

# MoleMash



פרק זה מראה לך כיצד ליצור את MoleMash, משחק בהשראת הקלאסי של ארקיד, A-Mole™, Whac-a-Mole. שבו יצורים מכניים יוצאים מהחורים, ושחקנים צוברים נקודות כשהם מצליחים להכות אותם עם פטיש. MoleMash בנוצרה על ידי חבר בצוות App Inventor, לכאורה כדי לבדוק את פונקציונליות ה-sprite (שהיא הטמיעה), אבל באמת בגלל שהיא מעריצה של המשחק.

כאשר אלן ספרטוס הצטרפה לצוות App Inventor בגול, היא הייתה להוטה להוסיף תמיכה ביצירת משחקים, אז היא התנדבה ליישם ספרייטים. המונח, שנבע במקור כדי לתאר יצורים מיתולוגיים כמו פיות ופיקסז, אומץ על ידי קהילת המחשוב בשנות ה-07 כדי להתייחס לתמונות המסוגלות לנוע על מסך מחשב (למשחקי וידאו). אלן קודם עבדה עם ספרייטים כשהשתתפה במחנה מחשבים בתחילת שנות השמונים ותכנתה TI 99/4. עבודתה על ספרייטים MoleMash-ו נבעה מנוסטלגיה כפולה - הן למחשבים והן למשחקים של ילדותה.

## מה תבנה

עבור אפליקציית MoleMash המוצגת באיור 3-1, תטמיע את הפונקציונליות הבאה:

- שומה צצה במקומות אקראיים על המסך, וזה פעם בשנייה.

- הקשה על השומה גורמת למכשיר לרטוט, להגדיל תצוגה של "מכות" (להגדיל אותה באחת), ולהזיז את השומה מיד למקום חדש.



איור 3-1. ממשק המשתמש של MoleMash

- הקשה על המסך אך החמצה של השומה מגדילה את התצוגה של "החמצות".
- לחיצה על כפתור איפוס מאפסת את ספירת הכניסות והחמצות.

## מה תלמד

המדריך מכסה את הרכיבים והמושגים הבאים:

- רכיב ImageSprite עבור תמונות נעות רגישות למגע.
- רכיב ה-Canvas, שפועל כמשטח עליו מניחים את ImageSprite.
- רכיב השעון כדי להזיז את הספרייט פעם בשנייה.
- רכיב הסאונד להפקת רטט בעת הקשה על השומה.
- רכיב הכפתור כדי להתחיל משחק חדש.
- נהלים ליישום התנהגות חוזרת, כגון הזזת השומה.
- הפקת מספרים אקראיים.
- שימוש בלוקים של חיבור (+) וחסור (-).

## מתחילים

התחבר לאתר App Inventor והתחל פרויקט חדש. תן לזה שם "MoleMash" וגם הגדר את כותרת המסך. "MoleMash" ללחץ על התחבר וחבר את המכשיר או האמולטור שלך לבדיקה חיה.

הורד את התמונה של שומה מ- <http://appinventor.org/bookFiles/MoleMash/> mole.png. ב-Component Designer, לחץ על "העלה קובץ", דפדף למקום שבו נמצא file-הבמחשב שלך, ולאחר מכן העלה אותו ל-Inventor. ppA.

## עיצוב הרכיבים

תשתמש ברכיבים האלה כדי ליצור MoleMash:

- קנבס המשמש כמגרש משחקים.
- ImageSprite • שמציג תמונה של שומה ויכול להסתובב ולחוש כשנוגעים בשומה.
- צליל רטט כשנוגעים בשומה.
- תוויות המציגות "היטים:", "החמצות:", "והמספרים בפועל של כניסות ו מתגעגע.
- סידורים אופקיים למיקום נכון של התוויות.
- כפתור כדי לאפס את מספר הכניסות והחמצות ל-0.

עיצוב הרכיבים 45

•שעון כדי לגרום לשומה לזוז פעם בשנייה.

טבלה 1-3מציגה את כל הרכיבים שבהם תשתמש.

טבלה 1-3רשימת הרכיבים המלאה עבור MoleMash

מָטְרָה	קבוצת צבעים איך תקרא לזה	סוג רכיב
המיכל עבור ImageSprite.	קנבס 1	ציור ו אנימציה
המשתמש ינסה לגעת בזה.	חפרפרת	ציור ו אנימציה
המשתמש ילחץ על זה כדי לאפס את ציון.	כפתור איפוס ממשק משתמש	לחצן
לשלוט בתנועת השומה.	שעון ממשק משתמש 1	שעון
רטט כשנוגעים בשומה.	קלי תקשורת צליל 1	נשמע
הצג את "היטים: "	ממשק משתמש HitsLabel	תווית
הצג את מספר הכניסות.	ממשק משתמש HitsCountLabel	תווית
מקם את HitsLabel ליד HitsCountLabel.	אופקי פריסת סידור אופקי הסדר	
הצג את "החמצות: "	ממשק משתמש MissesLabel	תווית
הצג את מספר ההחמצות.	ממשק משתמש MissesCountLabel	תווית
מקם MissesLabel ליד MissesCountLabel.	אופקי הסדר 2	פריסת סידור אופקי

הצבת רכיבי הפעולה

בחלק זה, תציב את הרכיבים הדרושים לפעולת המשחק. בתוך ה בסעיף הבא, תציב את הרכיבים להצגת הציון.

1.מהמגירה של ציור והנפשה, גרור פנימה רכיב , Canvasעזוב זה עם שם ברירת המחדל 1.Canvasהגדר את המאפיין Widthשלו ל-"llif" parent" שהוא הוא רחב כמו המסך, והגדר את גובהו ל-003 פיקסלים.

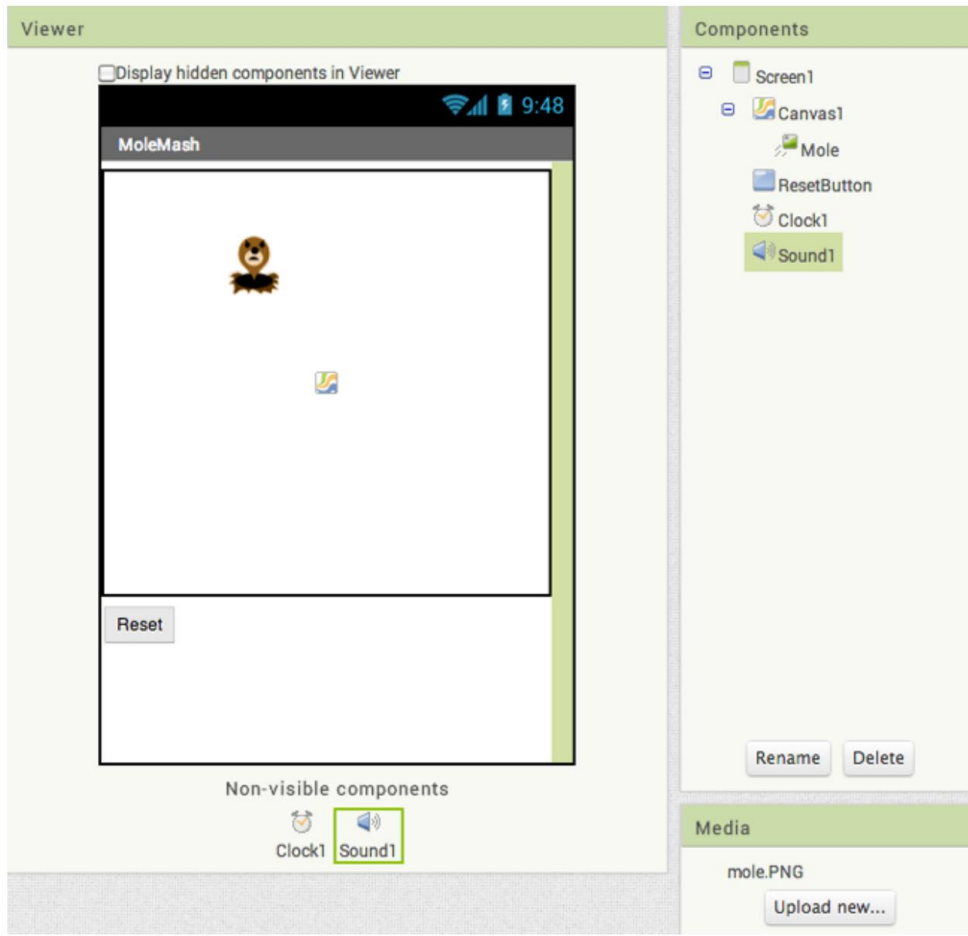
2.שוב ממגירת הציור וההנפשה, גרור ב- ImageSprite רכיב, הצבתו בכל מקום ב- 1.Canvasבתחתית הרכיבים ברשימה, לחץ על שנה שם ושנה את שמה ל"שומה". הגדר את מאפיין התמונה שלו ל mole.PNG,שהעלית קודם לכן.

3.מהמגירה של ממשק המשתמש, גרור פנימה רכיב , Buttonהצב אותו מתחת ל-savnaC1. שנה את שמו של "ResetButton"והגדר את מאפיין הטקסט שלו ל-"Reset".

4. גם מהמגירה של ממשק המשתמש, גרור פנימה רכיב שעון . הוא יופיע בחלק התחתון של Viewer-הבקטע "רכיבים שאינם נראים".

5. ממגירת המדיה, גרור פנימה רכיב סאונד . גם הוא יופיע בסעיף "רכיבים לא גלויים".

המסך שלך אמור כעת להיראות בערך כמו איור 2-3 (אם כי השומה שלך עשויה להיות בעמדה אחרת).



איור 2-3. תצוגת מעצב הרכיבים של רכיבי "פעולה".

הצבת רכיבי התווית

כעת תציב רכיבים להצגת הניקוד של המשתמש - באופן ספציפי, מספר הכניסות והחמצות.

1. ממגירת Layout-גורר פנימה HorizontalArrangement, הצב אותו מתחת ללחצן ושמו

על שם ברירת המחדל של HorizontalArrangement1.

2. מהמגירה של ממשק המשתמש, גרור שתי תוויות לתוך

סידור אופקי. 1.

□ שנה את שם התווית השמאלית "HitsLabel" והגדר את מאפיין ה-Text שלה

Hits: "ל-" (הקפד לכלול רווח אחרי הנקודתיים).

□ שנה את שם התווית הימנית "HitsCountLabel" והגדר את המאפיין Text שלה ל-

מספר 0.

3. גרור פנימה HorizontalArrangement שני, הצב אותו מתחת

HorizontalArrangement1.

4. גרור שתי תוויות לתוך Horizontal Arrangement2.

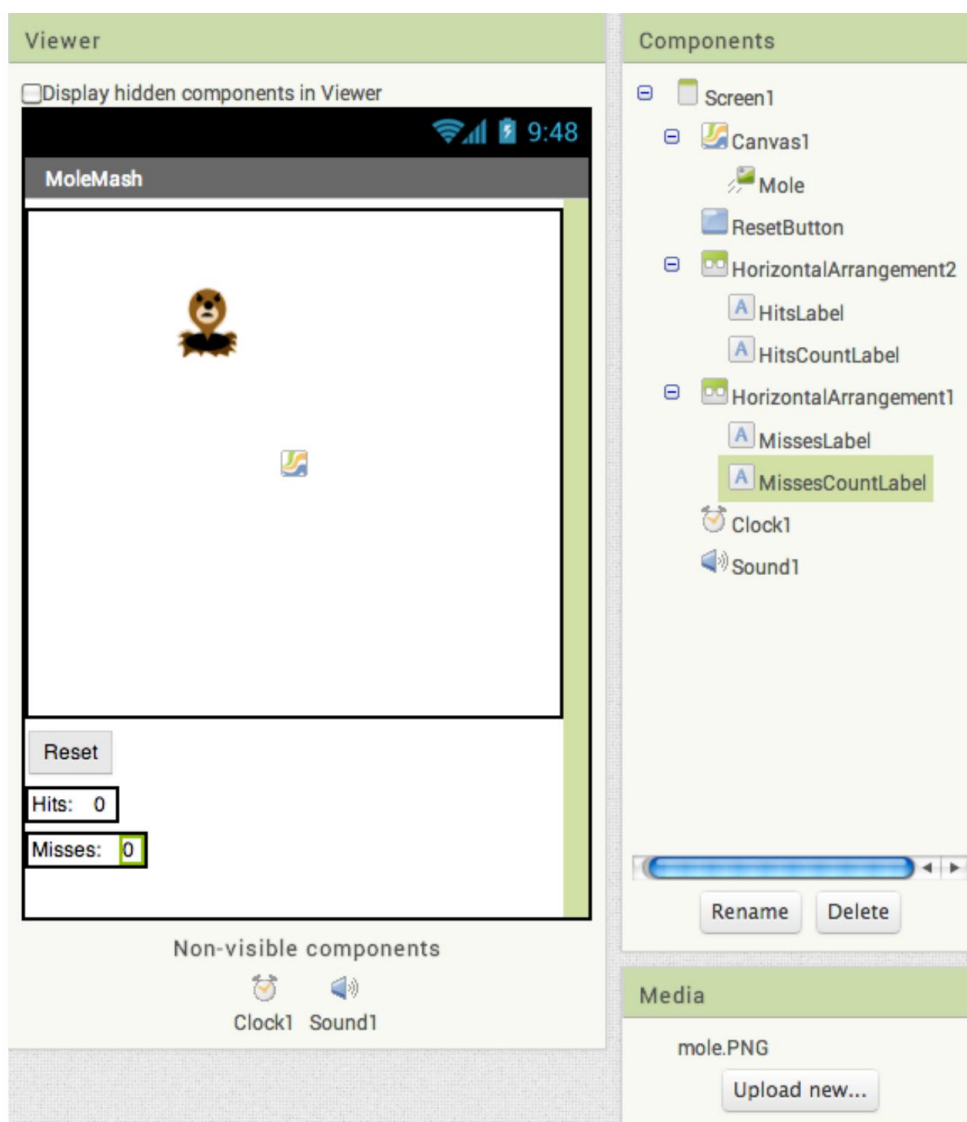
□ שנה את שם התווית השמאלית "MissesLabel" והגדר את מאפיין ה-Text שלה

Misses: "ל-" (הקפד לכלול רווח אחרי הקולון).

□ שנה את שם התווית הימנית "MissesCountLabel" והגדר את המאפיין Text שלה ל

המספר 0.

המסך שלך אמור כעת להיראות כמו איור 3-3.



איור 3-3. תצוגת מעצב הרכיבים של כל רכיבי MoleMash-ה

## הוספת התנהגויות לרכיבים

לאחר יצירת הרכיבים הקודמים, הבה נעבור לעורך הבלוקים כדי ליישם את התנהגות התוכנית. באופן ספציפי, אנחנו רוצים שהחפרפרת תעבור למיקום אקראי על הבד כל שנייה. המטרה של המשתמש היא להקיש על השומה בכל מקום שהיא מופיעה, והאפליקציה תציג את מספר הפעמים שהמשתמש פוגע או מפספס את השומה. (הערה: אנו ממליצים להשתמש ב-finger בשלך, לא בפטיש!) לחיצה על כפתור האיפוס מאפסת את מספר הפגיעות והחמצות ל-0.

## הזזת השומה

בתוכניות שכתבת עד כה, קראת לפרוצדורות מובנות כגון `רטט`. `HelloPurr`-בהאם לא יהיה נחמד אם ל-`ppA-Inventor` היה הליך שמעביר `ImageSprite` למיקום אקראי על המסך? החדשות הרעות: זה לא. החדשות הטובות: אתה יכול ליצור נהלים משלך! בדיוק כמו הנהלים המובנים, ההליך שלך יופיע במגירה ותוכל להשתמש בו בכל מקום באפליקציה.

באופן ספציפי, ניצור הליך להזזת השומה למיקום אקראי על המסך, שנקרא לו `MoveMole`. אנחנו רוצים לקרוא ל-`MoveMole` בתחילת המשחק, כשהמשתמש מקיש בהצלחה על השומה, ופעם בשנייה.

יצירת הליך `MOVEMOLE`

כדי להבין כיצד להזיז את השומה, עלינו להסתכל כיצד פועלת הגרפיקה של אנדרואיד.

ניתן לחשוב על הקנבס (והמסך) כעל רשת עם קואורדינטות x (אופקיות) ו-y (אנכיות), כאשר הקואורדינטות  $(x, y)$  של הפינה השמאלית העליונה הן  $(0, 0)$  וקואורדינטת ה-x גדלה ככל שאתה זז ימינה, וקואורדינטת ה-y הגדלה ככל שאתה זז למטה, כפי שמוצג באיור 3-4. המאפיינים x ו-y אשל `ImageSprite` מציינים היכן ממוקמת הפינה השמאלית העליונה שלו; לפיכך, לשומה בפינה השמאלית העליונה באיור 3-4 יש ערכי x ו-y אשל 0.

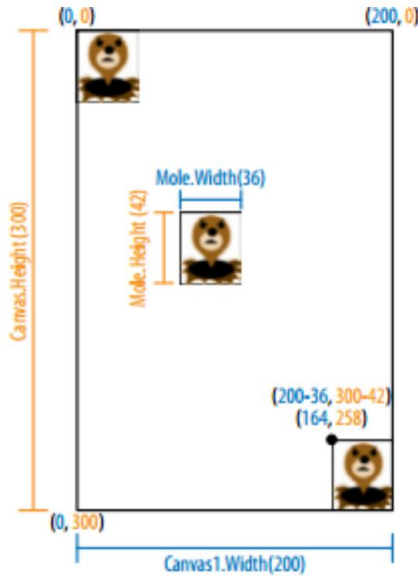
כדי לקבוע את ערכי ה-x וה-y המקסימליים הזמינים כך ש-`Mole.fts` על המסך, עלינו לעשות שימוש במאפייני הרוחב והגובה של שומה וקנבס. 1. (מאפייני הרוחב והגובה של השומה זהים לגודל התמונה שהעלית. כשיצרת את `Canvas1`, אתה מגדיר את הגובה שלו ל-300 פיקסלים ואת הרוחב שלו ל-`"parent", lliF` שמעתיק את הרוחב של אלמנט האב שלו, אשר במקרה זה המסך.) אם השומה ברוחב 36 פיקסלים והבד הוא ברוחב 200 פיקסלים, קואורדינטת ה-x של הצד השמאלי של השומה יכולה להיות נמוכה כמו 0 (כל הדרך שמאלה) או גבוהה כ-36, 461 (או `Canvas1.Width - Mole.Width`) ללא השומה המתרחבת בקצה הימני של המסך. באופן דומה, קואורדינטת ה-y השל החלק העליון של השומה יכולה לנוע בין 0 ל-`Canvas1.Height - Mole.Height`.

איור 3-5 מציג את ההליך שתיצור, עם הערות תיאוריות

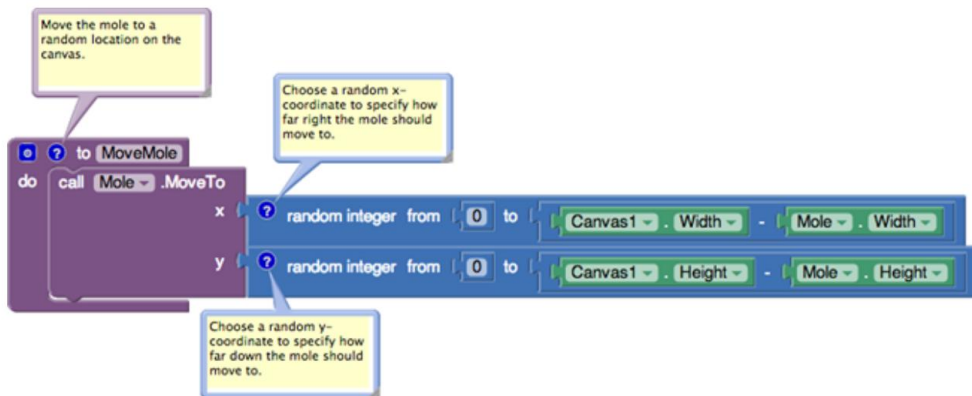
הערות (שאותן תוכל להוסיף להליך שלך באופן אופציונלי).

כדי למקם את השומה באופן אקראי, נרצה לבחור קואורדינטת אבטווה שבין `Mole.Width` - `Canvas1.Width` ל-0 באופן דומה, נרצה שקואורדינטת ה-y התהיה בטווח שבין `Mole.Height` - `Canvas1.Height` ל-0. אנחנו יכולים ליצור מספר אקראי באמצעות הפרוצדורה המובנית של מספר שלם אקראי, אותו תוכל למצוא במגירה של `Math`. יהיה עליך לשנות את פרמטר ברירת המחדל "מאת" מ-1 ל-0 ולהחליף את הפרמטרים "אל", כפי שמוצג באיור 3-5.

## 3: MoleMash פרק 50



איור 3-4. מיקומי השומה על המסך, עם מידע על קואורדינטות, גובה ורוחב; קואורדינטות x ורוחב מוצגים בכחול, ואילו קואורדינטות y וגובה מוצגים בכתום



איור 3-5. הליך MoveMole המציב את השומה במיקום אקראי

כדי ליצור את ההליך:

1. בעורך הבלוקים, לחץ על מגירת הפרוצדורה.

2. גרור החוצה את גוש ההליך (המכיל "עשה", לא "תוצאה").

3. בבלוק החדש, לחץ על הטקסט "פרוצדורה" והקלד "MoveMole" כדי להגדיר את שם ההליך.

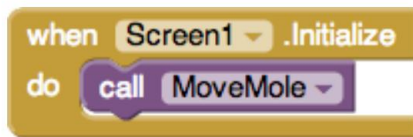
4. מכיוון שאנו רוצים להזיז את השומה, לחץ על מגירת השומה וגרור את הקריאה `Mole.MoveTo` להליך, מימין ל"עשה". שימו לב לשקעים הפתוחים בצד ימין שמציינים שעלינו לספק קואורדינטות `x` ו-`y`.
5. כדי לציין שקואורדינטת ה-`x` החדשה עבור השומה צריכה להיות בין 0 ל-`Canvas1.Width - Mole.Width`, כפי שנדון קודם לכן, בצע את הפעולות הבאות:
  - ממגירת המתמטיקה, גרור פנימה את המספר השלם האקראי מהגוש, והכנס את התקע (בליט) בצדו השמאלי לשקע "א" בעת השיחה `Mole.MoveTo`.
  - שנה את גוש המספר "1" בשקע "מאת" על ידי לחיצה עליו ולאחר מכן הזנת המספר "0".
  - הסר את המספר "100" על ידי לחיצה עליו ולחיצה על מקש `Delete` או `Del` של המקלדת, או על ידי גרירתו לפח האשפה.
  - ממגירת המתמטיקה, גרור פנימה בלוק חיסור (-) והנח אותו לתוך שקע "אל".
  - מהמגירה, `Canvas1`, בחר את בלוק `Canvas1.Width` וגרור אותו אל הצד השמאלי של פעולות החיסור.
  - באופן דומה, לחץ על מגירת החפרפרת וגרור את ה- `Mole.Width` אל סביבת העבודה. לאחר מכן, חבר את זה לצד הימני של בלוק החיסור.
  - 6. בצע הליך דומה כדי לציין שקואורדינטת ה-`y` הצריכה להיות אקראית מספר שלם בטווח שבין 0 ל- `Canvas1.Height - Mole.Height`.
  - 7. בדוק את התוצאות שלך מול איור 3-5.

מתקשר MOVEMOLE-לכאשר האפליקציה מתחילה

כעת לאחר שכתבת את הליך `MoveMole`, בואו נשתמש בו. מכיוון שכל כך נפוץ שמתכנתים רוצים שמשוה יקרה כשאפליקציה מתחילה, יש חסימה בדיוק למטרה זו: `Screen1.Initialize`.

1. לחץ על מגירת מסך 1 וגרור החוצה את מסך 1. אתחול.
2. לחץ על מגירת ההליכים, שבה תראה בלוק `MoveMole` שיחה. (שלה די מגניב שיצרת בלוק חדש, לא?!). גרור אותו החוצה והצב אותו ב- `Screen1.Initialize`, כפי שמוצג באיור 3-6.

## פרק 52 MoleMash 3:



איור 6-3. קורא לנוהל MoveMole כאשר היישום מתחיל

מתקשר MOVEMOLE לכל שנייה

לגרום לשומה לזוז בכל שנייה ידרוש את רכיב השעון. השארנו את המאפיין TimeInterval עבור Clock1 בערך ברירת המחדל שלו של 1,000 (מילישניות), או שנייה אחת. זה אומר שכל שנייה, כל מה שצוין בבלוק Clock1.Timer יתקיים. הנה איך להגדיר את זה:

1. לחץ על המגירה Clock1 וגרור החוצה Clock1.Timer.

2. לחץ על מגירת הפרוצדורות וגרור בלוק MoveMole שיחה לתוך בלוק Clock1.Timer, שמוצג באיור 7-3.

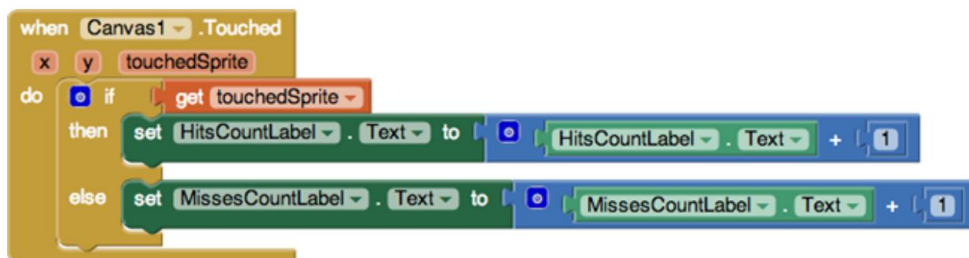


איור 7-3. קריאה להליך MoveMole כאשר הטיימר כבה (כל שנייה)

אם זה מהיר או איטי מדי עבורך, אתה יכול לשנות את המאפיין TimeInterval עבור Component Designer ב-Clock1 כדי לגרום לו לזוז בתדירות גבוהה יותר או פחות.

שמירה על ציון

מכור, יצרת שתי תוויות, MissesCountsLabel, ו-HitsCountsLabel, כזכור, שהיו להם ערכים התחלתיים של 0. ברצוננו להגדיל את המספרים בתוויות אלה בכל פעם שהמשתמש מקיש בהצלחה על השומה (מכה) או מקיש על המסך מבלי לגעת בשומה (פספוס). לשם כך, נשתמש בבלוק Canvas1.Touched, המציין שנגע בבד, קואורדינטות ה-x וה-y של המקום בו נוצר קשר (שאיננו צריכים לדעת), והאם הקישו על ספרייט (ש אנחנו צריכים לדעת). איור 8-3 מציג את הקוד שתיצור.



איור 8-3. הגדלת מספר ההיטים (HitsCountLabel) או החמצות (MissesCountLabel) כאשר נוגעים ב-Canvas1

אתה יכול לתרגם את הבלוקים באיור 8-3 באופן הבא: בכל פעם שהקנבס מוקשה, בדוק אם הקיש ספרייט. מכיוון שיש רק ספרייט אחד בתוכנית שלנו, זה חייב להיות Mole1. אם הקישו על Mole1, הוסף אחד למספר ב-HitsCountLabel.Text. לאחר מכן, הוסף אחד ל-MissesCountLabel.Text (הערך של touchedSprite הוא שקר אם לא נגעו ב-sprite).

כך יוצרים את הבלוקים:

1. לחץ על מגירת Canvas1 וגרור החוצה Canvas1.Touched.

2. לחץ על מגירת הבקרה וגרור החוצה את הבלוק אם-אז. לחץ על הסמל הכחול שלו והוסף ענף אחר. לאחר מכן, הנח אותו בתוך Canvas1.Touched.

3. העבר את העכבר מעל פרמטר האירוע touchedSprite ב-Canvas1.Touched, ולתוך מכן גרור החוצה את הבלוק get touchedSprite והנח אותו בשקע הבדיקה של if-then אחר.

4. כי אנחנו רוצים HitsCountLabel.Text שיוגדל אם הבדיקה הצליחה.

(אם נגע בשומה), בצע את הפעולות הבאות: מהמגירה של HitsCountLabel, גרור החוצה את הסט HitsCountLabel.Text לחסום, לשים אותו מימין ל"אז".

לחץ על מגירת המתמטיקה וגרור החוצה סימן פלוס (+) והצב אותו ב"אל" שקע.

לחץ על מגירת HitsCountLabel וגרור את בלוק HitsCountLabel משמאל לסימן הפלוס.

לחץ על מגירת המתמטיקה וגרור בלוק 0 מימין לסימן הפלוס. לחץ על 1 ושנה אותו ל-1.

5. חזור על שלב 4 עבור MissesCountLabel בקטע else של בלוק ifelse.



בדוק את האפליקציה שלך אתה יכול לבדוק את הקוד החדש הזה במכשיר שלך על ידי הקשה על הקנבס, הן על והן של השומה, וצפייה בניקוד משתנה.

אבסטרקציה פרוצדורלית

היכולת לתת שם ולקרוא מאוחר יותר לקבוצת הוראות כמו MoveMole היא אחד הכלים המרכזיים במדעי המחשב ומכונה הפשטה פרוצדורלית. זה נקרא "התפשטות" מכיוון שהמתקשר של ההליך (שסביר להניח שבפרויקטים בעולם האמיתי יהיה שונה ממחבר הפרוצדורה) רק צריך לדעת מה ההליך עושה (מזיז את השומה), לא איך זה עושה זאת (על ידי ביצוע שתי קריאות למחולל המספרים האקראיים). ללא הפשטה פרוצדורלית, תוכניות מחשב גדולות לא יהיו אפשריות, מכיוון שהן מכילות יותר מדי קוד כדי שאנשים יוכלו להחזיק אותם בראש בכל פעם. זה מקביל לחלוקת העבודה בעולם האמיתי, שבו, למשל, מהנדסים שונים מתכננים חלקים שונים של מכונת, אף אחד מהם לא מבין את כל הפרטים, והנהג צריך רק להבין את הממשק (למשל, לחיצה על דוושת בלם כדי לעצור את המכונת), לא היישום.

## הוספת התנהגויות לרכיבים 55

כמה יתרונות של הפשטה פרוצדורלית על פני העתקה והדבקה של קוד הם:

- קל יותר לבדוק קוד אם הוא מופרד בצורה מסודרת משאר התוכנית.
- אם יש טעות בקוד, יש לתקן אותה רק במקום אחד.
- כדי לשנות את היישום, כגון לוודא שהשומה לא זזה במקום שהוא הופיע לאחרונה, אתה רק צריך לשנות את הקוד במקום אחד.
- ניתן לאסוף נהלים לתוך ספרייה ולהשתמש בהם בתוכניות שונות. (למרבה הצער, פונקציונליות זו אינה נתמכת כעת (App Inventor ב-SB).
- שבירת קוד לחתיכות עוזרת לך לחשוב על האפליקציה וליישם אותה ("הפרד ומשול").
- בחירת שמות טובים לפרוצדורות עוזרת לתעד את הקוד, ומקלה על מישהו אחר (או לך, חודש לאחר מכן) לקרוא.

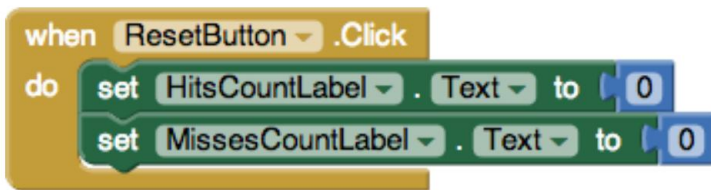
בפרקים מאוחרים יותר, תלמדו דרכים להפוך נהלים לחזקים עוד יותר: הוספת ארגומנטים, מתן ערכי החזרה וקיום נהלים לקרוא לעצמם. לסקירה כללית, ראה פרק 21.

איפוס הניקוד

חבר שרואה אותך משחק MoleMash-בכנראה ירצה לנסות את זה, אז טוב שתהיה לך דרך לאפס את מספר הכניסות וההחמצות ל-0. תלוי באילו מדריכים כבר עבדת, ייתכן שאתה מסוגל להבין כיצד לעשות זאת מבלי לקרוא את ההוראות הבאות. שקול לנסות את זה לפני שתקרא קדימה.

מה שאנחנו צריכים זה בלוק `ResetButton.Click` שנקבע את הערכים של `HitsCountLabel.Text` ו-`MissesCountLabel.Text` ל-0. צור את הבלוקים המוצגים ב

איור 9-3.



איור 9-3. איפוס מספר הכניסות (`HitsCountLabel`) וההחמצות (`MissesCountLabel`) כאשר כפתור האיפוס נלחץ

בשלב זה, כנראה שאינך זקוק להוראות שלב אחר שלב ליצירת מטפל באירוע לחיצת כפתור עם תוויות טקסט, אבל הנה טיפ שיעזור להאיץ את התהליך: במקום לקבל את המספר שלך ממגירת המתמטיקה, פשוט הקלד, וזאת

## פרק 56 MoleMash 3:

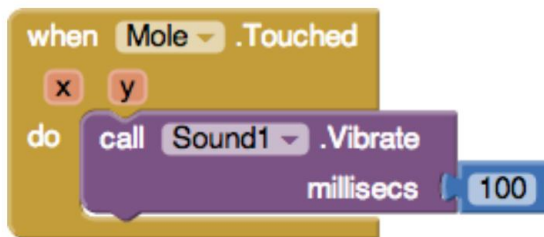
צריך ליצור עבורך בלוק. (סוגים אלה של קיצורי מקשים קיימים גם עבור בלוקים אחרים).



בדוק את האפליקציה שלך נסה להכות ולהחמיץ את השומה ולאחר מכן ללחוץ על כפתור האיפוס.

רוטט כשנוגעים בשומה

אמרנו קודם לכן שאנחנו רוצים שהמכשיר ירטוט כשהמשתמש מקיש על השומה, מה שאנחנו יכולים לעשות עם בלוק, `Sound1.Vibrate` כפי שמוצג באיור. 3-10



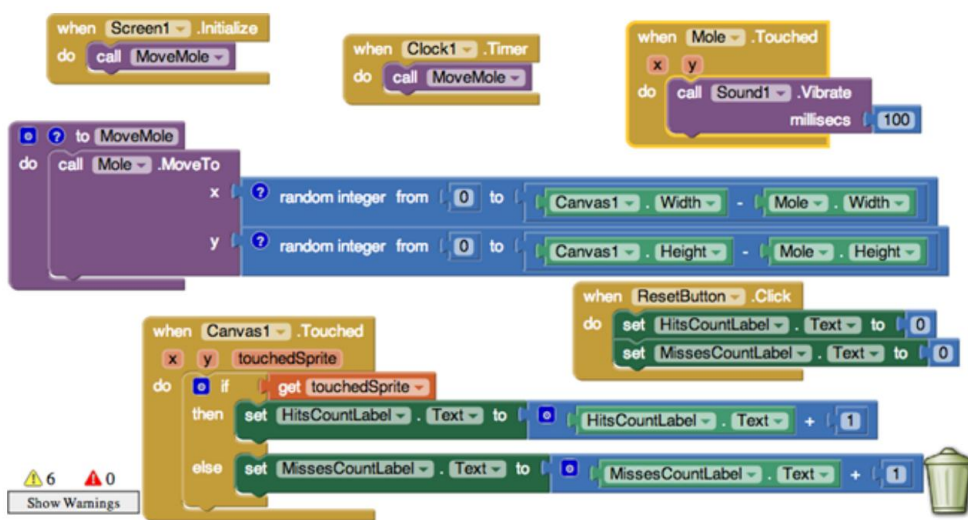
איור. 3-10. גורם למכשיר לרטוט קצר (למשך 100 מילישניות) כשנוגעים בשומה



בדוק את האפליקציה שלך ראה כיצד הרטט פועל כאשר אתה באמת מקיש על השומה. אם הרטט ארוך מדי או קצר מדי לטעמכם, שנה את מספר האלפיות השניות בבלוק `Sound1.Vibrate`.

## האפליקציה השלמה: MoleMash

איור 3-11 ממחיש את הבלוקים עבור אפליקציית MoleMash השלמה.



איור 11-3. אפליקציית MoleMash המלאה

## וריאציות

הנה כמה רעיונות לתוספות ל-MoleMash:

- הוסף כפתורים כדי לאפשר למשתמש לגרום לשומה לנוע מהר יותר או לאט יותר.
- הוסף תווית כדי לעקוב אחריה ולהציג את מספר הפעמים שיש לשומה הופיע (זז).
- הוסף ImageSprite שני עם תמונה של משהו שהמשתמש לא צריך מכה, כגון פרח. אם המשתמש נוגע בו, העניש אותו על ידי הפחתת הניקוד שלו או סיום המשחק.
- במקום להשתמש בתמונה של שומה, תן למשתמש לבחור תמונה עם ה-ImagePicker . רכיב .

## סיכום

בפרק זה, כיסינו מספר טכניקות שימושיות עבור אפליקציות בכלל ומשחקים בפרט:

- רכיב ה-Canvas עושה שימוש במערכת קואורדינטות, x, y, כאשר x מייצג את הכיוון האופקי (מ-0 משמאל ל-Canvas.Width-1 מימין), ו-y ואת הכיוון האנכי (מ-0 בחלק העליון ל-Canvas.Height-1 בתחתית). ניתן להפחית את הגובה והרוחב של ImageSprite מהגובה והרוחב של קנבס כדי לוודא שהספרייט מתאים לחלוטין על הקנבס.

• ניתן לנצל את מסך המגע של המכשיר דרך ה- `Canvas`

שיטות `Touched` של רכיבי `ImageSprite`.

• ניתן ליצור אפליקציות בזמן אמת המגיבות לא רק לקלט של המשתמש אלא גם בתגובה לטיימר הפנימי של המכשיר. באופן ספציפי, בלוק `Clock.Timer` פועל בתדירות המצוינת במאפיין `Clock.Interval` וניתן להשתמש בו כדי להזיז רכיבים של `ImageSprite` (או אחרים).

• אתה יכול להשתמש בתוויות כדי להציג ציונים, שעולים (או יורדים) בתגובה ל פעולות השחקן.

• ניתן לספק משוב מישוש למשתמשים באמצעות שיטת `Sound.Vibrate` שגורמת למכשיר לרטוט במשך המספר שצוין של אלפיות השנייה.

• במקום להשתמש רק בשיטות המובנות, אתה יכול ליצור נהלים לשמות לקבוצה של בלוקים שניתן לקרוא להם בדיוק כמו המובנים. זה נקרא הפשטה פרוצדורלית והוא מושג מפתח במדעי המחשב, המאפשר שימוש חוזר בקוד ומאפשר יישומים מורכבים.

• אתה יכול ליצור התנהגות בלתי צפויה עם בלוק השלם האקראי (ב- מגירת מתמטיקה), עושה משחק שונה בכל פעם שהוא משוחק.

תלמדו טכניקות נוספות למשחקים, כולל זיהוי התנגשויות ביניהם הזזת רכיבי `ImageSprite` בפרק 5.