות לטיפול במטלה החדשה ומדוע?

שם הקובץ: play-run4



## תרגיל 22: תרגיל זה הוא הרחבה של תרגיל 3 מפרק 6.

א. שנו את האנימציה שיצרתם בסעיף ב' של תרגיל 3 מפרק 6, כך שהצופים יוכלו לשלוט באופן ישיר רק על מספר הביצועים החוזרים של הרקדנית. מספר הביצועים החוזרים של הרקדן יהיה תלוי במספר הביצועים החוזרים של הרקדנית ויהיה גדול ממנו פי 3.

שם הקובץ: `dance-steps1`

ב. נכנה בשם "מסע" תנועה שלמה של הרקדן או הרקדנית ממרכז הבמה אל צדה, או מצד הבמה אל מרכזה. בסעיף זה יש להרחיב את האנימציה שיצרתם בסעיף הקודם כך שמספר המסעות של הרקדנית ומספר המסעות של הרקדן יימנו ויוצגו על הבמה. כך נוכל לוודא כי על כל שישה מסעות של הרקדנית מבצע הרקדן שני מסעות.

שם הקובץ: `dance-steps2`

ג. (מתקדם) במקום שהצופים יצטרכו לוודא כי נשמר היחס הרצוי בין מספר המסעות של הרקדנית לזה של הרקדן, הבדיקה תתבצע בתוך האנימציה. האחריות למטלה זו תהיה של הרקדן שיאמר "אני לא בקצב", אם לא מתקיים כי על כל שישה מסעות של הרקדנית הוא מבצע שני מסעות.



רמז: הרקדן יבצע את הבדיקה בכל פעם שהוא מסיים זוג מסעות. זאת יוכל לעשות על ידי שימוש בתסריט נוסף שיופעל בכל פעם שמסתיים זוג מסעות.

שם הקובץ: `dance-steps3`

ד. (מתקדם) קשה לסמוך על הרקדן שידווח באופן אמין על קצב תנועתו. כעת תבדוק הרקדנית כי נשמר היחס הרצוי בין מספר המסעות שלה לזה של הרקדן ואם הוא לא נשמר, היא תאמר "שמור על הקצב".



רמז: אין הרקדנית צריכה לבצע את הבדיקה בכל פעם שהיא מסיימת זוג מסעות, אלא רק בכל פעם שהיא מסיימת שישה מסעות. כיצד היא תדע מתי זה קורה?

שם הקובץ: `dance-steps4`

## תרגיל 23: תרגיל זה הוא הרחבה של תרגיל 4 מפרק 6.

א. שנו את האנימציות `rocket1` עד `rocket4` שיצרתם בסעיפים א' עד ד' של תרגיל 4 מפרק 6, כך שבכל אחת מהאנימציות יימנה ויוצג על הבמה מספר הצעדים שהטיל מבצע. חשבו (זו אינה שאלה פשוטה): מדוע בכל אחת מארבע האנימציות מתקבל בסוף הפעלת האנימציה ערך שונה עבור מספר הצעדים הכולל שביצע הטיל?

שמות הקבצים: `rocket1, rocket2, rocket3, rocket4`

ב. הרחיבו את האנימציות `rocket1` עד `rocket4` שיצרתם בסעיף הקודם, כך שבכל אחת מהאנימציות יימנה ויוצג על הבמה גם מספר הפעמים שמתבצעת הלולאה. הפעילו את ארבע האנימציות והסבירו את הערכים המתקבלים בכל אחת מההפעלות.

שמות הקבצים: `rocket1a, rocket2a, rocket3a, rocket4a`



## תרגיל 24: תרגיל זה הוא הרחבה של תרגיל 16 מפרק 7.

הרחיבו את האנימציות grass3 עד grass6 שיצרתם בסעיפים ג' עד ו', כך שבכל אחת מהאנימציות יימנו ויוצגו על הבמה מספר הפעמים בהן החגב הנע פוגש את החגב השני ומספר הפעמים בהן הוא פוגש את החגב השלישי. שמות הקבצים: grass1, grass2, grass3, grass4

## תרגיל 25: (מתקדם) תרגיל זה הוא הרחבה של תרגיל 16 מפרק 7.


א. נרצה להרחיב את האנימציה שיצרתם בסעיף א' של תרגיל 16 מפרק 7, כך שעל הבמה יוצג גם הערך הגדול ביותר על ציר ה- $x$  שאליו החגב מצליח להגיע.

**רמז:** צרו משתנה חדש – מקסימום- $x$  – שערכו יוצג על הבמה, והוא ישמור את הערך הגדול ביותר על ציר ה- $x$  שאליו החגב מצליח להגיע. המשתנה יאותחל ל-0. עדכנו של המשתנה החדש ייעשה באופן הבא: לאחר כל תזוזה של החגב, הוא יבדוק אם המיקום הנוכחי על ציר ה- $x$  גדול מהערך השמור במשתנה מקסימום- $x$ . אם כן, ערכו של המשתנה מקסימום- $x$  יעודכן לערך המיקום הנוכחי על ציר ה- $x$ .

בבעיות רבות נדרשת מציאת הערך הגדול ביותר (ערך מקסימום) מבין קבוצת ערכים. בכל פעם שיש למצוא ערך מקסימום, ניתן לעשות זאת על פי אותה שיטה שתוארה ברמז.

שם הקובץ: grass5

ב. באנימציה שהתקבלה בסעיף א' החגב נע ימינה ושמאלה על ציר ה- $x$ . כאשר הוא נע ימינה מנקודת ההתחלה שלו, ערכי המיקום על ציר ה- $x$  הם חיוביים, וכאשר הוא נע שמאלה מנקודת ההתחלה שלו, ערכי המיקום על ציר ה- $x$  הם שליליים. החישוב שערכנו בסעיף הקודם משווה ערכים של מספרים, ומשום כך הוא מתעלם לחלוטין מהתנועות שמאלה מנקודת ההתחלה. למעשה, מה שמעניין אותנו הוא לא הערך הגדול ביותר על ציר ה- $x$  שאליו החגב מצליח להגיע, אלא המרחק הגדול ביותר על ציר ה- $x$  מנקודת ההתחלה, שאותו הצליח החגב לעבור. שנו את האנימציה שיצרתם בסעיף הקודם כך שהשוואות שיתבצעו לא יהיו בין המיקום הנוכחי על ציר ה- $x$  ובין הערך השמור במשתנה מקסימום- $x$ , אלא יתעלמו מסימניהם של הערכים וישוו את ערכיהם המוחלטים.

לצורך פתרון הסעיף השתמשו בלבנת הערך  שבקבוצת המפעילים. כרגיל, לחיצה על החץ הקטן שבחלונית פותחת רשימה. במקרה זה זוהי רשימה של פעולות, וניתן לבחור מתוכה בפעולה abs המחשבת ערך מוחלט.

שם הקובץ: grass6

ג. בדומה למה שעשיתם בסעיף ז' של תרגיל 16 מפרק 7, שנו את האנימציה שיצרתם בסעיף הקודם כך שלאחר כל תזוזה יבחר החגב כיוון אקראי לסיבוב, בין  $-180^{\circ}$  ל- $180^{\circ}$ . כעת יחושבו ויוצגו על הבמה המרחקים הגדולים ביותר שחגב הצליח לעבור מנקודת ההתחלה, הן על ציר ה- $x$  והן על ציר ה- $y$ .

שם הקובץ: grass7

ד. שנו את האנימציה שיצרתם בסעיף הקודם, כך שבמקום שני ערכים נפרדים, אחד עבור ערכי  $x$  ואחד עבור ערכי  $y$ , יחושב ויוצג על הבמה ערך אחד, ערך של המרחק הגדול ביותר מנקודת ההתחלה שאותו הצליח החגב לעבור.



**רמז:** לצורך פתרון הסעיף תוכלו לבחור אחת משתי דרכים. דרך אחת היא לחשב את המרחק באופן ישיר על ידי נוסחת חישוב המרחק של נקודה מראשית הצירים. לשם כך אפשר להשתמש בלבנת הערך  של  שבקבוצת המפעילים, ולבחור בפעולה sqrt המחשבת שורש ריבועי. בדרך השנייה אפשר להגדיר דמות חדשה, של נקודת מרכז הבמה, שלא תעשה דבר חוץ מלהיות במרכז הבמה, בנקודה (0, 0). כעת השתמשו בלבנה  (ראו הערה בתחילת תרגיל 21) כדי לחשב ולהציג את הערך הדרוש.

שמות הקבצים: grass8, grass9

## תרגיל 26: תרגיל זה הוא הרחבה של סעיף ד' בתרגיל 17 מפרק 7.

שנו את האנימציה שיצרתם בסעיף ד' של תרגיל 17 מפרק 7, כך שיחושב ויוצג על הבמה הזמן שלוקח עד ששלושת זוגות הפרחים מוצאים זה את זה. כדי להפוך את האנימציה למעניינת יותר תוכלו גם להקטין את גודלם של הפרחים כך שמשך הזמן עד ששלושת זוגות הפרחים מוצאים זה את זה יהיה ארוך יותר.

**רמז:** השתמשו בשני משתנים חדשים, האחד למניית זוגות הפרחים שמצאו זה את זה, והאחד למניית הזמן החולף. מונה זוגות הפרחים שמצאו זה את זה יעודכן על ידי אחד מבני הזוג בכל פעם שזוג פרחים נפגש. את האחריות לעדכון מונה הזמן אפשר להטיל על דמות הכפתור start ולכלול בתסריטו ביצוע חוזר של הוראת המתנה של 0.1 ולאחריה עדכון מונה הזמן ב־0.1. איזה סוג של ביצוע חוזר מתאים במקרה זה?




שם הקובץ: flowers1

## סיכום

בפרק זה ראינו דרכים נוספות לעבודה עם מספרים.


ראינו כי לצורך השוואה בין מספרים ניתן להשתמש, בנוסף לתנאי השוויון הכללי, גם בתנאי אי־שוויון כלליים. בעזרת תנאי אי־שוויון ניתן לבדוק אם מספר אחד קטן ממספר אחר או אם מספר אחד גדול ממספר אחר.

ראינו כי ניתן ליצור תנאים מורכבים על ידי צירוף תנאים פשוטים. הכרנו בפרק זה פעולה אחת לצירוף תנאים, הפעולה וגם. פעולה זו מצרפת שני תנאים לקבלת תנאי אחד. תנאי זה מתקיים אם מתקיימים שני התנאים, ואין הוא מתקיים בכל מקרה אחר.


ב־Scratch נמצאים תנאי האי־שוויון הכלליים בקבוצת המפעילים, הירוקה־בהירה  ו־. בקבוצה זו נמצאת גם הלבנה המתאימה ליצירת תנאי "וגם" .


הכרנו משתנים בעלי תפקיד מיוחד: **צובר**, המשמש לצבירת ערכים, ו**מונה**, המשמש לספירה. לשימוש בשני סוגי המשתנים מאפיינים דומים: בשניהם יש שלב אתחול ל־0, ובשניהם יש שלבי עדכון של הערך השמור במשתנה. עבור משתנה שהוא מונה, עדכון הערך נעשה בכל פעם על ידי הגדלת הערך ב־1. עבור משתנה שהוא צובר, עדכון הערך מתבצע על ידי הגדלה במספר כלשהו.


בפרק זה השתמשנו במשתנה שיכול להכיל ערך מתוך **תחום מוגבל של ערכים**, ואשר הגדלת ערכו ב־1 היא **הגדלה מעגלית** בתוך התחום. לצורך ביצוע הגדלה מעגלית כזאת השתמשנו בחישוב השארית של חלוקת מספר שלם במספר שלם. השארית של פעולת חלוקה יכולה להיות 0 (אם המספר המחולק הוא כפולה של המספר המחלק), או כל מספר שלם שקטן מהמספר המחלק. הגדלה מעגלית בתחום יכולה להתבצע על ידי חישוב השארית של חלוקת הערך שהוגדל בגודל התחום המוגבל. ב־Scratch נמצאת הפעולה המתאימה לחישוב שארית בקבוצת המפעילים הירוקה־בהירה .

בפרק זה ראינו כי השימוש בפעולה מקבילית של דמויות שונות ושל תסריטים שונים מחייב זהירות, נושא חשוב שעליו נרחיב בפרק 10. הכרנו מנגנון בקרה שבו נעזרנו כדי ליצור תיאום זהיר יותר בין התסריטים: **שליחת מסר משולבת בהמתנה**. מנגנון זה מאפשר שליחת מסר, אך לאחר שהמסר נשלח, השולח ממתינ עד שהמסר מתקבל ומטופל, לפני שהוא ממשיך בביצוע ההוראות הבאות. ב־Scratch ניתן למצוא את ההוראה המתאימה למנגנון זה,  **שדר מסר** ו**חנה**, בקבוצת הבקרה.

עוד הכרנו בפרק זה כמה פעולות חדשות של Scratch:

דמות יכולה ליצור כמה תמונות שלה על הבמה על ידי שימוש בפעולת **החתמה**. לאחר ביצוע החתמה תישאר תמונתה של הדמות במקום שבו הייתה בעת ביצוע ההחתמה, גם אם לאחר מכן זזה הדמות למקום אחר. פעולת ההחתמה,  **חותמת**, נמצאת בקבוצת העט הירוקה.

דמות יכולה לנקות מהבמה את כל החותמות של כל הדמויות. פעולת ה**ניקיון** מתבצעת באמצעות ההוראה  **נקי** שבקבוצת העט הירוקה.

ראינו כיצד ניתן לבצע **הצמדה** של מילים ומספרים לרצף אחד. הוראת ההצמדה  **חבר** נמצאת בקבוצת המפעילים הירוקה־בהירה. ההוראה מאפשרת הצמדה של שני פריטים. להצמדה של יותר משני פריטים אפשר להשתמש בקינון של הצמדות.

## פרק 9

### רשימות – מנגנוני זיכרון מורכבים

בפרק זה ניצור אנימציה חלקית של מסעדה, שתתמקד בתהליך הזמנת מנות מהתפריט. כדי לבנות את האנימציה, נשתמש במנגנון חדש של Scratch שנכיר בפרק זה. בדומה למשתנים, גם זה מנגנון זיכרון, אך הוא מורכב יותר ממנגנון המשתנים ומאפשר לנו לזכור ערכים המסודרים ברשימות, כמו תפריטים מסוגים שונים או רשימת הזמנות.

### דוגמה: מה תרצה להזמין?

הנסיך, המוכר לנו כדורש התפוזים מפרק 8, ירד מנכסיו, וכעת אין לו ברירה אלא לעבוד לפרנסתו כמלצר. בהגיע לקוח – הלא הוא הצופה המפעיל את האנימציה – אל המסעדה, הוא יעיין בתפריטים המוצגים על הבמה ויקרא לנסיך המלצר. המלצר ירשום את הזמנתו של הלקוח וייצא אל המטבח להעביר את ההזמנה. מאחר שהמסעדה בהרצה, ייתכנו לצערנו (ובעיקר לצערך של הלקוח) מקרים שבהם חסרה במטבח של המסעדה מנה שהלקוח הזמין. המלצר הוא זה שייאלץ להתמודד עם מקרים כאלו ויבשר ללקוח כי המנה חסרה.

### בניית התפריטים

כרגיל, נבנה את האנימציה בשלבים, ובשלב ראשון נתמודד עם הצגת התפריטים על הבמה.

### משימה 1: מה בתפריט היום?

יש להציג על הבמה שלושה תפריטים של המנות שמציעה המסעדה: תפריט למנה עיקרית, תפריט לשתייה ותפריט קינוחים. משום שהמסעדה היא בשלבי הרצה, גם התפריט שלה מצומצם למדי: למנה עיקרית ניתן לבחור עוף, המבורגר, דג או טאקו, אפשר לבחור לשתות קולה, לימונדה או מים, ולקינוח אפשר לבחור פרי, גלידה או עוגה.



בנוסף יש להציג על הבמה דף מפנקס ההזמנות של המלצר שבו תירשם הזמנתו של הלקוח.

שם הקובץ: `import-list-contents`

כיצד נציג את התפריטים על הבמה?

אפשרות אחת היא להציג על הבמה תמונות של התפריטים השונים. למשל, על ידי שימוש בעורך הציור ליצירת רקע חדש הכולל את התפריטים, או על ידי שימוש בתוכנה אחרת כדי ליצור את הרקע הדרוש ואחר כך לייבא אותו כרקע. אבל אפשרות זו אינה מתאימה לצרכינו מכמה סיבות:

תפקודה הלקוי מעט של המסעדה יחייב גם הוצאה מהתפריט של מנות חסרות, ולכן במהלך האנימציה נרצה לעדכן את התפריטים המוצגים.

בנוסף נרצה להשתמש בתפריטים עצמם, כדי ליצור את רשימת ההזמנות של הלקוח. כלומר, נרצה

להעתיק מנות שהלקוח בחר מתוך התפריטים, אל פנקס ההזמנות של המלצר.

מה שאנו מחפשים הוא מנגנון שיאפשר לנו לזכור את התפריטים, להעתיק פריטים מתוכם, לעדכן אותם ולהציג את תוכנם על הבמה. מנגנון המשתנים מאפשר לנו ביצוע פעולות דומות, אבל עבור ערכים בודדים: ניתן לזכור ערך בתוך משתנה, להעתיק או להשתמש בערך השמור במשתנה, לשנות אותו ולהציג את תוכנו על הבמה.

לעזרתנו יבוא מנגנון זיכרון אחר, המאפשר לזכור בכפיפה אחת כמה פריטים השייכים כולם לקבוצה אחת. במקום לשמור בזיכרון ערך בודד בתוך משתנה, אנו שומרים קבוצת ערכים בתוך **רשימה**.

## הגדרת התפריטים

נפתח פרויקט חדש, בשלב זה ללא דמויות או תסריטים. נפנה לקבוצת המשתנים. שם נמצא, פרט ללבנה של יצירת משתנה, שבה השתמשנו בפרק 6, גם לבנה של יצירת רשימה **צור רשימה**.

כזכור, לבנה של יצירה אינה לבנה רגילה שנועדה לשילוב בתסריט. היא משמשת ליצירת שטח זיכרון ולהופעת לבנים שנועדו לעבודה עם שטח הזיכרון. לחצו על הלבנה **צור רשימה**. ייפתח חלון שבו תתבקשו לבחור שם לרשימה החדשה. זו תהיה הרשימה המתאימה לתפריט המנות העיקריות, ולכן כדאי לתת לה שם מתאים, כמו למשל, "מנה עיקרית", או "main". בנוסף תצטרכו להחליט אם רשימה זו היא לשימושן של כל הדמויות, בדומה לשאלה שהייתם נשאלים בעת יצירת משתנה. מאחר שכעת אף דמות אינה מוגדרת עדיין, נסמן כי הרשימה לא תהיה פרטית.

לאחר בחירת שם לרשימה קורים כמה דברים. ראשית, על הבמה מופיע מלבן המציג את הרשימה החדשה ובראשו רשום שמה. בתוך המלבן אמורים להופיע פריטי הרשימה. אבל מאחר שהרשימה עדיין ריקה ואינה מכילה אף פריט, מופיעה במקום זה המילה "ריק". בתחתית המלבן מצוין **אורך הרשימה**, כלומר, מספר הפריטים שבה. עבור רשימה ריקה, מופיע הערך 0.

בנוסף, בחלון ההוראות נוצרו לבנים חדשות בצבע אדום-כתום, לבנים שנועדו לשילוב בתסריטים. הראשונה היא הלבנה המייצגת את הרשימה. בדומה ללבנה המייצגת משתנה, גם ליד לבנה המייצגת רשימה יש ריבוע קטן. בתוך הריבוע שליד הלבנה החדשה מופיע כעת סימן "וי" ☒ **מנה עיקרית**. משום כך הרשימה מוצגת על הבמה. במקרים שבהם לא נרצה שרשימה תוצג על הבמה, יש ללחוץ על הריבוע שליד הלבנה המייצגת אותה, כך שלא יופיע בו הסימן "וי".

לבנים נוספות שנוצרו בחלון ההוראות בעקבות יצירת הרשימה מאפשרות לעבוד עם הרשימה, ובהמשך הפרק נכיר אותן.

**תרגיל 1:** צרו שלוש רשימות נוספות שיוצגו על הבמה. האחת עבור תפריט השתייה, השנייה עבור תפריט הקינוחים והשלישית עבור דף הזמנה מפנקסו של המלצר.

## הכנסת תוכן אל התפריטים

כעת עלינו למלא את התפריטים. לשם כך הכנו עבורכם שלושה קובצי טקסט, שכל אחד מהם מכיל את רשימת הפריטים המתאימה לתפריט מסוים. קובץ אחד עבור תפריט המנות העיקריות, קובץ שני עבור תפריט השתייה וקובץ שלישי עבור תפריט הקינוחים. נוכל להשתמש בקבצים אלה כדי למלא את התפריטים באופן הבא:

הזיזו את סמן העכבר כך שיעמוד על אחד ממלבני הרשימות שעל הבמה, למשל, מלבן הרשימה המתאימה לתפריט המנות העיקריות. עכשיו לחצו על הכפתור הימני של העכבר. נפתח תפריט ובו שלושה פריטים. בחרו בפריט השני, "בא...". כתוצאה מהבחירה ייפתח חלון, ובו תוכלו למצוא את הקובץ המתאים ולבחור בו. הקבצים שהכנו נמצאים יחד עם הפרויקטים של פרק זה, ושמו של הקובץ המתאים לתפריט המנות העיקריות הוא main. לאחר הבחירה תתמלא הרשימה, ועל הבמה יוצג תפריט המנות העיקריות המלא.

תוכלו לעיין בקבצים עצמם ולראות כיצד הם בנויים. אלו הם קובצי טקסט, ובכל קובץ מספר שורות כמספר המנות בתפריט המתאים: קובץ המנות העיקריות (main.txt) מכיל ארבע שורות, קובץ השתייה מכיל שלוש שורות (drink.txt) וכך גם קובץ הקינוחים (dessert.txt). כאשר קובץ כזה משמש כדי למלא רשימה, כל שורה במלואה מועתקת כפריט ברשימה.

אם תסתכלו היטב, תוכלו לראות כי בקבצים שיצרנו לצורך מילוי התפריטים, מופיעים שני רווחים בסופה של כל שורה. משום כך כאשר נכנס לרשימה פריט מהקובץ, הוא כולל גם רווחים בסופו. כך למשל, הפריט הראשון בתפריט המנות העיקריות אינו סדרת הסימנים **עוף** המורכבת משלושה סימנים, אלא סדרת הסימנים **עוף**, המורכבת מחמישה סימנים (אותיות המילה **עוף** ועוד שני רווחים). עשינו זאת במכוון, כדי להקל על ביצוע חלקים אחרים באנימציה, כפי שנראה בהמשך.

**תרגיל 2:** מלאו גם את תפריט השתייה ותפריט המנות העיקריות.

ובכן, יצרנו ארבע רשימות. רשימה אחת (דף מפנקס ההזמנות) היא ריקה, ואורכה 0. רשימה אחת (תפריט המנות העיקריות) מכילה ארבעה פריטים, ובהתאם לכך אורכה הוא 4. שתי הרשימות הנוספות (תפריט השתייה ותפריט הקינוחים) מכילות שלושה פריטים כל אחת, ולפיכך אורכן הוא 3. שמרו את הפרויקט.

### מושג חדש: רשימה

רשימה היא מנגנון זיכרון מורכב המאפשר לזכור ביחד כמה פריטים, השייכים כולם לקבוצה אחת.

לכל אחד מהפריטים ברשימה יש **מספר סידורי**, והוא מקומו ברשימה. ניתן להוסיף פריטים לרשימה קיימת, **למחוק** ממנה פריטים **ולהציף** לתוך מקום ברשימה כדי לראות מה הפריט שנמצא בו.

**האורך של רשימה** הוא מספר הפריטים שנמצא בה.

### פעולה חדשה ב־Scratch: יצירת רשימה

לחיצה על הלבנה חסרת הצבע **צור רשימה** שבקבוצת המשתנים תגרום להופעת חלון. בחלון זה ניתן לקבוע שם של רשימה, וכן לקבוע אם היא שייכת רק לדמות זו או לכל הדמויות. לאחר יצירת הרשימה תופיע במאגר ההוראות של קבוצת המשתנים לבנה המסומנת בשם הרשימה, היא הלבנה המייצגת את הרשימה.

אם זו הרשימה הראשונה בפרויקט, בעקבות יצירתה תופיע במאגר ההוראות של קבוצת המשתנים גם רשימה של לבנים שבעזרתן ניתן לשנות את הרשימה ולהסתכל בה.

### פעולה חדשה ב־Scratch: הצגת תוכנה של רשימה

כדי להציג על הבמה את תוכנה של רשימה, יש ללחוץ על הריבוע הצמוד ללבנה המייצגת את הרשימה, במאגר ההוראות של קבוצת המשתנים. כתוצאה מכך יופיע סימן "וי" בתוך הריבוע, ועל הבמה תופיע מסגרת ובה שם הרשימה, הפריטים הכלולים בה ואורכה.

### פעולה חדשה ב־Scratch: מילוי רשימה מקובץ

ניתן למלא רשימה מתוך קובץ טקסט שהוכן מראש. הקובץ יכול סדרת פריטים, כל פריט בשורה משלו. כדי למלא רשימה, יש להצביע עם העכבר על תצוגת הרשימה שעל הבמה, ללחוץ על הכפתור הימני של העכבר ולבחור בפריט "יבא..." ברשימה שנפתחה. בעקבות כך נפתח חלון ובו אפשר לבחור בקובץ הטקסט המתאים. הפעולה גורמת לרשימה להתמלא בפריטים שנמצאים בקובץ, על פי סדר הופעתם בקובץ.

## הזמנת ארוחה

כעת אנו מוכנים להרחבת האנימציה ולהכנסת המלצר לתמונה.

### משימה 2: ומה בשבילכם?

הרחיבו את האנימציה באופן הבא: בנוסף לתפריטים ולפנקס ההזמנות יהיה על הבמה גם כפתור הנועד לקריאה למלצר.

בעקבות לחיצה על הכפתור, יופיע המלצר הנסיכי וייקח מהלקוח את הזמנתו. ראשית ישאל מה ירצה הלקוח למנה עיקרית, אחר־כך מה ירצה לשתות, ולבסוף – מה ירצה למנה אחרונה. לאחר מכן, כדי למנוע טעויות, יחזור המלצר על ההזמנה הרשומה בפנקסו ויפנה למטבח למסור את ההזמנה.

שם הקובץ: prince-takes-order



## הדמויות

בפרויקט זה נזדקק לשתי דמויות: המלצר הנסיך וכפתור ההזמנה. הלקוח יהיה הצופה, ולכן איננו זקוקים לדמות עבורו.

את דמותו של הנסיך תוכלו לייבא כפי שעשיתם בפרק 8, מתיקיית People. לאור השינוי החד במעמדו, מתאים לתת לו שם חדש – מלצר במקום נסיך. את דמות הכפתור יצרנו עבורכם. היא נמצאת בתיקיית הפרויקטים של הפרק, ושם הקובץ המתאים הוא order-button. לאחר הגדרת הדמויות תוכלו לשנות את גודלן לגודל מתאים ולמקמן על הבמה במקום מתאים.

## הלקוח קורא למלצר – תפקידו של כפתור ההזמנות

כשהצופה מוכן להזמין, עליו לקרוא למלצר על ידי לחיצה על כפתור ההזמנות.

תפקידו של כפתור ההזמנות פשוט למדי: עליו לקרוא למלצר. בנוסף, ועוד לפני כן, נטיל עליו את האחריות "להפוך דף" בפנקס ההזמנות, כלומר, לדאוג שעל הבמה יופיע דף ריק שיאפשר למלצר לרשום את ההזמנה החדשה. את כל זה צריכה דמות הכפתור לבצע כאשר הכפתור נלחץ.

אם כך להתנהגותה של דמות הכפתור מתאים התיאור המילולי הבא:

0. אם לחיצה על דמותי:

1. נקה את הדף הפנקס ההזמנות על הבמה

2. קרא למלצר

אנו כבר יודעים כיצד לתרגם להוראה של Scratch את צעד 0. כדי לקרוא למלצר בצעד 2, נוכל להשתמש במנגנון המסרים. כיצד נתרגם את צעד 1?

## דף חדש בפנקס ההזמנות

הדף הנוכחי מפנקס ההזמנות מוצג על הבמה באמצעות מנגנון הרשימות. לכן כדי "לנקות" אותו, כלומר, למחוק את הכתוב בו, סביר שנוכל להיעזר בהוראות המטפלות ברשימות, ההוראות שנוצרו בקבוצת המשתנים לאחר יצירת הרשימה הראשונה.

ואכן, בין ההוראות החדשות שבקבוצה האדומה-כתומה נוכל למצוא הוראת מחיקה. הוראה זו מוחקת פריטים המופיעים ברשימה. הלבנה המתאימה להוראה זו היא  מתוך . להוראה זו שתי חלוניות שבצדן חץ קטן. כלומר, ניתן לשלב בחלוניות אלו פריט מתוך סדרה אפשרית של ערכים. הסדרה שמתאימה לחלונית השנייה כוללת את כל הרשימות שהוגדרו. הסדרה שמתאימה לחלונית הראשונה מכילה שלושה פריטים: 1, אחרון, הכול. אם כך, על ידי בחירת פריט מסדרה זו אפשר להורות על מחיקת הפריט הראשון ברשימה, על מחיקת הפריט האחרון ברשימה או על מחיקת כל פריטי הרשימה. ניתן גם להקליד באופן ישיר מספר בחלונית הראשונה, ובכך להורות על מחיקת פריט הנמצא במקום כלשהו ברשימה.

**תרגיל 3:** כתבו תסריט של Scratch עבור דמות הכפתור בהתאם לתיאור המילולי שניתן קודם.



## הוראה חדשה ב־Scratch: מחיקת פריט או פריטים מרשימה

ההוראה **מחק** **מתוך** גורמת למחיקת פריט או פריטים מתוך הרשימה ששמה מופיע בחלונית השנייה. זהותם של הפריט או הפריטים שנמחקים נקבעת לפי תוכנה של החלונית הראשונה:

אם מופיע בה ערך מספרי (1 או כל ערך אחר) – יימחק מהרשימה הפריט שמספרו הסידורי הוא הערך שבחלונית. אם אורכה של הרשימה אינו מגיע לערך זה, ולכן אין בה פריט מתאים עבורו, לא יימחק אף פריט.

אם מופיעה בחלונית הראשונה המילה "אחרון" – יימחק הפריט האחרון מהרשימה.

אם מופיעה בחלונית הראשונה המילה "הכול" – יימחקו כל הפריטים מהרשימה והיא תישאר ריקה.

## המלצר מגיע

המלצר מגיע בעקבות לחיצה על כפתור ההזמנות, ולמעשה – בעקבות המסר ששלח כפתור ההזמנות. אז עליו לקבל את ההזמנה, לרשום אותה בפנקסו ולחזור עליה בפני הלקוח. לכן למלצר מתאים התיאור המילולי הבא:

0. **אם קבלת מסר הקריאה:**
1. **הופע בפני הלקוח**
2. **אם הלקוח מה ירצה אומה עיקרי**
3. **רשום את גשוגו הפנקס ההזמנות**
4. **אם הלקוח מה ירצה אגוג**
5. **רשום את גשוגו הפנקס ההזמנות**
6. **אם הלקוח מה ירצה אומה אחרונה**
7. **רשום את גשוגו הפנקס ההזמנות**
8. **חזור אל ההזמנה בפני הלקוח**
9. **אך אטמה**

תיאור זה כולל כמה צעדים, שעדיין איננו יודעים כיצד לתרגמם להוראות של Scratch. אלו הם הצעדים 2 עד 7, שבהם מתנהל הדו־שיח בין המלצר ללקוח ותוצאותיו נרשמות בפנקס ההזמנות. אלו צעדים אלה נחזור בהמשך.

בצעד 1 המלצר צריך להופיע בפני הלקוח, ובצעד 9 עליו להסתלק אל המטבח. המטבח אינו נמצא מבחינתנו על הבמה, ולכן בעצם בצעד 9 המלצר צריך פשוט להיעלם מהבמה. אם כך, עבור שני הצעדים האלו נוכל להשתמש בצמד ההוראות שהכרנו בפרק 7: **הצג** ו־**הסתר** מקבוצת המראה.

**תרגיל 4:** התחילו לבנות את התסריט המתאים לתיאור המילולי של התנהגות המלצר. בשלב זה שלבו בו הוראות המתאימות לצעדים 0, 1, ו־9.

עבור צעד 8 נוכל להשתמש בהוראת דיבור. אך מה נשלב בחלונות של ההוראה? מה צריך המלצר לומר?

### הצצה לתוך רשימה – שימוש בערכו של פריט מתוך הרשימה

בצעד 8 המלצר צריך לומר ברצף את שמותיהן של שלוש המנות שהוזמנו: המנה העיקרית, השתייה והמנה האחרונה.

כזכור, את פנקס ההזמנות בחרנו לזכור כרשימה. אמנם, עוד לא כתבנו את ההוראות שמכניסות את בחירות הלקוח לתוך הרשימה של פנקס ההזמנות, כי החלטנו להתמודד עם בעיה זו מאוחר יותר. אבל בינתיים נוכל להניח שאכן הצלחנו בכך ושהרשימה המתאימה כבר מכילה את הפריטים המתאימים: המנה העיקרית מופיעה במקום הראשון ברשימה, השתייה במקום השני והקינוח מופיע במקום השלישי, כפי שאמור לקרות כתוצאה מביצוע הצעדים 3, 5 ו-7.



אם כך, המלצר צריך קודם כול להציץ ברשימה המתאימה ולראות מה הפריטים הנמצאים בה. הדבר דומה להצצה לתוך משתנה, כדי לראות מה ערכו ולהשתמש בו. אבל במקרה שלנו המשתנה הוא משתנה מורכב.

כדי להציץ לתוך משתנה, יכולנו להשתמש בלבנה המייצגת שלו. אבל, הלבנה המייצגת של רשימה מייצגת את הרשימה כולה. כשאנו רוצים להציץ לתוך רשימה עלינו לציין גם את המקום ברשימה שלתוכו אנו רוצים להסתכל. לשם כך נשתמש בלבנה אחרת, **פריט של**. בחלונות השנייה יש לבחור את הרשימה שאנו מעוניינים להסתכל לתוכה. בחלונות הראשונה יש לציין את מספרו הסידורי של הפריט שאנו מעוניינים לראות. נוכל לשלב בחלונות זה ערך מספרי בעצמנו או לבחור בעזרת החץ הקטן את הערך 1 או את המילה "אחרון". נוכל גם לבחור במילה "כול", ואז ייבחר מספר סידורי אקראי ברשימה.

הלבנה **פריט של** היא לבנת ערך. הערך הוא ערכו של הפריט הנמצא במקום המתאים ברשימה המתאימה. לכן נוכל לשלב את הלבנה בכל הוראה שמאפשרת שימוש בערך, בפרט בהוראת דיבור. אבל במקרה שלנו איננו מעוניינים באמירת שמו של פריט אחד מהרשימה, אלא באמירת שמם של שלושה. נוכל להשתמש פעמיים בהוראת ההצמדה **חבר** שהכרנו בפרק 8, כדי לומר את שמותיהן של שלוש המנות, זה אחרי זה.

כאן באים לעזרתנו הרווחים שהכנסנו בעת בניית קובצי הטקסט, והם כלולים כעת ברשימות כחלק מכל פריט. כאשר שמות המנות מוצמדים בעזרת ההוראה **חבר**, אין הם מופיעים ברצף, אלא מופרדים ברווחים.

## פעולה חדשה ב־Scratch: הצצה לתוך רשימה

הלבנה  של  היא לבנת ערך. הערך המתאים לה הוא ערכו של פריט מתוך הרשימה ששמה מופיע בחלונת השנייה. זהותו של הפריט נקבעת לפי תוכנה של החלונת הראשונה:

אם מופיע בה ערך מספרי (1 או כל ערך אחר), ערכה של הלבנה הוא ערכו של הפריט שמספרו הסידורי הוא הערך שבחלונת. אם אורכה של הרשימה אינו מגיע לערך זה, ולכן אין בה פריט מתאים עבורו, ללבנה ערך ריק.

אם מופיעה בחלונת הראשונה המילה "אחרון", ערכה של הלבנה הוא ערכו של הפריט האחרון ברשימה.

אם מופיעה בחלונת הראשונה המילה "הכול", ערכה של הלבנה הוא כערך פריט כלשהו ברשימה (כלומר, נבחר אקראית מספר סידורי, וערך הלבנה הוא הפריט שנמצא במקום שנבחר).

**תרגיל 5:** הוסיפו לתסריט של המלצר הוראה מתאימה לצעד 8.

### ד־שיח: זה שואל וזה עונה

בצעדים 2 עד 7 מתנהל ד־שיח בין המלצר ללקוח. המלצר שואל, הלקוח עונה והמלצר רושם את תשובתו בפנקס ההזמנות. כך שלוש פעמים עבור כל סוג של מנה. כיצד נתרגם זאת להוראות של Scratch?

למעשה, יש שני מרכיבים שונים בצעדים 2 עד 7. מרכיב אחד הוא שאילת שאלות וקבלת תשובות. המרכיב השני הוא רישום התשובות בפנקס ההזמנות. אנו נתמודד קודם עם המרכיב הראשון. ניתן כמובן להשתמש בהוראות דיבור כדי להציג ד־שיח, שאלות ותשובות. אבל במקרה שלנו אין זה מתאים. מדוע?

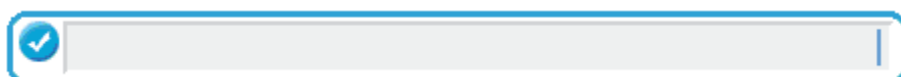
ראשית, איננו יודעים מראש מה יבחרו הלקוחות למנה עיקרית, שתייה וקינוח, ולכן איננו יכולים להכין משפטי דיבור מתאימים מראש. בחירותיהם של הלקוחות יתבררו רק תוך כדי האנימציה.

שנית, להוראות דיבור יש השפעה חיצונית בלבד. הצופים רואים אותן, אך פרט לכך אין להן כל השפעה על מהלך האנימציה. במקרה שלנו אנו רוצים להשתמש בתשובת הלקוח, ולעדכן בהתאם את פנקס ההזמנות.

אנו נשתמש במנגנון של Scratch, מנגנון המאפשר להציג שאלה על הבמה ולקלוט תשובה המוכנסת על ידי הצופים, תוך כדי ביצוע האנימציה. מאחר שזהו מנגנון שמתקשר עם הצופים תוך כדי ביצוע האנימציה, באמצעות לוח המקשים והעכבר, המקום שמתאים לחפש בו את ההוראות המתאימות למנגנון זה הוא קבוצת החיישנים.

## שאלת שאלות

ההוראה הרביעית בקבוצת החיישנים מאפשרת לשאול שאלה ולהציגה על הבמה. הלבנה המתאימה לה היא **המתן** **שאל**. בלבנה יש חלונית, שבה רושמים מילה, משפט, או כל דבר שנרצה שיופיע על הבמה כשאלה. נסו לגרור את הלבנה אל חלון התסריטים, אך השאירו אותה מנותקת, כשהיא אינה מחוברת לאף תסריט. שלבו בחלונית משפט מתאים. למשל, "מה תרצה למנה עיקרית?". כעת לחצו לחיצה כפולה על הלבנה בחלון התסריטים. תוכלו לראות על הבמה את תוצאת הפעלתה של הוראת השאלה: ראשית, מופיע בלון מלל ליד דמותו של הנסיך, ובו השאלה שנשאלה. בנוסף, מופיע על הבמה חלון קטן ומוארך, ובו שורה ריקה שבה הצופה יכול לכתוב את תשובתו. דבר נוסף לא יקרה עד אשר הצופה יענה לשאלה.



לאחר שהצופה מסיים לכתוב את תשובתו ולוחץ על סימן ה"י" שבחלק השמאלי של החלון או על המקש "Enter", החלון נעלם מהבמה.

על פי צורתה של הלבנה אנו רואים כי היא ניתנת לשילוב כהוראה רגילה בתסריט.

**תרגיל 6:** הוסיפו לתסריט של המלצר הוראות מתאימות לצעדים 2, 4, ו-6.

ובכן, הלקוח בחר את ארוחתו ובהתאם לכך עונה לכל השאלות המוצגות לו על הבמה. תשובותיו יהיו מספריות. למשל, כאשר הלקוח נשאל מה ירצה להזמין למנה עיקרית, הוא יענין בתפריט המנות העיקריות המוצג על הבמה; אם יבחר בעוף – יקליד את המספר 1, ואם יבחר בהמבורגר – יקליד את המספר 2. אם כך, תשובה חוקית עבור מנה עיקרית היא אחד מהערכים 1, 2, 3 או 4, ועבור שתייה או קינוח תשובה חוקית היא אחד מהערכים 1, 2 או 3.

## קליטת תשובות

מה קורה לתשובתו של הלקוח לאחר שהוא מסיים להקלידה והחלון נעלם? כיצד אפשר להשתמש בה בהמשך הביצוע?

Scratch דואגת לקרוא את התשובה שהוקלדה על הבמה בתוך חלון השאלה. היא שומרת אוטומטית את התשובה במשתנה מיוחד משלה, כמו משתנה גודל הדמות הנוכחי או משתנה התלבושת הנוכחית, המוגדרים ומעודכנים על ידי Scratch באופן אוטומטי. ובדיוק כמו למשתני הגודל והתלבושת, גם למשתנה התשובה יש לבנות ערך מתאימה לו, שנוכל למצוא בקבוצת החיישנים, מתחת להוראת השאלה, הלבנה **תשובה**.

כמו שקורה עבור כל לבנה שמייצגת משתנה, גם ליד הלבנה שמייצגת את התשובה יש ריבוע קטן, שלחיצה עליו גורמת להופעת סימן "י" או למחיקתו. אם בריבוע יש סימן "י" – יופיע על הבמה תוכנו של המשתנה, כלומר, תוכן התשובה האחרונה שהוכנסה.

בכל פעם אחרי שנשאלת שאלה ומוקלדת תשובה, התשובה נשמרת במשתנה המתאים, וניתן להשתמש בה, כמו בתוכנו של כל משתמש.



**שימו לב!** ברגע שתישאל שאלה נוספת, תשובתה תוכנס למשתנה התשובה, ובכך תימחק התשובה הקודמת שנשמרה שם. לכן עלינו לזכור לא לדחות את השימוש בתשובה, אלא להשתמש בה לפני שנשאלת שאלה נוספת.

כיצד נשתמש אנו בתשובת הלקוח?

## עדכון רשימה – הוספת פריט לרשימה

כל אחת משלוש תשובותיו של הלקוח צריכה להירשם בפנקס ההזמנות, כלומר, להיכנס אל הרשימה המתאימה לפנקס. לכן את הצעדים 3, 5, ו-7, "רשום את תשובתו בפנקס ההזמנות", ניתן לתאר מילולית גם כך:

הוסף אל גובעון הלקוחה הגשומה אל רשימת פנקס ההזמנות

תרגום צעד כזה להוראה של Scratch יתבצע בעזרת ההוראה **הוסף**  **ל-**  , המוסיפה את הפריט שמופיע בחלונית הראשונה אל הרשימה ששמה מופיע בחלונית השנייה. במקרה שלנו הרשימה שצריכה להופיע בחלונית השנייה בצעדים 3, 5, ו-7 היא הרשימה שנבנתה עבור פנקס ההזמנות.

מה צריך להופיע בחלונית הראשונה?

כזכור, הלקוח ענה לכל אחת משלוש השאלות תשובה מספרית, כלומר, רשם את מספרה הסידורי של המנה שבה בחר. בפנקס ההזמנות נרצה לכתוב את שם המנה ולא את מספרה הסידורי. לכן עלינו להציץ בכל אחד מהתפריטים המתאימים ולראות מה המנה שמתאימה למספר. זאת נוכל לעשות בעזרת לבנת הערך שהכרנו זה עתה, **פריט של** .

בחלונית השנייה בלבנת הערך יש לשלב את הרשימה שלתוכה אנו רוצים להציץ. עבור צעד 3 זו רשימת המנות העיקריות, עבור צעד 5 זוהי רשימת השתייה, ועבור צעד 7 זוהי רשימת הקינוחים.

מה נשלב בחלונית הראשונה עבור כל אחד מהצעדים? שם צריך להופיע מספרה הסידורי של המנה שנבחרה מכל תפריט, כלומר, תשובתו של הלקוח לכל אחת מהשאלות. זו, כזכור, שמורה במשתנה התשובה, וניתן להשתמש בה בעזרת הלבנה המייצגת שלו, **תשובה**.

**תרגיל 7:** הוסיפו לתסריטו של המלצר הוראות מתאימות לצעדים 3, 5, ו-7.

## פעולה חדשה ב-Scratch: ניהול דו-שיח

ההוראה **והמתן**  **שאל** גורמת להופעת בלון מלל ליד הדמות, ובו השאלה המופיעה בחלונית. בנוסף מופיע חלון על הבמה. החלון נשאר על הבמה, וביצוע האנימציה נעצר עד אשר הצופה מקליד בחלון תשובה ולוחץ על סימן ה"וי" שבצדו השמאלי של החלון, או על המקש "Enter". אז נעלם החלון, וביצוע האנימציה ממשיך, לא לפני שנשמרה תשובתו של הצופה.

תשובתו של הצופה נשמרת בתוך משתנה של Scratch שנועד לשמירת תשובות. לבנת הערך המייצגת את משתנה התשובה היא **תשובה**, וניתן לשלב אותה בכל מקום שנועד לשילוב ערך.

בכל פעם שנשאלת שאלה חדשה, ומוקלדת תשובה מתאימה לה, התשובה נשמרת במשתנה התשובה ובעקבות כך נמחקת התשובה האחרונה שנשמרה בו.

## הוראה חדשה ב־Scratch: הוספת פריט לרשימה

ההוראה הוספה גורמת להוספת הפריט המופיע בחלונת הראשונה אל הרשימה ששמה מופיע בחלונת השנייה.

הפריט יתווסף לאחר הפריט האחרון ברשימה, וכתוצאה מכך יגדל אורך הרשימה ב־1.

## אתחולים

כמעט סיימנו את משימה 2, אך כרגיל, עוד עלינו לבדוק אם דרושים אתחולים. התסריט שכתבנו עבור כפתור ההזמנות מתבצע כאשר הכפתור נלחץ, והתסריט שכתבנו עבור המלצר מתבצע כאשר הוא מקבל מסר מדמותו של כפתור ההזמנות.

האם יש פעולות שצריכות להתבצע עם הלחיצה על הדגל הירוק, עוד לפני שנלחץ כפתור ההזמנות?



התפריטים כבר נמצאים על הבמה, וכך גם פנקס ההזמנות. אבל אם זו אינה ההפעלה הראשונה של האנימציה, ייתכן שנשארה בפנקס ההזמנות הקודמת. דמות כפתור ההזמנות דואגת להפוך דף בפנקס ההזמנות בכל פעם שכפתור ההזמנות נלחץ; אלא שצריך לדאוג שבפנקס יופיע דף ריק מיד ברגע שנלחץ הדגל הירוק.

לכן עלינו לבנות גם תסריט אתחול, שיתבצע עם הלחיצה על הדגל הירוק וידאג להפוך דף בפנקס ההזמנות. מאחר שדמות כפתור ההזמנות כבר טיפלה גם בהפיכת דף, נטיל עליה גם את האחריות לביצוע האתחול הזה, שצריך לפעול לפי התיאור המילולי הבא:

0. אם אני צה על הדגל הירוק:

1. נקה את הדף הפנקס ההזמנות על הכמה

או בצורה מפורטת יותר:

0. אם אני צה על הדגל הירוק:

1. מחק את גופנה של הרשימה האנימציה הפנקס ההזמנות

**תרגיל 8:** הוסיפו לדמותו של כפתור ההזמנות תסריט אתחול מתאים.

אין זה האתחול היחיד שיש לבצע. כאשר אנימציה של Scratch מתחילה להתבצע, תמיד מוצגות כל הדמויות על הבמה. אבל הרי אנחנו מעוניינים שלקוח חדש המגיע למסעדה יוכל לעיין בתפריטים בשלווה, בלי שיופרע על ידי המלצר. אנו רוצים שהמלצר יופיע רק כאשר הלקוח יחליט להזמין וילחץ על כפתור ההזמנות.

לכן, כאשר האנימציה מתחילה להתבצע, על המלצר להעלים את עצמו בהתאם לתיאור המילולי הבא:

0. אם אני צה על הדגל הירוק:

1. הסג את דמותך

**תרגיל 9:** הוסיפו למלצר תסריט אתחול בהתאם לתיאור המילולי.

תעדו את הפרויקט, שמרו אותו בשם חדש ונסו להפעילו.

### סליחה אדוני, המנה חסרה

כפי שכבר הסברנו, המסעדה בהרצה, ולא הכול הולך חלק. למשל, לא פעם קורה שמנה שהזמין הלקוח חסרה במטבח. המלצר הוא זה שצריך להתמודד עם אי-הנעימות: עליו לחזור אל הלקוח, להסביר לו שחסרה מנה ולאפשר לו לבחור מנה אחרת במקומה.

מאחר שלצערם של הלקוחות תופעה זו אינה נדירה, בשלב זה יתרגל המלצר שוב ושוב את התנהלותו במצב זה. בכל פעם שמתבצעת הזמנה והמלצר חוזר אל המטבח, יודיעו לו כי אחת המנות שהוזמנה (מנה עיקרית, שתייה או קינוח) חסרה. המלצר יאלץ לפעול בהתאם.

### משימה 3: אופס, בדיוק נגמר...

הרחיבו את האנימציה באופן הבא: מיד לאחר שהמלצר נעלם אל המטבח עם הזמנתו של הלקוח, הוא חוזר להודיע לו כי אחת המנות חסרה. סוג המנה החסרה (מנה עיקרית, שתייה, או קינוח) ייקבע אקראית.

לאחר שהמלצר יודיע ללקוח על התקלה המצערת, הוא יאפשר לו להזמין ארוחה חדשה, כלומר, לבחון מחדש את כל החלטותיו הקודמות ולהזמין שוב את שלושת מרכיבי הארוחה על ידי לחיצה על כפתור ההזמנות. **שם הקובץ:** choice-not-available



על פי הגדרת המשימה, נראה כי צריך להרחיב את תסריטו של המלצר באופן הבא:

10. האמן מלט שכן במטבח

11. חזור שוב אל האקום

12. במר אקראי אל סוג המנה החסרה

13. הודע ללקוח כי חסרה המנה מאת המנה שמיאמה לסוג שבחר

14. הפוך דף הפנקס ההזמנה

15. הסבר ללקוח כי יוכל להזמין שוב על ידי לחיצה על כפתור ההזמנה

16. אך למטבח

בתרגיל 10 תתבקשו לתרגם להוראות של Scratch את המשך תסריטו של המלצר.

**תרגיל 10:** הרחיבו את תסריטו של המלצר כך שיכלול גם את ההוראות המתאימות להמשכו של

התיאור המילולי (צעדים 10-16). את צעד 12 תוכלו לממש על ידי בחירה אקראית של מספר בין 1 ל-3.

**תרגיל 11:** האם בעקבות הרחבת הפרויקט יש גם צורך בשינוי או בהרחבת האתחולים? נמקו את

תשובתכם.

אם עניתם בחיוב, הסבירו באיזה תסריט צריך להתבצע העדכון וכיצד, ושנו את הפרויקט בהתאם.

תעדו את הפרויקט, שמרו אותו בשם חדש ונסו להפעילו.

## עדכון התפריטים לפני הזמנה חוזרת

זה אכן נחמד מאוד מצדו של המלצר לאפשר ללקוח להזמין את ארוחתו מחדש, אבל יהיה נחמד עוד יותר אם הלקוח יערוך את הזמנתו המחודשת מתוך תפריט מעודכן, שבו המנה החסרה אינה מופיעה עוד.

### משימה 4: כדאי שתזכרו שהמנה הזאת חסרה



הרחיבו את האנימציה באופן הבא: לאחר שהמלצר מודיע ללקוח על המנה החסרה, ולפני שהוא מודיע לו כי הוא יכול ללחוץ שוב על כפתור ההזמנות, ידאג המלצר למחיקת המנה החסרה מן התפריט המתאים.

שם הקובץ: delete-a-menu-element1

לצורך פתרון המשימה החדשה עלינו להרחיב את חלקו השני של התיאור המילולי של המלצר ולהוסיף לו גם חלק המטפל בעדכון התפריטים, הוא צעד 15 החדש:

10. האמן מלט מן המטבח

11. תצור שוב אל האקום

12. במר אקראיג אל סוג המנה החסרה

13. הודע אלקום כי מתוך המנה חסרה המנה מהסוג שנבחר בצעד 12

14. הפוך דף הפנקס ההזמנות

15. מחק מהתפריט המאגיס אלסו שנבחר אל המנה שנבחר האקום

16. הסבר אלקום כי יוכל להזמין שוב אל ידי אחיזה אל כפתור ההזמנות

17. אל המטבח

מאחר שהתפריט שמור כרשימה, משמעות מחיקת מנה היא מחיקת הפריט המתאים מתוך הרשימה. כתוצאה מכך קורים כמה דברים: כדי שלא יישאר ברשימה מקום ריק, כל הפריטים שהיו ברשימה אחרי הפריט שנמחק, משנים את מקומם וזזים מקום אחד קדימה. אורך הרשימה קטן ב-1.

כבר הכרנו את ההוראה שמתאימה למחיקת פריט מתוך רשימה. זוהי ההוראה **מחק** **מתוך**

אנו השתמשנו בה למחיקת כל הרשימה, אך כפי שהסברנו, ניתן להשתמש בה גם למחיקת פריט בודד, על ידי כך שבחלונית הראשונה משלבים את מספרו הסידורי.

אם כך, עבור צעד 15 נוכל להשתמש בהוראת המחיקה. מה נשלב בחלוניות של ההוראה? נתחיל מהחלונית השנייה:

אם הסוג שנבחר בצעד 12 הוא מנה עיקרית, כלומר 1, נשלב בחלונית השנייה את רשימת המנות העיקריות. אם בצעד 12 נבחרה שתייה (כלומר, הערך 2), נשלב בחלונית השנייה את רשימת השתייה, ואם בצעד 12 הסוג שנבחר הוא קינוח (כלומר, הערך 3), נשלב בחלונית השנייה את רשימת הקינוחים.

מניסוח זה אנו מבינים כי ההוראה המתאימה לצעד 15 היא הוראה מורכבת של ביצוע מותנה.

**תרגיל 12:** הרחיבו את צעד 15 בתיאור המילולי לצעד מפורט יותר, המתאר ביצוע מותנה.

הוסיפו לתסריט הוראה המתאימה לצעד 15. בהוראות המחיקה מלאו בינתיים רק את החלונית השנייה עם שם הרשימה שממנה מתבצעת המחיקה.



מה יש לשלב בחלונית הראשונה של הוראות המחיקה?

עלינו לשלב בה מספר המתאים למספרה הסידורי של המנה העומדת להימחק. כלומר, זהו המספר הסידורי של המנה בתוך התפריט שאליו היא שייכת. למשל, אם הלקוח בחר בתחילה המבורגר, עלינו לשלב בחלונית הראשונה של הוראת המחיקה את הערך 2, כי זהו מספרה הסידורי של מנת ההמבורגר בתפריט המנות העיקריות.

האם אנו יודעים מהו המספר הזה? זהו המספר שהלקוח הקליד כתשובה כאשר נשאל מה ירצה להזמין למנה עיקרית. מיד לאחר השאלה המספר נשמר במשתנה התשובה, אך לאחר שהלקוח נשאל מה ירצה לשתות, התשובה הקודמת נמחקה וכבר אינה שמורה. כמובן, בפנקס ההזמנות שם המנה רשום עדיין, אך כבר "איבדנו" את מספרה הסידורי בתפריט שאליו היא שייכת.

נוכל להתמודד עם הבעיה בשתי דרכים: אפשרות אחת היא לנסות ולשחזר את המספר הסידורי האבוד על ידי הסתכלות בתפריט שאליו המנה החסרה שייכת. אפשרות נוספת היא לדאוג מראש לא לאבד אותו על ידי כך שנשמור היטב את תשובותיו של הלקוח בעזרת משתנים מתאימים שנגדיר בעצמנו.

קעת נבחר באפשרות השנייה, ובשלב מאוחר יותר נראה כיצד ניתן לפתור את הבעיה בדרך הראשונה.

**תרגיל 13:** עדכנו את התיאור המילולי של התנהגות המלצר כך שהתשובות יישמרו במשתנים, שבערך שלהם ייעשה שימוש בצעד 15.

שנו בהתאם את תסריטו של המלצר: הוסיפו משתנים מתאימים לשמירת שלוש תשובותיו של הלקוח לשאלותיו של המלצר (חשבו: האם צריך להציגם על הבמה?), דאגו שהתשובות אמנם יישמרו במשתנים החדשים והשתמשו במשתנים אלו כדי להשלים את ההוראה המתאימה לצעד 15. תעדו את הפרויקט שהתקבל ושמרו אותו בשם חדש.

## בנייה מחודשת של התפריטים

לשמחתם של לקוחות עתידיים, חסרונה של מנה במטבח אינו מצב קבוע. גם אם נמחקו מנות מהתפריט, ייתכן כי השפית של המסעדה תתאושש, תשיג את המצרכים המתאימים ותכין שוב את המנות שאזלו. כתוצאה ממאורע משמח זה, צריך יהיה לעדכן שוב את התפריטים, כדי שיכללו שוב את כל המנות שהוסרו מהם.

מאחר שבשלב זה אירועים כאלה אינם נדירים במסעדה, רכשו מנהלי המסעדה רובוט שכל תפקידו הוא לעדכן מחדש את התפריטים בכל פעם שמצב העניינים במסעדה חוזר לקדמותו וכל המנות זמינות שוב.

שימו לב, מאחר שתוכנם של התפריטים המקוריים שמור בתוך קובצי הטקסט שהוכנו לצורך כך, תמיד אפשר לשחזר תפריט באותו אופן שבו הוא נבנה בתחילה: על ידי הצבעה עם העכבר על תצוגת התפריט שעל הבמה, לחיצה על הכפתור הימני של העכבר, בחירה באפשרות "בא..." ובחירת הקובץ המתאים.

עם זאת, נרצה לספק למנהלי המסעדה אפשרות שהיא מבחינתם פשוטה ואינה מחייבת היכרות עם האופן שבו התפריטים מיוצגים ומטופלים. העובדה שהתפריטים מיוצגים כרשימות, ושנבנו קבצים

לצורך מילוי התפריטים – אינה מעניינת אותם כלל. זו כבר החלטה פנימית שלנו, ככותבי האנימציה, החלטה שאינה מעניינת של המשתמשים באנימציה – הלכוחות, מנהלי המסעדה ומפעילי המסעדה. מנהלי המסעדה יוכלו לעדכן מחדש את התפריטים על ידי הפעלת הרובוט.

## משימה 5: הכול בסדר, התפריט מתאושש

הרחיבו את האנימציה באופן הבא: צרפו דמות של רובוט. בכל פעם שהרובוט מופעל על ידי לחיצה על דמותו שעל הבמה הוא בונה מחדש את כל תפריטי המסעדה, כפי שהיו בעת שהמסעדה החלה לפעול.



שם הקובץ: restock-all

### ניסיון ראשוני

הגדירו דמות חדשה עבור הרובוט עם שם מתאים לתפקיד המיועד לה (תוכלו למצוא דמות מתאימה בתיקייה Fantasy, למשל, הדמות robot3).

מה צריכה להיות התנהגותה של דמות הרובוט?

בכל פעם שלוחצים עליה, היא צריכה לבנות מחדש את כל התפריטים. אין הוראה שמאפשרת לדמות לייבא קובץ לתוך רשימה, ולכן דמות הרובוט תצטרך לבנות את רשימות התפריטים בצורה מפורשת, כלומר, להכניס אליהן פריט אחר פריט.

אבל גם אם התפריטים אינם מלאים, סביר להניח כי לא כולם ריקים. אם תפריט אינו ריק, לאחר הוספת פריטי התפריט המקורי, יהיה התפריט ארוך מכפי שרצינו שיהיה: הוא יכיל גם את הפריטים שנותרו בו וגם את הפריטים שנוספו שוב. כלומר, חלק מהפריטים – אלו של המנות שלא היו חסרות במטבח – יופיעו בו פעמיים.

כיצד אפשר למנוע זאת?



לפני שהרובוט ממלא תפריט מחדש, עליו לנקות קודם את התפריט על ידי מחיקת כל הפריטים שנותרו בו.

אם כך, ניתן לתאר את התנהגות הרובוט על ידי התיאור המילולי הבא:

0. אם אחיצה על דמוג:

1. נקה את גפדיט המנוג העיקרי

2. הוסף אלגפדיט המנוג העיקרי אל המנוג שהיו בו בגמילה

3. נקה את גפדיט השג"ה

4. הוסף אלגפדיט השג"ה אל המנוג שהיו בו בגמילה

5. נקה את גפדיט הקינוחים

6. הוסף אלגפדיט הקינוחים אל המנוג שהיו בו בגמילה

**תרגיל 14:** תרגמו את התיאור המילולי של דמות הרובוט לתסריט של Scratch, אגב שימוש בהוראות המטפלות ברשימות שנלמדו עד כה בפרק. שימו לב כי לצעדים 2, 4, ו-6 מתאימה יותר מהוראה אחת (ארבע הוראות לצעד 2, שלוש הוראות עבור צעד 4 וכך גם עבור צעד 6). זכרו להצמיד שני רווחים אחרי שמה של כל מנה! בדקו את הפרויקט שבניתם, תעדו ושמרו בשם חדש.

שם הקובץ: restock-missing

### פתרון משופר – מה שכבר יש, יש

למעשה, דמות הרובוט מבצעת עבודה מיותרת. מדוע למחוק פריטים שנותרו ברשימות, כדי להכניסם אחר כך שוב? יותר הגיוני להכניס לתפריטים רק את הפריטים שחסרים בהם. כלומר, אנו מעוניינים לשנות את התיאור המילולי המתאים לדמות הרובוט כך:

0. אם אחיצה על דמויך:

1. אָפֵי כֹּא מִנָּה:

1.1 אם אינה נמצאת בגפריט שאֵליו היא שייכת

1.2 הוסף אותה לגפריט שאֵליו היא שייכת

שימו לב, בתיאור הנוכחי צעד 1 מייצג למעשה צעדים רבים, עשרה צעדים. לכל מנה ומנה יש צעדים מתאימים לה כדי לבדוק אם היא נמצאת בתפריט המתאים וכדי להוסיף אותה לתפריט אם אינה נמצאת בו.

למשל, עבור מנת העוף, התיאור המילולי צריך לכלול את הצעדים הבאים:

1. אם מנת העוף אינה נמצאת בגפריט המנות העיקריות

1.1 הוסף אותה לגפריט המנות העיקריות


עבור מנת ההמבורגר, התיאור המילולי צריך לכלול את הצעדים הבאים:

2. אם מנת ההמבורגר אינה נמצאת בגפריט המנות העיקריות


2.1 הוסף אותה לגפריט המנות העיקריות

### בדיקת שייכות לרשימה

כיצד נבדוק אם מנה מסוימת שייכת לתפריט מסוים?

נוכל להשתמש בתנאי חדש המתייחס לרשימות, התנאי . כמו כל תנאי, צורת הלבנה המתאימה לו היא מלבנית עם פינות קטומות. התנאי מתקיים אם הרשימה המופיעה בחלונת השמאלית מכילה את הפריט שמופיע בחלונת הימנית.

#### תנאי חדש ב-Scratch: שייכות של פריט לרשימה

התנאי  מתקיים אם הרשימה המופיעה בחלונת השמאלית מכילה את הפריט שמופיע בחלונת הימנית. אם הרשימה אינה מכילה את הפריט, התנאי אינו מתקיים.

## תנאי הפוך: שלילה של תנאי


הרובוט צריך לבדוק לכל מנה אם היא אינה נמצאת בתפריט המתאים. כלומר, צעד 1, למשל, צריך להיכתב בעצם כך:

1. אם לא מתקיים כי מנה העוף נמצא בגפדיט המנה העיקריות

1.1 הוסף אזהרה לגפדיט המנה העיקריות


לשם כך נוכל להשתמש בפעולה חדשה של Scratch, שיוצרת מתנאי אחד תנאי שהפוך לו. את הלבנה המתאימה נמצא בקבוצת המפעילים, והיא הלבנה .

שימו לב כי צורתה של הלבנה היא כצורת לבנת תנאי, מלבנית עם פינות קטומות. זאת משום שהלבנה יוצרת תנאי, תנאי הפוך לתנאי המקורי. כשהתנאי המקורי מתקיים, התנאי החדש אינו מתקיים, ולהיפך: אם התנאי המקורי אינו מתקיים, התנאי החדש מתקיים.

לכן יש לתרגם את צעד 1 המטפל במנת העוף, להוראת ביצוע מותנה ללא חלופה, שבה משובץ התנאי  (כרגיל, יש לזכור את הרווחים המוצמדים לשם המנה).

**תרגיל 15:** תרגמו את התיאור המילולי החדש והמלא של דמות הרובוט לתסריט של Scratch. בדקו את הפרויקט שבניתם, תעדו ושמרו.

### תנאי חדש ב־Scratch: תנאי הפוך

התנאי  הוא התנאי ההפוך לתנאי המשולב בחלופית: הוא מתקיים כאשר התנאי המשולב בחלופית אינו מתקיים, והוא אינו מתקיים כאשר התנאי המשולב בחלופית מתקיים.

## חיפוש ברשימה אחר מקומו של פריט שנמצא בה

לסימומו של הפרק, נשפר את אחד הפתרונות שבנינו קודם ותוך כדי כך נלמד כיצד לחפש אחר מקומו של פריט בתוך רשימה.

כזכור, כאשר המלצר מודיע ללקוח שמנה מסוימת חסרה, הוא אחראי גם למחוק אותה מהתפריט המתאים. משום כך המלצר שומר בתוך משתנים את תשובות הלקוח שניתנו במהלך הדו־שיח, וכך קל לו למחוק את הפריט המתאים למנה החסרה מרשימת התפריט המתאימה לסוג המנה. כאשר תיארנו את הפתרון שבנינו, הזכרנו כי ישנה דרך נוספת לדעת מה מקומה של המנה החסרה בתפריט שבו היא מופיעה: להסתכל בתפריט שלה ולחפש אחריה. כעת נבנה פתרון שמתאים לדרך זו.

פתרון זה אכן אינו מחייב את המלצר לזכור את תשובות הלקוח שניתנו במהלך הדו־שיח. במהלך הפתרון המלצר חולק את האחריות שקיבל עם השפית. אחרי הכול, היא אשמה יותר ממנו בכך שהמנה חסרה במטבח...

## משימה 6: היכן נמצאת המנה החסרה?

שנו את הפרויקט המתאים למשימה 4 כך שלצורך הסרת מנה שחסרה במטבח מתפריטה, ייעזר המלצר בשפית. השפית תצא מהמטבח ותלך גם היא אל שולחנו של הלקוח. אז תחפש את המנה החסרה בתפריט המתאים ותמחק אותה ממנו. לאחר מכן תשוב למטבח.

שם הקובץ: delete-a-menu-element2



### מה נשאר למלצר לעשות?

על פי הגדרת המשימה נראה כי צריך להחליף את צעד 15 בתיאור המילולי של התנהגות המלצר בקטע מורכב יותר של ביצוע מותנה, כך:

15. אם סוג המנה החסרה שבמנה הוא מנה עיקרי  
15.1 הודע אלפיג'לעדכן את גפריט המנה העיקרי

16. אחר

16.1 אם סוג המנה החסרה שבמנה הוא מנה עיקרי  
16.1.1 הודע אלפיג'לעדכן את גפריט המנה העיקרי

16.2 אחר

16.2.1 אם סוג המנה החסרה שבמנה הוא קינוח  
16.2.1.1 הודע אלפיג'לעדכן את גפריט הקינוח

לצורך העברת ההודעות מהמלצר לשפית נשתמש במנגנון המוכר לנו, המנגנון של שידור וקבלת מסרים. במקרה זה עלינו להשתמש בשלושה מסרים.

**תרגיל 16:** עדכנו את תסריטו של המלצר בהתאם לשינויים שתוארו ובהתאם לתיאור המילולי שניתן כעת. זכרו כי כעת המלצר כבר אינו צריך לשמור במשתנים את תשובות הלקוח במהלך הדרישה.

### תפקידה של השפית

בנוסף לתפקידי הבישול, כעת נדרשת השפית של המסעדה גם לעדכן את התפריטים. מאחר שכעת השפית נדרשת לצאת אל מחוץ למטבח ולבצע גם פעולות הנראות באנימציה, יש להגדיר לה דמות משלה. חפשו דמות מתאימה בתיקיית האנשים (People).

יש לבצע את פעולות העדכון של התפריטים לאחר שהמלצר מודיע כי תפריט מסוים זקוק לעדכון. אם כך השפית יכולה לצפות לקבלת שלושה סוגי הודעות מהמלצר באמצעות מנגנון המסרים, ולכן עליה להכין שלושה תסריטים שיגיבו על ההודעות האלו.

התסריטים יהיו דומים מאוד זה לזה ויבדלו רק בתפריט שבו הם מטפלים. נדון כעת בבניית התסריט המתאים למנה העיקרית, ואחר כך תוכלו לטפל באופן דומה גם בשני התפריטים האחרים.

בקווים כלליים, טיפולה של השפית בתפריט המנות העיקריות צריך להתאים לתיאור המילולי הבא:

0. אם קבאל המסר "בקשה להסרה מגפית מנות עיקריות":
1. אבי מהמטבח אל אולם המסעדה
2. מפשי אל מקומה של המנה המסרה בגוף גפית המנות העיקריות
3. מחקי מגפית המנות העיקריות אל המנה שמופיעה במקום שאלה 3א 2
4. מבי לגי שני
5. מצרי למטבח

## בעיית החיפוש ופתרונה

יש בידיכם כלים לתרגם ל-Scratch את כל הצעדים בתיאור הזה. אבל התרגום של צעד 2 אינו פשוט. זהו צעד מורכב המהווה בעיה בפני עצמה – בעיית החיפוש אחר מקומו של פריט בתוך רשימה. פעולה של חיפוש כזה היא פעולה שימושית כאשר עובדים עם רשימות, וכדאי לדעת כיצד לבצע זאת. כדי לבצע את החיפוש, השפית צריכה לדעת שני נתונים: באיזה תפריט היא צריכה לחפש ומהי המנה שהיא מחפשת.

הנתון הראשון ידוע לה על פי ההודעה שקיבלה מהמלצר, על פי המסר ששלח לה. מה לגבי הנתון השני? המלצר הוא זה שבחר אקראית את סוג המנה החסרה. אם דאג לשמור במשתנה את הסוג שבחר או את המנה החסרה עצמה, תוכל השפית להשתמש בערכו של משתנה כזה כדי לדעת מהי המנה החסרה. אם לא דאג להשתמש במשתנה כזה, אין לנו ברירה כעת אלא לתקן את תסריטו של המלצר כך שישמור במשתנה את זהות המנה החסרה.

כאשר ידועים לשפית שני הנתונים האלה, עליה לפנות אל הרשימה המתאימה ולהתחיל להסתכל בה פריט אחר פריט, עד אשר היא מוצאת את המנה שחסרה. כלומר, היא תסרוק את הרשימה פריט אחרי פריט, החל מהפריט הראשון, ותשווה כל פריט למנה החסרה. הסריקה מסתיימת כשנמצא הפריט המבוקש, המנה החסרה.

שימו לב: מובטח כי החיפוש של השפית יסתיים בהצלחה: הרי המנה החסרה היא מנה שהלקוח בחר לאחר עיון בתפריט המתאים, ולכן ברור שהיא נמצאת בתוך תפריט זה, במקום כלשהו. הנה תיאור מילולי ראשוני שמתאים לחיפוש של השפית ברשימת המנות העיקריות:

2. הסגלי אל המקום הראשון בגפית המנות העיקריות
3. בצע עד שמקיים כי המנה שנמצא במקום שאליו אל מסגל אל היא המנה החסרה
- 3.1 הסגלי אל המקום הבא בגפית המנות העיקריות

למעשה, למרות שהתיאור הזה מתייחס באופן מפורש לאנימציה שלנו, אנימציית המסעדה, הרי שאפשר להשתמש בתיאור שדומה לתיאור הזה כדי לתאר פעולה כללית של חיפוש מקומו של פריט הנמצא בתוך רשימה:

1. הסגל אל המקום הראשון ברשימה
2. בצע עד שמקיים כי הפריט שנמצא ברשימה במקום שאליו אל הוא הפריט שאני מחפש
- 2.1 הסגל אל המקום הבא ברשימה

זה אמנם נראה תיאור ברור, אבל אם נתעמק בו מעט, נבין שהוא עדיין אינו מפורט מספיק. כיצד יתבצע בכל סיבוב של הביצוע החוזר מעבֵר אל המקום הבא ברשימה?

אנו ניעזר במשתנה חדש, משתנה מסוג מיוחד. המשתנה ישמור בכל שלב את מספרו הסידורי של המקום ברשימה שעליו יש להסתכל. נוכל למשל לקרוא לו place. בצעד 1 יאותחל ערכו של המשתנה place ל-1, וכך מובטח שבפעם הראשונה שיתבצע צעד 2, ייבדק הפריט שנמצא במקום הראשון. בצעד 2.1 יעודכן ערכו של המשתנה place, כך שישמור את המספר הסידורי של המקום הבא ברשימה. כלומר, ערכו יוגדל ב-1:

1. אגמא אג ערכו על משנה המקום place ל-1
2. בצע עז שמקיים כי הפריט שנמצא במקום place ברשימה הוא הפריט שאגה מחפש
- 2.1 הגדל אג ערכו על place ב-1

הביצוע החוזר מסתיים כאשר מתקיים התנאי המשולב בו. כלומר, במקום place נמצא הפריט שאותו חיפשנו, ולכן משתנה המקום place שומר את התשובה המבוקשת.

שימו לב לאופן העבודה עם משתנה המקום: הוא מאותחל ל-1, ולאחר מכן יש ביצוע חוזר שבכל סיבוב בו מוגדל ערכו של משתנה המקום ב-1. תבנית פעולה זו מאפיינת את השימוש במשתנה מסוג כזה, **משתנה מקום** שנועד לשמש כעזר בפעולת חיפוש ברשימה.

**תרגיל 17:** בפרק 6 הכרנו משתנים מסוגים מיוחדים, משתנה **צובר** ומשתנה **מונה** (שהוא בעצם סוג מיוחד של צובר). השוו בין תבנית העבודה עם משתנה מקום לתבנית העבודה עם צובר ולתבנית העבודה עם מונה.

באנימצית המסעדה שלנו החיפוש של השפית אחר המנה החסרה בתפריט המנות העיקריות ייראה כך:

2. אגמאי אג ערכו על משנה המקום place ל-1
3. בצע עז שמקיים כי הפריט שנמצא במקום place ברשימת המנות העיקריות הוא המנה החסרה
- 3.1 הגדל אי אג ערכו על place ב-1

ובסך הכול מתקבל התיאור המילולי הבא:

0. עס קבאל המסר "בקשה להסרה מהפריט מנות עיקריות":
1. לבי מהמטבח אל אולם המסעדה
2. אגמאי אג ערכו על משנה המקום place ל-1
3. בצע עז שמקיים כי הפריט שנמצא במקום place ברשימת המנות העיקריות הוא המנה החסרה
- 3.1 הגדל אי אג ערכו על place ב-1
4. מחקי מהפריט המנות העיקריות אג המנה שנמצא במקום place
5. חבי שג' שניג
6. חצרי אמטבח

**תרגיל 18:** תרגמו לתסריט של Scratch את התיאור המילולי המתאים לעדכון תפריט המנות העיקריות. שימו לב כי יש צורך בהגדרת משתנה חדש, אך הוא עומד לרשותה הבלעדית של השפית. היא זו שמבצעת את החיפוש, ואין זה מעניינן של דמויות אחרות במה היא נעזרת לצורך ביצוע החיפוש.

צרו תסריטים דומים לעדכון תפריט השתייה ולעדכון תפריט הקינוחים.

### **בעיית החיפוש ברשימה אחר מקומו של פריט שנמצא בה, ופתרונה**

בעיה החוזרת על עצמה בעת עבודה עם רשימות היא בעיית החיפוש ברשימה אחר מקומו של פריט שנמצא ברשימה. את הבעיה ניתן לפתור על פי תבנית קבועה שמתאימה לכל רשימה ולכל פריט.

יש להשתמש **במשתנה מקום**, שתפקידו לזכור את המספר הסידורי של המקום ברשימה שעליו מסתכלים, ויש לבצע את הצעדים הבאים:

1. **אתחל** את ערכו של משתנה המקום ל-1

2. **בצע עד** שמתקיים כי הפריט שנמצא ברשימה במקום שמספרו הסידורי שמור במשתנה המקום, הוא הפריט שאותו אתה מחפש

2.1 **הגדל** את ערכו של משתנה המקום ב-1

בסיום הביצוע החוזר משתנה המקום מכיל את מקומו של הפריט שחיפשנו ברשימה.

**תרגיל 19:** האם בעקבות הוספת דמותה של השפית יש צורך בעדכון האתחולים בפרויקט? נמקו את תשובתכם.

אם עניתם בחיוב, הסבירו באיזה תסריט צריך להתבצע העדכון וכיצד ושנו את הפרויקט בהתאם.

תעדו את הפרויקט שהתקבל ושמרו אותו בשם חדש.

**תרגיל 20:** אפשר לחלק בין המלצר לשפית את האחריות לעדכון התפריטים באופן שונה במקצת. במקום שהמלצר יבדוק איזה תפריט זקוק לעדכון וישלח לשפית הודעה מתאימה לכך, הוא פשוט יודיע לשפית שנדרש עדכון של תפריט כלשהו, והיא זו שתצטרך לדאוג לעדכן את התפריט הנכון. שנו את התיאורים המילוליים של התנהגות המלצר והשפית בהתאם לגישה זו ועדכנו בהתאם גם את הפרויקט. תעדו ושמרו אותו בשם חדש.

**תרגיל 21:** בתרגיל זה נעסוק בחיפוש ברשימה.

א. בסעיף זה עליכם ליצור אנימציה שמציגה על הבמה רשימה של 10 מספרים בכל לחיצה על הדגל הירוק. כל אחד מעשרת המספרים נבחר אקראית מתוך תחום המספרים 1 עד 10. באנימציה משתתפת דמות של מורה למתמטיקה (דמות מתאימה היא למשל דמותו של ed מהתיקייה People). לאחר שהרשימה מוצגת על הבמה, המורה מבקש מהצופים



לבחור מספר מהרשימה. לאחר מכן הוא מחפש ברשימה את המספר שנבחר ומודיע באיזה מקום ברשימה הוא נמצא.

נסו להפעיל את האנימציה כמה פעמים:

בפעם הראשונה ענו לבקשתו של המורה עם המספר שנמצא במקום הראשון ברשימה.

בפעם השנייה השיבו עם המספר שנמצא במקום החמישי ברשימה.

בפעם השלישית השיבו עם המספר שנמצא במקום האחרון ברשימה.

מה קורה אם בהפעלה השנייה או השלישית המספר שנתתם כבר הופיע ברשימה קודם לכן?

כעת נסו לענות לבקשתו של המורה עם מספר שאינו מופיע ברשימה. הסבירו מה קורה

(רמז: יהיה לכם קל יותר להבין את המתרחש אם על הבמה יוצג המקום ברשימה שבו

המורה מחפש את המספר).

שם הקובץ: search1

ב. ההפעלות שערכתם בסעיף א' מראות כי תרגיל זה מזמן התמודדות עם בעיית חיפוש

מורכבת יותר מזו שהוצגה בפרק. ראשית, ייתכן שהפריט המבוקש מופיע ברשימה יותר

מפעם אחת. בנוסף, בבעיה שהוצגה בפרק החיפוש הוא בהכרח אחר פריט שנמצא

ברשימה. לכן מובטח שעד שנגיע לסוף הרשימה, הפריט יימצא. לעומת זאת, בבעיה

הנוכחית ייתכן שהפריט שאנו מחפשים אינו נמצא ברשימה כלל. במקרה זה עלינו לזהות

זאת ולתת הודעה מתאימה כמו "מצטער, המספר שנתת אינו מופיע ברשימה".

אם כך, בעיית החיפוש שאנו מתמודדים איתה כעת היא הבעיה הבאה: יש לחפש פריט

מבוקש בתוך רשימה נתונה. אם הפריט מופיע ברשימה, יש לתת הודעה המדווחת על

המקום הראשון ברשימה שבו הוא מופיע. אם הפריט אינו מופיע ברשימה, יש לתת

הודעה המדווחת שאין הוא נמצא ברשימה.

שנו את הפרויקט שכתבתם בסעיף א' כך שיטפל במקרה השני: במקום להשתמש

בביצוע חוזר מותנה, השתמשו בביצוע חוזר מוגבל מראש, כדי להבטיח שהמורה לא

ימשיך לחפש ברשימה מעבר לסופה, אחרי מספרים שאינם נמצאים בה.

שם הקובץ: search2

ג. הפעילו את האנימציה שיצרתם בסעיף ב' והשיבו למורה עם מספר שמופיע ברשימה. מה

מתרחש?

אם לא נהגתם זהירות מספקת בתכנון הפתרון שלכם, ייתכן שלאחר שהמספר נמצא

ולאחר שהופיעה הודעה על מציאתו תופיע הודעה נוספת שלפיה לא ניתן למצוא את

המספר!

האם כך קרה בפתרון שלכם? אם אכן כך קרה, הסבירו את הסיבה לכך. תקנו את הפתרון

כך שהמורה לא יודיע על אי־מציאתם של מספרים שמופיעים ברשימה.

שם הקובץ: search3

ד. בסעיף זה נתמודד עם בעיית חיפוש שונה: יש לחפש פריט מבוקש בתוך רשימה נתונה.

אם הפריט מופיע ברשימה, יש לתת הודעות המדווחות על כל המקומות ברשימה שבהם

הוא מופיע. אם הפריט אינו מופיע ברשימה, יש לתת הודעה המדווחת שאין הוא נמצא

ברשימה.

שנו את האנימציה האחרונה שיצרתם כך שאם המספר שסיפקתם למורה מופיע ברשימה, המורה יודיע על כל המקומות ברשימה שבהם הוא נמצא. אם המספר אינו מופיע ברשימה, המורה יודיע שלא הצליח למצוא אותו.

שם הקובץ: search4

ה. הפעילו את האנימציה שיצרתם בסעיף ד' וספקו למורה מספר שמופיע ברשימה. האם לאחר שהמורה דיווח על מיקומו (או מיקומיו) של המספר ברשימה, הוא נתן הודעה נוספת המדווחת שהמספר לא נמצא? אם אכן כך קרה, תקנו את הפרויקט כך שהודעה על אי-מציאת מספר תתבצע רק כאשר המספר אכן לא נמצא.

שם הקובץ: search5

רמז: השתמשו במשתנה נוסף.

ו. בסעיף ב' השתמשתם בביצוע מוגבל מראש במקום בביצוע חוזר מותנה כדי שתוכלו לפתור את בעיית החיפוש ברשימה, גם כאשר לא מובטח שהפריט המבוקש מופיע ברשימה. בגלל השימוש בביצוע חוזר מוגבל מראש, ייתכן שהביצוע החוזר ממשיך גם לאחר שהפריט המבוקש נמצא, משום שהוא טרם התבצע כמספר הפעמים שהוקצבו מראש. כתוצאה מכך ייתכן שלאחר שפריט נמצא ברשימה וניתנה הודעה על מציאתו, תופיע ההודעה שוב ושוב עד שיסתיים הביצוע החוזר.

בסעיף זה עליכם לפתור את סעיף ב' בדרך אחרת, המבטיחה שעבור איבר שמופיע ברשימה תינתן הודעה אחת בלבד על מיקומו, ועבור איבר שאינו מופיע ברשימה תינתן הודעה שלא ניתן למוצאו ברשימה. בפתרון זה לא תשתמשו בביצוע חוזר מוגבל מראש, אלא תשתמשו שוב בביצוע חוזר מותנה, בדומה למה שעשיתם בסעיף א'. אבל כדי שתוכלו לטפל היטב גם במספרים שאינם נמצאים ברשימה, יהיה הביצוע החוזר המותנה מסובך יותר מזה שכנראה נתתם בסעיף א'. התנאי המשולב בו יהיה תנאי מורכב. לצורך פתרון הסעיף מתאים להשתמש בתנאי המורכב **או** שבקבוצת המפעילים. התנאי **או** מתקיים אם התנאי המופיע בחלונית הראשונה מתקיים או אם התנאי המופיע בחלונית השנייה מתקיים או אם שני התנאים בחלוניות מתקיימים.

שם הקובץ: search6

ז. (מתקדם) ולבסוף, יש דרך נוספת לפתור את סעיף ב', דרך המשתמשת בביצוע חוזר מותנה, אך עם תנאי פשוט, ולא תנאי מורכב. הרעיון הוא כזה: נוודא שהמספר המבוקש בכל מקרה יופיע ברשימה. כיצד? לפני תחילת החיפוש נוסיף אותו בסופה של הרשימה. שנו את הפתרון שנתתם בסעיף א' כך שיפעל לפי הדרך המוצעת.

שם הקובץ: search7







## סיכום



בפרק זה ערכנו היכרות עם **רשימה**: מנגנון שמאפשר לנו לזכור ביחד קבוצה של ערכים מסוג מסוים, להוסיף ערכים לקבוצה, למחוק ממנה ערכים ולהציץ לתוכה.

רשימה היא מעין קופסה מודולרית בעלת כמה תאים המסודרים זה אחר זה, ולכל אחד מספר סידורי מתאים. כל תא יכול לשמור בתוכו ערך אחד בלבד. המודולריות של הקופסה מתבטאת בכך שניתן להוסיף לקופסה תאים ולהסיר ממנה תאים. **אורכה של רשימה** הוא מספר התאים שבה.


אחת הפעולות השימושיות בעבודה עם רשימות היא הפעולה של **חיפוש ברשימה אחר מקומו של פריט שנמצא בה**. לצורך ביצוע פעולת החיפוש יש להשתמש ב**משתנה מקום**. במהלך פעולת החיפוש נסרקת הרשימה פריט אחרי פריט, החל מהפריט הראשון, וכל פריט מושווה לפריט שמחפשים. הסריקה מסתיימת כשנמצא הפריט המבוקש. לשימוש במשתנה מקום לצורך חיפוש יש תבנית קבועה: הוא מאותחל ל-1, וערכו גדל בכל שלב ב-1.




**יצירת רשימה** מתבצעת ב-Scratch על ידי הוראת יצירה שנמצאת בקבוצת המשתנים האדומה-כתומה, . זאת אינה הוראה שנועדה לשילוב בתוך תסריט, אלא הוראה שנועדה להיות מופעלת בתוך Scratch, כשבוחרים בה בעזרת העכבר. כאשר מפעילים את ההוראה, נוספות לקבוצת המשתנים לבנים המתאימות להוראות שנועדו לשילוב בתסריטים והמאפשרות לשנות רשימה ולהשתמש בערכים השמורים בה. כמו כן נוספת לבנה שמייצגת את הרשימה עצמה.

**מחיקת פריט** או פריטים מתוך רשימה מתבצעת בעזרת ההוראה . **הצצה לתוך רשימה** מתבצעת בעזרת לבנת הערך . **שימוש בערכו של פריט מתוך רשימה** נעשה על ידי שילוב לבנת הערך  בתוך הוראות אחרות.

**הוספת פריט לרשימה** מתבצעת על ידי ההוראה . **בדיקת שייכות של פריט לרשימה** מתבצעת בעזרת התנאי . הלבנה המייצגת רשימה היא סוג של לבנת ערך. היא מאפשרת יצירת **תצוגה של הרשימה** על הבמה. כאשר מסומנת החלונת הקטנה שנמצאת ליד הלבנה שמייצגת רשימה, הרשימה מוצגת על הבמה. התצוגה כוללת את שם הרשימה, כל הפריטים שבה ואורכה.

ב-Scratch ניתן לבצע **מילוי של רשימה מתוך קובץ טקסט**. הקובץ יכול סדרת פריטים, כל פריט בשורה נפרדת. כדי למלא את הרשימה מתוך הקובץ, יש להצביע עם העכבר על הרשימה שעל הבמה, ללחוץ על הכפתור הימני של העכבר וברשימה שנפתחה לבחור בפריט "יבא...". בעקבות כך נפתח חלון ובו אפשר לבחור בקובץ הטקסט המתאים. הפעולה גורמת לרשימה להתמלא בפריטים שנמצאים בקובץ על פי סדר הופעתם בקובץ.

בפרק זה הכרנו **פעולת היפוך** שיוצרת מתנאי אחד את התנאי ההפוך לו. הלבנה המתאימה לפעולה זו ב-Scratch היא לבנת התנאי  מקבוצת המפעילים.

ב-Scratch ניתן לנהל דו-שיח בעזרת ההוראה  מקבוצת החיישנים ובעזרת הלבנה  (אף היא מקבוצת החיישנים). הלבנה  מייצגת את **משתנה התשובה**, משתנה מיוחד של Scratch שבו נשמרת התשובה לשאלה האחרונה שנשאלה.

## פרק 10 ביצוע מקבילי

האנימציות הנוצרות בסביבת Scratch משלבות מקביליות באופן טבעי. כבר בפרק 2 ראינו דוגמה לאנימציות שבהן פועלות מספר דמויות שתסריטיהן מבוצעים במקביל. בפרק 3 ראינו מקביליות מסוג שונה מעט, אולי מבלבל יותר, שבה לדמות אחת מספר תסריטים הפועלים במקביל. אין זה מפתיע שהמקביליות ב־Scratch היא טבעית ושימושית: גם בחיי היומיום הדמויות המקיפות אותנו פועלות במקביל. ובחיי היומיום גם קורה שכל אחד מאתנו מבצע מספר פעולות במקביל. ייתכן שאנו צופים בטלוויזיה ובמקביל משוחחים בטלפון או אוכלים.

למרות ההקשר הטבעי של המקביליות, כאשר אנו מתכננים אנימציה שבה פועלים במקביל כמה תסריטים, של אותה הדמות או של דמויות שונות, עלינו לעבוד בזהירות, כפי שראינו בפרק הקודם. השילוב בין תסריטים הפועלים במקביל עלול להביא לעתים לתוצאות שקשה לצפות אותן מראש ואשר עלולות לפגוע בהשגת מטרת האנימציה. בפרק זה נתמקד בנושא המקביליות, נדגים את התוצאות האפשריות משילוב מקבילי לא זהיר ונראה כיצד ניתן למנוע אותן.

### דוגמה: ארנבים מלומדים

באנימציה שניצור בפרק זה ישתתפו שני ארנבים, שיקראו מספר הכתוב על לוח ויכתבו אחר במקומו. כל ארנב מצויד בלוח קטן משלו, ובנוסף יש בחדר לוח גדול שעליו רשום מספר. כל ארנב ירוץ אל הלוח הגדול, יעתיק את המספר הרשום בו אל הלוח הקטן שלו ויחזור למקומו. כעת יגדיל כל ארנב, על הלוח הקטן שלו, את המספר הרשום בו ב־1, ירוץ שוב אל הלוח הגדול ויכתוב עליו, במקום המספר שמופיע עליו, את המספר החדש שחישב על הלוח שלו.

### הבסיס הגרפי של האנימציה

הכנו עבורכם פרויקט המכיל את כל הדמויות המתאימות ותלבושות עבורן, אך עדיין ללא תסריטים. בפרויקט כלולות שלוש דמויות: ארנב 1, ארנב 2 והמספר הכתוב על הלוח הגדול. הלוח הגדול אינו דמות בעצמו. הוא נמצא בחדר כחלק מהרקע שיצרנו עבור הבמה. [שם הקובץ: costumes-rabbits](#)

לכל אחת מהדמויות כמה תלבושות, שאותן תוכלו לראות אם תבחרו בדמות ותלחצו על לשונית התלבושות בחלון המרכזי.

לדמות המספר יש תלבושות המתאימות לערכים שונים של המספר, כלומר, ערכים שונים שיכולים להיכתב על הלוח הגדול. התלבושת הראשונה מתאימה לערך 0, השנייה לערך 1 והשלישית לערך 2.

לכל אחת מדמויות הארנבים כמה תלבושות המתאימות למצבים שונים של הלוח הקטן שמחזיק הארנב בידו, כלומר, לערכים שונים שיכולים להיות כתובים עליו במהלך ביצוע האנימציה. התלבושות – התלבושת הראשונה, השנייה והשלישית – של הארנבים מתאימות לערכים 0, 1 ו־2, בסדר זה. התלבושת הרביעית של כל אחד מהארנבים מתאימה ללוח ריק, שדבר לא כתוב עליו.

נשתמש בפרויקט זה כבסיס לאנימציה שניצור.

## לפי הסדר: קודם האחד ואחר כך האחר

בשלב ראשון ניצור גרסה פשוטה יחסית של האנימציה, שבה הארנבים אינם פועלים במקביל, אלא זה אחרי זה. רק אחר כך נאפשר להם לעבוד במקביל ונראה מה יכולות להיות תוצאותיה של המקביליות.

### משימה 1: ארנב 1 ואז ארנב 2

יש לבנות אנימציה שבה משתתפים שני הארנבים, ולכל אחד לוח קטן. בתחילה כל אחד מהארנבים יתייצב במקומו בפינת החדר, הלוח שבידו ריק, ועל הלוח במרכז החדר מופיע המספר 0.

לאחר מכן כל ארנב בתורו (קודם ארנב 1 ואחר כך ארנב 2) ירוץ אל הלוח הגדול, יעתיק את המספר הרשום שם אל הלוח שלו, יחזור לפינתו ויגדיל את המספר שעל הלוח שלו ב-1. אחר כך יחזור אל הלוח המרכזי, יעתיק אליו את המספר שעל הלוח שלו ויחזור למקומו.

שם הקובץ: sequential-rabbits



ובכן, ארנב 1 יתחיל. הוא ירוץ אל הלוח הגדול, יעתיק את המספר 0 אל הלוח שלו, יחזור למקומו, יגדיל את המספר על הלוח שלו ב-1, כך שעל הלוח שלו יופיע המספר 1. כעת ירוץ שוב אל הלוח הגדול ויכתוב עליו, במקום המספר שכתוב עליו, 0, את המספר שרשום על הלוח שלו, 1. הוא יחזור למקומו ויסמן לארנב 2 שהגיע תורו.

ארנב 2 יבצע בתורו סדרת פעולות דומה. הוא ירוץ אל הלוח הגדול ויעתיק אל הלוח שבידו את המספר שכתוב כעת על הלוח הגדול, הלוא הוא 1. הוא ירוץ למקומו ושם יגדיל ב-1 את המספר הכתוב על הלוח שלו. לכן כעת יהיה כתוב על הלוח שלו המספר 2. אז ירוץ שוב אל הלוח הגדול, יכתוב עליו, במקום המספר שכתוב עליו, 1, את המספר שרשום על הלוח שלו, 2, ויחזור למקומו.

אם כן, כאשר הארנבים פועלים האחד לאחר האחר, לפי תור, המספר על הלוח ישתנה מ-0 ל-1 ואחר כך מ-1 ל-2.

### התנהגות ארנב 1 – תיאור מילולי כללי

נתחיל לפתח את הפרויקט בהדרגה ונפנה בשלב ראשון לתיאור של התנהגותו של ארנב 1, תיאור שתואם את התיאור הכללי שנתנו קודם, אך מעט יותר מפורט ובנוי בצעדים.

0. אם אחיזה על הדגל היחיד;

1. הגי"צ בפני החדר

2. נקה את הלוח שלך

3. הוצא את הלוח שבמרכז החדר

4. העגק את המספר שבגובה הלוח הגדול אל הלוח שלך

5. חזור לפינת החדר

6. הגדל את המספר הרשום על הלוח שלך ב-1

7. חזור אל הלוח שבמרכז החדר

8. בגובה הלוח הגדול את המספר שרשום על הלוח שלך

9. חצור אפינה המדר

10. סמן לאינה 2 שהיזה גיו

התיאור הזה אמנם מזכיר את התיאורים המילוליים שכתבנו עד כה, אבל הוא מעט יותר כללי. הוא כולל חמישה צעדי תנועה: צעד 1 של ההתייבבות בפינת החדר לקראת תחילת הביצוע, וצעדים 3, 5, 7, 9 של ריצה למרכז החדר, חזרה לפינה ושוב ריצה למרכז החדר וחזרה לפינה. התרגום של צעדים אלה להוראות של Scratch, הוראות תנועה, הוא פשוט למדיי. גם את צעד 10 קל למדיי לתרגם על ידי שימוש במנגנון המוכר לנו, של העברת מסרים. אבל לא כך כשמדובר בתרגום של הצעדים 2, 4, 6 ו-8 להוראות של Scratch. לפני שנוכל לפנות לביצוע התרגום, נצטרך לפרק אותם לצעדים ממוקדים יותר.

### העבודה עם הלוחות

צעדים 2, 4, 6 ו-8 מתייחסים כולם ללוחות: הלוח האישי של הארנב או הלוח הגדול שבמרכז החדר. הפעולות המתבצעות כוללות שינוי של הערך הרשום על לוח זה או אחר והעתקת ערך מלוח ללוח. כיצד נוכל "לייצג" את שני הלוחות השונים? למעשה, הייצוג של כל לוח צריך להיות כפול: צריך לשמור ולזכור את הערך הרשום בו, כדי שנדע, למשל, להגדילו ב-1. לשם כך נוכל להשתמש במשתנים, משתנה עבור כל לוח, ש"יזכרו" את הערכים הרשומים על הלוחות ויעודכנו בהתאם. את הערכים הרשומים על כל לוח יש לא רק לזכור, אלא גם "להציגם" על הבמה, כשהם כתובים על הלוחות. להזכירם, הכנו לשם כך לכל דמות כמה תלבושות, אשר ההבדל ביניהן הוא בערך הרשום על הלוח שלה. לכן שינוי הערכים הכתובים על הלוחות שעל הבמה יתבצע על ידי החלפת תלבושת. הנה שוב התיאור המילולי המתאים להתנהגותו של ארנב 1, אך כאשר הצעדים המתאימים לעבודה עם הלוחות מפורטים יותר וכוללים את הטיפול במשתני הלוחות ואת החלפת התלבושות:

0. אם אמיצה לא הדגל היזוק:

1. הגי'צב הפינה המדר

2. אגמא אג ערך המשנה "האום ש'א" א-0

3. אגמא אג אגאבאג אגאבאג האום היזוק

4. כוזא אא האום שבארכ המדר

5. העגק אג עכבו ש המשנה "האום הקדא" אא המשנה "האום ש'א"

6. שנה אג אגאבאג אא המכילה אג הערך המדש ש המשנה "האום ש'א" (מספרה הסידורי הוא ערך המשנה +1).

7. חצור אפינה המדר

8. הקדא אג עכבו ש המשנה "האום ש'א" ב-1

9. שנה אג אגאבאג אא המכילה אג הערך המדש ש המשנה "האום ש'א" (מספרה הסידורי הוא ערך המשנה +1).

10. חצור אל האום שבמרכז החדר
11. העגק אג עככו של המעגנה "האום של" אל המעגנה "האום הקדול"
12. חצור אפינג החדר
13. עדי אאנה 2 מסר "גויך הקיע"

**תרגיל 1:** בתיאור המילולי החדש שינוי הערך הכתוב על הלוח הגדול שבמרכז החדר מתבטא רק בשינוי ערך המשתנה המתאים, אך לא בהחלפת תלבושת. מדוע?

**תרגיל 2:** תרגמו את התיאור המילולי שהתקבל לתסריט של Scratch. מיקום ארנב 1 בפינת החדר יהיה בנקודה  $(-180, -130)$ , וכאשר הוא רץ למרכז החדר עליו להגיע לנקודה  $(0, -60)$ . את צעדי הריצה מפינת החדר למרכז ובחזרה כדאי לבצע בעזרת הוראות גלישה. כדאי לשלב בתסריט הוראות המתנה לאחר הוראות הגלישה והוראות שינוי התלבושת, כדי שהאנימציה המתקבלת לא תהיה מהירה מדי וניתן יהיה לעקוב אחריה.

שימו לב: לצורך בניית התסריט תצטרכו ליצור שני משתנים, האחד עבור הלוח המרכזי והאחר עבור הלוח של ארנב 1. את משתנה הלוח המרכזי הגדירו לשימושן של כל הדמויות, אך את משתנה הלוח של ארנב 1 הגדירו רק לשימושה של דמותו של ארנב 1. הרי אף דמות אחרת לא תתעניין בערך זה! אין צורך להציג את ערכם של המשתנים על הבמה: השינוי על הבמה מתבטא בשינוי התלבושת, והשימוש במשתנים הוא החלטה פנימית שלנו שאינה מעניינה של הצופה.

## התנהגותו של ארנב 2

כיצד צריך לפעול ארנב 2? התנהגותו דומה מאוד לזו של ארנב 1, אך ישנם שני הבדלים: ראשית, את התנועה אל הלוח הגדול ובחזרה, תנועה המשולבת בשינוי הערכים הכתובים על הלוחות, עליו לבצע רק כאשר מגיע תורו. כלומר, כאשר מתקבל המסר שנשלח על ידי ארנב 1. בנוסף לאחר סיום תנועתו הוא אינו צריך לשלוח מסר, כי אין דמות נוספת שאמורה לפעול אחריו.

אם כך כאשר נלחץ הדגל הירוק כל שעליו לעשות הוא להתייבב בפינת החדר המיועדת לו, "לנקות" את הלוח שלו ולאתחל את המשתנה שמייצג את הלוח שלו. שאר פעילותו תתבצע בתסריט אחר המותנה בקבלת המסר. התיאור המילולי שמתאים לארנב 2 הוא זה:

0. עס אמיצה על הדגל היחיד:
1. הגי"צ הפינג החדר
2. אגמל אג עיך המעגנה "האום של" א-0
3. אגמל אג גאבועג אגאבועג האום היק

0. עם קבלת המסר "גורן היצ"ע":
1. הוצא אל האות שבמרכז המדר
2. העגק את ערכו של המשגנה "האות הקדו" אל המשגנה "האות של"י"
3. שנה את גאבוסק אלו המכילה את הערך המדש של המשגנה "האות של"י" (מספנה הסידוקי הוא ערך המשגנה +1).
4. חצור אפינא המדר
5. הקדא את ערכו של המשגנה "האות של"י" ב-1
6. שנה את גאבוסק אלו המכילה את הערך המדש של המשגנה "האות של"י" (מספנה הסידוקי הוא ערך המשגנה +1).
7. חצור אל האות שבמרכז המדר
8. העגק את ערכו של המשגנה "האות של"י" אל המשגנה "האות הקדו"
9. חצור אפינא המדר

## משתנים פרטיים

שימו לב כי בדומה לתיאור המילולי של ארנב 1, גם בתיאור המילולי של ארנב 2 נעשה שימוש בשני משתנים: המשתנה "הלוח הגדול" והמשתנה "הלוח שלי". המשתנה "הלוח הגדול" הוא אותו משתנה שבו משתמש ארנב 1, משום שיש רק לוח גדול אחד, והמשתנה שמתאים לו מותר לשימושן של כל הדמויות. אבל משתנה "הלוח שלי" של ארנב 2 אינו משתנה "הלוח שלי" של ארנב 1. הרי לכל אחד מהם לוח פרטי משלו!

האם אפשר להשתמש באותו שם – "הלוח שלי" – עבור שני המשתנים? כיצד נדע בכל פעם לאיזה משני המשתנים להתייחס?

הדבר אפשרי משום שכל אחד משני המשתנים האלו מותר לשימושה הבלעדי של דמות ארנב אחת בלבד. לכן בכל פעם שהמשתנה "הלוח שלי" מופיע בתסריטה של דמות ארנב, ברור שהכוונה היא למשתנה של ארנב זה. המשתנה "הלוח שלי" בתסריטו של ארנב 1 מתייחס תמיד למשתנה המותר לשימושו הבלעדי של ארנב 1, והמשתנה "הלוח שלי" בתסריטו של ארנב 2 מתייחס תמיד למשתנה המותר לשימושו הבלעדי של ארנב 2. נוכל לדמיין כאילו לשני המשתנים שם פרטי זהה – "הלוח שלי" – אבל לכל אחד מהם גם שם משפחה, הוא שם הדמות שעבורה נוצר, ושמות המשפחה מבדילים בין שני המשתנים.

### מושג חדש: משתנה פרטי

משתנה פרטי הוא משתנה שהוגדר לשימושה הבלעדי של דמות מסוימת. משום כך, רק היא מכירה אותו. ניתן לדמיין זאת כאילו שם הדמות הוא חלק משמו של המשתנה. משום כך אפשר שלדמות אחרת יהיה משתנה פרטי משלה הנושא את אותו שם, כי שמם המלא של שני המשתנים (הכולל גם את שם הדמות שעבורה נוצר) יהיה שונה.



### תרגיל 3:

תרגמו את התיאור המילולי שהתקבל לתסריטים של Scratch. מיקום ארנב 2 בפינת החדר יהיה בנקודה (130, -180), וכאשר הוא רץ למרכז החדר, עליו להגיע לנקודה (0, 60). זכרו לשלב בתסריט הוראות המתנה כדי שניתן יהיה לעקוב בנוחות אחרי האנימציה המתקבלת.

### שינוי הערך הכתוב על הלוח המרכזי

פרט לשני הארנבים משתתפת באנימציה גם דמות המספר. תפקידה לא גדול, אבל חשוב מאוד. הרי, כפי שעלה בתרגיל 1, כאשר שני הארנבים "מעתיקים" את הערך הרשום על הלוח שלהם אל הלוח הגדול, הם יכולים רק לעדכן את ערכו של המשתנה המתאים, המשתנה "הלוח הגדול". אף דמות אינה יכולה לשנות את תלבושתה של דמות אחרת, ולכן תסריטיהם של שני הארנבים אינם גורמים להצגת הערך החדש על הלוח הגדול, על הבמה. שינוי מראה הלוח הגדול הוא באחריותה של דמות המספר.

כיצד תדע דמות המספר לאיזו תלבושת עליה להחליף? את התלבושת החדשה תוכל דמות המספר לזהות על פי ערכו של המשתנה "הלוח הגדול". כאשר ערכו 0, עליה "ללבוש" את התלבושת הראשונה; כאשר ערכו 1, עליה "ללבוש" את התלבושת השנייה; וכאשר ערכו 2 – את התלבושת השלישית. התלבושת המתאימה היא תמיד זו שמספרה הסידורי גדול ב-1 מערך המשתנה.

מתי צריכה דמות המספר לבצע החלפת תלבושת? כיצד תדע הדמות מתי שינה אחד הארנבים את ערך המשתנה המתאים ללוח הגדול?

דרך אחת היא לפתור זאת על ידי תקשורת בין הדמויות שמשתמשת בהעברת מסרים: על פי שיטה זו, בכל פעם שאחד הארנבים משנה את ערכו של המשתנה "הלוח הגדול" הוא ישלח מסר של "ערך חדש". לדמות המספר יהיה תסריט המותנה בקבלת המסר "ערך חדש" שמחליף את תלבושתה בהתאם לערך המשתנה "הלוח הגדול".

בפרק 7 ראינו כי דמויות יכולות להתקשר ביניהן גם דרך תיבת דואר, בעזרת משתנים. במקרה שלנו כבר מוכנה עבורנו תיבת דואר כזו, המשתנה "הלוח הגדול". כל שעל דמות המספר לעשות הוא לבדוק שוב ושוב את התיבה ולשנות את תלבושתה בהתאם לערך שבתיבת הדואר. אם כן על דמות המספר לפעול לפי התיאור המילולי הבא:

0. אם אחיזה על הדגל הימני:

1. אגמא אג ערך המשנה "הלוח הגדול" ל-0

2. אגמא אג גלאבוסגך אגלאבוסגך הכאשונה (מספרה הסידורי הוא ערך המשנה "הלוח הגדול" + 1)

3. תצור שוב ושוב:

3.1 שנה אג אגלאבוסגך אגלאבוסגך שמספרה הסידורי הוא ערך המשנה "הלוח הגדול" + 1

### תרגיל 4:

תרגמו את התיאור המילולי של דמות המספר לתסריט של Scratch. בדקו את הפרויקט שבניתם, תעדו ושמרו.

## ועכשיו כולם ביחד

כעת נגרום לארנבים לפעול במקביל, ולא זה אחר זה.

### משימה 2: הארנבים פועלים במקביל

שנו את האנימציה כך ששני הארנבים יעבדו במקביל. כלומר, ארנב 2 לא ימתין עד שארנב 1 יסיים את חלקו, אלא שניהם יתחילו לפעול באותו זמן, מיד עם הלחיצה על הדגל הירוק.

שם הקובץ: `concurrent-rabbits`



השינוי הנדרש בהתנהגותם של הארנבים נראה פשוט, ועיקרו הוא בזה שאין עוד צורך בשימוש במסר לשם הסדרת התור. לכן יש להוריד את הוראת שידור המסר מתסריטו של ארנב 1. בנוסף ארנב 2 לא צריך להמתין עוד לקבלת המסר, ולכן נוכל לבטל את תסריטו השני. את ההוראות הכלולות בו נעביר לתסריט המתבצע עם הלחיצה על הדגל הירוק, אחרי הוראות האתחול שכבר כלולות בו. למעשה, כעת יש לשני הארנבים תסריטים זהים, הנבדלים ביניהם רק בנקודות היעד שמשולבות בהוראות התנועה.

**תרגיל 5:** בצעו את השינויים שתוארו כאן כך ששני הארנבים יפעלו במקביל. תעדו את הפרויקט החדש ושמרו אותו. הפעילו אותו וראו מה הערך הרשום על הלוח הגדול בתום ביצוע האנימציה.

בגרסה זו בסיום האנימציה רשום על הלוח הגדול הערך 1. זאת בגלל שהארנבים פועלים במקביל ובתיאום מושלם: שניהם מגיעים ביחד אל הלוח הגדול ומעתיקים אל הלוחות שלהם את הערך 0, שניהם מגדילים את הערך על הלוח שלהם מ-0 ל-1, ושניהם מגיעים ביחד אל הלוח הגדול ומשנים את הערך הרשום עליו ל-1.

### עדיין ביחד, אבל הקצב שונה

התיאום המושלם להפליא בהפעלה המקבילית של הארנבים אינו סביר. הרי בדרך כלל דמויות אמיתיות אינן פועלות בדיוק באותו קצב. בואו ננסה לראות מה קורה כששני הארנבים פועלים במהירויות שונות.

### משימה 3: ארנב אחד מהיר, והאחר אטי

שנו את האנימציה כך ששני הארנבים נעים במהירויות שונות. למשל, נחליט באופן שרירותי שהדרך של ארנב 1 מהפינה שלו אל מרכז החדר אורכת מחצית מפרק הזמן הקודם. ארנב 2 אטי יותר, והדרך מהפינה שלו אל מרכז החדר אורכת פי 5 מזו של ארנב 1. כך גם לגבי הדרך בחזרה, ממרכז החדר אל הפינה.

שם הקובץ: `different-speeds`



**תרגיל 6:** בצעו את השינויים שתוארו כאן, כך ששני הארנבים יפעלו במקביל, בעוד האחד אטי פי 5 מרעהו. תעדו את הפרויקט החדש ושמרו אותו. הפעילו אותו וראו מה הערך הרשום על הלוח הגדול בתום ביצוע האנימציה.

בגרסה זו למרות ששני הארנבים פועלים במקביל, כתוב על הלוח הגדול בסיום האנימציה אותו הערך שנכתב עליו בסיום ביצוע האנימציה הסדרתית, הערך 2. זאת בגלל שארנב 2 אטי מאוד ביחס לארנב

1. אמנם שניהם יוצאים לדרך ביחד, אבל עד שארנב 2 מספיק להגיע ללוח הגדול, כבר הספיק ארנב 1 להגיע אל הלוח הגדול, להעתיק את הערך 0 הכתוב עליו ללוח שלו, להגדיל את הערך על הלוח שלו ל-1, לרוץ שוב למרכז החדר ולשנות את הכתוב על הלוח הגדול ל-1. כשארנב 2 מגיע, הוא מעתיק ללוח שלו את הערך 1 ובסופו של דבר משנה את הכתוב על הלוח הגדול ל-2.

## עדיין ביחד, הקצב שונה, אבל מפתיע

למעשה, גם הגרסה הקודמת אינה סבירה. בדרך כלל אין דמויות אמיתיות מצליחות לשמור בדיוק על אותו הקצב. פעם הן מהירות יותר ופעם אטיות יותר. אם כך ננסה לראות מה קורה כאשר מהירותו של כל אחד מהארנבים משתנה באופן אקראי מהפעלה להפעלה. עדיין לעולם יהיה ארנב 1 מהיר יותר מארנב 2, אך ייתכן כי יהיה מהיר מאוד, וייתכן כי יהיה מהיר ממנו רק במעט.

## משימה 4: ארנב אחד מהיר והאחר אטי, אבל ההפרש משתנה

שנו את האנימציה כך ששני הארנבים נעים במהירויות שונות אך אקראיות. עם זאת עדיין מובטח כי לעולם יהיה ארנב 1 מהיר מארנב 2.

שם הקובץ: random-speed



נוכל להשתמש במנגנון ההגרלה של מספר אקראי שבו השתמשנו בפרקים 7, 8 ו-9. לפני תחילת התנועה, כחלק משלב האתחול, יבחר לו כל ארנב באופן אקראי "מהירות" תנועה משלו. למעשה, הוא יבחר באופן אקראי את פרק הזמן שדרוש לו כדי להגיע מפנית החדר שלו אל מרכזו של החדר, פרק הזמן שבו ישתמש בהוראות הגלישה שבהמשך תסריטו. נחליט כי עבור ארנב 1 פרק זמן זה יכול להיות 1, 2 או 3 שניות, ואילו עבור ארנב 2 פרק זמן זה יכול להיות 6, 7 או 8 שניות.

לאחר שהגריל מספר אקראי בתחום "שלו", ישמור אותו כל ארנב במשתנה חדש, ששמו למשל "speed". בכל הוראות הגלישה שבהמשך התסריט, תשולב הלבנה המייצגת את המשתנה החדש בחלונות המתאימה לזמן הגלישה.

**שימו לב:** גם הפעם ניתן להשתמש באותו שם עבור שני המשתנים של שני הארנבים, משום שהם יהיו משתנים פרטיים. כלומר, המשתנים החדשים צריכים להיות מותרים לשימוש בלעדי – כל ארנב יוכל להשתמש רק במשתנה שלו. בכך הם הופכים למשתנים שיש להם גם שם משפחה, הוא שם הדמות שאליה הם שייכים; ולכן למרות ששם הפרטי זהה, שם המלא אינו זהה.



בצעו את השינויים שתוארו כאן, כך ששני הארנבים יפעלו במקביל, אך בקצב אקראי. הציגו את מהירותם של הארנבים על הבמה (שימו לב לשמות שליד התצוגות). תעדו את הפרויקט החדש ושמרו אותו. הפעילו אותו כמה פעמים וראו מה הערך הרשום על הלוח הגדול בתום ביצוע האנימציה.

## תרגיל 7:

בגרסה זו למרות ששני הארנבים פועלים במקביל, ייתכן שבתום הביצוע של ההפעלות השונות נמצא ערכים שונים על הלוח הגדול. בגלל גורם האקראיות איננו יכולים לומר מראש אילו ערכים אלו יהיו בכל הפעלה והפעלה, אך סביר להניח שבמספר הפעלות רצופות נקבל ערכים שונים. אילו ערכים אפשר שנקבל?

אם ארנב 2 אטי מאוד ביחס לארנב 1, בדומה לגרסה הקודמת (למשל, כאשר מהירותו של ארנב 1 היא 1 ומהירותו של ארנב 2 היא 6), יתקבל הערך 2.

אבל אם הפער ביניהם אינו כה גדול, למשל כאשר מהירותו של ארנב 1 היא 3 ומהירותו של ארנב 2 היא 6, יכול להתקבל הערך 1: אמנם ארנב 1 מגיע ללוח הגדול ראשון ומעתיק ללוח שלו את הערך 0, אבל עוד לפני שהוא מספיק לחזור למרכז החדר ולשנות את הכתוב על הלוח הגדול ל-1, מגיע ארנב 2 אל הלוח הגדול, ואף הוא מעתיק אל הלוח שלו את הערך 0. ארנב 1 יחזור ראשון אל הלוח הגדול ויכתוב עליו את הערך 1, אבל אז יגיע אליו ארנב 2 וירשום ערך שגדול ב-1 מהכתוב על הלוח האישי שלו; לכן גם הוא יחזור ויכתוב על הלוח הגדול את הערך 1.

## ביצוע מקבילי ושילובים אפשריים שונים

ראינו שבביצוע מקבילי מהירותן היחסית של הדמויות יכולה להשפיע על התוצאה המתקבלת, משום שההוראות השונות המתבצעות על ידי כל אחת מהדמויות משתלבות בסדר שונה. למשל, כאשר ארנב 1 מהיר מאוד ביחס לארנב 2, מתקבל הסדר הבא המביא לכך שבסיום האנימציה כתוב על הלוח הגדול הערך 2 (פעולות של ארנב 1 צבועות בכחול, ופעולות של ארנב 2 צבועות בירוק):

- ארנב 1 מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- ארנב 1 חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב-1.
- ארנב 1 חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.
- ארנב 2 מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- ארנב 2 חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב-1.
- ארנב 2 חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.

אבל כאשר ארנב 1 מהיר רק במעט ביחס לארנב 2, יכול להתקבל הסדר הבא, שיביא לכך שבסיום האנימציה יהיה כתוב על הלוח הגדול הערך 1:

- ארנב 1 מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- ארנב 2 מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- ארנב 1 חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב-1.
- ארנב 2 חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב-1.
- ארנב 1 חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.
- ארנב 2 חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.

עבור יחסי מהירויות מסוימים יכול גם להתקבל הסדר הבא, וגם הוא יביא לכך שבסיום האנימציה יהיה כתוב על הלוח הגדול הערך 1:

- **ארנב 1** מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- **ארנב 1** חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב־1.
- **ארנב 2** מגיע אל הלוח הגדול ומעתיק אל הלוח שלו את הערך שכתוב עליו.
- **ארנב 1** חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.
- **ארנב 2** חוזר לפינת החדר ומגדיל את הערך על הלוח שלו ב־1.
- **ארנב 2** חוזר ללוח הגדול ומעתיק אליו את הערך שכתוב על הלוח שלו.

כל ארנב מבצע בכל הפעלה בדיוק אותה סדרת פעולות, אבל מהפעלה להפעלה יכול להשתנות ה**שילוב** בין שתי הסדרות – זו של ארנב 1 וזו של ארנב 2 – ובסך הכול הסדרה המתקבלת תהיה שונה, והיא עלולה גם להביא לתוצאות שונות.

## ביצוע מקבילי בעולם האמיתי

בדוגמת הארנבים יצרנו באופן מלאכותי את התופעה של שילובים שונים. עשינו זאת על ידי כך שנתנו לכל ארנב מהירות משלו, מהירות שיכולה להשתנות מעט מהפעלה להפעלה. אבל בעצם, תופעה דומה יכולה להתרחש בכל פעם שתסריטים פועלים במקביל.

הסיבה היא שהמחשב שעליו אנו מפעילים את האנימציה אינו באמת מחשב מקבילי. הוא יודע לבצע בכל רגע ורגע פעולה אחת בלבד. כמובן, משום שמחשבים הם מהירים מאוד, פרק הזמן שבו מתבצעת פעולה יחידה הוא קצר מאוד. התסריטים מבוצעים "במקביל" כך שבכל פעם מבוצעת ההוראה הבאה בתור (או כמה הוראות) מתסריט אחר. הביצוע "קופץ" מתסריט לתסריט: יש התקדמות מסוימת בתסריט אחד, אחר כך נעצר הביצוע של תסריט זה, ויש התקדמות בתסריט אחר; לאחר מכן מגיע תורו של תסריט שלישי, או אולי שוב של התסריט הראשון וכך הלאה, עד שכל התסריטים מגיעים לסיומם. בגלל שפרקי הזמן הם קצרים כל כך, נראה לנו שהתסריטים השונים אכן מבוצעים במקביל. למעשה, סדרות ההוראות של כל התסריטים משולבות לסדרה אחת ארוכה, אבל סדר השילוב יכול להיות שונה מהפעלה להפעלה. מובטח שהוראותיו של כל תסריט ותסריט יבוצעו לפי הסדר שלו, אבל האופן שבו יתבצע השילוב הכולל אינו מובטח.

### מושג חדש: שילובים אפשריים

כאשר תסריטים מבוצעים במקביל, הוראותיהן משולבות לסדרת הוראות אחת. ייתכנו שילובים רבים, ומובטח רק כי הסדר המקורי נשמר בין ההוראות הלקוחות מאותו תסריט.

מדוע זה חשוב? אם בסופו של דבר זה נראה לנו כביצוע מקבילי לכל דבר, מדוע סדר השילוב יכול לשנות?

הדוגמה המלאכותית של הארנבים הראתה לנו כי שילובים שונים יכולים להביא גם לתוצאות שונות. כאשר אנו בונים פרויקט ובו תסריטים העובדים במקביל, עלינו לנסות לחשוב מראש על כל השילובים האפשריים ולוודא כי בכל שילוב אפשרי תמיד תתקבל תוצאה שמתאימה לנו. נדגים זאת בעזרת אחת מהאנימציות מהפרקים הקודמים.

## מבט נוסף אל פקמן – דגש על הביצוע המקבילי

בפרק 7 בנינו משחק מחשב נחמד ובו דמותו של פקמן. כבר בשלבים התחלתיים למדיי של פיתוח המשחק היו לדמות של פקמן שני תסריטים שעבדו במקביל: תסריט אחד שאחראי על פתיחה וסגירה של פיו של פקמן (בעזרת שינוי תלבושות), ותסריט אחר שאחראי על התנועה במבוך.

להזכירכם, בשני התסריטים שילבנו את הוראות האתחול (התייצבות בפינת המבוך בכיוון המתאים). לכאורה, מספיק היה למנות את אחד התסריטים כאחראי על פעולת האתחול. מדוע "שכפלנו" את אותה סדרת הוראות אתחול בשני התסריטים?

נתבונן למשל בגרסת הביניים של המשחק שבה כבר ידע פקמן לזהות התנגשות בקיר. כאשר התנגש בקיר אמר "איי", וכל עוד לא זיהה התנגשות בקיר, המשיך בהתקדמותו. נניח שהיינו משלבים את הוראות האתחול רק בתסריט המטפל בהחלפת התלבושות. בתסריט האחראי על התנועה לא היינו כוללים הוראות אתחול, אלא רק את הביצוע החוזר האינסופי שבו נבדקת שוב ושוב נגיעה בקיר.

כעת נדמיין שהיינו מפעילים פרויקט זה על ידי לחיצה על הדגל הירוק. פקמן היה נע ישר במבוך, אגב פתיחה וסגירה של פיו חליפות, עד שהיה מתנגש בקיר, עוצר ואומר "איי" שוב ושוב. בלחיצה חוזרת על הדגל הירוק היינו מצפים שיתבצע שוב אתחול, ושפקמן יחזור לתחילת המבוך לפני שהוא מתחיל בתנועתו, המשולבת בבדיקת התנגשות בקיר.

אבל, התסריטים פועלים במקביל, והאופן שבו הם משולבים אינו ידוע!

ייתכן שמיד עם הלחיצה על הדגל הירוק תתבצע התקדמות דווקא בתסריט האחראי על התנועה. מאחר שטרם בוצע אתחול, כי טרם התבצעה הוראה מתסריט החלפת התלבושות, פקמן נמצא עדיין במקום שבו סיים את הביצוע הקודם, כלומר, לאחר שהתנגש בקיר. אם כך תסריט התנועה יזהה מיד התנגשות, וכתוצאה מכך יאמר פקמן "איי". עכשיו תתבצע התקדמות בתסריט האחראי על החלפת התלבושות ויתבצע אתחול מיקום וכיוון. תסריט התנועה יזהה כי כעת אין נגיעה בקיר והוא יתחיל לקדם את פקמן, בעודו אומר עדיין "איי"...

בואו נראה בפירוט כיצד נוצר שילוב זה. תסריט החלפת התלבושות מייצר את הסדרה האינסופית הבאה של הוראות:

- אתחול מיקום
- אתחול כיוון
- אתחול תלבושת
- המתן
- שנה תלבושת
- המתן
- שנה תלבושת
- המתן
- 
- 
-

לעומתו תסריט התנועה מייצר את סדרת ההוראות האינסופית הבאה:

- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- 
- 
- 

אנו מעוניינים בשילוב שנותן סדרה שבה קודם כל מופיעות הוראות האתחול, למשל כך (פעולות של תסריט החלפת התלבושות צבועות בכחול ופעולות של תסריט התנועה צבועות בירוק):

- אתחל מיקום
- אתחל כיוון
- אתחל תלבושת
- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- המתן
- שנה תלבושת
- המתן
- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- שנה תלבושת
- המתן
- 
- 
-

אבל מאחר שסדר השילוב אינו ידוע ודבר אינו מובטח לגביו, ייתכן בהחלט גם השילוב הבא:

- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- אתחל מיקום
- אתחל כיוון
- אתחל תלבושת
- המתן
- שנה תלבושת
- המתן
- אם נוגע בקיר
- אמור "איי"
- אחרת זוז 2 צעדים
- שנה תלבושת
- המתן
- 
- 
- 

שילוב כזה המתבצע לאחר הפעלה שבה פקמן התנגש בקיר, יכול להביא לאנימציה שבה פקמן אומר "איי" מיד עם הלחיצה על הדגל הירוק וממשיך בכך כל עוד הוא נע במבוך.

**תרגיל 8:** שנו על פי המתואר את הפרויקט המתאים מפרק 7 (בלי לפגוע בפרויקט המקורי). כלומר, מחקו את אתחול המיקום והכיוון מתסריט התנועה והשאירו אותו רק בתסריט החלפת התלבושות. נסו להפעיל את הפרויקט כמה פעמים על ידי לחיצה חוזרת על הדגל הירוק. אם לא קיבלתם את ההתנהגות שתוארה כאן, דאגו שבחלון התסריטים יופיע תסריט החלפת התלבושות לפני תסריט התנועה (על ידי גרירת תסריט החלפת התלבושות מעל תסריט התנועה), ונסו להפעיל את הפרויקט שנית.





**שימו לב:** המסמכים המצורפים לסביבת Scratch אינם מפרטים את האופן שבו תסריטים העובדים במקביל משולבים לסדרת הוראות אחת. ניסיונו מראה כי השילובים אינם אקראיים לגמרי, וכך למשל, ייתכן כי הסדר של התסריטים בחלון התסריטים ישפיע על האופן שבו הם ישולבו זה בזה.

בעצם, אין זה חשוב בדיוק אילו שילובים אפשריים ב־Scratch. הרי תמיד ייתכן שבגרסאות חדשות יותר יחליטו על דרך אחרת לשילוב. הדבר הנכון הוא לא להסתמך על כך שב־Scratch תסריטים משולבים כך או אחרת, אלא להניח כי כל השילובים ייתכנו ולדאוג כי הפרויקטים שלנו יעבדו תמיד נכון, יהיה השילוב שנוצר אשר יהיה.

כדי למנוע ביצועים כאלו אשר אינם רצויים מבחינתנו, הכנסנו את סדרת האתחול לשני התסריטים. בכך וידאנו כי כאשר לוחצים על הדגל הירוק, תמיד תתבצע בדיקת ההתנגשות רק לאחר החזרה לתחילת המבוק.

### מושג חדש: נכונות של תוכניות המשתמשות במקביליות

כדי שנוכל לקבוע שפרויקט הוא נכון, עלינו לוודא שהוא יפעל נכון בכל שילוב אפשרי של תסריטיו.

## פתרון הבעיה על ידי הכנסת סדר

ראינו כי הדרך למנוע שילובים לא רצויים היא לכתוב את התסריטים כך שלא יתאפשרו שילובים כאלה. בגישה דומה אפשר לנקוט ביחס לאנימציות הארנבים.

בגרסתה האחרונה של אנימציה זו הארנבים פועלים במקביל, כך שייתכנו שילובים אפשריים רבים של פעולותיהם (שילובים שהשגנו באופן מלאכותי על ידי הכנסת אקראיות ביחס למהירותם). כתוצאה מכך ייתכנו תוצאות רבות. אנו מעוניינים שהתוצאה תהיה תמיד אותה התוצאה. כלומר, תמיד בתום ביצוע האנימציה יופיע על הלוח הערך 2, כאילו הארנבים פעלו זה לאחר זה, יהיה היחס בין מהירותם של הארנבים אשר יהיה.

לפיכך עלינו למנוע בדרך כלשהי שילובים שבהם אחד הארנבים מספיק להעתיק את הערך מהלוח הגדול ללוח שלו; אך לפני שהוא מספיק לעדכן את הלוח הגדול, הארנב השני כבר הספיק להעתיק את אותו ערך ללוח שלו.

## משימה 5: הכנסת סדר בבלגן

שנו את הפרויקט שנכתב עבור משימה 2 כך שלא יתאפשרו ביצועים בהם אחד הארנבים מתחיל לעבוד עם הלוח הגדול, והארנב השני ניגש אל הלוח הגדול עוד לפני שהארנב הראשון סיים לכתוב עליו את הערך החדש שחישב.



שם הקובץ: `key-for-synchronization`

כיצד נמנע שילובים כאלה?

נכניס סדר בבלגן ונאלץ כל ארנב לבקש רשות לפני שהוא ניגש ללוח הגדול. הרשות תינתן לו רק אם הארנב השני אינו נמצא כרגע בתהליך עבודה עם הלוח הגדול. כלומר, לעולם לא יקרה מצב שבו שני

הארנבים עובדים עם הלוח הגדול במקביל.

כדי להבין זאת, נדמיין כי מעל הלוח הגדול יש נורה גדולה, אדומה (תפוס) או ירוקה (פנוי). ארנב יוכל להתחיל עבודה עם הלוח הגדול (כלומר, העתקת הכתוב על הלוח הגדול אל הלוח שלו) רק אם הלוח פנוי, אם הנורה ירוקה. כאשר ארנב מתחיל לעבוד עם הלוח הגדול, הנורה הופכת לאדומה ומסמנת בכך לארנב השני שהוא אינו יכול כרגע לגשת אל הלוח הגדול. רק כאשר הארנב הפועל עם הלוח הגדול סיים להעתיק את הערך החדש אל הלוח הגדול, הוא יסמן שהשטח פנוי, הנורה תהפוך שוב לירוקה, והארנב השני יקבל רשות להתקדם. בכך אנו בעצם מונעים ביצוע מקבילי של קטעים מסוימים, ומכריחים את הקטעים האלה להתבצע באופן סדרתי. ניתן לומר כי אנו הופכים את הפרויקט למעט פחות מקבילי.

אם כך נוכל לתאר את פעולתו של כל ארנב כך:

0. אם למי3ה על הדגל היחיד:
1. הגי3ה הפינה המדר
2. אגמא אל ערך המשנה "הלוח שלי" 0-1
3. אגמא אל גאבוסגך אגאבוסג הלוח היק
4. בקש רשוג אלגשג אללוח והמגן עד שהרשוג ניגמ
5. חזר אל הלוח שהמכס המדר
6. העגק אל ערכו של המשנה "הלוח הגדול" אל המשנה "הלוח שלי"
7. שנה אל גאבוסגך אלזו המכילה אל הערך המדרש של המשנה "הלוח שלי" (מספנה הסידורי הוא ערך המשנה + 1).
8. חזר אלפינה המדר
9. הגדל אל ערכו של המשנה "הלוח שלי" ב-1
10. שנה אל גאבוסגך אלזו המכילה אל הערך המדרש של המשנה "הלוח שלי" (מספנה הסידורי הוא ערך המשנה + 1).
11. חזר אל הלוח שהמכס המדר
12. העגק אל ערכו של המשנה "הלוח שלי" אל המשנה "הלוח הגדול"
13. חזר אלפינה המדר
14. חודע כי סיימא אלכוד עם הלוח הגדול

## מפתח שמנהל את העניינים

כיצד נייצג על הבמה את תהליך בקשת הרשות, הענקתה והודעת סיום עבודה?

אפשרות אחת היא להשתמש, כמו בהסבר שנתנו, בנורות שיופיעו מעל הלוח – נורה אדומה ונורה ירוקה. בהתאם למצב הנורות אפשר יהיה לראות אם הלוח פנוי או תפוס. החיסרון של דרך הצגה זו הוא שהיא אינה מאפשרת לנו לראות בבירור מי הוא זה שקיבל את רשות העבודה עם הלוח, ארנב 1 או ארנב 2.

לכן נבחר באופן הצגה אחר ונייצג על הבמה את הענקת הרשות לפעולה בעזרת מפתח. המפתח ימוקם בין שני הארנבים, וכאשר אחד מהם מקבל רשות לעבודה עם הלוח הגדול, יפנה המפתח לכיוונו. כאשר הלוח פנוי, יפנה המפתח למרכז. כך נוכל לראות על הבמה אם הלוח פנוי או תפוס, ואם הוא תפוס – נוכל לראות גם מי הארנב שפועל ומי ממתין.

כלומר, במקום נורות נייצג את הרשות על ידי חפץ, במקרה זה מפתח. מי שמחזיק בחפץ – או במקרה שלנו, מי שהמפתח פונה לכיוונו – רשות הפעולה היא שלו. שיטה זו נהוגה לעתים בדיונים; על השולחן יש מקל, ומי שמחזיק במקל – רשות הדיבור היא שלו.

דמות המפתח היא זו שתהיה אחראית לנהל את המנגנון של בקשת הרשות והענקתה. למשל, התקשורת של דמות המפתח עם ארנב 1 תתבצע על פי התיאור המילולי הבא:

0. כאשר ארנב 1 מקבל רשות

1. מכה על שולחן הגדול מאחורי

2. סמן אלצמך שולחן הגדול גפוס

3. פנה לכיוון ארנב 1

0. כאשר ארנב 1 מסיים לעבוד

1. פנה לכיוון המרכז

2. סמן אלצמך שולחן הגדול פנוי

### מסרים ככלי עזר להשלטת סדר

בקשת הרשות ואיתות כי העבודה הסתיימה יתבצעו בעזרת מסרים. כלומר, כאשר ארנב 1 למשל רוצה לבקש רשות לעבוד עם הלוח הגדול, הוא ישלח מסר מתאים, וכאשר הוא מסיים את עבודתו, הוא ישלח מסר אחר. ארנב 2 ישתמש בשני מסרים דומים משלו.

בנוסף תשתמש דמות המפתח במשתנה משלה, busy, שנועד לשימושה הבלעדי, ובעזרתו תסמן לעצמה אם הלוח הגדול תפוס או פנוי. כאשר ערכו של המשתנה 0 – הלוח פנוי, וכאשר ערכו 1 – הלוח תפוס.

בהתאם לבחירה במפתח לצורך בקשת רשות והענקתה, ובהתאם להחלטה על שימוש במסרים, נשנה את התיאור המילולי שלפיו יפעל ארנב 1. נשתמש באותו תיאור מילולי שכתבנו קודם, אך את צעדים 4 ו-14, של בקשת הרשות וקבלתה ושל הודעה על סיום עבודה, נכתוב שוב בהתאם לשיטה שבה בחרנו:

0. אם אחיזה על הדגל הידוק;

1. הגייה בפיה המדבר

2. אגמא אל ערך המשנה "הלוח שלי" ל-0

3. אגמא אל גאבושג אלבושג הלוח הידוק

4. שלח מסר "ארנב 1 מקבל רשות" והמתן

5. כוץ אל הלוח שבמרכז המדבר

6. העגק אג ערכו של המשנה "האומ הקדוא" אל המשנה "האומ שאי"
7. שנה אג גאקושהג אצו המכילה אג הערך המדש של המשנה "האומ שאי" (מספנה הסידוי הוא ערך המשנה +1).
8. חצור אפינג המדר
9. הקדא אג ערכו של המשנה "האומ שאי" ב-1
10. שנה אג גאקושהג אצו המכילה אג הערך המדש של המשנה "האומ שאי" (מספנה הסידוי הוא ערך המשנה +1).
11. חצור אל האומ שהמרכש המדר
12. העגק אג ערכו של המשנה "האומ שאי" אל המשנה "האומ הקדוא"
13. חצור אפינג המדר
14. שאלח מסר "אנה 1 סיימ אעבוד עס האומ הקדוא"

נעדכן גם את התיאור המילולי של התקשורת של דמות המפתח עם ארנב 1:

0. עס קבאל המסר "אנה 1 מקקש רשג"
1. המגן עד שערכו של המשנה busy הוא 0
2. שנה אג ערכו של המשנה busy א-1
3. פנה לכיוון אנה 1

0. עס קבאל המסר "אנה 1 סיימ אעבוד עס האומ הקדוא"
1. פנה לכיוון המרכש
2. שנה אג ערכו של המשנה busy א-0

עבור ארנב 2 והתקשורת שלו עם דמות המפתח יהיו שלושה תיאורים מילוליים דומים.

## השילובים האפשריים

כדאי לחשוב מעט על השילובים האפשריים, כדי להבין את המנגנון שיצרנו כאן. נראה למשל מה יכול לקרות, כאשר ארנב 2 הוא הראשון שמבקש רשות לעבוד עם הלוח הגדול:

כשארנב 2 שולח מסר בקשת רשות הוא ממתין עד אשר המסר יטופל. עם קבלת המסר, דמות המפתח ממתנה עד שערכו של busy שווה ל-0. מאחר שהלוח פנוי, ערכו של המשתנה busy כבר שווה ל-0, ולכן דמות המפתח ממשיכה הלאה. היא משנה את ערכו של busy ל-1 ומפנה עצמה לכיוונו של ארנב 2. בכך מסתיים הטיפול במסר, וארנב 2 יכול להמשיך לעבוד.

כעת נניח שבזמן שארנב 2 עובד עם הלוח הגדול, ארנב 1 מבקש רשות פעולה. גם הוא שולח מסר בקשת רשות וממתין עד שהמסר יטופל. עם קבלת המסר דמות המפתח נכנסת להמתנה עד שערכו של busy יהיה שווה ל-0. כל עוד ערכו של busy אינו משתנה, גם התסריט של המפתח המטפל בבקשת רשות של ארנב 1 וגם תסריטו של הארנב נמצאים בהמתנה.

מצב זה ישתנה רק כאשר ארנב 2 מתקדם בביצוע תסריטו ומגיע לשליחת מסר סיום עבודה. עם קבלת

המסר הזה יופעל אצל דמות המפתח תסריט סיום עבודה של ארנב 2. דמות המפתח תפנה לכיוון המרכז ותשנה את ערכו של busy ל-0. כתוצאה מכך תסתיים ההמתנה בתסריט המפתח המטפל בבקשת רשות של ארנב 1. ערכו של busy ישתנה שוב ל-1, דמות המפתח תפנה עצמה לעברו של ארנב 1, הטיפול במסר יסתיים, וארנב 2 יוכל להמשיך לעבוד.

**תרגיל 9:** לדמות המפתח נדרש גם תסריט אתחול. הכינו תיאור מילולי מתאים.

**תרגיל 10:** הכינו 7 תסריטים המתאימים לתיאורים המילוליים (חמישה לדמות המפתח ואחד לכל ארנב). כדי שיהיה נוח לראות את האנימציה, שלבו הוראות המתנה של שנייה אחת: לאחר כל פנייה של דמות המפתח ולפני בקשת הרשות אצל כל ארנב. תעדו את הפרויקט החדש ושמרו אותו. נסו להפעיל אותו.

**תרגיל 11:** כדי שאכן נראה ביצועים שונים, נכניס גם הפעם אקראיות באופן מלאכותי. לשם כך שנו בפרויקט האחרון את תסריטיהם של שני הארנבים, כך שהוראת ההמתנה הראשונה, לפני בקשת הרשות, תהיה של המתנה למשך מספר אקראי של שניות, מספר הנע בין 1 ל-3. כתוצאה מכך לא ניתן לדעת מי מהארנבים יהיה הראשון שיבקש רשות. נסו להפעיל את הפרויקט החדש כמה פעמים. תעדו ושמרו אותו.

שם הקובץ: `key-with-random-initialization`

### מושג חדש: מניעת שילובים לא רצויים על ידי הפחתת מידת המקביליות

כדי לוודא שפרויקט הוא נכון, יש למנוע שילובים המביאים לתוצאה לא נכונה. ניתן לעשות זאת על ידי הכנסת סדר, במחיר של ויתור על מעט מקביליות.

**תרגיל 12:** בתרגיל זה נבנה משחק טילים. לצורך בניית הפרויקטים שתידרשו לבנות בסעיפיו השונים של התרגיל, תוכלו להתבסס על הפרויקט `costumes-monster`. פרויקט זה כולל את התלבושות המתאימות לדמויות שיש להשתמש בהן בסעיפים השונים.

א. צרו אנימציה פשוטה שבה ניתן לשגר טיל מתחתית הבמה אל חלקה העליון. בעקבות הלחיצה על הדגל הירוק הטיל מתייצב בתחתית הבמה, למשל, בנקודה (-150, -140), כשהוא מכוון כלפי מעלה. כדי לשגר את הטיל, יש ללחוץ על כפתור השיגור שמופיע על הבמה. כתוצאה מהלחיצה על כפתור השיגור הטיל מופעל (זאת ניתן לראות על פי שובל האש שמופיע בחלקו האחורי) ומתחיל בתנועה אל חלקה העליון של הבמה, עד המסגרת. הצופים יכולים לשלוט על כיוון תנועתו של הטיל בעזרת ה־צצים: לחיצה על מקש החץ הימני תגרום לטיל להסתובב 10 מעלות עם כיוון השעון, ולחיצה על מקש החץ השמאלי תגרום לטיל להסתובב 10 מעלות נגד כיוון השעון.

שם הקובץ: `launch-rocket`

ב. הרחיבו את האנימציה שיצרתם בסעיף א' והוסיפו טיל נוסף. לחיצה על כפתור השיגור תגרום כעת לשיגורם של שני הטילים, ושימוש במקשי ה־צצים ישפיע על כיוון תנועתם של שני הטילים.

שם הקובץ: `launch-two-rockets`


ג. נקבל אנימציה מעניינת יותר אם נאפשר לצופים לשלוט על כיוון תנועתם של הטילים בנפרד. המשמעות של שינוי כזה היא שבכל רגע ניתן להשפיע רק על אחד הטילים ולא על שניהם, כי במקלדת יש רק קבוצה אחת של מקשי חצים (כמו שבדוגמת הארנבים יש רק לוח אחד במרכז החדר).

הרחיבו את האנימציה שיצרתם בסעיף ב', כך שלחיצה על מקשי החצים תשפיע רק על טיל אחד. מיד לאחר הלחיצה על הדגל הירוק, לכל אחד מהטילים יהיה סיכוי שווה להיות זה שיושפע מהחצים. הצופים יוכלו לשלוט על זהות הטיל המושפע בעזרת לחיצה על כפתור חדש. על הכפתור יופיע תמיד מספרו של הטיל שמושפע מהחצים.

**רמז:** השתמשו במשתנה ששומר את מספרו של הטיל המושפע.

שם הקובץ: `button-controls-one-rocket`

ד. הרחיבו את המשחק: כעת משתתפת בו גם מפלצת הנעה לרוחב הבמה מצדה הימני, למשל, מהנקודה (130, 190) אל גבולה השמאלי. הצופים יכולים לנווט את הטילים כך שיפגעו במפלצת. כאשר טיל פוגע במפלצת, היא מתפוצצת והמשחק מסתיים.

לצורך עצירת המשחק בעקבות פגיעה במפלצת, תוכלו להשתמש בהוראת הבקרה  הפסק הנל, שחייבת להיות הוראה אחרונה בתסריט, כפי שמעידה צורת הלבנה שמתאימה לה. הוראה זו מפסיקה את ביצועי כל התסריטים שפועלים בעת ביצועה.

שם הקובץ: `rockets-with-monster`

**תרגיל 13:** בתרגיל זה תבנו אנימציה של טילים שמשוגרים ומתודלקים שוב ושוב. המשימה היא פשוטה יחסית כאשר מדובר בטיל אחד, אך כאשר יש יותר מטיל אחד, מורכבות המשימה עולה בגלל הצורך לתאם בין הטילים.

לצורך בניית הפרויקטים שתידרשו לבנות בסעיפיו השונים של התרגיל, תוכלו להתבסס על הפרויקט `costumes-rockets`. פרויקט זה כולל את התלבושות המתאימות לדמויות שיש להשתמש בהן בסעיפים השונים.

א. צרו אנימציה פשוטה שבה טיל משוגר מתחתית הבמה אל חלקה העליון (עד למסגרת). כאשר הטיל מקבל הוראת שיגור, הוא מתכוון למשך פרק זמן שנע בין שנייה אחת לשלוש שניות, ואז יוצא לדרכו. כאשר הוא מגיע אל חלקה העליון של הבמה, הוא עוצר לתדלוק. זמן התדלוק יכול לנוע בין שנייה אחת ל-5 שניות. תדלוק של טיל בא לידי ביטוי באנימציה כך שהטיל עומד ללא תזוזה ו"אומר" כי הוא בתדלוק. לאחר התדלוק הטיל חוזר לנקודת המוצא, משוגר שוב, עוצר לתדלק בתום טיסתו, וכך הלאה, באופן אינסופי. שימו לב: הטיל משוגר באופן אוטומטי, ולא הצופים הם ששולטים על שיגורו. מצד שני, הטיל אינו משגר את עצמו! על איזה רכיב בפרויקט תטילו את האחריות לשיגור הטיל?

שם הקובץ: `rocket1`

ב. הרחיבו את האנימציה שיצרתם בסעיף א': כאשר הטיל נמצא בתנועה, יוצא שובל אש מחלקו האחורי. שובל כזה אינו נראה כאשר הטיל נמצא בעמדת השיגור שלו או כאשר הוא מתדלק.

שם הקובץ: `rocket2`



ג. **(מתקדם)** כעת האנימציה תכלול ארבעה טילים ולא רק טיל אחד. כל הטילים ישוגרו אל חלקה העליון של הבמה (אל המסגרת) מתחתית הבמה, מקו שיגור דמיוני שהוא קו מאוזן המקביל לציר ה-x. ארבע עמדות השיגור יהיו בארבע נקודות שונות לאורך קו השיגור. גם הפעם כל טיל צריך לעבור תדלוק כאשר הוא מגיע לחלקה העליון של הבמה. אבל הבעיה היא שבגלל מגבלות טכניות של המערכת, בכל רגע שני טילים לכל היותר יכולים לעבור תדלוק בו זמנית. אם יש טילים נוספים שזקוקים לתדלוק, עליהם להמתין עד שאחד משני הטילים שמתדלקים יסיים לתדלק. כדי שלצופים יהיה קל לעקוב אחר מהלך האנימציה, השתמשו באפקט הצבע כדי שיהיה קל להבחין בין הטילים השונים וגרמו לטיל הממתין לתדלוק "לומר" כי הוא ממתין לתדלוק.

**רמז:** למעשה הבעיה היא בעיית תיאום בין הטילים ובין מנגנון התדלוק. היא דומה לבעיה שהוצגה בפרק, שדנה בתיאום בין הארנבים ובין הלוח הגדול: בבעיה שהוצגה בפרק בכל רגע רק ארנב אחד יכול היה לעבוד עם הלוח הגדול; אם גם הארנב השני רצה לגשת אל הלוח, היה עליו להמתין עד שהארנב הראשון יסיים את העבודה עם הלוח. הבעיה שמוצגת בסעיף זה היא מורכבת יותר, כי כאן שניים (ולא אחד) יכולים לעבוד בו זמנית עם מנגנון התדלוק. לכן במקרה הזה המנגנון שישלוט על ההמתנה והשחרור מההמתנה יהיה שונה – השימוש במפתח לא יתאים משום שמפתח אינו יכול ל"הצביע" לכיוונו של שני טילים בבת אחת. במקום זאת השתמשו במשתנה שיזכור כמה טילים מתדלקים כעת, והשחרור מההמתנה יהיה תלוי בערך המשתנה. אין צורך להציג את ערך המשתנה על הבמה, מאחר שהוא חלק מהמנגנון הפנימי של האנימציה ואינו מעניינם של הצופים.

**שם הקובץ:** rocket3

ד. **(מתקדם)** וכעת נסבך את האנימציה אף יותר! הרחיבו את האנימציה שיצרתם בסעיף הקודם כך שכאשר טיל מתייצב לשיגור, הוא בוחר אקראית עמדה מבין עמדות השיגור. כמובן, לא ניתן לשגר שני טילים מאותה עמדה, ולכן הבחירה יכולה להיות רק מבין העמדות הפנויות.



**רמז:** השתמשו ברשימה שתשמור את עמדות השיגור הפנויות.

**שם הקובץ:** rocket4

## סיכום

בפרק זה העמקנו בביצוע מקבילי.

ראינו כי כאשר תסריטים מתבצעים במקביל, מתבצע למעשה **שילוב** של ההוראות המבוצעות בכל אחד מהתסריטים. מובטח כי השילוב שומר על הסדר בין ההוראות של אותו תסריט, אך לא מובטח דבר לגבי השילוב, ובדרך כלל ייתכנו שילובים רבים.

כדי שנהיה בטוחים כי פרויקט הכולל תסריטים שמתבצעים במקביל הוא **נכון** ופועל כנדרש, עלינו לבחון כל שילוב אפשרי.

במקרים שבהם ייתכן שייווצרו שילובים המביאים לתוצאה לא רצויה, עלינו לתקן את הפרויקט כך שתימנע יצירתם של שילובים כאלו. דרך אפשרית למנוע שילובים כאלה היא להפחית את מידת המקביליות ולהבטיח שקטעים מסוימים יתבצעו באופן סדרתי. לשם כך ניתן להשתמש בהעברת מסרים.

עוד ראינו כי כאשר משתנה מוגדר לשימושה הבלעדי של דמות מסוימת, הוא הופך להיות **משתנה פרטי**, וניתן לדמיין ששם הדמות הוא חלק משמו. משום כך אין מניעה להגדיר אצל דמויות שונות משתנים פרטיים הנושאים את אותו השם.





## פרק 11 (רשות) הרחבה והעמקה – במדעי המחשב ובסביבת Scratch

לאורך מסע החקר שלנו ב־Scratch בחרנו בגישה שלפיה אנו מלמדים הוראה רק כאשר היא נדרשת לצורך פתרון של משימה ולצורך הדגמת רעיון או מושג במדעי המחשב. לכן למרות שהכרנו הוראות רבות שניתנות לשילוב בתוך תסריטים של Scratch, לא סקרנו באופן שיטתי את כל ההוראות על פי סדר הקבוצות שאליהן הן משתייכות ועל פי סדר הופעתן במאגר ההוראות (למעוניינים, תוכלו למצוא רשימה כזו של כל ההוראות במדריך למשתמש של Scratch, שאליו ניתן להגיע דרך התפריט "עזרה"). המסע שלנו הגיע כעת לסיומו, אך עדיין נותרו כמה וכמה הוראות מעניינות ושימושיות של Scratch שטרם הזדמן לנו להכיר.

בפרק זה נסקור עוד כמה מהן, אך בהחלט לא את כולן. מידת ההיכרות שלכם עם סביבת Scratch בוודאי מאפשרת לכם להכיר בעצמכם הוראות אחרות, אם תרצו בכך. אנו נישאר נאמנים לגישה שלנו ולא נסתפק רק בהיכרות עם רכיבים של Scratch, אלא ננצלם כדי להכיר ולהדגים רעיונות במדעי המחשב.

פרק זה יהיה שונה מהפרקים שקדמו לו, משום שהוא לא יכלול פיתוח מלא ומפורט של פתרונות למשימות הכלולות בו. עבור כל משימה נציג את המושגים החדשים שבעזרתם ניתן לפתור אותה, ואת מרכיבי Scratch המתאימים להם, ונשאיר לכם לפתח את הפתרון המלא.

### דוגמה 1: רדאר למניעת התנגשויות – עוד על חיישנים ב־Scratch

התנגשויות מטוסים הן מאורעות טרגיים, והתנגשות אחת עלולה לגבות את חייהם של מאות אנשים. בשנים האחרונות ירד מספר ההתנגשויות בין מטוסים, בין השאר הודות להתקנות של מערכות רדאר למניעת התנגשויות בכל מטוס. מערכת רדאר כזו סורקת את השמיים באזור המטוס ומזהירה את הטייס אם מטוס אחר מתקרב אליו. מערכות כאלו יכולות אף להציע לטייס באיזה כיוון כדאי לו לפנות כדי להימנע מהתנגשות.

#### משימה 1: שני מטוסים מתנגשים זה בזה

יש לבנות אנימציה שבה משתתפים שני מטוסים. מטוס אחד ממריא מהפינה השמאלית התחתונה של הבמה וטס לכיוון הפינה הימנית העליונה. המטוס השני מגיע לנחיתה, כשהוא מתקרב מכיוון הפינה הימנית העליונה של הבמה ומכוון לנחות בחלקה התחתון השמאלי. יש לתכנן את מסלולם של המטוסים כך שבשום שלב לא יגעו במסגרת הבמה. כאשר שני המטוסים מתקרבים מאוד זה לזה, הם מתנגשים ונופלים.



שם הקובץ: collision


## משימה 2: מניעת התנגשות

יש לשנות את הפרויקט שפותח עבור משימה 1 כך שכאשר המטוסים מתקרבים זה לזה, כל אחד מהם סוטה ימינה ובכך נמנעת ההתנגשות. **שם הקובץ: collision-avoidance**

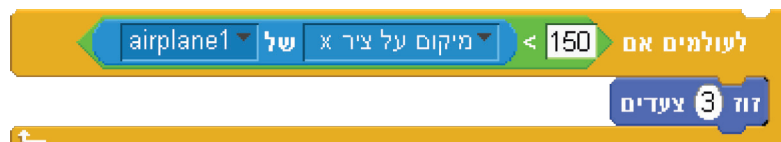


### קווים כלליים של פתרון המשימות


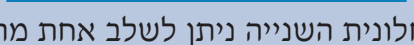
כבר השתמשנו בפרקים קודמים במערכת החיישנים של Scratch כדי לזהות מאורעות הדומים להתנגשות. למשל, בפרק 2 זיהינו פגישה בין דמויות על ידי שימוש בחיישנים המזהים נגיעה של דמות בדמות אחרת, ובפרק 7 זיהינו התנגשות של פקמן בקיר על ידי שימוש בחיישנים המזהים נגיעה בצבע. משימות 1 ו-2 דורשות זיהוי שני סוגים של התנגשות: ראשית, אנו נדרשים לזהות התקרבות של כל אחד מהמטוסים אל מסגרת הבמה כדי למנוע נגיעה של מטוס במסגרת. שנית, אנו נדרשים לזהות קרבה מוגזמת בין המטוסים כדי להגיב עליה בפיצוץ (כלומר, שינוי תלבושת) או בשינוי כיוון.

פתרון המשימה מתבסס על מאפיין יסודי של כל דמות ודמות ב-Scratch. כזכור מפרק 1, אחד ממאפייניה של דמות ב-Scratch הוא **המיקום** שלה. המיקום מתואר על ידי שני ערכים שמתאימים לערך ה-x ולערך ה-y של הנקודה על הבמה שבה נמצא מרכז של הדמות. סביבת Scratch מאפשרת לדמות "לחוש" דברים רבים, בין השאר גם את ערכי ה-x וה-y של מיקומה שלה ושל דמויות אחרות. ניתן לעשות זאת בעזרת ההוראה  שבקבוצת החיישנים הכחולה-בהירה. בחלונית הראשונה ישולב המאפיין שאותו רוצים לחוש; במקרה שלנו הפריט המתאים ברשימה הצמודה לחלונית הראשונה הוא "מיקום על ציר x". בחלונית השנייה תשולב הדמות שאותה רוצים לחוש, במקרה שלנו זהו המטוס עצמו.

לכן מטוס יוכל לזהות אם הוא קרוב מדי למסגרת הבמה על ידי השוואת ערך ה-x של מיקומו לערך ה-x של מסגרת הבמה בכיוון שאליו הוא טס. כדי לוודא שאינו נוגע במסגרת, ימשיך המטוס את תנועתו רק כל עוד ערך ה-x של מיקומו רחוק מספיק מערך ה-x של מסגרת הבמה בכיוון שאליו הוא טס. אבל שלא כמו הדמויות של Scratch, מסגרת הבמה היא קבועה במקומה, ונקודות הציון שלה קבועות וידועות. אם כך המטוס צריך להשוות את ערך ה-x של מיקומו לערך קבוע. הוראת בקרה מתאימה לתיאור המילולי הזה היא למשל:





## הוראה חדשה ב-Scratch: חישה של מאפיין של דמות



ההוראה  של  שבקבוצת החיישנים מאפשרת לחוש מאפיין של דמות. בחלונת השנייה ניתן לשלב אחת מהדמויות המשתתפות בפרויקט. בחלונת הראשונה ניתן לשלב מאפיין מתוך רשימת מאפיינים הכוללת את ערך  $x$  של מיקום הדמות, ערך  $y$  של מיקום הדמות, כיוון הדמות, תלבושתה של הדמות, גודלה ועוצמת הקול שלה. הלבנה המתאימה היא לבנת ערך וניתן לשלבה בכל הוראה, בכל חלונת שמתאימה ללבנת ערך (כלומר, שפינותיה מעוגלות).

ניתן להשתמש בהוראה גם כדי לחוש במאפיין של הבמה על ידי בחירת הפריט "רקע" בחלונת השנייה. רשימת המאפיינים המתאימה לבמה היא שונה מרשימת המאפיינים שמתאימה לדמות, והיא כוללת את הרקע ועוצמת הקול.

למאפיין המיקום יש חשיבות גם כדי למנוע התנגשות בין המטוסים: בעולם שבו לדמויות הפועלות יש מאפיין של מיקום, ניתן גם לדבר על **מרחק** בין שתי דמויות. לצורך מניעת התנגשות חשוב לעקוב אחרי המרחק בין שני המטוסים. על המטוס להמשיך את תנועתו בכיוון ההתחלתי רק כל עוד המרחק אל המטוס השני אינו קטן מדי. כאשר הוא מזהה כי המרחק קטן מדי עליו לסטות ימינה כדי להימנע מההתנגשות.

אם ידוע מיקומן של שתי דמויות, ניתן לחשב את המרחק ביניהן על ידי שימוש בנוסחה מתאימה שבה יש להציב את ערכי ה- $x$  וה- $y$  של מיקומן של שתי הדמויות. לכן אם המטוס יודע את מיקומו ואת מיקום המטוס השני בעזרת חיישני המיקום, הוא יכול לחשב בעזרת ערכי המיקום את המרחק בינו ובין המטוס השני על פי הנוסחה, ולסטות אם מרחק זה קטן מדי.

משום שפעולת חישוב המרחק בין דמויות היא פעולה שימושית כל כך, סביבת Scratch כוללת הוראה ישירה לחישוב מרחק בין דמויות, הוראה שחוסכת את הצורך לכלול בפרויקט חישוב מפורש על פי נוסחת המרחק. ההוראה היא  **המרחק אל**  שנמצאת בקבוצת החיישנים. בחלונת של ההוראה יש חץ קטן שגורם לפתיחת רשימת ערכים לשילוב בחלונת. הרשימה כוללת את כל הדמויות של הפרויקט (פרט לדמות הפעילה שבתסריטה משולבת ההוראה) וגם את מצביע העכבר. המרחק המחושב הוא המרחק מהדמות הפעילה אל הדמות (או מצביע העכבר) ששמה מופיע בחלונת.

כאמור, במקרה שלנו מטוס צריך לבדוק אם הוא קרוב מדי למטוס השני. תנאי שמתאים לתיאור כזה הוא למשל התנאי  **המרחק אל** .

לעובדה שסביבת Scratch נותנת לנו להשתמש בהוראה אחת שמגלמת מאחוריה חישוב לא פשוט, יש משמעות רבה ויתרונות רבים. תסריט שמשתמש בהוראת המרחק הוא בוודאי פשוט יותר לכתבה מתסריט שבו מחושב המרחק באופן מפורש. התסריט הוא גם פשוט יותר להבנה שהרי כעת אין הקוראים נדרשים לקרוא כמה הוראות מורכבות מאוד, להבין אותן ורק אז לקבל את התמונה הכללית של המתרחש בתסריט; במקום זאת ברור להם ממבט ראשון שהתסריט משתמש בחישוב מרחק אל דמות אחרת. בתסריט כזה מובלט הרעיון של שימוש במושג המרחק, אבל הפרטים של אופן חישוב המרחק, שאינם מהותיים כרגע מבחינת הקוראים או הכותבים, נותרים חבויים ומוסתרים.

### הוראה חדשה ב־Scratch: חישוב מרחק

ההוראה **המרחק אל** שבקבוצת החיישנים מאפשרת לדמות לחוש, או לחשב, את מרחקו אל דמות אחרת או אל מצביע העכבר (על פי תוכן החלונת). הלבנה המתאימה היא לבנת ערך, וניתן לשלבה בכל הוראה, בכל חלונת שמתאימה ללבנת ערך (כלומר, שפינותיה מעוגלות).

### מושג חדש: חיישני מיקום כדרך להעברת מידע בין דמויות

בעולם (כמו Scratch) שבו לדמויות הפועלות יש מאפיין של מיקום, המתואר על ידי ערך  $x$  וערך  $y$ , ידיעה של המיקום היא חשובה. אם דמות יכולה לחוש במיקומה של כל דמות אחרת, היא תוכל להשתמש במידע זה, למשל, לצורך חישוב מרחק. לכן חיישני מיקומה של דמות יכולים לשמש ככלי להעברת מידע בין דמויות (בדומה לחיישני נגיעה, למסרים ולמשתנים).

### מושג חדש: פישוט של פתרון על ידי החבאת פרטים

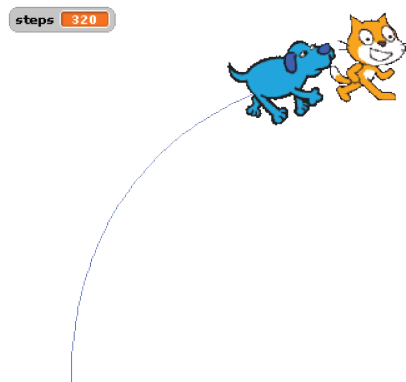
לעתים קרובות פתרון של משימה או בעיה אחת משתמש בפתרון של בעיה אחרת, שניתן להתייחס אליה כאל תת־משימה של המשימה הנתונה. למשל, כאשר לצורך פתרון בעיה מחשבים מרחק (פותרים את הבעיה של חישוב מרחק) או מוצאים ערך גדול ביותר מתוך קבוצת ערכים (פותרים את הבעיה של מציאת מקסימום). במקרים מסוימים במקום לכלול בפתרון חלק שבו נפתרת התת־משימה באופן מפורש, אפשר להחליף חלק זה של הפתרון בהוראה אחת שמבצעת את הדרוש. הפתרון הכולל הוא פשוט יותר, הן לכתיבה והן להבנה, משום שמוסתרים בו הפרטים של פתרון התת־משימה, פרטים שיכולים לסרב את הכתיבה, להעמיס ולהפריע לראיית התמונה הכללית.

## דוגמה 2: הנחיית טילים – או כלבים...

טיל שמשוגר כדי לפגוע במטוס צריך במהלך תנועתו לשנות את כיוונו כל הזמן כדי להצליח במשימתו. זאת משום שהמטוס עצמו נע ואינו נשאר בנקודה שבה היה כאשר שוגר הטיל. הדרך הפשוטה ביותר לעשות זאת נקראת "רדיפת כלב", משום שכך רודף כלב אחרי עצם נע כלשהו, למשל, אחרי חתול ה-Scratch. הוא רץ תמיד בכיוון מיקומו הנוכחי של החתול.

## משימה 3: כלב טיפש


יש לבנות אנימציה שבה משתתפים כלב וחתול. החתול נע מהפינה השמאלית העליונה של הבמה ועד לפינה הימנית התחתונה שלה. הכלב רודף אחרי החתול, כך שכיוון תנועתו הוא תמיד בכיוון מיקומו הנוכחי של החתול. מסלולו של הכלב יוצג על הבמה, וכאשר הכלב משיג את החתול, יוצג על הבמה מספר הצעדים שעשה הכלב עד שהצליח להשיג את החתול, כפי שמודגם בתמונה.



שם הקובץ: stupid-dog


## קווים כלליים של פתרון המשימה




כדי לפתור את המשימה צריך להתמודד עם שלוש תת-בעיות: ראשית, כיצד לגרום לדמות לפנות לכיוונה של דמות אחרת; שנית, כיצד לצייר מסלול תנועה או להשאיר עקבות תוך כדי תנועה; ושלישית, כיצד להציג ערך על הבמה רק מרגע מסוים ולא קודם לכן.

מהרגע שדמות יודעת את מיקומה של דמות אחרת, היא יכולה לחשב גם את הכיוון שבו הדמות האחרת נמצאת ביחס אליה. ניתן לעשות זאת בעזרת נוסחה גאומטרית מתאימה. מאחר שהפעולה שימושית למדיי, ומאחר שהנוסחה היא מורכבת למדיי, סביבת Scratch כוללת הוראה ישירה לפנייה בכיוון אל דמות (או אל מצביע העכבר). הוראה זו חוסכת את הצורך לכלול בפרויקט חישוב מפורש על פי הנוסחה הגיאומטרית המתאימה. ההוראה היא **פנה לכיוונו של**  שנמצאת בקבוצת התנועה הכחולה. כמו בהוראת חישוב המרחק, הרשימה שנפתחת עם הלחיצה על החץ הקטן כוללת את כל הדמויות של הפרויקט (פרט לדמות הפעילה שבתסריטה משולבת ההוראה) וגם את מצביע העכבר.

בדומה לשימוש בהוראת המרחק, לשימוש בהוראת הפנייה לכיוון יש יתרון כפול: תסריט שכולל את הוראת הפנייה לכיוונה של דמות הוא פשוט יותר לכתובה ופשוט יותר להבנה מתסריט שבו הכיוון מחושב במפורש.

### הוראה חדשה ב־Scratch: פנייה לכיוונה של דמות או לכיוון מצביע העכבר

ההוראה **פנה לכיוונו של**  שבקבוצת התנועה גורמת לדמות לפנות לכיוונה של דמות אחרת או לכיוון מצביע העכבר (על פי תוכן החלונת).

**תרגיל 1:** הוראת החישה של מאפיין של דמות מאפשרת גם לחוש בכיוונה של דמות, למשל כך:  **כיוון של**  Cat. האם הכלב יכול להשתמש בהוראה זו כדי לפנות לכיוונו של החתול במקום בהוראה **פנה לכיוונו של** ? הסבירו את תשובתכם.

תת־הבעיה השנייה היא השארת עקבות של מסלול תנועה. כפי שאנו יודעים, הדמויות של Scratch הן דמויות בעולם גאומטרי דו־ממדי, דמויות שיכולות לנוע. מכך נגזרים מאפיינים של הדמויות ב־Scratch, מאפיינים שבהם השתמשנו כמו מיקום וכיוון. בעולמות גאומטריים כאלו (כגון Scratch או Logo, לאלו מביניכם שמכירים סביבה זו) מקובל לייחס לדמויות מאפיין נוסף: נהוג לחשוב על הדמויות כאילו כל אחת מהן מצוידת בעט שצמוד אליה כל הזמן. בין השאר, מאפיין העט של דמות מתואר על ידי אחד משני ערכים: "למעלה" או "למטה". כאשר העט של דמות כלשהי הוא במצב "למעלה", הדמות אינה משאירה אחריה סימן במהלך תנועתה. כאשר העט של דמות כלשהי הוא במצב "למטה", הדמות משאירה אחריה סימן במהלך תנועתה.


ואכן גם ב־Scratch מאפיין זה קיים וניתן להשתמש בו. עד כה, אצל כל הדמויות בכל הפרויקטים שיצרנו היה העט מורם, במצב "למעלה", אבל ניתן לשנות את מצב העט אצל כל דמות. ההוראות המתאימות נמצאות בקבוצת העט שצבעה ירוק כהה. אלו הן ההוראות **עט מטה** ו־**עט מעלה**. ביצוע ההוראה **עט מטה** גורם לכך שכל תנועותיה של הדמות מרגע הביצוע משאירות סימן על הבמה, ולכן מסלול תנועתה מצויר. כאשר מבוצעת ההוראה **עט מעלה**, ציור מסלול תנועתה של הדמות מפסיק, אך עדיין נשארים על הבמה ציורי מסלולים קודמים. בקבוצת העט יש הוראות נוספות המשפיעות על העט של דמות, למשל על צבע העט או על עובי הקו המצויר על ידי העט.

### הוראות חדשות ב־Scratch: הרמה והורדה של עט


ההוראה **עט מטה** גורמת לדמות להתחיל לצייר את מסלול תנועתה, כלומר להשאיר על הבמה קו העוקב אחר מסלול תנועתה.


ההוראה **עט מעלה** גורמת לדמות להפסיק את ציור מסלול תנועתה. תנועות שיבוצעו לאחר מכן לא ישאירו סימן על הבמה.

תת־הבעיה השלישית דורשת להציג על הבמה את מספר הצעדים שהכלב עשה, אך רק כאשר הוא משיג את החתול. ניתן להשתמש במשתנה כדי לשמור את מספר הצעדים שהכלב מבצע, ואנו כבר יודעים כי אם הלבנה המייצגת של משתנה מסומנת, ערכו של המשתנה מוצג על הבמה; בעוד שאם הלבנה המייצגת של משתנה אינה מסומנת, ערכו אינו מוצג על הבמה. אך כיצד נוכל לגרום לכלב להציג את ערך המשתנה רק כאשר הוא משיג את החתול ולא קודם לכן?

לצורך כך מספקת לנו Scratch שתי הוראות ששולטות על התצוגה של משתנה על הבמה. שילוב ההוראות האלו בתוך תסריטים מאפשר "להדליק" ו"לכבות" את התצוגה תוך כדי הביצוע. אלו הן ההוראות  ו-  מקבוצת המשתנים האדומה־כתומה.

#### הוראות חדשות ב־Scratch: הצגת והסתרת משתנה

ההוראה  גורמת לתצוגה של המשתנה שבחלונית להופיע על הבמה.

ההוראה  גורמת להיעלמות של תצוגת המשתנה שבחלונית מהבמה.

### משימה 4: כלב חכם

יש לשנות את האנימציה שנבנתה עבור משימה 3 כך שהכלב יפעל בצורה חכמה יותר: הוא יפנה לכיוון שבו הוא צופה כי החתול יימצא עוד מעט. כלומר, במקום לפנות אל כיוון מיקומו הנוכחי של החתול הוא יפנה לכיוון נקודה שהיא עוד מעט הלאה משם, הווה אומר, הוא יגדיל מעט את זווית הפנייה שלו.

התנסו בערכים שונים עבור ההגדלה בזווית הפנייה ובחנו אם הכלב החכם יכול להשיג את החתול בפחות צעדים מאשר הכלב הטיפש.

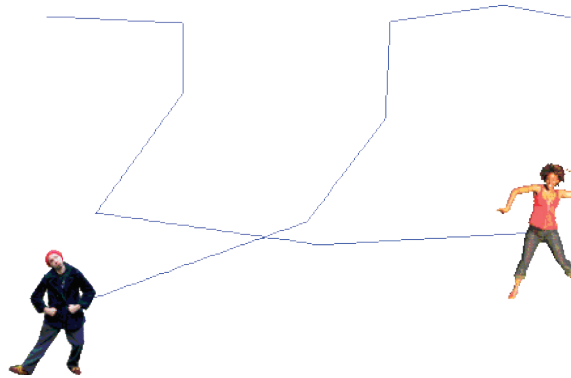
שם הקובץ: smart-dog





### דוגמה 3: כוראוגרפיה – לעולם גרפי יש גם עומק

כוראוגרף עוסק בהמצאה ובתכנון של ריקודים. כוראוגרפיים משתמשים בכלים שונים כדי לסמן את הריקודים שהם מתכננים. בדוגמה זו נפתח כלי שמאפשר לכוראוגרף לתכנן ריקודים ולסמנם, בדומה למוצג בתמונה:



### משימה 5: כוראוגרפיה לרקדן יחיד

יש לבנות אנימציה אשר מציגה דמות של רקדנית. לאחר הלחיצה על הדגל הירוק, ממתינים שהמשתמש יבצע 5 לחיצות על העכבר בנקודות שונות על הבמה. לאחר סדרה של 5 לחיצות, דמות הרקדנית נעה על פני הבמה, במסלול בן חמישה קטעים שמחבר בין נקודת ההתחלה ובין 5 נקודות הלחיצה (על פי סדר הלחיצה), אגב ציור מסלול תנועתה על הבמה.



שם הקובץ: choreography1

### קווים כלליים של פתרון המשימה

רוב המרכיבים הדרושים לפתרון המשימה כבר ידועים לכם, אך יש מרכיב אחד חדש. באנימציה שיש ליצור נדרשת תקשורת עם המשתמשים. כבר בניתם אנימציות שכוללות תקשורת עם הצופים או המשתמשים – על ידי שימוש בגולל שהוא חלק מתצוגה של משתנה, על ידי לחיצה על מקשים שונים במקלדת, או על ידי הקלדת תשובות לשאלות. באנימציה הזאת התקשורת עם המשתמשים היא בעזרת העכבר.

בכל עולם גרפי ממוחשב פרט לדמויות הפועלות בו, קיימת תמיד גם דמות העכבר, המיוצגת בעולם על ידי מצביע העכבר. למצביע העכבר יש תמיד מיקום משלו, והמשתמשים יכולים להשפיע על ערכי המיקום בכך שהם מזיזים את העכבר. בנוסף לכך המשתמשים גם יכולים ללחוץ על העכבר. ראינו כי ב־Scratch החיישנים שיודעים לחוש את הדמויות השונות מסוגלים לחוש גם את מצביע העכבר. למשל, תנאי הנגיעה יכול להתייחס גם לנגיעה במצביע העכבר, ניתן לחשב מרחק אל מצביע העכבר, וניתן לפנות לכיוונו.

בקבוצת החיישנים יש שלוש לבנים נוספות המתייחסות לעכבר באופן מפורש. ישנן שתי הוראות חישה אשר חשות את מיקומו של העכבר: ההוראה **ערך ענבר בציר x** חשה את ערך ה־x של מיקומו, וההוראה **ערך ענבר בציר y** חשה את ערך ה־y של מיקומו. בנוסף יש תנאי, **ענבר נלחץ?**, הבודק אם כפתור העכבר לחוץ.

#### מושג חדש: עכבר כישות גרפית המיוצגת על ידי מצביע העכבר

בעולם גאומטרי ממוחשב משתתפת תמיד דמות מיוחדת המשמשת לתקשורת עם הצופים. זוהי דמות **העכבר** המיוצגת על ידי **מצביע העכבר**. כמו לדמויות אחרות, לדמות העכבר מאפיין של מיקום. המשתמש יכול להשפיע על מיקומו של מצביע העכבר על ידי הזזת העכבר, וביכולתו גם ללחוץ על העכבר.

#### הוראה חדשה ב־Scratch: חישה של מיקום מצביע העכבר

ההוראה **ערך ענבר בציר x** שבקבוצת החיישנים מאפשרת לחוש את ערך ה־x של מיקום מצביע העכבר.

ההוראה **ערך ענבר בציר y** שבקבוצת החיישנים מאפשרת לחוש את ערך ה־y של מיקום מצביע העכבר.

#### תנאי חדש ב־Scratch: לחיצה על העכבר

התנאי **ענבר נלחץ?** מייצג שאלה שהתשובה עליה היא "כן" אם כפתור העכבר לחוץ, ו"לא" אם אין הוא לחוץ.

- כעת אתם מכירים כל שדרוש לפתרון המשימה, אך זו אינה משימה פשוטה. היעזרו בהנחיות הבאות:
- בשלב הראשון יש לשמור את חמש הנקודות שבהן נלחץ העכבר, הנקודות שמגדירות את המסלול.
  - הנקודות יכולות להישמר בעזרת שתי רשימות, רשימה לערכי ה־x של הנקודות ורשימה מקבילה עבור ערכי ה־y שלהן.
  - כדי לשמור נקודה אחת, הרקדנית צריכה להמתין עד שהעכבר נלחץ, ואז לחוש את המיקום שבו נמצא מצביע העכבר ולשמור את ערכיו ברשימות. על תהליך זה יש לחזור 5 פעמים.
  - אם הרקדנית תהיה זריזה מדיי והמשתמש יהיה אטי מדיי, ייתכן כי הרקדנית תספיק לשמור את ערכי המיקום ולשוב ולבדוק אם העכבר לחוץ, בעוד שהמשתמש כלל לא הספיק לשחרר את הלחיצה הקודמת! במקרה כזה שוב יישמרו – כנקודה חדשה – ערכי המיקום שמתאימים

לנקודה הקודמת. ניתן לפתור זאת בכך שלאחר שמירת ערכי המיקום תמתין הרקדנית שהעכבר לא יהיה לחוץ, ורק אז תמשיך לטפל בנקודה הבאה.

- לאחר שחמש הנקודות שמורות, כל שנותר לרקדנית הוא לנוע מהמיקום ההתחלתי שלה בהתאם למסלול שמגדירות חמש הנקודות, אגב ציור מסלול תנועתה.

בהמשך הדוגמה נרצה להרחיבה כך שריקוד יוכל לכלול יותר מרקדן אחד. כאשר בריקוד משתתפים מספר רקדנים, הצגתו על הבמה מסובכת יותר, משום שהרקדנים יכולים לעתים להסתיר זה את זה, ותצוגה אמינה צריכה להביע זאת. המשימה הבאה מתייחסת להיבט זה של הסתרת דמויות.

## משימה 6: פעם זה מלפנים, ופעם זה מלפנים

בנו אנימציה של ריקוד שבו משתתפים שני רקדנים. הרקדנית רוקדת שוב ושוב מצדה השמאלי של הבמה אל צדה הימני ובחזרה. הרקדן רוקד שוב ושוב מצדה הימני של הבמה אל צדה השמאלי ובחזרה. כאשר הרקדנית נעה ימינה והרקדן שמאלה, הרקדנית עוברת לפני הרקדן בעוד שבכיוון ההפוך הוא עובר על פניה. [שם הקובץ: choreography2](#)



## קווים כלליים של פתרון המשימה

Scratch היא דוגמה לעולם גאומטרי שטוח, משום שלבמה יש רק שני ממדים. אבל גם בעולם דו־ממדי אפשר לעתים לדמות עולם תלת־ממדי שיש בו גם עומק. למשל, נוכל לדמות תנועה של חפץ המתרחק מאיתנו על ידי כך שנשלב בתנועה גם הקטנה של החפץ.

דרך נוספת המאפשרת להכניס עומק לעולם גאומטרי דו־ממדי היא לתת לכל דמות מאפיין נוסף: ניתן לדמיין כאילו בעולם יש שכבות, שכבה מאחורי שכבה. מספר השכבות הוא כמספר הדמויות הפועלות, ולכל דמות שכבה משלה שבה היא נעה. כאשר שתי דמויות נעות חופפות בחלקן (כלומר, בעולם הדו־ממדי יש נקודות על הבמה ששתי הדמויות נוגעות בהן) הדמות שנמצאת בשכבה הקדמית יותר (הקרובה יותר לצופים) מסתירה את החלקים החופפים של הדמות שנמצאת בשכבה האחורית יותר.

אם כך, דמות יכולה להתאפיין גם על ידי השכבה שבה היא נמצאת, וזהו לאו דווקא מאפיין קבוע. דמות יכולה להשפיע על המאפיין הזה על ידי תזוזה בין השכבות.

סביבת Scratch מאפשרת עבודה עם שכבות, וקבוצת המראה, הקבוצה הסגולה, כוללת שתי הוראות המסייעות בכך. ההוראה **העבר לשינבה הקדמית** גורמת לדמות לעבור אל השכבה הקדמית (שכבה 1). כתוצאה מכך כל הדמויות שהיו קודם לפנייה זזות שכבה אחת אחורה. למשל, נניח שבשכבה הראשונה יש כלב, בשנייה יש חתול, בשלישית פיל וברביעית סוס. אם הפיל עובר לשכבה הקדמית, הרי שהוא נמצא כעת בשכבה הראשונה, הכלב בשנייה, החתול בשלישית והפיל נשאר ברביעית. אם החיות הן בגודל רגיל, הפיל מסתיר את החיות שמאחוריו.

ההוראה **העבר שנייה לאחור** גורמת לדמות לזוז אחורה, כמספר השכבות שמופיע בחלונית. כתוצאה

מכך כמה מהדמויות שהיו קודם אחריה נמצאות כעת לפניה. למשל, נניח שלאחר שהפיל עבר לשכבה הקדמית, עבר הכלב שכבה אחת אחורה. כעת הפיל נמצא בשכבה הראשונה, החתול בשכבה השנייה, הכלב נמצא בשכבה השלישית, והסוס ברביעית. אם כעת הפיל עובר שתי שכבות אחורה, החתול נמצא בשכבה הראשונה, הכלב בשנייה, הפיל בשלישית והסוס עדיין בשכבה הרביעית. מאחר שהפיל גדול ביחס לחתול וגם ביחס לכלב, סביר להניח שעדיין ניתן לראות חלקים ממנו, למרות שהוא בשכבה השלישית.

### מושג חדש: שכבות כדרך להוסיף עומק לעולם שטוח

בעולם גאומטרי ממוחשב דמויות לעתים מתאפיינות גם על ידי השכבה שבה הן נמצאות. ככל שמספרה של שכבה הוא נמוך יותר, כך השכבה קדמית יותר. כאשר דמויות הן חופפות, דמות שנמצאת בשכבה קדמית יותר מסתירה דמות שנמצאת בשכבה אחורית יותר.

### הוראות חדשות ב־Scratch: מעבר בין שכבות

ההוראה **העבר לשיכבה הקדמית** שבקבוצת המראה גורמת לדמות לעבור לשכבה הקדמית ולכל הדמויות שהיו לפניה לזוז שכבה אחת אחורה.

ההוראה **העבר שכה לאחור** שבקבוצת המראה גורמת לדמות לעבור אחורה כמספר השכבות שבחלונית ולכל הדמויות שהיו מאחוריה – עד השכבה שאליה היא עוברת ועד בכלל – לעבור שכבה אחת קדימה.

מספיק להטיל על אחת משתי דמויות הרקדנים את האחריות למעבר בין השכבות. למשל, נניח שהאחריות מוטלת על הרקדנית. כאשר היא נעה שמאלה, עליה לזוז לשכבה הקדמית, בעוד שבעת התנועה ימינה, עליה לזוז שכבה אחת אחורה.

## משימה 7: כוראוגרפיה לשני רקדנים

הרחיבו את האנימציה שיצרתם במשימה 5. כעת משתתפים גם רקדנית וגם רקדן. לאחר הלחיצה על הדגל הירוק ממתנינים לעשר לחיצות על העכבר. 5 נקודות הלחיצה הראשונות מגדירות מסלול עבור הרקדנית וחמש האחרונות מגדירות מסלול עבור הרקדן. לאחר עשר הלחיצות שני הרקדנים רוקדים במקביל, כל אחד את המסלול שהוגדר עבורו. כאשר מסלוליהם מובילים אותם לחלוף זה על פני זה, הרקדן מאפשר לרקדנית לעבור לפניו.



שם הקובץ: choreography3

## קווים כלליים של פתרון המשימה

בפתרון משימה זו חשוב לשמור את נקודות הלחיצה באופן שיאפשר אחר כך לשני הרקדנים לשחזר בנוחות ובצורה נכונה את המסלולים המוגדרים להם. נוח לשמור ארבע רשימות: שתיים עבור הנקודות של הרקדנית ושתיים עבור הנקודות של הרקדן. אם נרצה לחלק את האחריות לשמירת הנתונים בין הרקדנים, כך שכל אחד יהיה אחראי לשמירה ברשימות שלו, נצטרך לתאם בזירות את עבודתם: קודם הרקדנית תשמור את הנקודות שלה, אחר כך הרקדן ישמור את שלו, ורק אז יעבדו שניהם במקביל. אפשרות אחרת היא להטיל על הבמה את האחריות השמירה של הנקודות, חמש נקודות ברשימות של הרקדנית וחמש ברשימות של הרקדן. לאחר מכן יתחילו הרקדנים לעבוד במקביל.

### תרגיל 2:

באנימציה האחרונה שיצרתם הרקדנים רוקדים במקביל במסלול בן 5 קטעים. הפכו את האנימציה למעניינת יותר באופן הבא: בקטעי המסלול האי-זוגיים (ראשון, שלישי וחמישי) הרקדן עובר לפני הרקדנית ובקטעי המסלול הזוגיים (שני ורביעי) הרקדנית עוברת לפני הרקדן.

**רמז:** השתמשו בפעולת השאריט. ניתן לבדוק זוגיות של מספר על ידי בדיקת שאריט החלוקה שלו ב-2.

שם הקובץ: choreography4

### תרגיל 3:

הרחיבו את האנימציה שיצרתם בתרגיל 2: בכל פעם שהעכבר נלחץ בנקודה מסוימת, נקודה זו תסומן על הבמה כך שהמשתמשים יוכלו לראות בבירור את נקודות הלחיצה שסימנו. נקודות הלחיצה שנועדו להגדרת המסלול של הרקדנית יסומנו באופן שונה (למשל, בצבע שונה) מאלו המתאימות למסלולו של הרקדן.



**רמז:** הוסיפו שתי דמויות חדשות – האחת של סימון נקודת לחיצה של הרקדנית, והאחרת של סימון נקודת לחיצה של הרקדן. במקום שהבמה תקלוט את נקודות הלחיצה ותשמור אותן ברשימות המתאימות, האחריות תעבור אל הדמויות החדשות: כל אחת מהן תקלוט את הנקודות המתאימות לה ותדאג לשמירתן. תוך כדי קליטת הנקודות תשתמש כל אחת מהדמויות החדשות בחותמת כדי לסמן את נקודות הלחיצה.

שם הקובץ: choreography5




## סיכום



בפרק זה העמקנו בפתרון בעיות והכרנו מאפיינים של עולמות גאומטריים ממוחשבים.



בעת **פתרון בעיה המשתמשת בפתרון בעיה אחרת** כתת-משימה, ניתן לעתים לפשט את הפתרון על ידי החלפה של החלק שפותר את התת-בעיה בהוראה שמבצעת אותו הדבר באופן ישיר. בכך מוסתרים פרטי הפתרון של התת-בעיה, והפתרון המתקבל קצר יותר ופשוט יותר לכתובה ולהבנה. כך אנו מתמקדים בשימוש בפתרון של התת-בעיה לצורך הפתרון הכולל ולא באופן שבו התת-בעיה נפתרה.

בעולם גאומטרי ממוחשב יש לכל דמות מאפיין של מיקום, המתואר על ידי ערך  $x$  וערך  $y$ . אם דמות יכולה לחוש את המיקום של דמות אחרת, **חיישני המיקום** יכולים לשמש ככלי להעברת מידע בין דמויות. בפרט יכולה דמות לחשב בעזרתם את **המרחק** שלה מדמויות אחרות. באופן דומה ניתן להשתמש גם **בחיישנים של מאפיינים** נוספים, כגון כיוון. ב־Scratch דמות יכולה לחוש במאפיינים של דמות, או של הבמה, בעזרת ההוראה  שבקבוצת החיישנים. המאפיינים שאותם ניתן לחוש עבור דמות הם מיקום, כיוון, תלבושת, גודל ועוצמת קול. עבור הבמה ניתן לחוש ברקע ובעוצמת הקול. ב־Scratch ניתן לחשב באופן ישיר מרחק אל דמות או אל מצביע העכבר על ידי שימוש בהוראה  שבקבוצת החיישנים.



תפקיד מיוחד שמור בעולם גאומטרי ממוחשב **לעכבר** שמאפשר תקשורת עם המשתמשים ומיוצג על ידי **מצביע העכבר**. גם מצביע העכבר מאופיין על ידי מיקום, וניתן להשפיע על מיקומו על ידי הזזת העכבר. תקשורת עם המשתמשים יכולה להתבצע גם על ידי לחיצות על העכבר.

ב־Scratch אפשר לחוש במיקומו של העכבר על ידי הוראות החישה  ו־. התנאי  שבקבוצת החיישנים בודק אם כפתור העכבר לחוץ.

בעולם גאומטרי ממוחשב שטוח ניתן לדמות **עומק** על ידי שימוש ב**שכבות**. דמויות שנמצאות בשכבות קדמיות מסתירות דמויות הנמצאות בשכבות אחוריות יותר. ב־Scratch דמויות יכולות לעבור בין שכבות בעזרת ההוראות  ו־ שבקבוצת המראה.

דמויות בעולם גאומטרי ממוחשב מצוידות ב**עט**, המאפשר להן להשאיר ציור מסלול העוקב אחר תנועתן. עט יכול להיות מורם במצב "למעלה" וכך הוא אינו מצייר, והוא יכול להיות גם מורד במצב "למטה" ואז הוא מצייר בעת התנועה. ב־Scratch ניתן לשלוט על מצב העט בעזרת ההוראות  ו־ שבקבוצת העט הירוקה-כהה. בקבוצת העט יש הוראות נוספות ששולטות על העט, כגון על צבעו, עובי הקו שלו ועוד.

הכרנו הוראת תנועה חדשה של **פנייה לכיוון יחסי**, . בעזרת הוראה זו יכולה דמות להפנות עצמה לכיוונה של דמות אחרת או לכיוון מצביע העכבר.

ב־Scratch ניתן לשלוט על תצוגה של משתנה תוך כדי ביצוע בעזרת ההוראות  ו־ שבקבוצת המשתנים.



## רעיונות ומושגים במדעי המחשב דרך Scratch – מבט אחורה

מטרתו של ספר זה, אשר אל סיומו הגעתם כעת, היא לערוך לכם היכרות עם רעיונות ומושגים במדעי המחשב. ייתכן כי יש ביניכם כאלה השואלים: האם אנשי מדעי המחשב עושים דברים דומים לאלה שעשינו אנו במהלך הלימוד?

במידה רבה, כן. אמנם Scratch אינה משמשת בדרך כלל את אנשי מדעי המחשב בעבודתם. רובם גם אינם עוסקים בכתיבת אנימציות דומות לאלו שאתם בניתם במהלך הלימוד. אבל העקרונות שאליהם נחשפתם במהלך הלימוד בהחלט משמשים את אנשי מדעי המחשב בעבודתם.

מדעי המחשב הוא תחום של **פתרון בעיות**. בדומה למשימות שהוצגו בפניכם בספר זה, הבעיות לקוחות מתחומים רבים, והן מאופיינות בדרגות משתנות של מורכבות וקושי. בתהליך של פתרון בעיה משתמשים בעקרונות ובשיטות שהכרתם וחוויתם או טעמתם במהלך הלימוד:

- **פתרון בשלבים:** המשימה מחולקת לכמה חלקים, והפותר מתמודד עם כל חלק לחוד. בכך מקטין הפותר את דרגת המורכבות ואולי גם את דרגת הקושי של הבעיה.

במקרים רבים החלקים אינם נפתרים על פי סדרם. כאשר הפותר מעדיף להתמודד עם חלק מאוחר לפני שפתר חלק שקודם לו, הוא מניח שהחלק הקודם כבר פתור, והוא יכול להשתמש בהנחה זו בעת פתרון החלק המאוחר.

- **תיאור מילולי:** הפתרון מתחיל בתכנון מילולי וחופשי. רק לאחר מכן מתורגם התיאור המילולי להוראות מדויקות שהמחשב – שיצטרך בסופו של דבר לבצע את הפתרון – יכול "להבין". עבור בעיות מורכבות במקרים רבים עיקר הקושי הוא בתכנון הפתרון ובבניית התיאור המילולי (הקרוי בפי אנשי מדעי המחשב **אלגוריתם**); ואילו תרגום התיאור המילולי להוראות מדויקות הוא פשוט יחסית. לכן יש גם מי שעוסקים במדעי המחשב ומתמחים בתכנון תיאורים מילוליים (אלגוריתמים) לבעיות מורכבות, אך אינם עוסקים בדרך כלל בתרגומם לשפה שאותה המחשב יכול להבין.

- **פיתוח הדרגתי של התיאור המילולי:** התיאור המילולי נבנה בהדרגה, מן הכלל אל הפרט. בתחילה הוא מורכב מצעדים כלליים יחסית. לאחר מכן הצעדים הכלליים מוחלפים בהדרגה בצעד או בסדרה של צעדים מפורטים יותר (למשל, צעד אתחול כללי מוחלף בשלושה צעדי אתחול מפורטים, אשר כל אחד מהם מטפל באתחול של פרט אחר). בסוף התהליך מתקבל תיאור מילולי שהוא עדיין חופשי, אך קל יותר לתרגם כל צעד בו להוראה בשפת המחשב.

- **שימוש בתבניות מוכרות:** לעתים מזהה הפותר כי לצורך פתרון חלק מהבעיה אפשר להיעזר בפתרונות שכבר נבנו בעבר – על ידו או על ידי אחרים. למשל, אם הפותר מזהה כי חלק מהבעיה מחייב חיפוש ברשימה אחר מקומו של איבר שנמצא בה, הוא יכול להשתמש בתבנית המוכרת של חיפוש מסוג זה, ורק להתאימה לבעיה הנוכחית. בדומה לכך, אם הפותר מזהה כי לצורך פתרון חלק מהבעיה הוא נדרש למנות או לצבור ערכים, הוא יכול להשתמש בתבניות המוכרות של מנייה או צבירה (שנעזרות במונה או בצובר). השימוש בתבניות מוכרות מקל על תהליך הפתרון.



● **“החבאת” מידע:** בתהליך הפתרון של בעיה מתקבלות החלטות כאלו ואחרות. במקרים רבים החלטות אלו חשובות למתכנני הפתרון, אך הן אינן מעניינם של המשתמשים בפתרון, ולכן גם לא כדאי לחשוף אותן בפניהם. למשל, בדרך כלל המשתמשים בפתרון אינם צריכים להיות מודעים למשתנים שבהם החליט הפותר להשתמש. עקרון ההחבאה משמש גם בתוך פתרון, כאשר “מחביאים” מרכיבים מסוימים בתוך חלק של הפתרון ואין חושפים אותם בפני חלקים אחרים (למשל, על ידי משתנים פרטיים). כך פוחתת התלות בין חלקי הפתרון לבין עצמם. מידה נמוכה של תלות היא חשובה: אם למשל, נרצה לשפר או לשנות חלק של הפתרון, חלקי פתרון אחרים, שאינם תלויים בו, לא יושפעו מהשינויים.

● **תיקון תקלות:** ככל שהבעיה מורכבת יותר, כך סביר יותר כי ניסיונות הפתרון הראשונים לא יצלחו. טעויות הן חלק בלתי נפרד מתהליך פתרון בעיות, וחלק מהמיומנות הנדרשת היא הבנת מקורן של טעויות ותקלות, ובהתאם לכך תיקון.

● **תיעוד:** כאשר איש מדעי המחשב מספק פתרון לבעיה, בדרך כלל הפתרון אינו חד־פעמי, אלא אמור לשמש לאורך זמן. במהלך השימוש ייתכן שיידרשו שיפורים ועדכונים. ייתכן גם שתקלות כאלו או אחרות יתגלו רק במהלך השימוש, ולא במהלך פיתוח הפתרון. כדי שיהיה קל יחסית לחזור אל פתרון לאחר זמן ולהבין היטב את פרטיו, חשוב מאוד לתעד פתרון בזמן הפיתוח על ידי שילוב הערות מפורטות במקומות המתאימים.

בנוסף נחשפתם למושגים חשובים במדעי המחשב, חלקם מושגים מתקדמים ולא פשוטים, כגון:

● ביצוע סדרתי

● ביצוע מקבילי

● ביצוע חוזר (אינסופי או סופי, מוגבל מראש או מותנה)

● ביצוע מותנה

● תקשורת ותיאום בעזרת מסרים

● משתנים

● משתנים מורכבים: רשימות

● בחירה אקראית

אם הבנתם והפנמתם את העקרונות, השיטות והמושגים שנמנו כאן, ואף השתמשתם בהם תוך כדי פתרון בעיות – הרי שערכתם היכרות מייצגת עם תחום מדעי המחשב ועם רעיונות מרכזיים שלו.

# רעיונות ומושגים במדעי המחשב דרך Scratch

מיכל ארמוני ומוטי בן-ארי

מכון ויצמן למדע

2010



תנאי השימוש בגרסה האלקטרונית של הספר כפופים לרישיון

"ייחוס-שימוש לא מסחרי-איסור יצירת נגזרות 2.5 ישראל" של Creative Commons.

ניתן לעיין בתקציר הרישיון באתר:

<http://creativecommons.org/licenses/by-nc-nd/2.5/il/>

<http://creativecommons.org/licenses/by-nc-nd/2.5/il/legalcode>