שלום פורר

פרק זה גורם לך להתחיל לבנות אפליקציות. הוא מציג את מרכיבי המפתח של Inventor, App מעצב הרכיבים ועורך הבלוקים, ומוביל אותך בשלבים הבסיסיים של יצירת האפליקציה הראשונה שלך, HelloPurr. כשתסיים, תהיה מוכן לבנות עליהם אפליקציות משלך.

איור 1-1. אפליקציית HelloPurr

תוכנית ראשונה טיפוסית עם מערכת מחשב חדשה מדפיסה את ההודעה "Hello World" כדי להראות שהכל מחובר כהלכה. מסורת זו חוזרת לשנות ה-07 ולעבודתו של בריאן קרניגאן על שפת התכנות C ב-Bell Labs. עם App Inventor, אפילו האפליקציות הפשוטות ביותר עושות יותר מסתם הצגת הודעות: הן משמיעות צלילים ומגיבות כשאתה נוגע במכשיר. אז, אנחנו הולכים להתחיל מיד עם משהו יותר מרגש: האפליקציה הראשונה שלך (כמתואר באיור 1-1) תהיה "HelloPurr", תמונה של חתול שמייאם כשאתה נוגע בו ומגרגר כשאתה רועד המכשיר שבו הוא נצפה.

מה תלמד

הפרק מכסה את הנושאים הבאים:

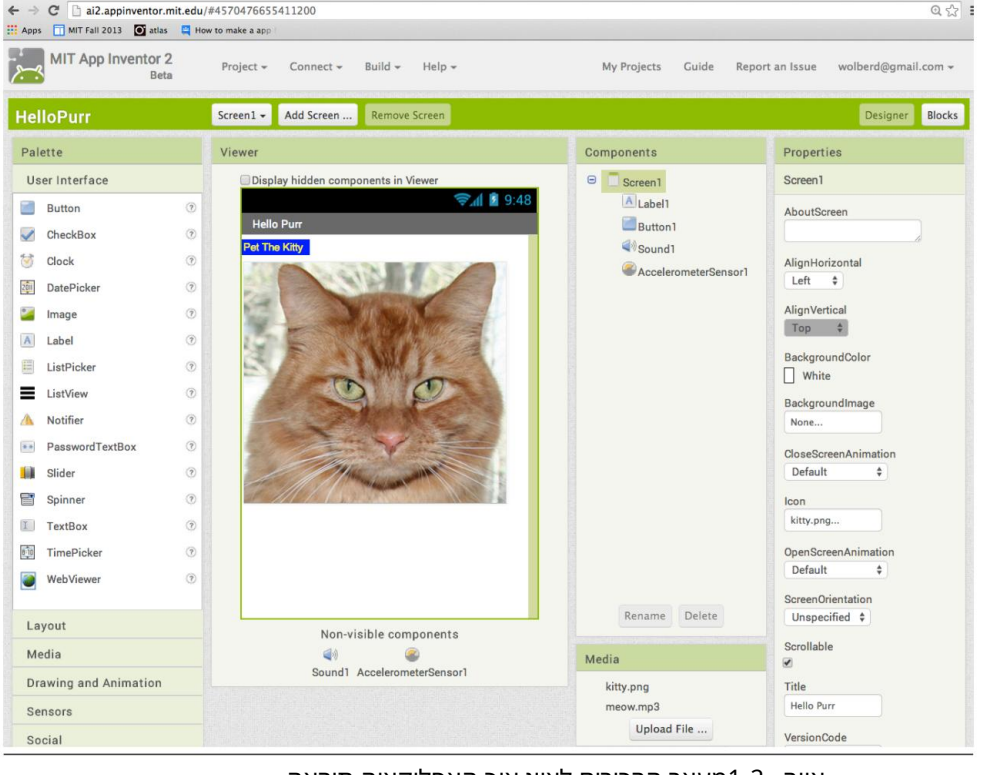
- בניית אפליקציות על ידי בחירת רכיבים וציון התנהגותם.
- שימוש Component Designer- בלבחירת רכיבים. חלק מהמרכיבים הם גלויים על מסך המכשיר וחלקם לא.
- הוספת מדיה (צלילים ותמונות) לאפליקציות על ידי העלאתן מהמכשיר שלך מַחשֵב.
- שימוש בעורך הבלוקים כדי להרכיב בלוקים שמגדירים את הרכיבים' התנהגות.
- בדיקת אפליקציות עם הבדיקות החיות של App Inventor. זה מאפשר לך לראות כיצד יישומים ייראו ויתנהגו במכשיר, צעד אחר צעד, גם בזמן שאתה בונה אותם.
- אריזת האפליקציות שאתה בונה והורדתן למכשיר.

סביבת ממציא האפליקציה

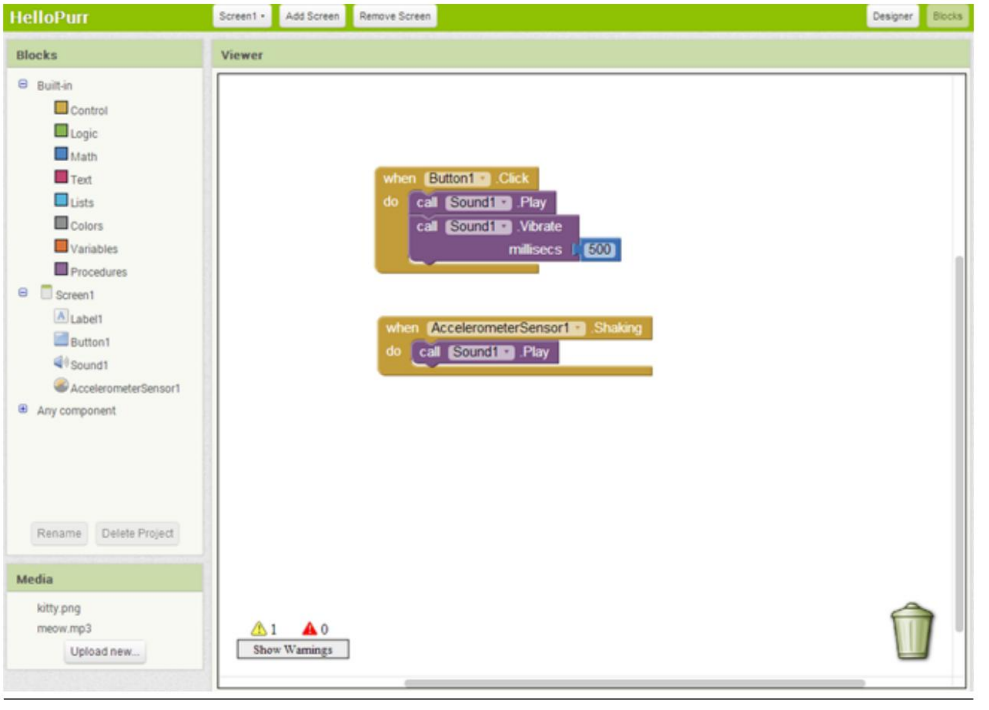
אתה יכול להתחיל לתכנת עם App Inventor על ידי פתיחת דפדפן ל-
ai2.appinventor.mit.edu. זה פותח את הגרסה החדשה ביותר של App Inventor, אשר
שוחררה בדצמבר, 2013. יש אנשים שקוראים לזה App Inventor 2, אבל היא נקראת
רשמית רק App Inventor, והגרסה הקודמת נקראת App Inventor Classic. בספר זה
תשתמש בגרסה החדשה.

לסביבת התכנות App Inventor יש שלושה חלקים מרכזיים:

- מעצב הרכיבים (איור 2-1). אתה משתמש בו כדי לבחור רכיבים עבורך האפליקציה וציינו את המאפיינים שלהם.
- עורך הבלוקים (איור 3-1). אתה משתמש בו כדי לציין כיצד הרכיבים יתנהגו (למשל, מה קורה כאשר משתמש לוחץ על כפתור).
- מכשיר אנדרואיד שבעזרתו תוכל להפעיל ולבדוק את האפליקציה שלך תוך כדי פיתוחה. אם אין לך מכשיר אנדרואיד בהישג יד, תוכל לבדוק את האפליקציות שאתה בונה באמצעות אמולטור אנדרואיד שמגיע עם המערכת.



איור 2-1. מעצב הרכיבים לציון איך האפליקציה תיראה



איור .3-1עורך הבלוקים לציון כיצד האפליקציה תתנהג

בפעם הראשונה שאתה גולש אל ,ai2.appinventor.mit.eduתראה את דף הפרויקטים, שיהיה ברובו ריק מכיוון שעדיין לא יצרת אף פרויקט. כדי ליצור פרויקט, בפינה השמאלית העליונה של העמוד, לחץ על "פרויקט חדש", הזן את שם הפרויקט "rruPolleH" (מילה אחת ללא רווחים), ולאחר מכן לחץ על אישור.

החלון הראשון שנפתח הוא מעצב הרכיבים. עורך הבלוקים זמין על ידי לחיצה על כפתור "בלוקים" בפינה השמאלית העליונה של החלון.

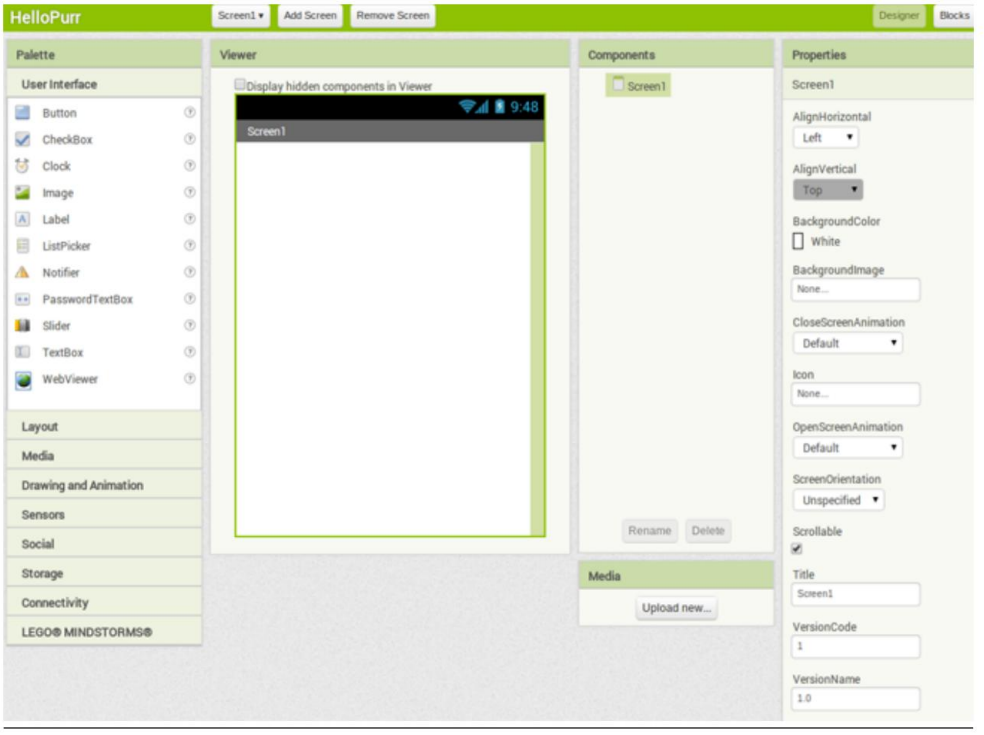
App Inventorהוא כלי מחשוב ענן, כלומר האפליקציה שלך מאוחסנת בשרת מקוון תוך כדי עבודה. אז אם תסגור את ,App Inventorהאפליקציה שלך תהיה שם כשתחזור; אתה לא צריך לשמור שום דבר במחשב שלך כפי שהיית עושה עם, למשל, .Microsoft Word flee

עיצוב הרכיבים

הכלי הראשון שבו תשתמש הוא מעצב הרכיבים (או רק מעצב). רכיבים הם האלמנטים שאתה משלב כדי ליצור אפליקציות, כמו מרכיבים במתכון. חלק מהרכיבים פשוטים מאוד, כמו רכיב תווית , שמציג טקסט על המסך, או רכיב כפתור , עליו אתה מקיש כדי להתחיל פעולה. רכיבים אחרים משוכללים יותר: קנבס ציור שיכול להחזיק תמונות סטילס או אנימציות; מד תאוצה, שהוא חיישן תנועה שמזהה מתי אתה מזיז או מנער את

התקן; או רכיבים שיוצרים או שולחים הודעות טקסט, מנגנים מוזיקה ווידאו, מקבלים מידע מאתרים וכן הלאה.

כאשר אתה פותח את המעצב, הוא יופיע כפי שמוצג באיור .1-4



איור .1-4מעצב הרכיבים של ממציא האפליקציה

המעצב מחולק למספר תחומים:

•לכיוון המרכז נמצא אזור לבן הנקרא .Viewerזה המקום שבו אתה מציב רכיבים וסדר אותם כדי למפות איך אתה רוצה שהאפליקציה שלך תיראה. ה-reweiV מציג רק אינדיקציה גסה של איך האפליקציה תיראה, כך למשל, שורת טקסט עלולה להישבר במקום אחר במכשיר שלך מאשר ב-reweiV. כדי לראות איך האפליקציה שלך תראה באמת , תצטרך לבדוק אותה במכשיר שלך או באמולטור (עוד מעט נראה לך איך לעשות זאת).

•משמאל ל-reweiV נמצאת ה- ,Paletteשהיא רשימה של רכיבים שמהם ניתן לבחור. הפלטה מחולקת לחלקים; בשלב זה, רק רכיבי ממשק המשתמש גלויים, אך ניתן לראות רכיבים בחלקים אחרים של הפלטה על ידי לחיצה על הכותרות שכותרתן Media ,Layoutוכן הלאה.

•מימין ל-reweiV נמצאת רשימת הרכיבים , שמפרטת את הרכיבים בפרויקט שלך. כל רכיב שתגרור ל-reweiV יופיע גם הוא

ברשימה זו. נכון לעכשיו, לפרוייקט מופיע רק רכיב אחד: ,Screen1המייצג את מסך המכשיר עצמו.

•מתחת לרשימת הרכיבים נמצא אזור המציג את המדיה (תמונות ו צליל) בפרויקט. לפרויקט הזה אין עדיין מדיה, אבל בקרוב תוסיף כמה.

•בקצה הימני יש קטע המציג את מאפייני הרכיבים; כאשר תלחץ על רכיב ב-reweiV, תראה את המאפיינים שלו רשומים כאן. מאפיינים הם פרטים על כל רכיב שניתן לשנות. (לדוגמה, בעת לחיצה על רכיב תווית , ייתכן שתראה מאפיינים הקשורים לצבע, טקסט, גופן וכן הלאה.) כרגע, הוא מציג את המאפיינים של המסך (הנקרא ,(Screen1הכוללים צבע רקע, א. תמונת רקע וכותרת.

עבור אפליקציית ,HelloPurrתזדקק לשני רכיבים גלויים (תחשוב על אלה כרכיבים שאתה באמת יכול לראות באפליקציה): רכיב התווית הקורא "Pet the Kitty"ורכיב Buttonעם תמונה של חתול בתוכו. תזדקק גם לרכיב סאונד שאינו נראה לעין שיודע לנגן צלילים, כגון "מיאו", ורכיב תאוצה לזיהוי מתי המכשיר מנער. אל תדאג -אנו נדריך אותך דרך כל רכיב, צעד אחר צעד.

יצירת תווית

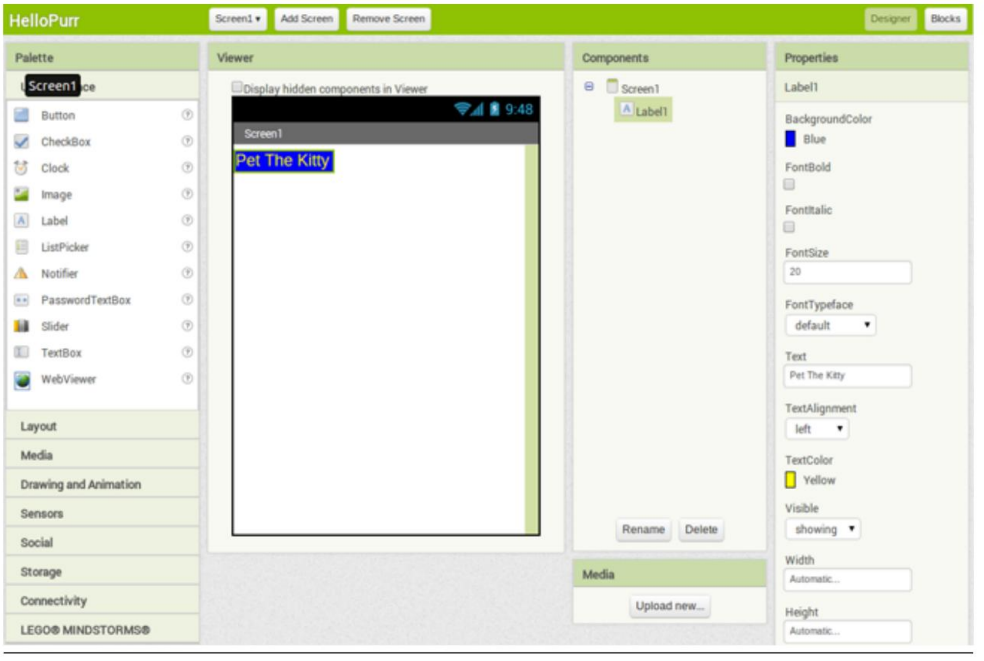
הרכיב הראשון שיש להוסיף הוא תווית:

1.עבור ,Palette-לפתח את מגירת ממשק המשתמש אם היא לא פתוחה, לחץ על Label(שמופיע בערך שישה נקודות למטה ברשימת הרכיבים), וגרור אותה ל-reweiV. תראה צורה מלבנית מופיעה ב-reweiV, המכילה את המילים "טקסט לתווית ."1

2.הסתכלו על תיבת המאפיינים בצד ימין של המעצב. זה מראה את מאפייני התווית. בערך באמצע הדרך למטה, יש מאפיין שנקרא טקסט, עם תיבה לטקסט של התווית. שנה את הטקסט ל-"teP "the Kitty"ולחץ על Return.תראה את הטקסט משתנה ב-reweiV.

3.שנה את צבע הרקע של התווית על ידי לחיצה על התיבה, אשר כעת קורא ללא, כדי לבחור צבע מהרשימה שמופיעה. בחר כחול. שנה גם את TextColor של התווית לצהוב. לבסוף, שנה את גודל הגופן ל-02.

המעצב אמור להופיע כעת כפי שמוצג באיור .5-1



איור .1-5לאפליקציה יש כעת תווית

הוספת הכפתור

הקיטי עבור HelloPurrמיושם כרכיב - Buttonאתה יוצר כפתור רגיל ואז משנה את תמונת הכפתור לקיטי. כדי להפוך את הכפתור הבסיסי תחילה, עבור Palette-לב-rengiseDולחץ על Button(בראש רשימת הרכיבים). גרור אותו אל ה-reweiV, הצב אותו מתחת לתווית. תראה כפתור מלבני מופיע ב-reweiV.

עכשיו יש לך כפתור שתשתמש בו כדי להפעיל את אפקט הצליל כאשר מישהו מקיש עליו, אבל אנחנו באמת רוצים שזה ייראה כמו התמונה של החתלתול, לא מלבן פשוט וישן. כדי לגרום לכפתור להיראות כמו חתלתול:

1. ראשית, עליך להוריד תמונה של הקיטי ולשמור אותה על שולחן העבודה של המחשב. אתה יכול להוריד אותו בכתובת bookFiles/HelloPurr/ kitty.png. http://appinventor.org/הסיומת .gnp היא פורמט תמונה סטנדרטי הדומה ל- gpj .- ו-fig; כל סוגי הקבצים הללו יעבדו ב-ppA Inventor, וכך גם רוב קבצי הקול הסטנדרטיים כגון .gpm או .3pm. אתה יכול גם להוריד את קובץ הקול שתצטרך כדי להפוך את הקיטי מיאו בכתובת http://appinventor.org/bookFiles/HelloPurr/meow.mp3.

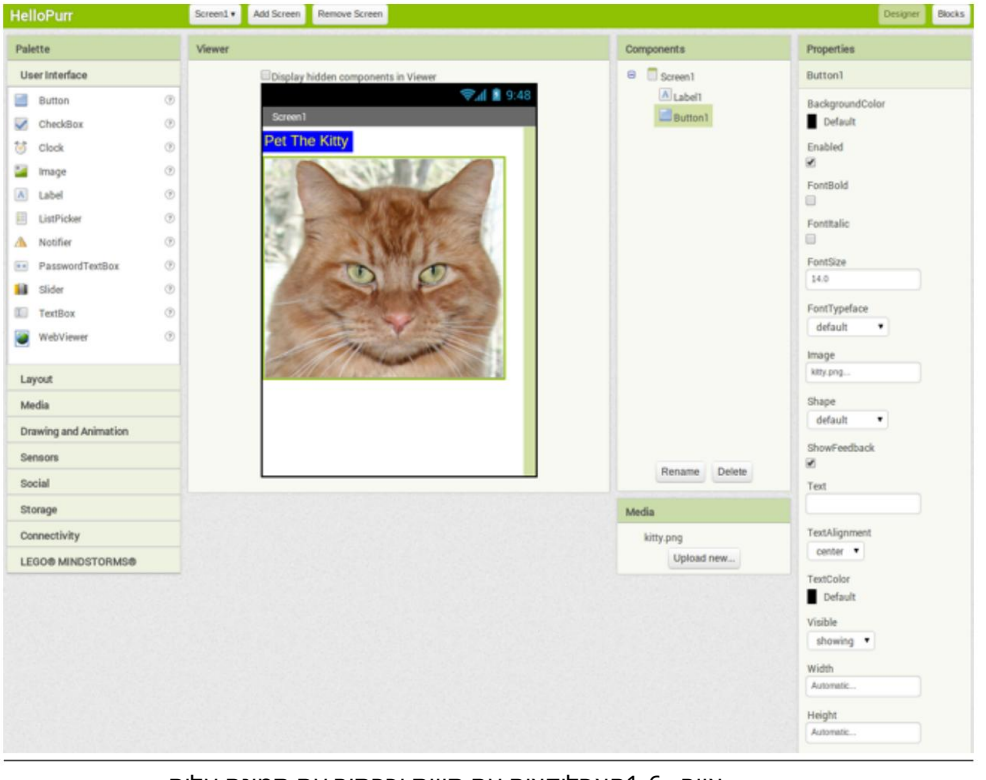
2. תיבת המאפיינים אמורה להציג את המאפיינים של הכפתור. אם לא, לחץ על הלחצן ב-reweiV כדי לחשוף את מאפייני הכפתור בצד ימין. בתיבה מאפיינים, לחץ על האזור תחת תמונה (שנכתב כרגע "ללא...").

3.לחץ על "העלה קובץ". לאחר מכן, לחץ על "בחר קובץ" ודפדף כדי לבחור את הקובץ kitty.png שהורדת למחשב שלך קודם לכן, ולאחר מכן לחץ על אישור.

4.לאחר העלאת התמונה, יש לרשום kitty.png כאופציה למאפיין התמונה של הכפתור. לחץ על אישור כדי לבחור בו. תראה את הקובץ גם ברשימה באזור המדיה של חלון המעצב, ממש מתחת לרשימת הרכיבים. ואם תסתכל על הכפתור במעצב, תראה את תמונת הקיטי מוצגת -הכפתור נראה כעת כמו חתלתול.

5.אולי גם שמתם לב שבתמונת הקיטי עדיין יש את המילים "טקסט עבור לחצן 1" מוצג עליו. אתה כנראה לא רוצה את זה באפליקציה שלך, אז קדימה ותרוקן את מאפיין הטקסט של Button1.

כעת, המעצב אמור להופיע כפי שמוצג באיור 1-6.



איור 1-6.האפליקציה עם תווית וכפתור עם תמונה עליה

הוספת צליל המיאו

באפליקציה שלך, אתה רוצה שהקיטי יבש מיאו כשאתה מקיש על הכפתור. לשם כך, תצטרך להוסיף את צליל המיאו ולתכנת את התנהגות הכפתור כדי להשמיע את הצליל הזה כאשר הלחצן נלחץ:

1.אם לא הורדת את קובץ meow.mp3לשולחן העבודה של המחשב שלך, עשה זאת כעת באמצעות קישור זה: http://appinventor.org/bookFiles/HelloPurr/meow.mp3.

2.עבור ללוח הצבעים משמאל לחלון המעצב ולחץ על הכותרת מסומן מדיה כדי להרחיב את קטע המדיה. גרור החוצה רכיב סאונד ו מקם אותו ב-reweiV. לא משנה היכן תפיל אותו, הוא יופיע באזור ב בחלק התחתון של ה-reweiV מסומן "רכיבים שאינם נראים". לא נראה לעין רכיבים הם אובייקטים שעושים דברים עבור האפליקציה אך אינם מופיעים בוויזואל ממשק משתמש.

3.לחץ על Sound1כדי להציג את המאפיינים שלו. לחץ על מאפיין המקור ולאחר מכן עבור דרך השלבים כדי להעלות ולבחור את קובץ meow.mp3שהורדת מוקדם יותר. כשתסיים, אתה אמור לראות גם kitty.pngוגם meow.mp3ברשימה מדור המדיה של המעצב.

טבלה 1-1מפרטת את הרכיבים שאספתם עבור האפליקציה שלכם עד כה.

טבלה 1-1. הרכיבים שהוספת לאפליקציית HelloPurr

סוג רכיב	קבוצת לוח שם הרכיב	מטרה	
כפתור	ממשק משתמש 1	לחץ כדי לגרום לקיטי להגיב	לַחְצָן
תווית	ממשק משתמש 1	מציג את הטקסט "Pet the kitty".	תווית
נשמע	רכיבים שאינם נראים	השמע את צליל המיאו.	צליל 1

בדיקה חיה

עם App Inventor, אתה יכול להציג ולבדוק את האפליקציה שלך במכשיר אנדרואיד בזמן שאתה ליצור אותו. בדיקת האפליקציה שלך באופן מצטבר היא תרגול המשמש את efective מפתחי תוכנה ויחסכו לכם שעות עבודה.

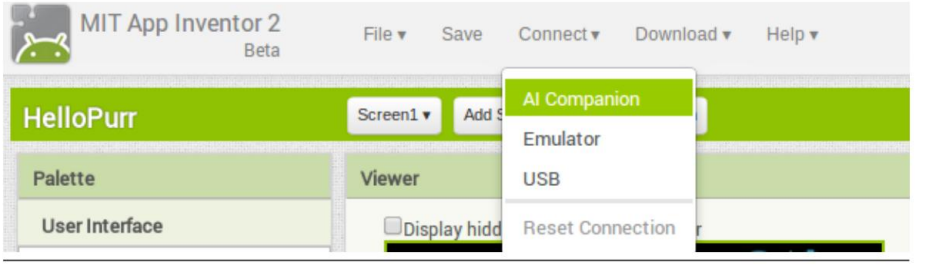
אם יש לך מכשיר אנדרואיד וחיבור לאינטרנט עם WiFi, אתה יכול להגדיר בדיקה חיה תוך דקות, ואתה לא צריך להוריד שום תוכנה שלך מחשב (רק אפליקציה בטלפון שלך). אם אין לך מכשיר אנדרואיד, יהיה לך צריך לבצע הגדרה נוספת על מנת להשתמש באמולטור, הפרטים של אשר מכוסים בכתובת http://appinventor.mit.edu/explore/2ia/setup.html.

אם יש לך מכשיר אנדרואיד, בצע את הפעולות הבאות:

1.במכשיר שלך, הורד והתקן את האפליקציה "MIT 2IA" Companionמה-חנות הגוגל סטור. הפעל את האפליקציה כשהיא מותקנת.

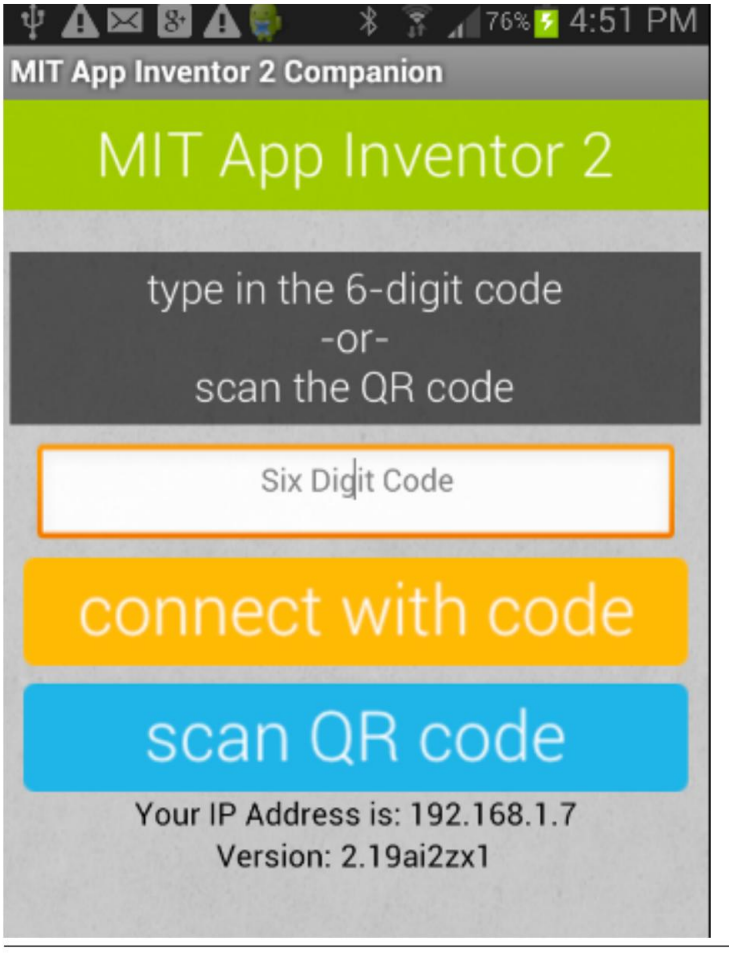
2.חבר גם את המחשב וגם את המכשיר שלך לאותו חיבור WiFi.

3. ב-ppA Inventor(בדפדפן), מהתפריט העליון, בחר Connectולאחר מכן בחר AI Companion, כפי שמוצג באיור 1-7.



איור 1-7. לחץ על התחבר ולאחר מכן בחר AI Companion

4. במכשיר שלך, הפעל את האפליקציה שהתקנת, MIT-הAI 2, Companionכמו מוצג באיור 1-8. בחר "סרוק קוד QR" ולאחר מכן החזק את המכשיר שלך עד לקוד ה-RQ על מסך המחשב כדי לסרוק אותו.



איור 1-8. במכשיר שלך, פתח את אפליקציית Companion ולחץ על "סרוק קוד QR"

אם הכל הולך כשורה, אתה אמור לראות את אפליקציית HelloPurr פועלת במכשיר שלך, כולל כל הרכיבים שהוספת. כאשר אתה מבצע שינויים ב-App Inventor Designer או Blocks Editor, שינויים אלה יופיעו גם במכשיר.

הגדרת בדיקה חיה אם אתה מתקשה להגדיר בדיקה חיה, בקר בכתובת http://appinventor.mit.edu/explore/help/2ia-setup.html.

אם האפליקציה שלך אכן מופיעה במכשיר, קדימה הקש על הכפתור. אתה חושב משהו יקרה? זה לא יקרה, מכיוון שעדיין לא הורית ללחצן לעשות שום דבר. זו הנקודה החשובה הראשונה שצריך להבין על App Inventor: עבור

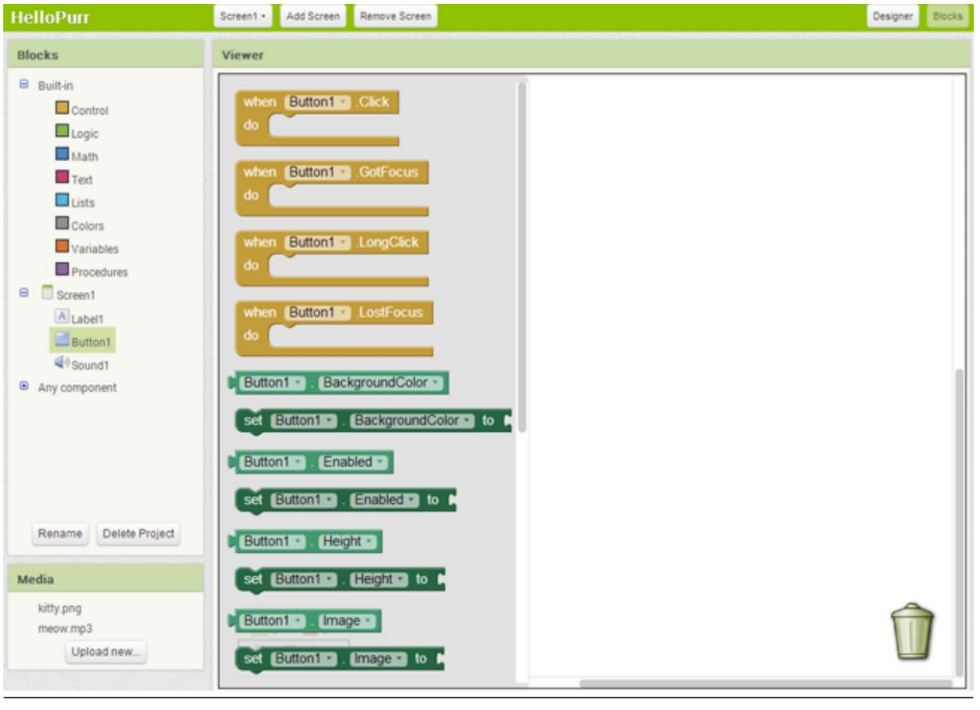
כל רכיב שתוסיף ב-rengiseD, עליך לעבור Blocks Editor-לוליצור את הקוד כדי לגרום לרכיב הזה לעשות מה שאתה רוצה שהוא יעשה.

הוספת התנהגויות לרכיבים

זה עתה הוספת רכיבי לחצן, תווית וסאונד כאבני הבניין של האפליקציה הראשונה שלך. עכשיו, בואו נגרום לקיטי מיאו כשאתה מקיש על הכפתור. אתה עושה זאת עם עורך הבלוקים. בפינה השמאלית העליונה של מעצב הרכיבים, לחץ על "בלוקים". תסתכל בחלון עורך בלוקים. זה המקום שבו אתה מורה לרכיבים מה לעשות ומתי לעשות זאת. אתה הולך לכוון את כפתור הקיטי להשמיע צליל כשהמשתמש מקיש עליו. אם רכיבים הם מרכיבים במתכון, אפשר לחשוב על בלוקים כעל הוראות הבישול.

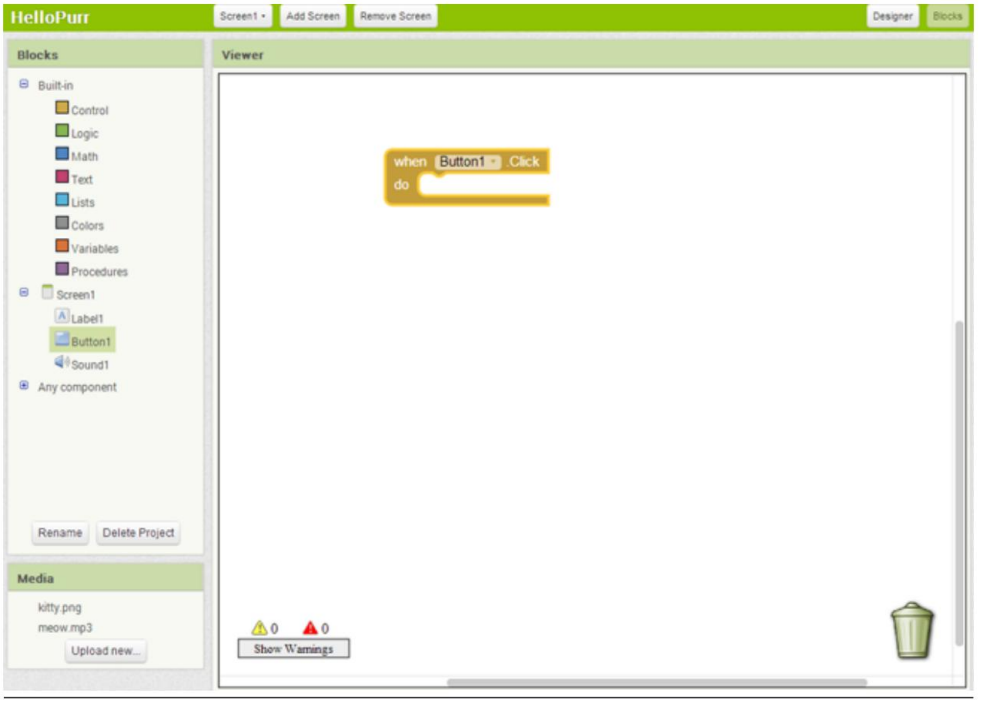
עושה את הקיטי מיאו

בפינה השמאלית העליונה של החלון, מתחת לכותרת ,Blocksתראה עמודה הכוללת מגירה מובנית ומגירה עבור כל רכיב שיצרת ב-rengiseD: Button1, Label1, Screen1 ו- .Sound1כאשר אתה לוחץ על מגירה, אתה מקבל חבורה של אפשרויות (בלוקים) עבור הרכיב הזה. לחץ על המגירה עבור כפתור . 1המגירה נפתחת, מציגה מבחר של בלוקים שבהם אתה יכול להשתמש כדי לבנות את התנהגות הכפתור, החל מ- . 1Button1לחץ בחלק העליון, כפי שמוצג באיור .1-9



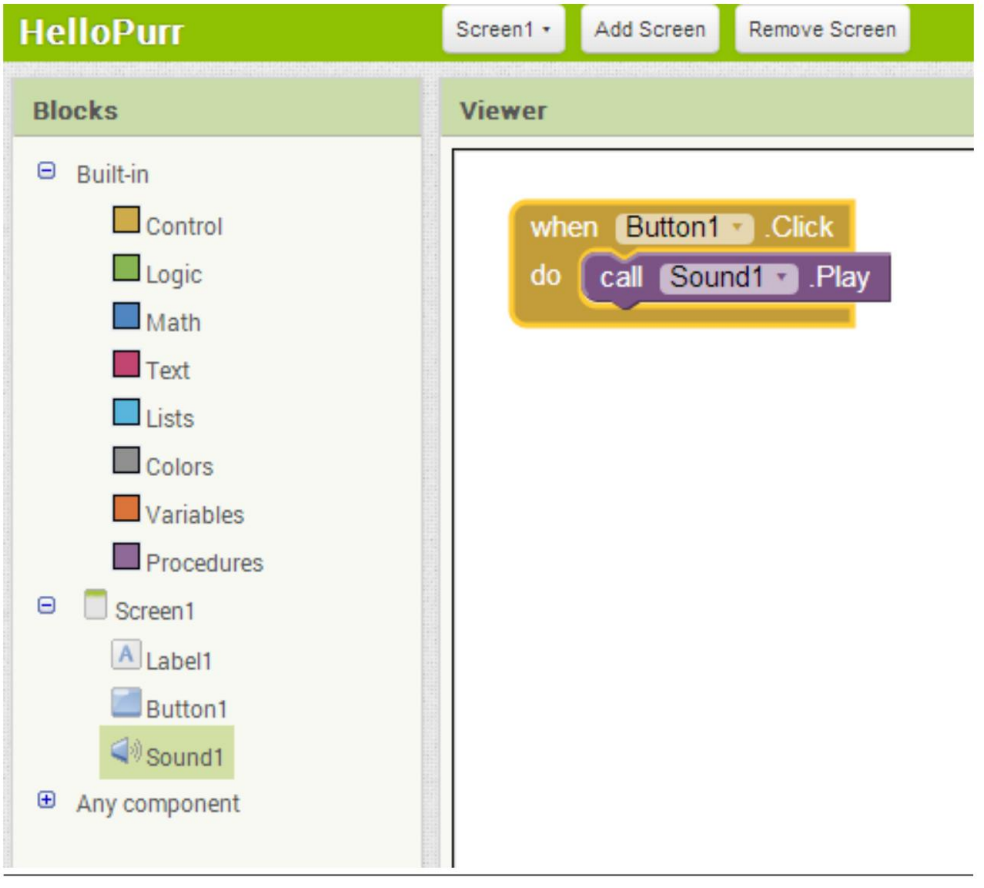
איור 1-9. לחיצה על כפתור1 מציגה את בלוקים של הרכיב

לחץ על הבלוק שכותרתו Button1. וגרור אותו אל סביבת העבודה. אתה תשים לב שהמילה "מתי" כלולה בבלוק Button1.Click. בלוקים הכוללים את המילה "מתי" נקראים מטפלי אירועים; הם מציינים מה רכיבים צריכים לעשות כאשר מתרחש אירוע מסוים. במקרה זה, האירוע שאנו מעוניינים בו מתרחש כאשר משתמש האפליקציה מקיש על התמונה של הקיטי (שזה באמת כפתור), כפי שמוצג באיור 1-10. לאחר מכן, תוסיף כמה בלוקים כדי לתכנת את מה שיקרה בתגובה לאותו אירוע.



איור .10-1תציין תגובה ללחיצה של המשתמש בתוך בלוק Button.Click

לחץ על Sound1כדי לפתוח את המגירה עבור רכיב הסאונד, ולאחר מכן גרור החוצה את התקשר .Sound1.Play block(זכור, קודם לכן הגדרנו את המאפיין עבור Sound1לwoem-sound fleשהורדת למחשב שלך.) בשלב זה, אולי שמת לב שגוש ה- call Sound1.Play מעוצב כך שהוא יכול להיכנס לרווח המסומן " do"בבלוק Button1 . Click. App Inventor מוגדר כך שרק בלוקים מסוימים נמצאים יחד; בדרך זו, אתה תמיד יודע שאתה מחבר בלוקים שבאמת עובדים יחד. במקרה זה, בלוקים עם המילה "קריאה" גורמים לרכיבים לעשות דברים. שני הבלוקים צריכים להיצמד זה לזה כדי ליצור יחידה, כפי שמוצג באיור ,11-1ותשמע צליל נקישה כשהם מתחברים.



איור 1-11. עכשיו כשמישהו לוחץ על הכפתור, צליל המיאו יתנגן

שלא כמו קוד תכנות מסורתי (שלעתים קרובות נראה כמו בלגן מבולגן של gobbledygook"מילים")), בלוקי התגובה לאירועים ב-ppA Inventorמפרטים את ההתנהגויות שאתה מנסה ליצור בצורה פשוטה ומובנת. במקרה הזה, אנחנו בעצם אומרים, "היי, ממציא אפליקציות, כשמישהו מקיש על כפתור הקיטי, השמע את צליל המיאו."

בדוק את האפליקציה שלך בדוק כדי לוודא שהכל עובד כמו שצריך -
חשוב לבדוק את האפליקציה שלך בכל פעם שאתה מוסיף משהו חדש.
הקש על הכפתור במכשיר (או לחץ עליו אם אתה משתמש באמולטור).
אתה צריך לשמוע את הקיטי מיאו.
מזל טוב, האפליקציה הראשונה שלך פועלת!

הוספת גרגר

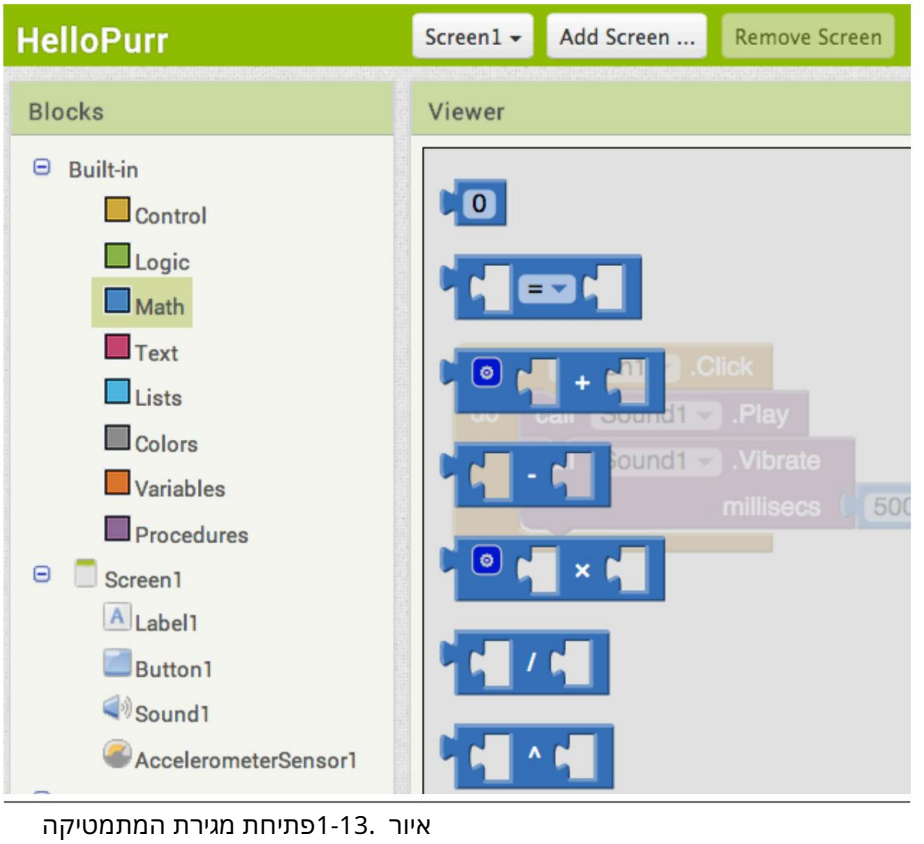
עכשיו אנחנו הולכים לגרום לקיטי לגרגר ולמיאו כשאתה מקיש על הכפתור. נדמה את הגרגר על ידי גרימת רטט של המכשיר. זה אולי נשמע קשה, אבל למעשה, קל לעשות זאת מכיוון שרכיב הסאונד שבו השתמשנו כדי להשמיע את צליל המיאו יכול לגרום למכשיר לרטוט, גם כן. App Inventorעוזר לך לנצל סוג זה של פונקציונליות ליבה של המכשיר מבלי שתצטרך להתמודד עם האופן שבו המכשיר רוטט בפועל. אתה לא צריך לעשות שום דבר שונה במעצב; אתה יכול פשוט להוסיף בלוק קריאת פונקציה שני ללחוץ על הכפתור בעורך הבלוקים:

1. עבור אל עורך הבלוקים ולחץ על Sound1כדי לפתוח את המגירה.

2. בחר התקשר סאונד1.רטט וגרור אותו מתחת לחסימת השיחה Sound1.Playבלוק חריץ לחצן .1קליק . הבלוק צריך להיצמד למקומו, כפי שמוצג באיור .12-1אם לא, נסה לגרור אותו כך שהחריץ הקטן בקצה העליון של השיחה Sound1.Vibrateנוגע בבליטה הקטנה בתחתית השיחה .Sound1.Play

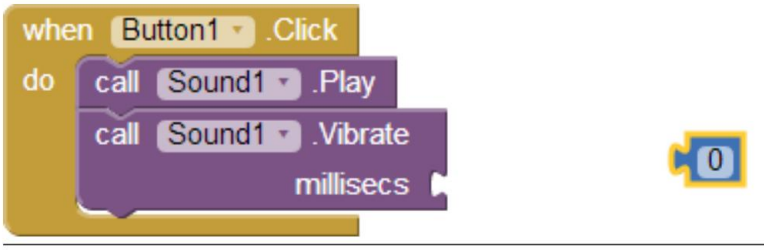
איור .12-1השמעת הצליל והרטט באירוע הקליק

3.אולי שמתם לב שה- call Sound1.Vibrate בלוק כולל את הטקסט "מיליסק" בפינה הימנית התחתונה, ולצדו יש שקע פתוח הבולט פנימה מקצה הבלוק. שקע פתוח בבלוק אומר שאתה צריך לחבר משהו לתוכו כדי לציין יותר כיצד ההתנהגות אמורה לפעול. במקרה זה, עליך לומר לבלוק הרטט כמה זמן עליו לרטוט. אתה צריך לציין את הזמן הזה באלפיות השנייה (מילישניות), וזה די נפוץ בשפות תכנות רבות. לכן, כדי לגרום למכשיר לרטוט למשך חצי שנייה, עליך להזין ערך של 500אלפיות השנייה. כדי לעשות זאת, אתה צריך לתפוס בלוק מספר. לחץ על מגירת המתמטיקה ותראה רשימה של בלוקים כחולים מופיעים, כפי שמוצג באיור 1-13.



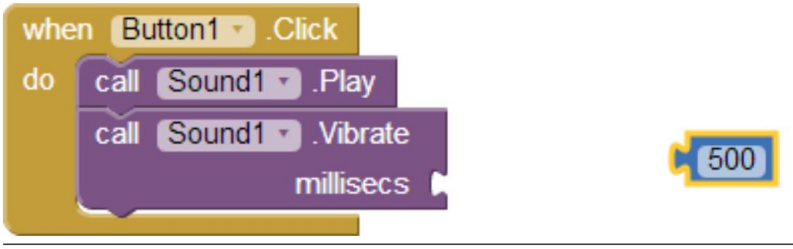
איור 1-13. פתיחת מגירת המתמטיקה

4.בראש הרשימה, אתה אמור לראות בלוק עם "0" בתוכו. אתה יכול לגרור את הבלוק הזה החוצה ואז לשנות את ה-0 לכל מספר שתרצה. קדימה וגרור החוצה את בלוק המספרים, כפי שמוצג באיור 1-14.



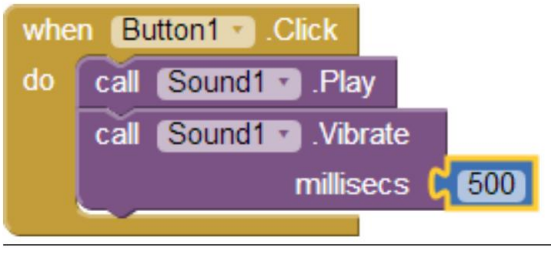
איור 1-14. בחירת בלוק מספר 0 (הוא ערך ברירת המחדל)

5. לחץ על ה-0 והקלד את הערך החדש, 500, כפי שמוצג באיור 1-15.



איור 1-15. שינוי הערך ל-005

6. חבר את בלוק המספרים 500 לשקע בצד ימין של שיחה Sound1.Vibrate, כפי שמוצג באיור 1-16.



איור 1-16. חיבור הערך 500 לשקע המילי-שניות

בדוק את האפליקציה שלך נסה אותה! הקש על הכפתור במכשיר, ותרגיש את הרגרגר למשך חצי שנייה.

מנער את המכשיר

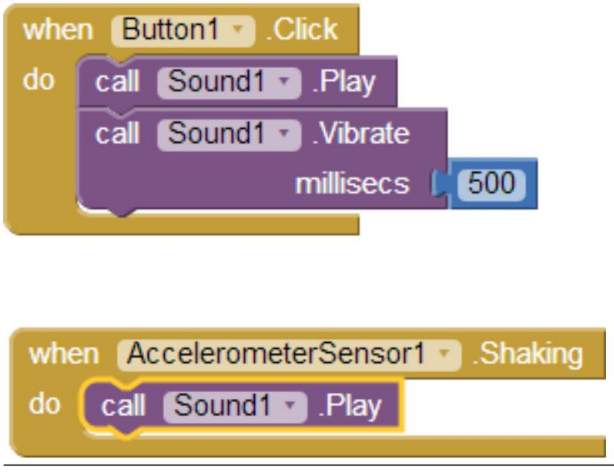
כעת, בואו נוסיף אלמנט סופי שמתחבר לתכונה מגניבה נוספת של אנדרואיד: לגרום לקיטי מיאו כאשר אתה מנער את המכשיר. לשם כך, תשתמש ברכיב הנקרא AccelerometerSensor שיכול לחוש מתי אתה מנער או מזיז את המכשיר.

1.ב-rengiseD, ברשימת רכיבי הצבעים, הרחב את אזור החיישנים וגרור החוצה חיישן תאוצה. אל תדאג לאן אתה גורר את זה. כמו בכל רכיב שאינו גלוי, לא משנה היכן תציב אותו ב-reweiV, הוא יעבור לקטע "רכיבים שאינם גלויים" בתחתית ה-reweiV.

2.תרצו להתייחס למישהו שמנער את המכשיר כאל אירוע שונה ונפרד מהלחצן ללחוץ. זה אומר שאתה צריך מטפל חדש באירועים. עבור אל עורך הבלוקים. צריכה להיות מגירה חדשה עבור AccelerometerSensor1.פתח אותו וגרור החוצה את גוש ה- AccelerometerSensor1.Shaking. זה צריך להיות הבלוק השני ברשימה.

3.בדיוק כפי שעשית עם הצליל ולחיצת הכפתור, גרור שיחה צליל .1הפעל בלוק והכנס אותו למרווח ב- AccelerometerSensor1. Shaking. נסה זאת על ידי ניעור המכשיר.

איור 1-17מציג את הבלוקים עבור אפליקציית HelloPurrשהושלמה.



איור .1-17הבלוקים עבור HelloPurr

הורדת האפליקציה למכשיר האנדרואיד שלך

תכונת הבדיקה החיה של App Inventorמאפשרת לך לבדוק בקלות את האפליקציה תוך כדי חיבור למכשיר שלך. הבעיה היחידה היא שאם תנתק את המכשיר שלך מ-ppA ,Inventor האפליקציה שפועלת במכשיר תיפסק, ולא תמצא את האפליקציה בשום מקום במכשיר מכיוון שהיא מעולם לא הותקנה באמת; זה פשוט רץ בתוך האפליקציה .App Inventor Companion

ניתן להוריד ולהתקין את האפליקציה שהושלמה כך שהיא תפעל בכל מכשיר, גם כשהוא לא מחובר למחשב. כדי להתכונן לזה, תחילה הגדר את האפליקציה

סמל כך שכאשר תתקין אותו במכשיר, הוא יופיע עם תמונה מבדלת ברשימת האפליקציות.
אתה יכול לעשות זאת ב-rengiseD על ידי בחירת רכיב המסך , לחיצה על מאפיין הסמל שלו , ולאחר מכן העלאת תמונה כסמל (למשל, תמונת הקיטי).

לאחר מכן, ודא שהמכשיר שלך מאפשר הורדת אפליקציות ממקומות אחרים
מאשר שוק אנדרואיד. עבור רוב מכשירי האנדרואיד, אתה עושה זאת על ידי מעבר
להגדרות יישומים ולאחר מכן סימון התיבה שליד "מקורות לא ידועים".
לאחר מכן, בחזרה ב-ppA ,Inventorבמעצב, לחץ על Buildובחר "אפליקציה (ספק קוד QRעבור .kpa)." אתה אמור לראות הודעת "סרגל התקדמות" בחלון, תהליך שנמשך עד דקה. כאשר מוצג קוד ה-RQ של האפליקציה המוגמרת, סרוק אותו אל המכשיר שלך באמצעות אפליקציית סורק ברקוד.1 לאחר סריקת קוד ה-RQ, ייתכן שהמכשיר יבקש ממך להזין את הסיסמה שלך עבור חשבון Googleשלך. לאחר שתסיים להזין את הסיסמה שלך, האפליקציה שלך תתחיל להוריד למכשיר שלך ותראה סמל הורדה בהודעות של המכשיר שלך. עבור אל ההתראות שלך, המתן עד שההורדה תסתיים, ולאחר מכן בחר את האפליקציה כדי להתקין אותה.
לאחר שהתקנת אותה, עיין באפליקציות הזמינות במכשיר שלך, וכעת תראה את ,HelloPurr האפליקציה שזה עתה בנינו. אתה מפעיל אותה בדיוק כמו כל אפליקציה אחרת. (ודא שאתה מפעיל את האפליקציה החדשה שלך, לא את אפליקציית .(App Inventor Companionכעת תוכל לעצור את אפליקציית Companionאו לנתק את המכשיר מהמחשב, והאפליקציה הארוזה החדשה שלך עדיין תהיה שם.
חשוב להבין שמשמעות הדבר היא שהאפליקציה הארוזה שלך נפרדת כעת מהפרויקט על .App Inventorאתה יכול לעשות יותר עבודה על הפרויקט ב-ppA Inventor על ידי חיבור המכשיר עם AI Companionכמו קודם. אבל זה לא ישנה את האפליקציה הארוזה שמותקנת כעת במכשיר שלך. אם תבצע שינויים נוספים באפליקציה שלך ב-ppA ,Inventorתרצה לארוז את התוצאה ולהוריד את הגרסה החדשה כדי להחליף את הגרסה הישנה במכשיר.

שיתוף האפליקציה

אתה יכול לשתף את האפליקציה שלך בכמה דרכים. כדי לשתף את אפליקציית ההפעלה (קובץ ה-.kpa), תחילה לחץ על Buildובחר "יישום (שמור במחשב שלי)." פעולה זו תיצור פלי עם סיומת .kpa במחשב שלך. אתה יכול לשתף את המטוס הזה עם אחרים על ידי שליחתו אליהם כקובץ מצורף לדוא"ל, שאותו הם יפתחו עם אפליקציית האימייל שלהם במכשיר שלהם. לחלופין, אתה יכול להעלות את קובץ ה- .kpa למקום כלשהו באינטרנט (למשל, .(Dropboxב רק הקפד ליידע את האנשים שמתקינים את האפליקציה שלך שהם צריכים לעשות זאת

1ישנם סורקי QR Codeרבים עבור אנדרואיד. אם אין לך אחד במכשיר שלך, עבור אל חנות Play ולהתקין אחד.

אפשר "מקורות לא ידועים" בהגדרות האפליקציה של המכשיר שלהם כדי להתקין אפליקציות שאינן מ-Market. diordnA

אתה יכול גם ליצור קוד QR עבור האפליקציה כדי שאנשים יוכלו לסרוק אותו למכשיר שלהם מהאינטרנט או אפילו פוסטר פיזי. ישנם כלים רבים שייצרו קוד QR מכתובת אתר (למשל, בדוק את qrcode.kaywa.com). לאחר מכן תוכל לגזור ולהדביק את קוד ה-RQ בדף אינטרנט או במסמך להדפסה ולפרסום.

אתה יכול גם לשתף את קוד המקור (בלוקים) של האפליקציה שלך עם מפתח אחר של App Inventor. כדי לעשות זאת, לחץ על הפרויקטים שלי, סמן את האפליקציה שברצונך לשתף (במקרה זה, HelloPurr), ולאחר מכן בחר פרויקט ייצא פרויקט (לקובץ שנוצר במחשב שלך תהיה סיומת .aia. אתה יכול לשלוח את הקובץ הזה באימייל למישהו, והוא יכול לפתוח את App Inventor, לבחור פרויקט פרויקט ייבוא ולאחר מכן לבחור ב-.aia. flee זה ייתן למשתמש עותק שלם של האפליקציה שלך, שאותו ניתן לערוך ולהתאים אישית מבלי להשפיע על הגרסה שלך.

ל-ppA Inventor תהיה בקרוב גלריית אפליקציות משלה שבה תוכל לשתף את האפליקציות שלך ולערבב מחדש את האפליקציות ממפתחים בכל העולם.

וריאציות

לאחר שתבנה את האפליקציות בספר זה, סביר להניח שתחשוב על דרכים לשפר אותן. בסוף כל פרק, נציע לך גם רעיונות להתאמה אישית שתוכל לנסות.
התאמה אישית של האפליקציות תוביל אותך לחקור את הרכיבים והבלוקים הזמינים, וללמוד לתכנת בעצמך ללא ההוראות המפורטות המופיעות במדריכים.

הנה כמה דברים שכדאי לנסות עבור HelloPurr:

•בזמן שאתה מנער את המכשיר, המיאוים יישמעו מוזרים, כאילו הם מהדהדים. הסיבה לכך היא שחיישן מד התאוצה מפעיל את אירוע הרעד פעמים רבות בשנייה בתגובה לכל תנועה של מעלה ומטה, כך שהמיאוים חופפים. אם תסתכל על רכיב הסאונד ב-rengiseD, תראה מאפיין שנקרא Minimum Interval. מאפיין זה קובע כמה קרוב זה לזה יכולים להתחיל צלילים עוקבים. כרגע הוא מוגדר על קצת פחות מחצי שנייה (400 אלפיות שניות), שזה פחות ממשך מיאו בודד.

על ידי התאמת המרווח המינימלי, אתה יכול לשנות את מידת החפיפה של המיאוים.

•אם תפעילו את האפליקציה הארוזה ותסתובבו עם המכשיר בכיס, המכשיר שלכם ימאם בכל פעם שתזוזו בפתאומיות, דבר שעשוי להיות מביך. אפליקציות אנדרואיד נועדו בדרך כלל להמשיך לפעול גם כשאתה לא מסתכל עליהן; האפליקציה שלך ממשיכה לתקשר עם מד התאוצה והמיאו פשוט ממשיך. כדי באמת לצאת מהאפליקציה, העלה

HelloPurrולחץ על כפתור התפריט של המכשיר. תוצע לך אפשרות לעצור את היישום. בחר באפשרות זו כדי לסגור את האפליקציה לחלוטין.

סיכום

להלן כמה מהמושגים שכיסינו בפרק זה:

•אתה בונה אפליקציות על ידי בחירת רכיבים ב-rengiseD, ולאחר מכן Blocks-ב עורך, אתה אומר לרכיבים מה לעשות ומתי לעשות את זה.

•חלק מהרכיבים גלויים וחלק לא. הגלויים מופיעים בממשק המשתמש של האפליקציה. אלה שאינם נראים לעין עושים דברים כמו השמעת צלילים.

•אתה מגדיר את התנהגות הרכיבים על ידי הרכבת בלוקים בעורך הבלוקים. תחילה אתה גורר החוצה מטפל באירועים, כגון ,Button1.Clickולאחר מכן מציבים בתוכו בלוקי פקודות כמו . Sound.Playכל חסימות בתוך Button1.Click יתבצעו כאשר המשתמש יקיש על הכפתור.

•פקודות מסוימות זקוקות למידע נוסף כדי לגרום להן לעבוד. דוגמה לכך היא רטט, שצריך לדעת לכמה אלפיות שניות לרטוט. ערכים אלו נקראים ארגומנטים או פרמטרים.

•מספרים מיוצגים כקוביות מספרים. אתה יכול לחבר את אלה לפקודות שלוקחים מספרים כטיעונים.

•לאפליקציה Inventorיש רכיבי חיישן. חיישן התאוצה יכול לזהות מתי המכשיר מוזז או מטלטל.

•אתה יכול לארוז את האפליקציות שאתה בונה ולהוריד אותן לטלפון, איפה הם פועלים ללא תלות ב-Inventor. ppA

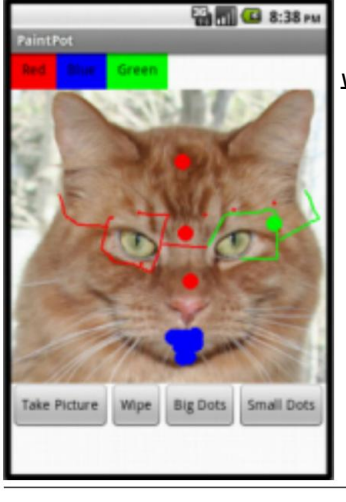
PaintPot

איור 2-1.

מדריך זה מציג את רכיב ה-savnaC ליצירת גרפיקה פשוטה ודו-ממדית .(2D)
אתה תבנה את ,PaintPotאפליקציה המאפשרת למשתמש לצייר על המסך בצבעים שונים, ולאחר מכן לעדכן אותו כך שהמשתמש יוכל לצלם תמונה ולצייר על זה במקום. בנימה היסטורית, PaintPotהייתה אחת התוכנות הראשונות שפותחו כדי להדגים את הפוטנציאל של מחשבים אישיים, עוד בשנות ה-07. אז, יצירת משהו כמו אפליקציית הציור הפשוטה הזו הייתה משימה מורכבת מאוד, והתוצאות היו די לא מלוטשות. אבל עכשיו, עם ,App Inventorכל אחד יכול להרכיב במהירות אפליקציית ציור די מגניבה, שהיא נקודת התחלה מצוינת לבניית דו-ממד משחקים.

עם אפליקציית PaintPotהמוצגת באיור ,2-1אתה יכול:

- טבלו את האצבע בסיר צבע וירטואלי כדי לצייר בצבע זה.
- גרור את האצבע לאורך המסך כדי לצייר קו.
- לתקוע את המסך כדי ליצור נקודות.
- השתמש בלחצן בתחתית כדי למחוק את המסך לְנַקוֹת.
- שנה את גודל הנקודה לגדול או קטן בעזרת הכפתורים בתחתית.
- צלם תמונה עם המצלמה ואז צייר עליה תְמוּנָה.



איור 2-2.אפליקציית PaintPot

מה תלמד

מדריך זה מציג את המושגים הבאים:

•שימוש ברכיב Canvasלציור.

•טיפול באירועי מגע וגרירה על פני המכשיר.

•שליטה בפריסת המסך עם רכיבי סידור.

•שימוש במטפלי אירועים בעלי ארגומנטים.

•הגדרת משתנים כדי לזכור דברים כמו גודל הנקודה עבורו בחר המשתמש צִיוּר.

מתחילים

נווט לאתר App Inventor.התחל פרויקט חדש וקרא לו "PaintPot".לחץ על התחבר והגדר את המכשיר (או האמולטור) שלך לבדיקה חיה (ראה http://appinventor.mit.edu/explore/2iah/setupלעזרה בהגדרה).

לאחר מכן, בצד ימין של המעצב, עבור לחלונית המאפיינים ושנה את כותרת המסך ל-"PaintPot".אתה אמור לראות את השינוי הזה במכשיר, כשהכותרת החדשה מוצגת בשורת הכותרת של האפליקציה שלך.

אם אתה מודאג מבלבול בין שם הפרויקט שלך לשם המסך, אל תדאג! ישנם שלושה שמות מפתח ב-App Inventor:

•השם שתבחרו לפרויקט שלכם בזמן שאתם עובדים עליו. זה יהיה גם שם האפליקציה כשתארוז אותה עבור המכשיר. שים לב שאתה יכול ללחוץ על פרוייקט ושמירה בשם Component Designer-בכדי להתחיל גרסה חדשה או לשנות שם של פרוייקט.

•שם הרכיב, Screen1,שתראה בחלונית Components.לא ניתן לשנות את השם של מסך ראשוני זה בגרסה הנוכחית של App Inventor.

•הכותרת של המסך, שהיא מה שתראה בשורת הכותרת של האפליקציה. זה מתחיל להיות זהה לשם הרכיב, Screen1,וזה מה שהשתמשת בו HelloPurr.-באבל אתה יכול לשנות את זה, כפי שעשינו זה עתה עבור PaintPot.

עיצוב הרכיבים

תשתמש ברכיבים הבאים כדי ליצור את האפליקציה:

•שלושה רכיבי כפתור לבחירת צבע אדום, כחול או ירוק, וכן א
רכיב סידור אופקי לארגון אותם.

•רכיב לחצן אחד לניגוב הציור, שניים לשינוי גודל הנקודות המצויירות ואחד להפעלת המצלמה לצלם.

•רכיב , Canvasשהוא משטח הציור. לקנבס יש א
מאפיין , BackgroundImageאותו תגדיר לקובץ kitty.pngמתוך המדריך של HelloPurrבפרק .1בהמשך פרק זה, תשנה את האפליקציה כך שניתן להגדיר את הרקע לתמונה שהמשתמש מצלם.

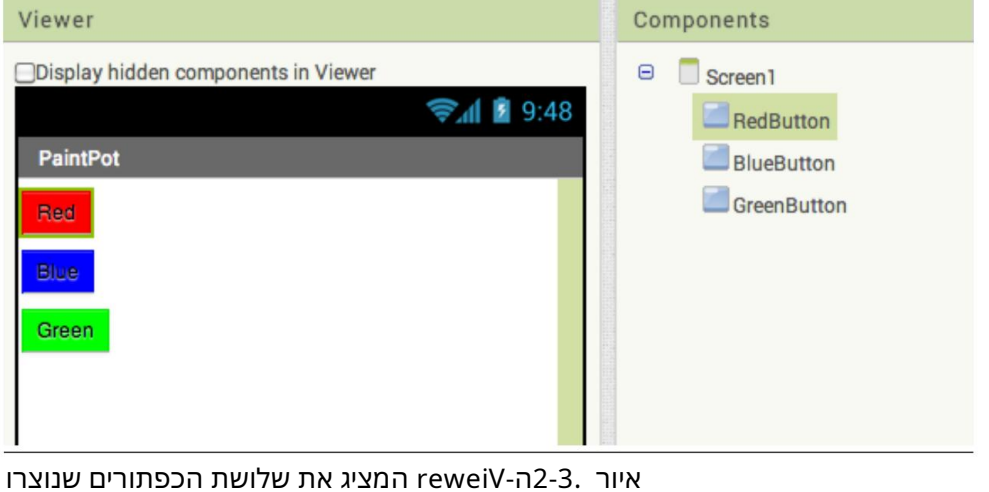
יצירת כפתורי הצבע

ראשית, צור את שלושת הכפתורים הצבעוניים על ידי ביצוע ההוראות הבאות:

.1גרור רכיב לחצן אל ה-reweiV, שנה את תכונת הטקסט שלו ל"אדום", ולאחר מכן להפוך את צבע הרקע שלו לאדום.

.2ב-reweiV, ברשימת הרכיבים, לחץ על Button1כדי להדגיש אותו (יכול להיות כבר מודגש) ולחץ על שנה שם כדי לשנות את שמו מ- Button1ל- RedButton. שים לב שרווחים אינם מותרים בשמות הרכיבים, ולכן מקובל להשתמש באות רישיות באות הראשונה של כל מילה בשם.

.3באופן דומה, צור שני כפתורים נוספים עבור כחול וירוק, בשם BlueButtonו GreenButton, הצבתם מתחת לכפתור האדום בצורה אנכית. בדוק את העבודה שלך עד לנקודה זו מול איור 2-2.



איור 2-3. המציג את שלושת הכפתורים שנוצרו ה-reweiV

שימו לב שבפרויקט הזה אתם משנים את שמות הרכיבים במקום זאת מאשר להשאיר אותם בתור שמות ברירת המחדל, כפי שעשית עם .HelloPurrמשתמש יותר

עיצוב הרכיבים

שמות משמעותיים הופכים את הפרויקטים שלך לקריאה יותר, וזה באמת יעזור כשאתה עובר לעורך הבלוקים וחייב להתייחס לרכיבים בשמם. בספר זה, נשתמש במוסכמה ששם הרכיב מסתיים בסוג שלו (לדוגמה, .(RedButton

בדוק את האפליקציה שלך אם לא לחצת על התחבר וחיברת את המכשיר שלך, עשה זאת כעת ובדוק כיצד האפליקציה שלך נראית במכשיר שלך או באמולטור.

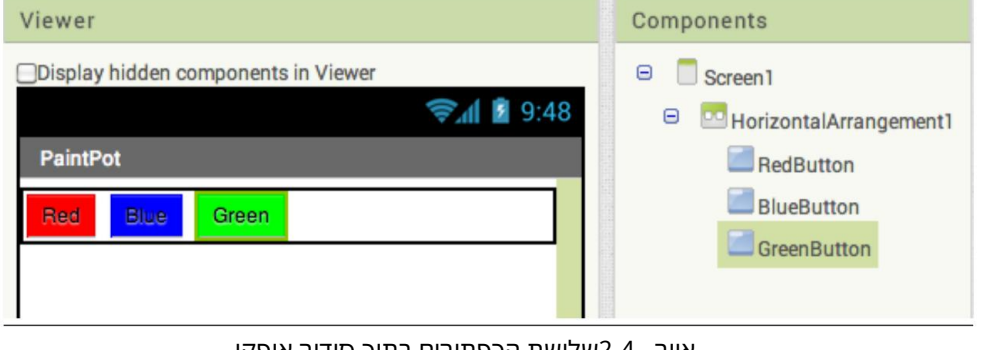
שימוש בהסדרים לפריסות טובות יותר

כעת אמורים להיות לך שלושה כפתורים מוערמים אחד על גבי השני. אבל עבור האפליקציה הזו, אתה רוצה שכולם יהיו מסודרים זה לצד זה, על פני החלק העליון של המסך, כפי שמוצג באיור .2-3אתה עושה זאת באמצעות רכיב :Horizontal Arrangement

1. ממגירת ה-tuoyaL של הפלטה, גרור החוצה רכיב HorizontalArrangement והנח אותו מתחת ללחצנים.

2. בחלונית Properties, שנה את Width of Horizontal Arrangement ל-"lliF parent" כך שימלא את כל רוחב המסך.

3. הזז את שלושת הכפתורים בזה אחר זה לתוך ה- Horizontal Arrangement רְכִיב. רמז: תראה קו אנכי כחול המראה לאן ילך היצירה שאתה גורר.



איור .2-4שלושת הכפתורים בתוך סידור אופקי

אם תסתכל ברשימת רכיבי הפרויקט, תראה את שלושת הכפתורים מבולגנים תחת הרכיב .Horizontal Arrangementזה מציין שהלחצנים הם כעת רכיבי משנה של רכיב .1 HorizontalArrangementשימו לב שכל הרכיבים מוכנסים מתחת למסך

אתה יכול למרכז את כל שורת הכפתורים על המסך על ידי שינוי מסך 1
יישר מאפיין אופקי ל"מרכז".

בדוק את האפליקציה שלך במכשיר, אתה אמור לראות את שלושת הכפתורים שלך מסודרים בשורה בחלק העליון של המסך, אם כי ייתכן שהדברים לא ייראו בדיוק כפי שהם נראים ב-rengiseD. לדוגמה, המתאר סביב HorizontalArrangementמופיע ב-reweiV אך לא במכשיר.

באופן כללי, אתה משתמש בסידורי מסך כדי ליצור פריסות אנכיות, אופקיות או טבלאות פשוטות. ניתן גם ליצור פריסות מורכבות יותר על ידי הכנסת (או קינון) רכיבי סידור מסך זה בתוך זה.

הוספת הקנבס

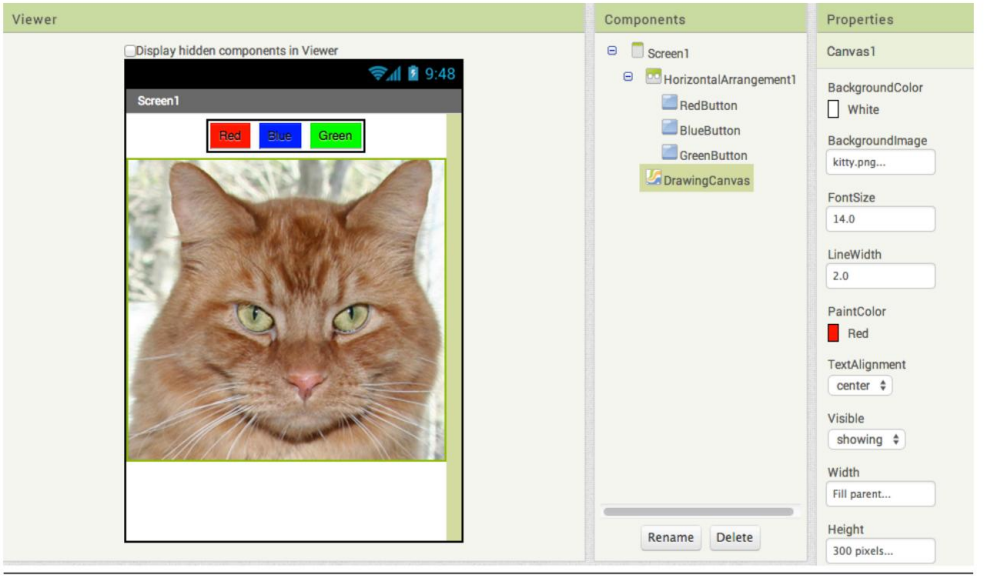
השלב הבא הוא להגדיר את הקנבס שבו יתרחש הציור:

1. ממגירת הציור וההנפשה של הפלטה, גרור רכיב Canvas אל ה-reweiV. שנה את שמו ל- DrawingCanvas. הגדר את הרוחב שלו ל"מלא הורה" כך שהוא יתפרש על כל רוחב המסך. הגדר את הגובה שלו ל-003 פיקסלים, מה שישאיר מקום לשתי שורות הכפתורים.

2. אם השלמת את המדריך של HelloPurr (פרק 1), כבר
הורד את הקובץ kitty.png. אם לא עשית זאת, תוכל להוריד אותו בכתובת
http://appinventor.org/bookFiles/HelloPurr/kitty.png.

3. הגדר את תמונת הרקע של DrawingCanvas לקובץ kitty.png. בתוך ה בקטע מאפיינים של מעצב הרכיבים, תמונת הרקע תוגדר ל-enoN. לחץ על השדה והעלה את הקובץ kitty.png.

4. הגדר את PaintColor של DrawingCanvas לאדום כך שכאשר המשתמש יתחיל את האפליקציה אך עדיין לא לחץ על כפתור, צבע ברירת המחדל שלו יהיה אדום. בדוק שמה שבנית נראה כמו איור 4-2.



איור .2-5לרכיב DrawingCanvasיש תמונת רקע של תמונת הקיטי

סידור הכפתורים התחתונים ורכיב המצלמה

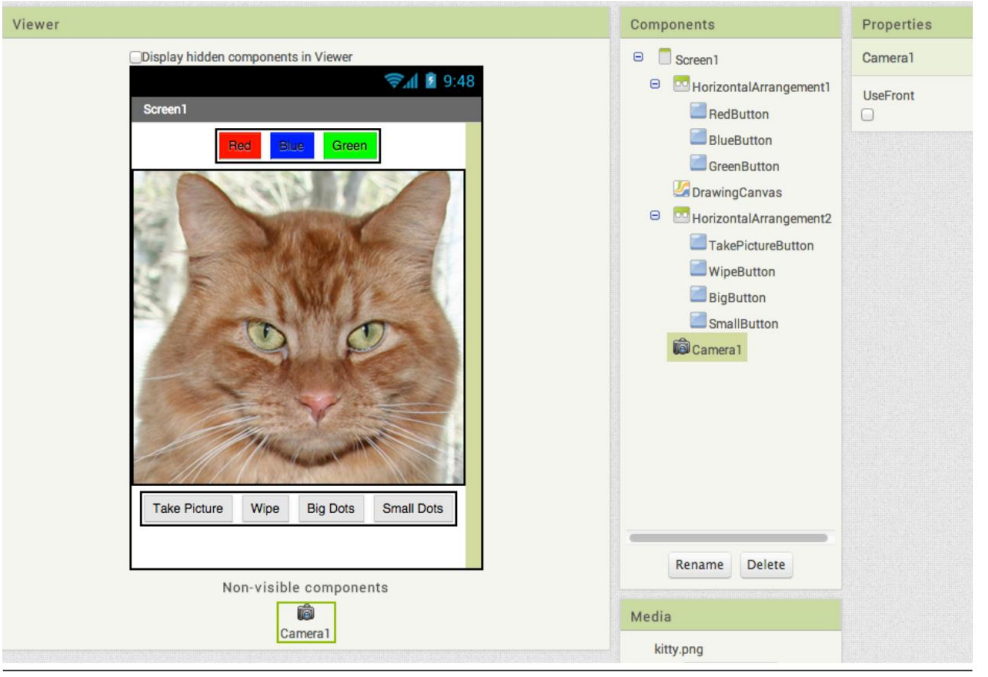
.1מהלוח, גרור סידור אופקי שני והנח אותו מתחת לקנבס. לאחר מכן, גרור שני רכיבי כפתור נוספים על המסך והצב אותם בסידור אופקי תחתון זה. שנה את שם הכפתור הראשון ל- TakePictureButtonואת תכונת הטקסט שלו ".Take Picture"-לשנה את שם הכפתור השני ל- WipeButtonואת מאפיין הטקסט שלו ל-"epiW".

.2גרור שני רכיבי כפתור נוספים מהלוח לתוך ה- HorizontalArrangement,והצב אותם ליד WipeButton.

.3תן שם ללחצנים BigButtonו- SmallButton, והגדר את הטקסט שלהם ל-"stoDgiB" ו"נקודות קטנות", בהתאמה.

.4ממגירת המדיה, גרור רכיב מצלמה אל ה-reweiV. זה יהיה מופיעים באזור הרכיבים הלא גלויים.

כעת השלמת את השלבים להגדרת המראה של האפליקציה שלך, שאמור להיות נראה כמו איור .2-5



איור .2-6משק המשתמש המלא עבור PaintPot

בדוק את האפליקציה שלך בדוק את האפליקציה במכשיר. האם תמונת הקיטי מופיעה כעת מתחת לשורת הכפתורים העליונה? האם שורת הכפתורים התחתונה נמצאת מתחת לתמונה?

הוספת התנהגויות לרכיבים

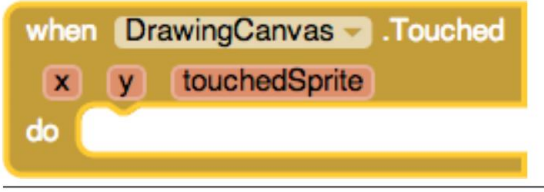
השלב הבא הוא להגדיר כיצד הרכיבים מתנהגים. יצירת תוכנית ציור עשויה להיראות מכריעה, אבל תהיו בטוחים ש-ppAInventor עשה עבורכם הרבה מהעבודה הכבדה: ישנם בלוקים קלים לשימוש לטיפול בפעולות המגע והגרירה של המשתמש, ולציור וצילום תמונות.

ב-rengiseD, הוספת רכיב Canvas בשם .DrawingCanvasכמו כל רכיבי ה-savnaC, ל-DrawingCanvasיש אירוע נגע ואירוע נגרר . אתה תתכנת את אירוע DrawingCanvas.Touchedכך שיצויר עיגול בתגובה לנגיעה של המשתמש ב-regnf שלה על המסך. תתכנת את האירוע DrawingCanvas.Draggedכך שקו יצויר כשהמשתמש גורר את ה-regnF שלה על פני הבד. לאחר מכן תתכנת את הכפתורים לשנות את צבע הציור, לנקות את הבד ולשנות את רקע הבד לתמונה שצולמה עם המצלמה.

הוספת אירוע מגע כדי לצייר נקודה

ראשית, תסדר את הדברים כך שכאשר אתה נוגע ב-savnaCgniwarD, תצייר נקודה בנקודת המגע:

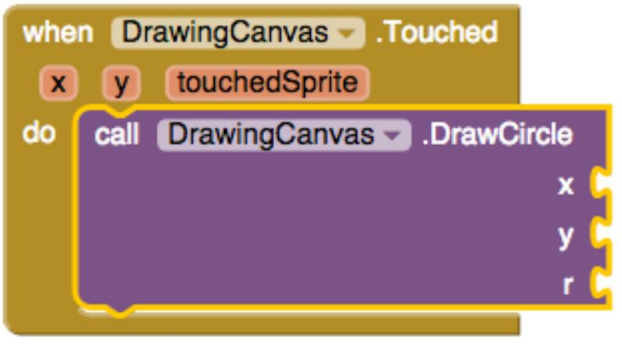
1. בעורך בלוקים, בחר את המגירה עבור DrawingCanvasולאחר מכן גרור את הבלוק DrawingCanvas.Touchedאל סביבת העבודה. לבלוק יש פרמטרים עבור y ,x-, touchedSpriteכפי שמוצג באיור .2-6פרמטרים אלה מספקים מידע על מיקום המגע.



איור .2-7האירוע מגיע עם מידע על היכן נוגעים במסך

הערה אם השלמת את אפליקציית HelloPurrבפרק ,1אתה מכיר את אירועי , Button.Clickאך לא את אירועי Canvas. אירועי Button.Clickהם פשוטים למדי מכיוון שאין מה לדעת על האירוע מלבד זה שהוא קרה.
עם זאת, חלק ממטפלי האירועים מגיעים עם מידע על האירוע הנקרא ארגומנטים. האירוע DrawingCanvas.Touchedמספק את קואורדינטות ה-x וה-y של המגע בתוך .DrawingCanvasזה גם מאפשר לך לדעת אם אובייקט בתוך ציור קנבס (בדוגמה שלנו, Inventorזה נקרא ספרייט) היה נגע, אבל לא נצטרך את זה עד פרק .3קואורדינטות ה-x וה-y הן הארגומנטים שבהם נשתמש כדי לזהות היכן המשתמש נגע במסך. אז נוכל לצייר את הנקודה על זה עמדה.

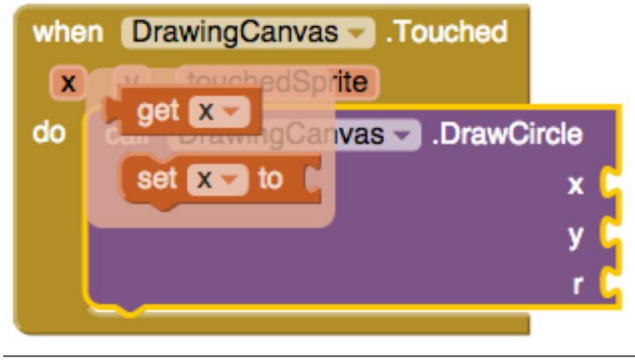
2. מהמגירה, DrawingCanvasגרור החוצה פקודת DrawingCanvas.DrawCircle והצב אותה בתוך המטפל באירוע , DrawingCanvas.Touchedכפי שמוצג באיור .2-7



איור 2-8. כאשר המשתמש נוגע בקנבס, האפליקציה מציירת עיגול

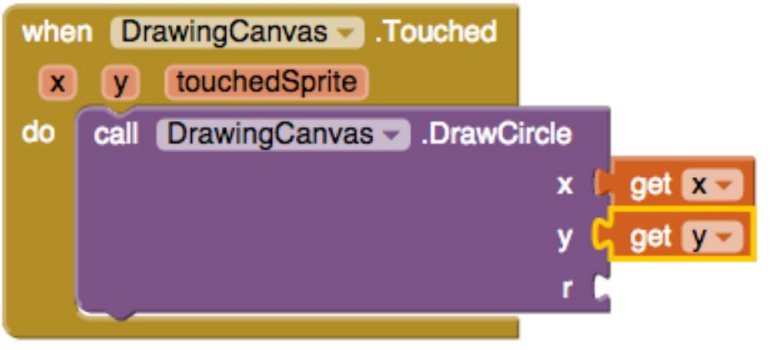
בצד ימין של בלוק DrawingCanvas.DrawCircle, תראה שלושה שקעים עבור הארגומנטים שאנו צריכים לחבר: x, y, r. הארגומנטים x ו-y מציינים את המיקום שבו יש לצייר את העיגול, ו-r קובע את הרדיוס (או הגודל) של המעגל. מטפל באירועים זה יכול להיות מעט מבלבל מכיוון שלאירוע DrawingCanvas.Touched יש גם x ו-y; רק זכור שה-x ו-y עבור האירוע DrawingCanvas.Touched מציינים היכן המשתמש נגע, בעוד שה-x ו-y עבור האירוע DrawingCanvas.DrawCircle הם שקעים פתוחים עבורך כדי לציין היכן יש לצייר את העיגול. מכיוון שאתה רוצה לצייר את העיגול שבו המשתמש נגע, תחבר את ערכי ה-x וה-y מ- DrawingCanvas.Touched כערכים לשימוש עבור הפרמטרים x ו-y ב- DrawingCanvas.DrawCircle.

הערה ניתן לגשת לערכי פרמטר האירוע על ידי העברת העכבר מעליהם בבלוק When, כפי שמוצג באיור 2-8.



איור 2-9. העבר את העכבר מעל פרמטר אירוע כדי לגרור בלוק גט להשגת הערך

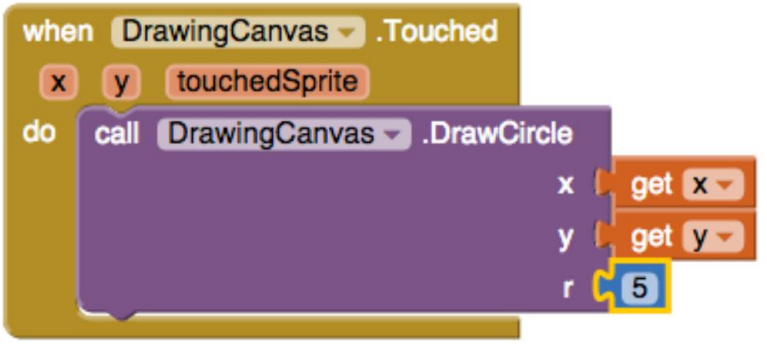
3. גרור הוצאת בלוקים עבור ערכי xו- yוחבר אותם לשקעים פנימה בלוק , DrawingCanvas.DrawCircleכפי שמוצג באיור 2-9.



איור 2-10. האפליקציה יודעת כעת היכן לצייר (x,y), אבל אנחנו עדיין צריכים לציין כמה גדול המעגל צריך להיות

4. כעת, תצטרך לציין את הרדיוס, r, של המעגל לציור. הרדיוס נמדד בפיקסלים, שהיא הנקודה הקטנה ביותר שניתן לצייר על מסך המכשיר. לעת עתה, הגדר אותו ל-5: לחץ על אזור ריק במסך, הקלד 5 ולאחר מכן הקש Return (זה יצור בלוק מספרים באופן אוטומטי), ואז חבר את זה לשקע r. כאשר תעשה זאת, התיבה הצהובה בפינה השמאלית התחתונה תחזור ל-0 מכיוון שכל השקעים מלאים כעת. איור 2-10 מחיש כיצד צריך להיראות המטפל הסופי באירוע DrawingCanvas.Touched.

הערה אם תקליד "5" בעורך בלוקים ותלחץ על Return, יופיע בלוק מספר עם "5". בו. תכונה זו נקראת חסימת סוג: אם אתה מתחיל להקליד, עורך הבלוקים מציג רשימה של בלוקים ששמותיהם תואמים למה שאתה מקליד; אם אתה מקליד מספר, זה יוצר בלוק מספר.



איור 2-11. כאשר המשתמש נוגע ב-DrawingCanvas, מעגל ברדיוס 5 מצויר במיקום המגע (x,y)

בדוק את האפליקציה שלך נסה את מה שיש לך עד כה במכשיר. כאשר אתה נוגע ב-DrawingCanvas, ה-sprite שלך צריך להשאיר נקודה בכל מקום שאתה נוגע בו. הנקודות יהיו אדומות אם תגדיר את המאפיין DrawingCanvas.PaintColor לאדום Designer ב-Component (אחרת, הוא שחור, מכיוון שזו ברירת המחדל).

הוספת אירוע הגרירה שמצייר קו

לאחר מכן, תוסיף את המטפל באירועי גרירה. הנה ההבדל בין נגיעה לגרירה:

- מגע הוא כאשר אתה מניח את האצבע שלך על DrawingCanvas ולאחר מכן מרים אותו מבלי להזיז אותו.
- גרירה היא כאשר אתה מניח את האצבע שלך על DrawingCanvas ומזיז אותו מסביב תוך שמירה על מגע עם המסך.

בתוכנית צביעה, נראה שגרירת ה-sprite שלך בקשת על פני המסך מצייר קו מעוקל שעוקב אחר הנתיב של ה-sprite שלך. מה שאתה בעצם עושה זה לצייר קווים זעירים וישרים רבים; בכל פעם שאתה מזיז את ה-sprite שלך, אפילו קצת, אתה משרטט את הקו מהמיקום האחרון של ה-sprite שלך למיקום החדש שלו.

1. מהמגירה, DrawingCanvas גרור את הבלוק DrawingCanvas.Dragged אל סביבת העבודה. אתה אמור לראות את המטפל באירוע כפי שהוא מוצג באיור 2-11.

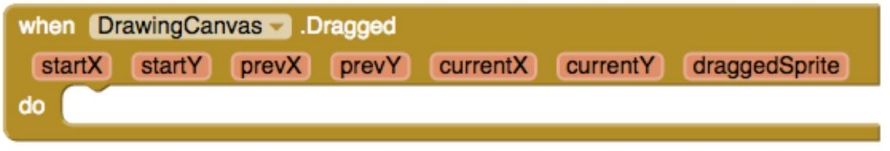
האירוע DrawingCanvas.Dragged מגיע עם הטיעונים הבאים:

- startX, startY: מיקום האצבע בנקודה שבה הגרירה התחיל.

currentX, currentY: המיקום הנוכחי של האצבע שלך

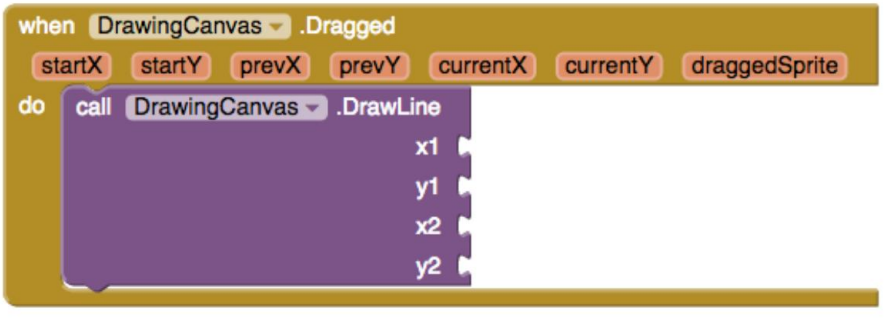
prevX, prevY: המיקום הקודם של האצבע שלך.

draggedSprite: ערך בוליאני, זה יהיה נכון אם המשתמש יגרור ישירות על ספרייט תמונה. לא נשתמש בטיעון זה במדריך זה.



איור 2-12. לאירוע שנגרר יש אפילו יותר טיעונים מאשר ב-dehcuoT

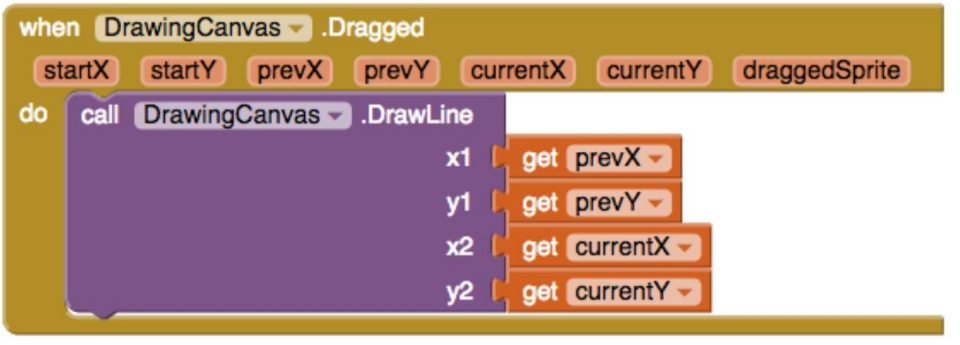
2. מהמגירה DrawingCanvas, גרור את הבלוק DrawingCanvas.DrawLine לתוך הבלוק DrawingCanvas.Dragged, כפי שמוצג באיור 2-12.



איור 2-13. הוספת היכולת לשרטט קווים

לבלוק DrawingCanvas.DrawLine יש ארבעה ארגומנטים, שניים לכל נקודה שקובעת את הקו. (x1,y1) היא נקודה אחת, ואילו (x2,y2) היא השנייה. האם אתה יכול להבין אילו ערכים צריכים להיות מחוברים לכל ארגומנט? זכור, האירוע הנגרר ייקרא פעמים רבות כאשר אתה גורר את ה-regnF שלך על פני DrawingCanvas. האפליקציה מצייר קו קטנטן בכל פעם שה-regnf שלך עובר, מ- (prevx,prevy) ל- (currentX,currentY).

3. גרור החוצה קבל בלוקים עבור הארגומנטים שאתה צריך. get prevX A וקבל prevY צריך להיות מחובר לשקעי 1x-1y, בהתאמה. יש לחבר את get currentX- ו get currentY לשקעי 2x-2y, בהתאמה, כפי שמוצג באיור 2-13.



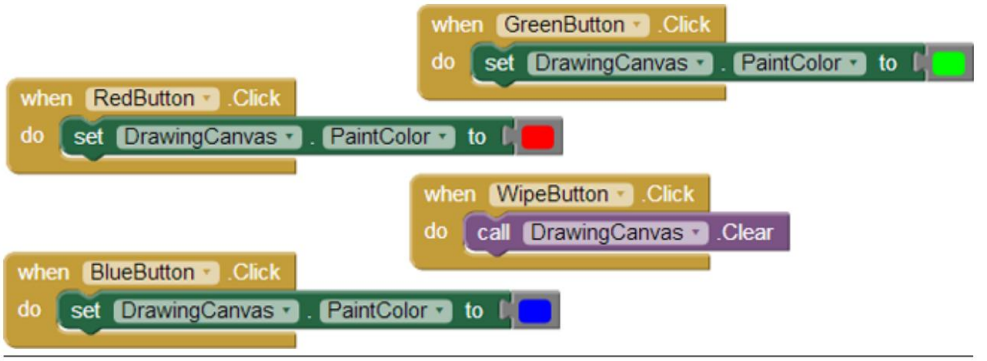
איור 2-14. כשהמשתמש גורר, האפליקציה תשרטט קו מהנקודה הקודמת לנקודה הנוכחית

בדוק את האפליקציה שלך נסה התנהגות זו במכשיר. גרור את ה-regnF שלך על המסך כדי לצייר קווים ועיקולים. גע במסך כדי ליצור נקודות.

שינוי הצבע

האפליקציה שבניתם מאפשרת למשתמש לצייר, אבל עד כה הכל היה באדום. לאחר מכן, הוסף מטפלי אירועים עבור כפתורי הצבע כדי שמשתמשים יוכלו לשנות את צבע הצבע, ועוד אחד עבור ה-nottuBepiW כדי לאפשר להם לנקות את המסך ולהתחיל מחדש.
בעורך בלוקים:

1. פתח את המגירה עבור RedButton וגרור החוצה את בלוק RedButton.Click .
2. פתח את מגירת DrawingCanvas. גרור החוצה את הסט DrawingCanvas.PaintColor לחסימה (ייתכן שתצטרך לגלול ברשימת הבלוקים במגירה כדי למצוא אותו) והצב אותו בקטע "עשה" של RedButton.Click.
3. פתחו את מגירת הצבעים וגררו החוצה את הבלוק עבור הצבע האדום וחברו אותו ל-DrawingCanvas.PaintColor להגדיר לחסום.
4. חזור על שלבים 2-4 עבור הלחצנים הכחולים והירוקים.
5. הכפתור האחרון להגדרה הוא WipeButton. גרור החוצה nottuBepiW. לחץ מהמגירה WipeButton, גרור החוצה את DrawingCanvas מהמגירה DrawingCanvas.Clear והנח אותו בבלוק WipeButton.Click. ודא שהבלוקים שלך מופיעים כפי שהם מופיעים באיור 2-14.



איור 2-15. לחיצה על לחצני הצבע משנה את צבע הציור של DrawingCanvas; לחיצה על מחק מנקה את המסך

בדוק את האפליקציה שלך נסה את ההתנהגויות על ידי לחיצה על כל אחד מכפתורי הצבע וראה אם אתה יכול לצייר עיגולים בצבעים שונים. לאחר מכן, לחץ על הלחצן מחק כדי לראות אם בד הציור נוקה.

מתן אפשרות למשתמש לצלם תמונה

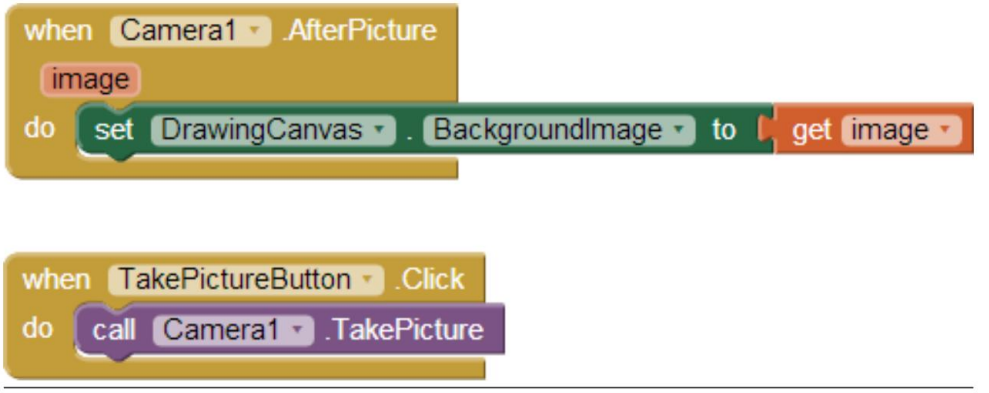
אפליקציות Inventor של אפליקציות יכולות ליצור אינטראקציה עם התכונות החזקות של מכשיר אנדרואיד, כולל המצלמה. כדי לתבל את האפליקציה, תנו למשתמש להגדיר את הרקע של הציור על ידי צילום תמונה עם המצלמה.

לרכיב המצלמה שני בלוקים מרכזיים. בלוק Camera.TakePicture מפעיל את אפליקציית המצלמה במכשיר. האירוע Camera.AfterPicture מופעל כאשר המשתמש סיים לצלם את התמונה. תוסיף בלוקים במטפל האירועים Camera.AfterPicture כדי להגדיר את DrawingCanvas.BackgroundImage לתמונה שהמשתמש צילם זה עתה.

1. פתח את מגירת TakePictureButton וגרור את TakePictureButton.Click מטפל באירועים לתוך סביבת העבודה.
2. מתוך Camera1, גרור החוצה את Camera1.TakePicture והצב אותה ב- TakePictureButton.Click מטפל באירועים.
3. מתוך Camera1, גרור את המטפל באירועים Camera1.AfterPicture לתוך סביבת עבודה.
4. מתוך DrawingCanvas, גרור את הסט DrawingCanvas.BackgroundImage לחסימה והנח אותו במטפל האירועים Camera1.AfterPicture .
5. למצלמה1 AfterPicture יש ארגומנט בשם תמונה, שהיא התמונה שצולמה זה עתה. אתה יכול לקבל הפניה אליו באמצעות בלוק get מה-

Camera1.AfterPictureבלוק, ולאחר מכן חבר אותו
DrawingCanvas.BackgroundImage. -ל

הבלוקים צריכים להיראות כמו איור 2-15.



איור 2-16. כאשר התמונה צולמה, היא מוגדרת כתמונת רקע עבור DrawingCanvas

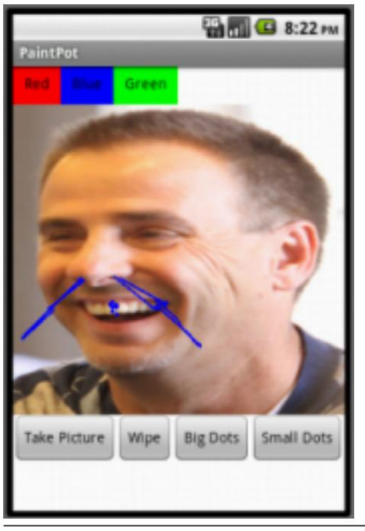
בדוק את האפליקציה שלך נסה את ההתנהגות הזו על ידי לחיצה על צלם תמונה במכשיר שלך וצילום תמונה. התמונה של החתול צריכה להשתנות לתמונה שזה עתה צילמת, ואז תוכל לצייר על התמונה הזו. (ציור על פרופסור וולבר הוא הבילוי המועדף על תלמידיו, כפי שמוצג באיור 2-16.)

שינוי גודל הנקודה

גודל הנקודות המצוירות ב-savnaCgniwarD נקבע בקריאה ל- DrawingCanvas.DrawCircle, כאשר ארגומנט הרדיוס rמוגדר ל-5. כדי לשנות את הגודל, ניתן להכניס ערך שונה עבור r. כדי לבדוק זאת, נסה לשנות את ה-5 ל-01 ולבדוק אותו במכשיר כדי לראות איך הוא נראה.

הקאץ' כאן הוא שהמשתמש מוגבל לכל גודל שתגדיר ברדיוס
טַעֲנָה. מה אם המשתמש רוצה לשנות את גודל הנקודות? בואו נשנה את התוכנית כך שהמשתמש, לא רק המתכנת, יוכל לשנות את גודל הנקודה. נתכנת את הכפתור שכותרתו "נקודות גדולות" לשנות את גודל הנקודה ל-8, ונתכנת את הכפתור שכותרתו "נקודות קטנות" כדי להתאים את הגודל ל-2.
כדי להשתמש בערכים שונים עבור ארגומנט הרדיוס, האפליקציה צריכה לדעת איזה מהם אנחנו רוצים ליישם. אנחנו צריכים להורות לו להשתמש בערך ספציפי, והוא צריך לאחסן (או לזכור) את הערך הזה איכשהו כדי שהוא יוכל להמשיך להשתמש בו. כאשר האפליקציה שלך צריכה לזכור משהו שאינו מאפיין, אתה יכול להגדיר משתנה. משתנה הוא תא זיכרון; אתה יכול לחשוב על זה כמו דלי שבו אתה יכול לאחסן נתונים שיכולים להשתנות,

שבמקרה זה הוא גודל הנקודה הנוכחית (למידע נוסף על משתנים, ראה פרק 16).



איור 2-17. אפליקציית PaintPot עם תמונה "מוערת" של פרופסור וולבר

נתחיל בהגדרת משתנה בשם dotSize:

1. בעורך הבלוקים, ממגירת המשתנים של הבלוקים המובנים, גרור החוצה שם גלובלי לאתחל לחסימה. בתוך בלוק האתחול, שנה את הטקסט "שם" ל-"eziStod".

2. שים לב שלאתחול dotSize גלובלי לחסום יש שקע פתוח. זה המקום שבו אתה יכול לציין את הערך הראשוני עבור המשתנה, או את הערך שאליו הוא מוגדר כברירת מחדל כאשר האפליקציה מתחילה. (זה מכונה לעתים קרובות "אתחול משתנה" במונחי תכנות.) עבור אפליקציה זו, אתחול ה- dotSize ל-2 על ידי יצירת בלוק מספר 2 (השתמש בתכונת חסימת הסוג: הקלד "2" בעורך הבלוקים ולאחר מכן הקש Return) ולאחר מכן חבר אותו לאתחל dotSize to, כפי שמוצג באיור 2-17.

איור 2-18. אתחול המשתנה dotSize בערך של 2

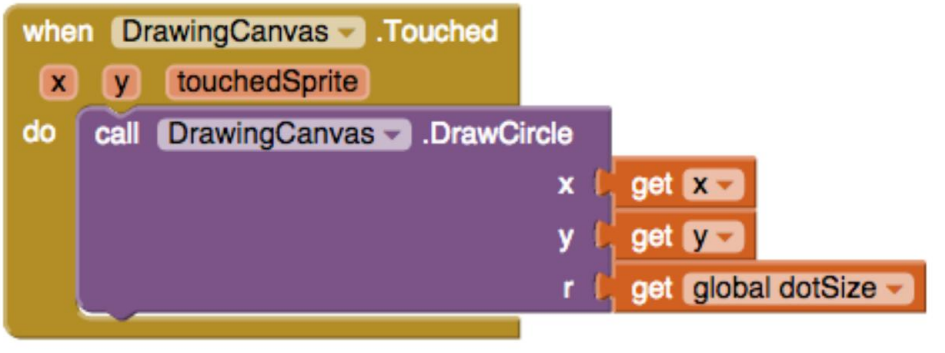
התייחסות למשתנה ה-EZISTOD במעגל השרטוט

לאחר מכן, אנו רוצים לשנות את הארגומנט של DrawingCanvas.DrawCircle במטפל האירוע DrawingCanvas.Touched כך שישתמש בערך של dotSize ולא

תמיד באמצעות מספר קבוע. (זה עשוי להיראות כאילו "שיפצנו" את dotSizeלערך 2במקום להפוך אותו למשתנה כי אתחלנו אותו כך, אבל תראה תוך דקה איך אנחנו יכולים לשנות את הערך של , dotSizeולכן, לשנות את גודל הנקודה שמצוירת).

1.גרור החוצה בלוק קבל מגודל הנקודות הגלובלי לאתחל לחסימה. אתה אמור לראות בלוק get global dotSizeשמספק את הערך של המשתנה.

2.עבור אל הבלוק , DrawingCanvas.DrawCircleגרור את הבלוק מספר 5אל מחוץ לחריץ rולאחר מכן הנח אותו לפח. לאחר מכן, החלף אותו בגוש get global dotSize (ראה איור .(18-2כאשר המשתמש נוגע ב-savnaCgniwarD, האפליקציה תקבע כעת את הרדיוס מהמשתנה .dotSize



איור .19-2כעת הגודל של כל עיגול תלוי במה שמאוחסן במשתנה dotSize

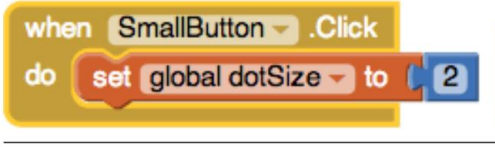
שינוי הערך של DOTSIZE

האפליקציה שלך תצייר כעת עיגולים בגודל לפי הערך במשתנה ,dotSizeאבל אתה עדיין צריך קוד כדי ש- dotSizeישתנה (כרגע הוא יישאר כ-2) לפי מה שהמשתמש בוחר. תוכל ליישם התנהגות זו על ידי תכנות המטפלים באירוע SmallButton.Click ו- : BigButton.Click

1.גרור החוצה מטפל באירוע SmallButton.Click מהמגירה של .SmallButton לאחר מכן, העבר את העכבר מעל "גודל הנקודות" בתוך הבלוק הגלובלי האתחול וגרור החוצה את ה- dotSizeהגלובלי המוגדר לחסימה. חבר אותו ל- SmallButton.Click.לבסוף, צור בלוק מספר 2וחבר אותו ל- dotSizeהגלובלי המוגדר כדי לחסום.

2.צור מטפל אירועים דומה עבור ,BigButton.Clickאך הגדר את dotSizeל-8. שני מטפלי האירועים אמורים להופיע כעת בעורך הבלוקים, כפי שמוצג באיור .19-2

הערה ה"גלובל" set global dotSize ב- מתייחס לעובדה שניתן להשתמש במשתנה בכל מטפלי האירועים של התוכנית (גלובלית). ב-Inventor, ppA אתה יכול גם להגדיר משתנים שהם "מקומיים" לחלק מסוים של התוכנית (ראה פרק 21לפרטים).

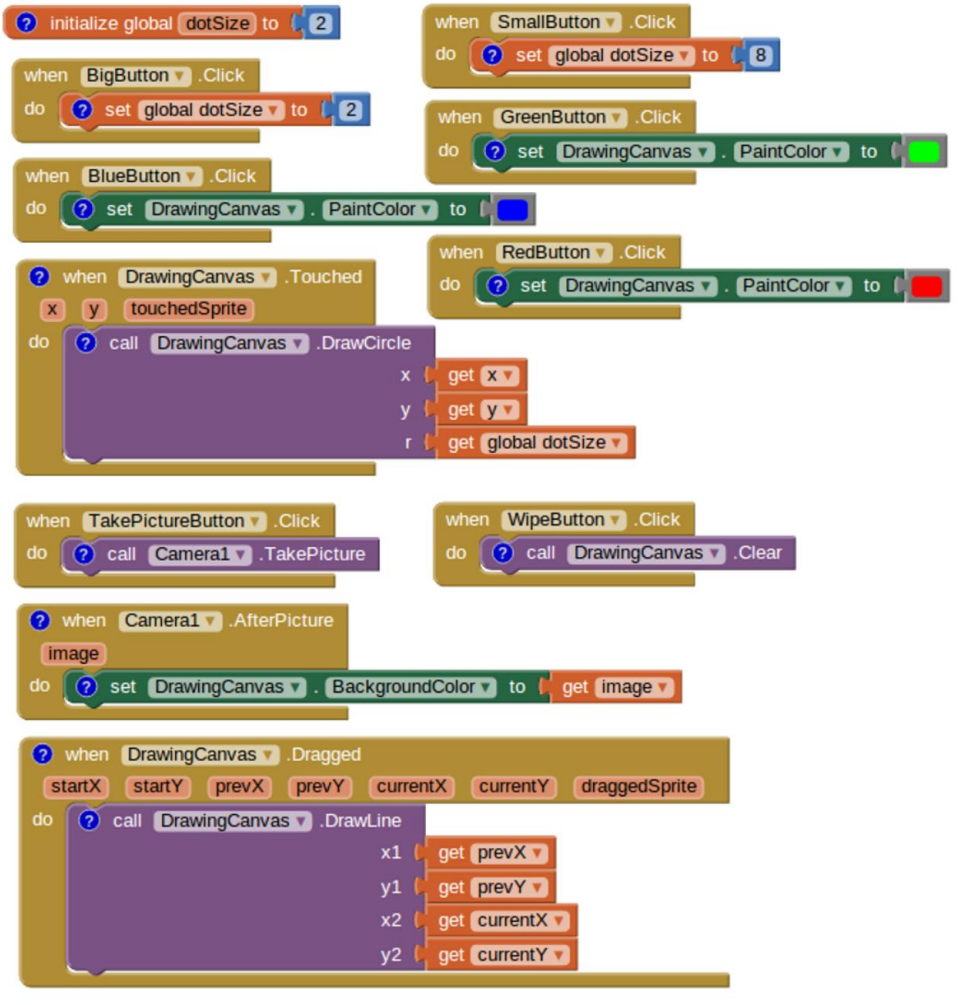


איור 2-20. לחיצה על הכפתורים משנה את גודל הנקודה; נגיעות לאחר מכן יציירו בגודל זה

בדוק את האפליקציה שלך נסה ללחוץ על לחצני הגודל ולאחר מכן לגעת ב-savnaCgniwarD. האם העיגולים מצוירים בגדלים שונים? האם הקווים? גודל הקו לא אמור להשתנות מכיוון שתכנת את dotSize לשימוש רק בבלוק DrawingCanvas.DrawCircle. בהתבסס על זה, האם אתה יכול לחשוב כיצד תוכל לשנות את הבלוקים שלך כך שמשתמשים יוכלו לשנות גם את גודל הקו? (רמז: ל- DrawingCanvas יש מאפיין בשם LineWidth.)

האפליקציה השלמה: PaintPot

איור 2-20 ממחיש את אפליקציית PaintPot שהושלמה.



איור 21-2. ערכת הבלוקים הסופית עבור PaintPot

וריאציות

הנה כמה וריאציות שתוכל לחקור:

•ממשק המשתמש של האפליקציה אינו מספק מידע רב על ההגדרות הנוכחיות (לדוגמה, הדרך היחידה לדעת את גודל הנקודה או הצבע הנוכחיים היא לצייר משהו). שנה את האפליקציה כך שהגדרות אלה יוצגו ב-

מִשְׁתַמֵשׁ.

•תן למשתמש לציין ערכים שאינם 2 ו-8 עבור גודל הנקודה באמצעות סליידר רְכִיב.

סיכום

הנה כמה מהרעיונות שכיסינו בפרק זה:

•רכיב ה-savnaC מאפשר לך לצייר עליו. הוא יכול גם לחוש נגיעות וגרירות, ואתה יכול למפות אירועים אלה לפונקציות ציור.

•אתה יכול להשתמש ברכיבי סידור מסך כדי לארגן את פריסת הרכיבים במקום פשוט למקם אותם אחד מתחת לשני.

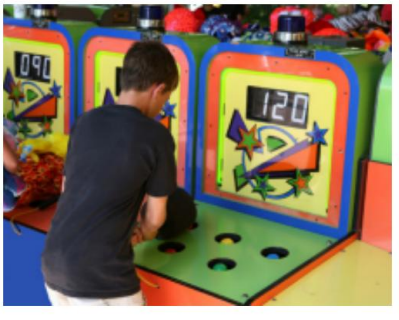
•מטפלי אירועים מסוימים מגיעים עם מידע על האירוע, כגון
קואורדינטות של מקום נגיעה במסך. מידע זה מיוצג על ידי טיעונים. כאשר אתה גורר מטפל באירועים שיש לו ארגומנטים, App Inventorיוצר פריטים קבל וקבע בתוך הבלוק לשימוש כדי להתייחס אליהם
טיעונים.

•אתה יוצר משתנים באמצעות אתחול שם גלובלי לבלוקים מה-
מגירת משתנים. משתנים מאפשרים לאפליקציה לזכור מידע, כגון גודל הנקודה, שאינו מאוחסן במאפיין רכיב.

•עבור כל משתנה שאתה מגדיר, App Inventorמספק אוטומטית הפניה גלובלית של get המעניקה את הערך של המשתנה, ובלוק גלובלי מוגדר לשינוי הערך של המשתנה. כדי לגשת לאלה, העבר את העכבר מעל שם המשתנה בבלוק האתחול שלו.

פרק זה הראה כיצד ניתן להשתמש ברכיב Canvasעבור תוכנית ציור. אתה יכול גם להשתמש בו כדי לתכנת אנימציות, כמו אלה שתמצא במשחקי דו-ממד. למידע נוסף, עיין במשחק MoleMashבפרק ,3במשחק Ladybug Chaseבפרק ,5ובדיון על אנימציה בפרק 17.

MoleMash



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

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

מה אתה תבנה

איור 3-1. ממשק המשתמש של MoleMash

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

- שומה צצה במקומות אקראיים על המסך, זה פעם בשנייה.
- הקשה על השומה גורמת למכשיר לרטוט, להגדיל תצוגה של "מכות" (להגדיל אותה באחת), ולהזיז את השומה מיד למקום חדש.
- 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 ImageSpriteעבור תמונות נעות רגישות למגע.

•רכיב ה- , Canvasשפועל כמשטח עליו מניחים את ImageSprite.

•רכיב השעון להזיז את הספרייט פעם בשנייה.

•רכיב הצליל 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 (+)וחיסור .(–)

מתחילים

התחבר לאתר App Inventorוהתחל 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.בקטע מדיה, לחץ על "העלה קובץ", דפדף למקום Component Designer,-ב שבו נמצא ה-elf במחשב שלך ולאחר מכן העלה אותו ל-App Inventor.

עיצוב הרכיבים

אתה תשתמש ברכיבים האלה כדי ליצור :MoleMash

•קנבס המשמש כמגרש משחקים.

•שמציג תמונה של שומה ויכול להסתובב ו ImageSprite לחוש כשנוגעים בשומה.

•צליל רוטט כשנוגעים בשומה.

•תוויות המציגות "היטים: ,""החמצות: ,"והמספרים בפועל של כניסות ו מתגעגע.

•הסדרים אופקיים למיקום נכון של התוויות.

•כפתור כדי לאפס את מספר הכניסות והחמצות ל-0.

•שעון כדי לגרום לשומה לזוז פעם בשנייה.

טבלה 3-1מציגה את כל הרכיבים שבהם תשתמש.

טבלה 3-1. הרשימה המלאה של הרכיבים עבור MoleMash

מַטָרָה	קבוצת צבעים איך תקרא לזה		סוג רכיב
המיכל עבור .ImageSprite	בד 1	ציור ו אנימציה	בַּד
המשתמש ינסה לגעת בזה.	חֲפַרפֶּרֶת	ציור ו אנימציה	ImageSprite
המשתמש ילחץ על זה כדי לאפס את ציון.	כפתור איפוס ממשק משתמש		לַחְצָן
לשלוט בתנועת השומה.	שעון ממשק משתמש 1		שָׁעוֹן
רטט כשנוגעים בשומה.	כְּלֵי תִקְשׁוֹרֶת צליל 1		נשמע
הצג את "היטים: ."	ממשק משתמש HitsLabel		תווית
הצג את מספר הכניסות.	ממשק משתמש HitsCountLabel		תווית
מקם את HitsLabelליד HitsCountLabel.	אופקי פריסת סידור אופקי סידור 1		
הצג את "החמצות:".	ממשק משתמש MissesLabel		תווית
הצג את מספר ההחמצות.	ממשק משתמש MissesCountLabel		תווית
מקם MissesLabelליד 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.

.1מהמגירה של ציור והנפשה, גרור פנימה רכיב Canvasועזוב
זה עם שם ברירת המחדל .1Canvasהגדר את המאפיין Widthשלו ל-"lliF" "parentכך שהוא
הוא רחב כמו המסך, והגדר את הגובה שלו ל-003 פיקסלים.

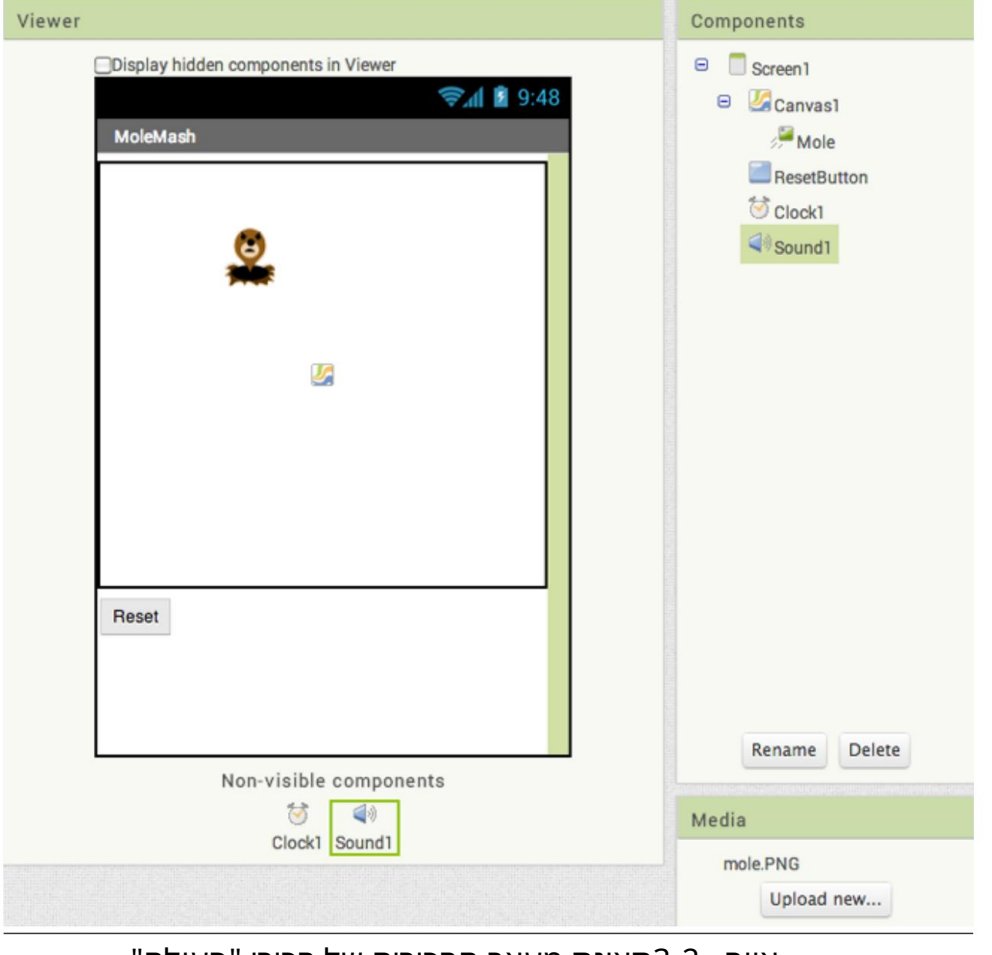
.2שוב מהמגירה של ציור והנפשה, גרור ב- ImageSprite
רכיב, הצבתו בכל מקום על .1Canvasבתחתית הרכיבים
ברשימה, לחץ על שנה שם ושנה את שמו ל"שומה". הגדר את מאפיין התמונה שלו ל
,mole.PNGשהעלית קודם לכן.

.3מהמגירה של ממשק המשתמש, גרור פנימה רכיב , Buttonהצב אותו
מתחת ל-1savnaC. שנה את שמו של "ResetButton"והגדר את מאפיין הטקסט שלו ל-"teseR".

4.גם מהמגירה של ממשק המשתמש, גרור פנימה רכיב שעון . הוא יופיע בתחתית ה-reweiV בקטע "רכיבים שאינם נראים".

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

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



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

הצבת רכיבי התווית

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

1. ממגירת ה-Layout, גרור פנימה HorizontalArrangement והצב אותו מתחת ללחצן ושמור על שם ברירת המחדל של HorizontalArrangement1.

2. מהמגירה של ממשק המשתמש, גרור שתי תוויות לתוך HorizontalArrangement1. סידור אופקי

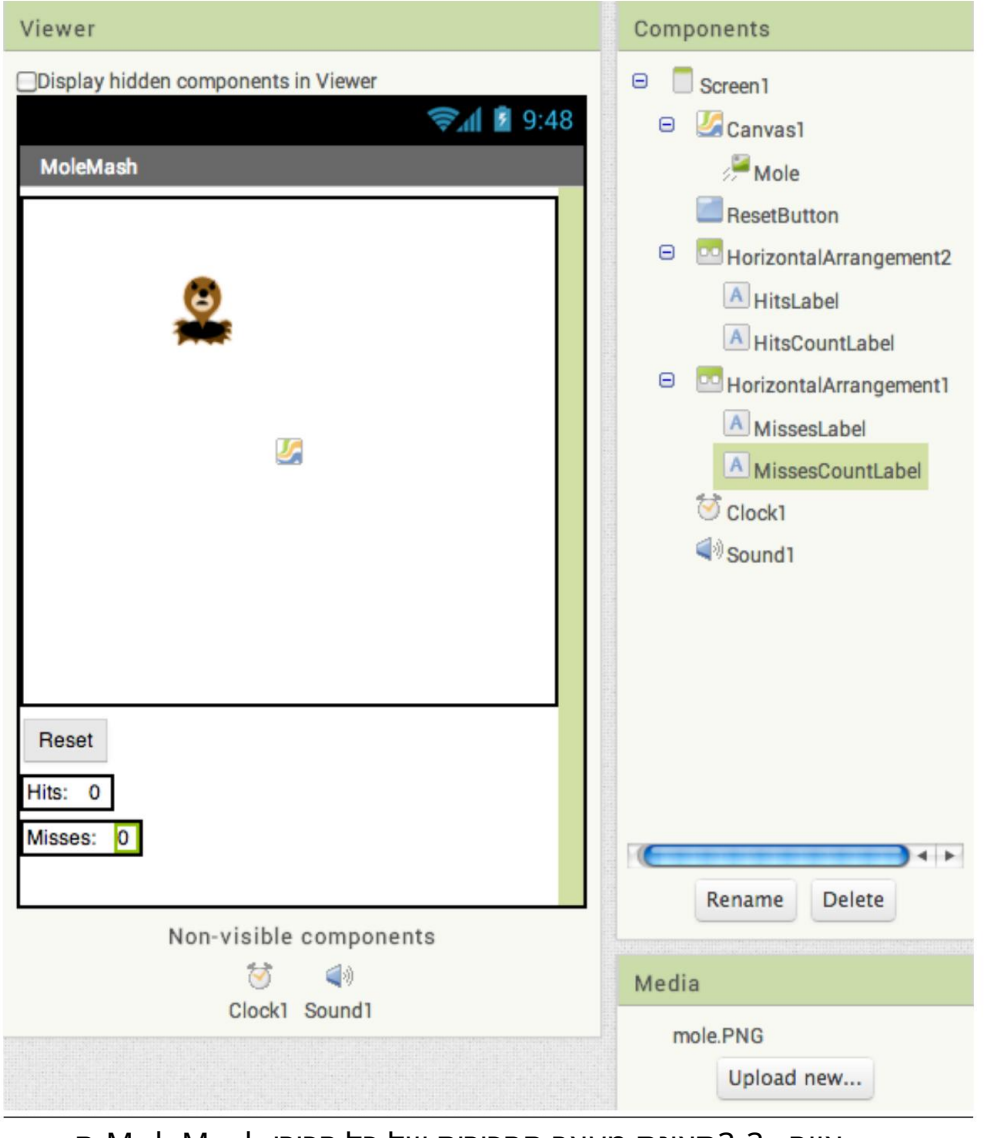
- שנה את שם התווית השמאלית "HitsLabel" והגדר את מאפיין ה-Text שלה ל-"Hits:" (הקפד לכלול רווח אחרי הנקודתיים).
- שנה את שם התווית הימנית "HitsCountLabel" והגדר את המאפיין Text שלה ל-0.

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 ל-"Misses: " (הקפד לכלול רווח אחרי הקולון).
- שנה את שם התווית הימנית "MissesCountLabel" והגדר את המאפיין Text שלה ל-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 מחטיא את השומה. (הערה: אנו ממליצים להשתמש ב-regnf שלך, 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 Inventorהיה הליך שמעביר ImageSpriteלמיקום 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- MoveMoleבתחילת המשחק, כשהמשתמש מקיש בהצלחה על השומה, ופעם בשנייה.

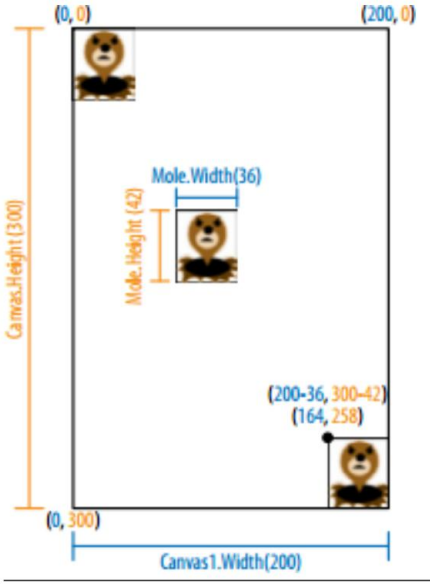
יצירת הליך MOVEMOLE

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

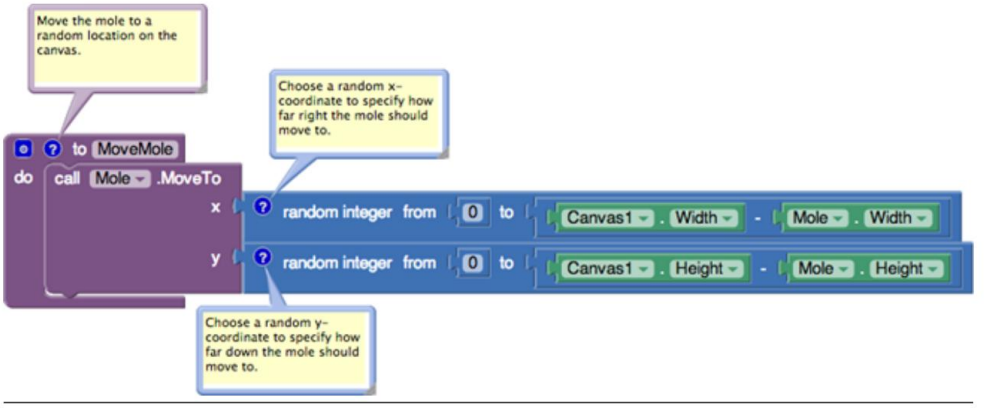
ניתן לחשוב על הקנבס (והמסך) כעל רשת עם קואורדינטות x(אופקיות) ו-y (אנכיות), כאשר הקואורדינטות (x, y)של הפינה השמאלית העליונה הן .(0, 0)
קואורדינטת ה-x גדלה ככל שאתה זז ימינה, וקואורדינטת ה-y גדלה ככל שאתה זז למטה, כפי שמוצג באיור .3-4המאפיינים x-וy של ImageSpriteמציינים היכן ממוקמת הפינה השמאלית העליונה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

כדי לקבוע את ערכי ה-x וה-y המקסימליים הזמינים כך ש- Mole ftsעל המסך, עלינו לעשות שימוש במאפייני הרוחב והגובה של שומה וקנבס .1(מאפייני הרוחב והגובה של השומה זהים לגודל התמונה שהעלית. כשיצרת את ,Canvas1אתה מגדיר את הגובה שלו ל 300-פיקסלים ואת הרוחב שלו ל-"lliF", "parentמה שמעתיק את הרוחב של אלמנט האב שלו, אשר במקרה זה הוא המסך.) אם השומה ברוחב 36פיקסלים והבד הוא ברוחב 200 פיקסלים, קואורדינטת ה-x של הצד השמאלי של השומה יכולה להיות נמוכה כמו 0(כל הדרך שמאלה) או גבוהה כ-461 ,36 - 200)או (Canvas1.Width - Mole.Widthללא השומה המתרחבת בקצה הימני של המסך. באופן דומה, קואורדינטת ה-y של החלק העליון של השומה יכולה לנוע בין 0ל- .Canvas1.Height – Mole.Height

איור 3-5מציג את ההליך שתיצור, עם הערות תיאוריות
הערות (שאותן תוכל להוסיף להליך שלך באופן אופציונלי).
כדי למקם את השומה באופן אקראי, נרצה לבחור קואורדינטת xבטווח שבין .Mole.Width – Canvas1.Width-ל 0באופן דומה, נרצה שקואורדינטת ה-y תהיה בטווח שבין .Mole.Height – Canvas1.Height-ל 0אנו יכולים ליצור מספר אקראי באמצעות הפרוצדורה המובנית של מספר שלם אקראי, אותו תוכל למצוא במגירה של .Mathיהיה עליך לשנות את פרמטר ברירת המחדל "מאת" מ-1 ל-0 ולהחליף את הפרמטרים "אל", כפי שמוצג באיור 3-5.



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



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

כדי ליצור את ההליך:

1. בעורך הבלוקים, לחץ על מגירת ההליך.

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.MoveToלהליך, 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 0ל-
Canvas1.Width – Mole.Width, כפי שנדון קודם לכן, בצע את הפעולות הבאות:

- ממגירת המתמטיקה, גרור פנימה את המספר השלם האקראי מהגוש, והכנס את התקע (בליטה) בצדו השמאלי לתוך שקע "x"בעת קריאה 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 Delאו Delete של המקלדת שלך, או על ידי גרירתו לפח האשפה.
- ממגירת המתמטיקה, גרור פנימה בלוק חיסור (–)והנח אותו לתוך שקע "אל".
- מהמגירה Canvas1, בחר את בלוק Canvas1.Widthוגרור אותו אל הצד השמאלי של פעולות החיסור.
- באופן דומה, לחץ על מגירת החפרפרת וגרור את ה- Mole.Widthאל סביבת העבודה. לאחר מכן, חבר את זה לצד הימני של בלוק החיסור.

6.בצע הליך דומה כדי לציין שקואורדינטת ה-y צריכה להיות אקראית
מספר שלם בטווח שבין 0ל- Canvas1.Height – Mole.Height.

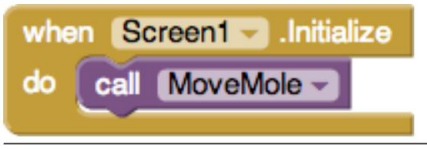
7.בדוק את התוצאות שלך מול איור 3-5.

מתקשר ל-ELOMEVOM כאשר האפליקציה מתחילה

כעת לאחר שכתבת את הליך , MoveMoleבואו נשתמש בו. בגלל שזה כל כך נפוץ שמתכנתים רוצים שמשהו יקרה כשאפליקציה מתחילה, יש חסימה בדיוק למטרה הזו: .Screen1.Initialize

1.לחץ על מגירת מסך1 וגרור החוצה את מסך1. אתחול.

2.לחץ על מגירת ההליכים, שבה תראה בלוק MoveMoleשיחה . (שֶׂלָה
די מגניב שיצרת בלוק חדש, לא?!) גרור אותו החוצה והנח אותו במסך .1אתחול, כפי
שמוצג באיור 3-6.

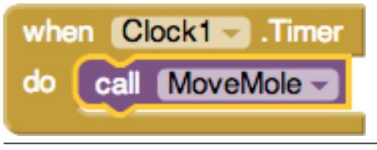


איור 3-6. קורא לנוהל MoveMole כאשר היישום מתחיל

מתקשר ל-ELOMEVOM כל שנייה

העברת השומה כל שנייה תדרוש את רכיב השעון . השארנו את המאפיין TimerInterval 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, כפי שמוצג באיור 3-7.

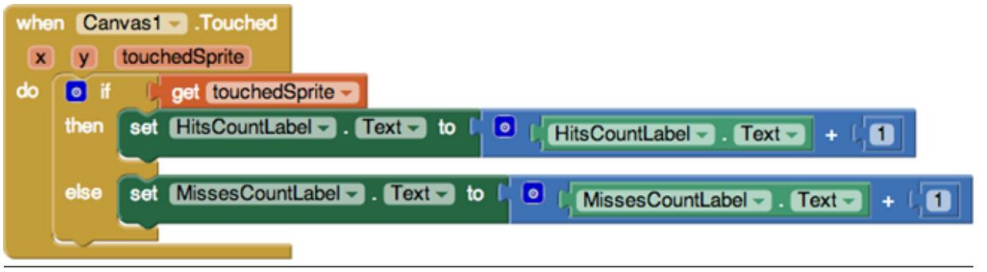


איור 3-7. קריאה להליך MoveMole כאשר הטיימר כבה (כל שנייה)

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

שמירה על ציון

כזכור, יצרת שתי תוויות, HitsCountsLabel ו-MissesCountsLabel, 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 בגוש 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 3-8 מציג את הקוד שתיצור.



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

אתה יכול לתרגם את הבלוקים באיור 8-3בצורה הבאה: בכל פעם שהקנבס מוקשה, בדוק אם הקיש ספרייט. מכיוון שיש רק ספרייט אחד בתוכנית שלנו, זה חייב להיות .Mole1אם הקישו על , Mole1הוסף אחד למספר ב-HitsCountLabel.Text; אחרת, הוסף אחד ל- .MissesCountLabel.Text(הערך של touchedSpriteהוא שקר אם לא נגעו בספרייט.)

כך יוצרים את הבלוקים:

1.לחץ על מגירת Canvas1וגרור החוצה את 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, ולאחר מכן גרור החוצה את הבלוק get touchedSpriteוהנח אותו בשקע הבדיקה של if-then אַחֵר.

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" שֶׁקַע.
- לחץ על מגירת HitsCountLabelוגרור את בלוק HitsCountLabel.Text משמאל לסימן הפלוס.
- לחץ על מגירת המתמטיקה וגרור בלוק 0מימין לסימן הפלוס. לחץ על 0ושנה אותו ל-1.

5.חזור על שלב 4עבור MissesCountLabelבקטע elseשל בלוק . ifelse

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

אבסטרקציה פרוצדורלית

היכולת לתת שם ולקרוא מאוחר יותר לקבוצת הוראות כמו MoveMoleהיא אחד הכלים המרכזיים במדעי המחשב ומכונה הפשטה פרוצדורלית. זה נקרא "הפשטה" מכיוון שהקורא של ההליך (שבפרויקטים בעולם האמיתי, כנראה שונה ממחבר ההליך) רק צריך לדעת מה ההליך עושה (מזיז את השומה), לא איך הוא 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 מכונית, אף אחד מהם לא מבין את כל הפרטים, והנהג צריך רק להבין את הממשק (למשל, לחיצה על דוושת בלם כדי לעצור את המכונית), לא היישום.

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 ב-ppA (.Inventor

•שבירת קוד לחתיכות עוזרת לך לחשוב על האפליקציה וליישם אותה ("הפרד ומשול").

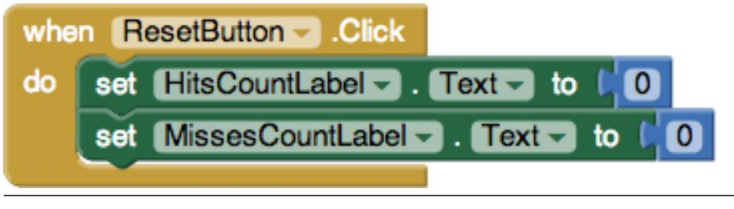
•בחירת שמות טובים לפרוצדורות עוזרת לתעד את הקוד, ומקלה על מישהו אחר (או לך, חודש לאחר מכן) לקרוא.

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 שקובע את הערכים של
HitsCountLabel.Text ו- MissesCountLabel.Text to 0. צור את הבלוקים המוצגים ב
איור 3-9.



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

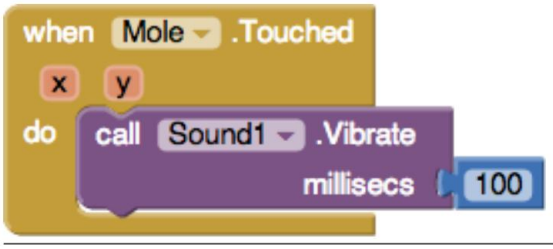
בשלב זה, סביר להניח שאינך זקוק להוראות שלב אחר שלב ליצירת מטפל באירוע לחיצת כפתור עם תוויות טקסט, אבל הנה טיפ שיעזור להאיץ את התהליך: במקום לקבל את המספר שלך ממגירת המתמטיקה, פשוט הקלד 0, ואת

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.Vibrateכפי שמוצג באיור 3-10.

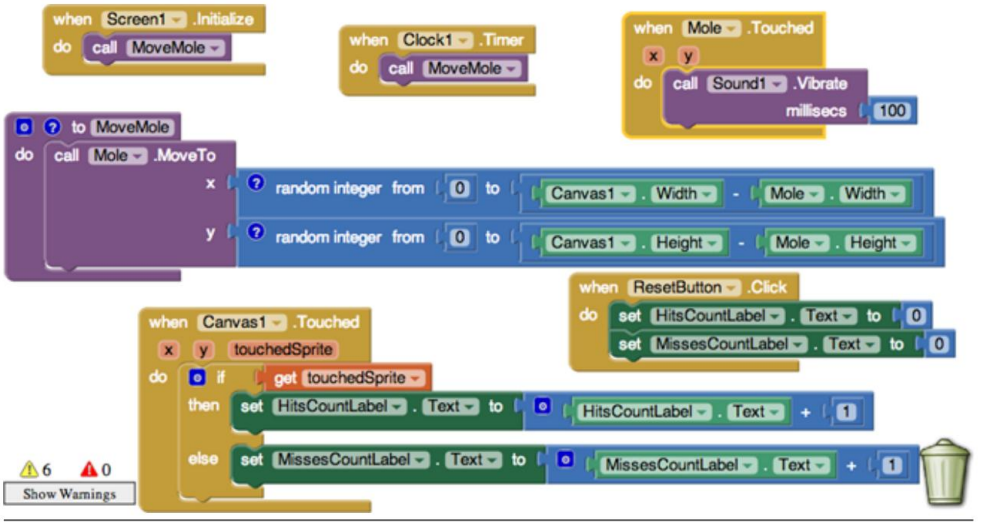


איור 3-10. גורם למכשיר לרטוט קצר (למשך 100מילישניות) כשנוגעים בשומה

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 MoleMashהשלמה.



איור 3-11. אפליקציית MoleMash המלאה

וריאציות

הנה כמה רעיונות לתוספות ל-MoleMash:

•הוסף לחצנים 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, הענש אותו על ידי הפחתת הניקוד שלו או סיום המשחק.

•במקום להשתמש בתמונה של שומה, תן למשתמש לבחור תמונה עם ה-רכיב ImagePicker.

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 בשמאל ל-Canvas.Width–1 מימין), ו-y את 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- Canvasו
שיטות Touchedשל רכיבי . ImageSprite

•ניתן ליצור אפליקציות בזמן אמת המגיבות לא רק לקלט המשתמש אלא גם בתגובה לטיימר הפנימי של המכשיר. באופן ספציפי, בלוק Clock.Timerפועל בתדירות המצוינת במאפיין Clock.Intervalוניתן 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.Vibrateשגורמת 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.

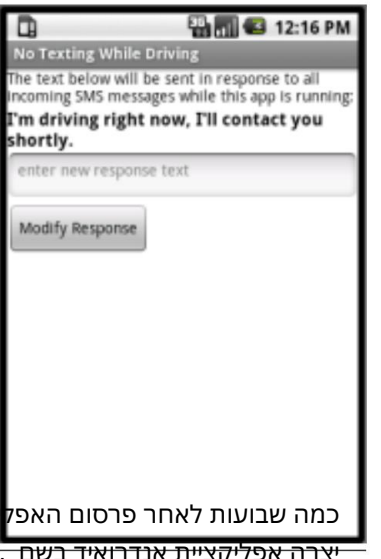
•אתה יכול ליצור התנהגות בלתי צפויה עם בלוק השלם האקראי (ב- מגירת מתמטיקה), עושה משחק שונה בכל פעם שהוא משוחק.

תלמדו טכניקות נוספות למשחקים, כולל זיהוי התנגשויות ביניהם הזזת רכיבי , ImageSpriteבפרק 5.

אין לשלוח הודעות טקסט בזמן נהיגה

פרק זה ילווה אותך דרך היצירה של ללא הודעות טקסט בזמן נהיגה, אפליקציית "משיבון טקסט" המגיבה אוטומטית להודעות טקסט שאתה מקבל בזמן נהיגה (או במשרד וכו'), מדברת הודעות טקסט בקול רם, ואפילו שולחת מידע מיקום כחלק מה- תשובת טקסט אוטומטית. האפליקציה מדגימה כיצד ניתן לשלוט בכמה מהתכונות הנהדרות של טלפון אנדרואיד, כולל הודעות ,SMSטקסט לדיבור, נתונים מתמשכים וחישת מיקום .GPS

בינואר ,2010המועצה הלאומית לבטיחות של ארצות הברית (NSC)הכריזה על תוצאות מחקר שמצא שלפחות 28אחוז מכל תאונות הדרכים -קרוב ל-6.1 מיליון תאונות בכל שנה -נגרמות על ידי נהגים המשתמשים בטלפונים סלולריים, ולפחות 200,000מהתאונות הללו התרחשו בזמן שנהגים שלחו הודעות טקסט.1 כתוצאה מכך, מדינות רבות אסרו על נהגים להשתמש בטלפונים סלולריים לחלוטין.



איור 4-1. האפליקציה ללא הודעות טקסט בזמן נהיגה

דניאל פינגן, סטודנט באוניברסיטת סן פרנסיסקו שלמד בשיעור תכנות App Inventor, הגה רעיון לאפליקציה כדי לעזור במגפת הנהיגה וההודעות הטקסט. האפליקציה שהוא יצר, שמוצגת באיור 4-1, מגיבה אוטומטית (ודיבורית) לכל טקסט עם הודעה כגון "אני נוהג עכשיו, אצור איתך קשר בקרוב."

מאוחר יותר האפליקציה הורחבה כך שתעשה זאת דברו את הטקסטים הנכנסים בקול רם והוסיפו את מיקום ה-SPG של הנהג לטקסט התגובה האוטומטית, וזה הפך למדריך עבור האפליקציה
אתר ממציאים.

כמה שבועות לאחר פרסום האפליקציה באתר App Inventor, State Farm Insurance יצרה אפליקציית אנדרואיד בשם On the Move, בעלת פונקציונליות דומה ל- While Driving. No Texting

אנחנו לא יודעים אם האפליקציה של דניאל או ההדרכה באתר App Inventor השפיעו על On the Move, אבל מעניין לשקול את האפשרות שאולי יש לאפליקציה שנוצרה בקורס תכנות מתחיל (על ידי סטודנט לכתיבה יוצרת, לא פחות!) היווה השראה לתוכנה בייצור המוני הזה, או לפחות תרם למערכת האקולוגית שהביאה אותה. זה בהחלט הדגים כיצד App Inventor הוריד את מחסום הכניסה כך שכל אחד עם רעיון טוב יוכל להפוך את הרעיון שלו במהירות ובזול לאפליקציה מוחשית ואינטראקטיבית. קלייב תומפסון מהמגזין Wired קלט את החידוש וכתב את זה:

> תוכנה, אחרי הכל, משפיעה כמעט על כל מה שאנחנו עושים. בחר כל בעיה מרכזית - התחממות כדור הארץ, שירותי בריאות, או, במקרה של Finnegan, בטיחות בכבישים מהירים - ותוכנה חכמה היא חלק מהפתרון. עם זאת, רק חלק קטנטן של אנשים שוקל אי פעם ללמוד לכתוב קוד, מה שאומר שאנחנו לא מקישים על היצירתיות של חלק גדול מהחברה.[2] App Inventor עוסק בלחיצה על היצירתיות שמזכיר תומפסון, על פתיחת עולם יצירת התוכנה ל כל אחד.

מה תלמד

זוהי אפליקציה מורכבת יותר מאלו שבפרקים הקודמים, כך שתבנה אותה חתיכת פונקציונליות אחת בכל פעם, החל בהודעת התגובה האוטומטית. תלמד על:

2 קלייב תומפסון, "קלייב תומפסון על קידוד להמונים", http://wrd.cm/1uT25O5

- רכיב הטקסט לשליחת טקסטים ועיבוד טקסטים שהתקבלו.
- טופס קלט לשליחת הודעת התגובה המותאמת אישית.
- רכיב מסד הנתונים TinyDB לשמירת ההודעה המותאמת גם לאחר סגירת האפליקציה.

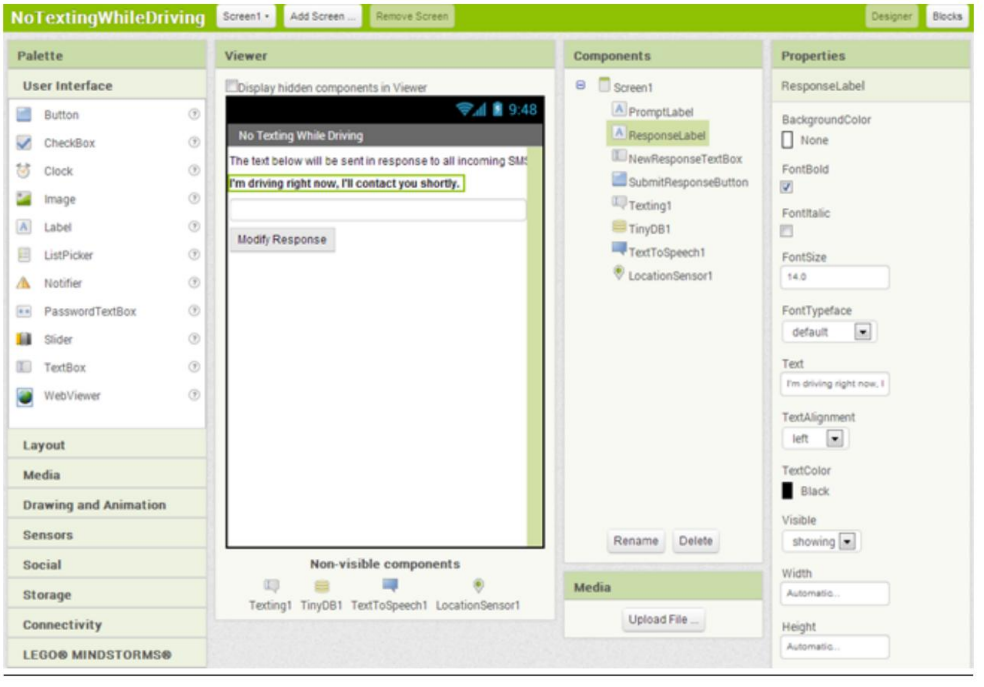
- האירוע Screen.Initialize לטעינת התגובה המותאמת אישית כאשר האפליקציה השקות.
- רכיב TextToSpeech להקראת טקסטים בקול רם.
- רכיב ה- LocationSensor לדיווח על מיקומו הנוכחי של הנהג.

מתחילים

פתח את הדפדפן שלך לאתר App Inventor והתחל פרויקט חדש. תן לזה שם "gnivirDelihWgnitxeToN" (זכור, שמות פרויקטים לא יכולים לכלול רווחים) והגדר את כותרת המסך ל"אין לשלוח הודעות טקסט בזמן נהיגה". לאחר מכן, לחץ על התחבר והגדר בדיקה חיה במכשיר שלך או באמולטור.

עיצוב הרכיבים

ממשק המשתמש של האפליקציה פשוט יחסית: יש לו תווית המורה למשתמש כיצד האפליקציה פועלת, תווית המציגה את הטקסט שאמור להישלח אוטומטית בתגובה לטקסטים נכנסים, תיבת טקסט לשינוי התגובה ו- כפתור להגשת השינוי. תצטרך גם לגרור פנימה רכיב טקסט , רכיב , TinyDB רכיב TextToSpeech ורכיב חיישן מיקום , כולם יופיעו באזור "רכיבים שאינם נראים". אתה יכול לראות איך זה אמור להיראות בתמונת המצב של מעצב הרכיבים באיור 4-2.



איור 4-2. האפליקציה ללא הודעות טקסט בזמן נהיגה ב-Component Designer

אתה יכול לבנות את ממשק המשתמש המוצג באיור 4-2 על ידי גרירת ה-רכיבים המפורטים בטבלה 4-1.

טבלה 4-1. כל הרכיבים לאפליקציית No Texting

מַטָרָה	קבוצת צבעים איך תקרא לזה	סוג רכיב
תן למשתמש לדעת כיצד האפליקציה פועלת.	ממשק משתמש PromptLabel	תווית
התגובה שתישלח בחזרה ל שׁוֹלֵחַ.	ממשק משתמש ResponseLabel	תווית
ממשק משתמש newResponseTextBox המשתמש יזין כאן את התגובה המותאמת אישית.		תיבת טקסט
ממשק משתמש SubmitResponseButton המשתמש לוחץ על זה כדי לשלוח תגובה.		לַחְצָן
עבדו את הטקסטים.	שליחת הודעות טקסט 1	חֶבְרָתִי	הודעות טקסט
אחסן את התגובה במסד הנתונים.	TinyDB1	אִחסוֹן	TinyDB
דבר את הטקסט בקול רם.	TextToSpeech1	מדיה	TextToSpeech
חוש היכן המכשיר נמצא.	חיישן מיקום 1	חיישני חיישן מיקום

הגדר את המאפיינים של הרכיבים בצורה הבאה:

•הגדר את הטקסט של PromptLabelל"הטקסט למטה יישלח כתגובה לכולם הודעות SMSהתקבלו בזמן שהאפליקציה הזו פועלת."

•הגדר את הטקסט של ResponseLabelל"אני נוהג עכשיו, אצור איתך קשר בְּקָרוֹב." בדוק את תכונת ההעזה שלו.

עבור הגדר את הטקסט של NewResponseTextboxל- ." "(זה משאיר את תיבת הטקסט ריקה קלט המשתמש.)

•הגדר את הרמז של NewResponseTextboxל"הזן טקסט תגובה חדש".

•הגדר את הטקסט של SubmitResponseButtonל"שנה תגובה".

הוספת התנהגויות לרכיבים

תתחיל בתכנות התנהגות התגובה האוטומטית שבה תשובת טקסט נשלחת לכל טקסט נכנס. לאחר מכן תוסיף בלוקים כדי שהמשתמש יוכל לציין תגובה מותאמת אישית ולשמור את התגובה הזו בהתמדה. לבסוף, תוסיף בלוקים שקוראים בקול את הטקסטים הנכנסים ותוסיף מידע מיקום לטקסטים של התגובה האוטומטית.

תגובה אוטומטית לטקסט

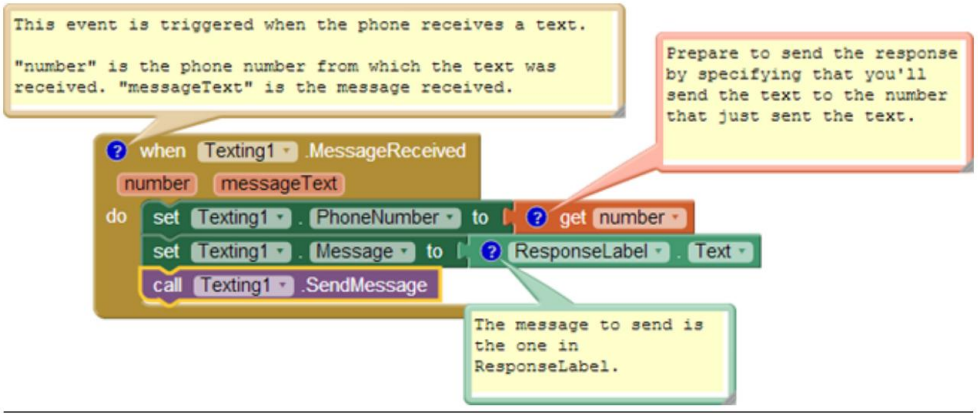
עבור התנהגות התגובה האוטומטית, תשתמש ברכיב הטקסט של .App Inventorאתה יכול לחשוב על הרכיב הזה כעל אדם קטן בתוך הטלפון שלך שיודע לקרוא ולכתוב טקסטים. לקריאת טקסטים, הרכיב מספק בלוק אירוע .Texting.MessageReceivedאתה יכול לגרור את הבלוק הזה החוצה ולהציב בתוכו בלוקים כדי להראות מה צריך לקרות כשמתקבל טקסט. במקרה של אפליקציה זו, אנו רוצים לשלוח בחזרה הודעת טקסט באופן אוטומטי בתגובה.

אתה יכול לשלוח הודעת טקסט עם שלושה בלוקים. ראשית, אתה מגדיר את מספר הטלפון שאליו יש לשלוח את הטקסט, שהוא מאפיין של רכיב .Texting1לאחר מכן, אתה מגדיר את ההודעה להישלח, גם היא מאפיין של .SMS1לבסוף, אתה למעשה שולח את הטקסט עם בלוק .Texting1.SendMessage טבלה 4-2מפרטת את כל הבלוקים שתזדקק להתנהגות התגובה האוטומטית הזו, ואיור 4-3מראה כיצד הם צריכים להיראות בעורך הבלוקים.

טבלה .4-2החסימות לשליחת תגובה אוטומטית

מַטָרָה	מִגזָרָה	סוג בלוק
המטפל באירועים שמופעל כאשר הטלפון מקבל הודעת טקסט.	שלח הודעה	.1התקבלה הודעה
הגדר את המאפיין PhoneNumberלפני השליחה.	הגדר את Texting1.PhoneNumberל-SMS	
גרור ממתי חסימה מספר הטלפון של האדם ששלח את הטקסט.	מספר ערך	

סוג בלוק	מִגרָה	מַטָרָה
הגדר ל-1SMS.הודעה ל	הודעות טקסט	הגדר את מאפיין ההודעה לפני השליחה.
ResponseLabel.Text	ResponseLabel	ההודעה שהמשתמש הזין.
שליחת הודעות .1שלח הודעה	הודעות טקסט	שלח את ההודעה.



איור .4-3תגובה לטקסט נכנס

איך הבלוקים עובדים

כאשר הטלפון מקבל הודעת טקסט, האירוע Texting1.MessageReceivedהוא מופעל. מספר הטלפון של השולח נמצא במספר הארגומנט, וה-ההודעה שהתקבלה נמצאת בארגומנט .messageText

מכיוון שטקסט התגובה האוטומטית צריך להישלח בחזרה אל שולח, הודעות טקסט. PhoneNumberמוגדר למספר. הוגדרה הודעות טקסט ל- ,ResponseLabel.Textשזה מה שהקלדת בזמן המעצב: "אני נוהג ברגע זה, אצור איתך קשר בקרוב." כאשר אלה מוגדרים, האפליקציה מתקשרת Texting.SendMessageכדי לשלוח את התגובה בפועל.

בדוק את האפליקציה שלך תצטרך שני טלפונים כדי לבדוק התנהגות זו, אחד כדי להפעיל את האפליקציה ואחד כדי לשלוח את הטקסט הראשוני. אם לא אם יש לך טלפון שני בהישג יד, אתה יכול להשתמש Google Voice-באו א שירות דומה במחשב שלך ולשלוח ממנו הודעות טקסט שירות לטלפון שמריץ את האפליקציה. אחרי שתגדיר את הדברים, שלח הודעת טקסט לטלפון שמריץ את האפליקציה. עושה את הטלפון הראשון לקבל את טקסט התגובה?

הזנת תגובה מותאמת אישית

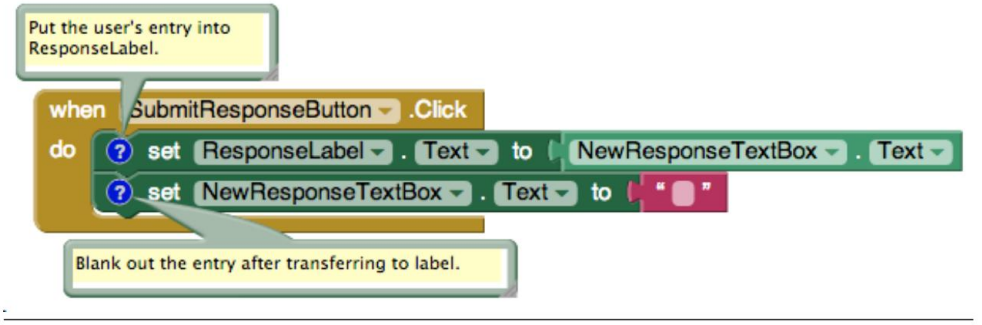
לאחר מכן, בואו נוסיף בלוקים כדי שהמשתמש יוכל להזין תגובה מותאמת אישית משלה. בתוך ה
מעצב רכיבים, הוספת רכיב TextBox בשם NewResponseTextbox;
זה המקום שבו המשתמש יקליד את התגובה המותאמת אישית. כאשר המשתמש לוחץ על
SubmitResponseButton, עליך להעתיק את הערך (NewResponseTextbox) לתוך
ResponseLabel, המשמש להגיב לטקסטים. טבלה 4-3 מפרטת את הבלוקים שתזדקק להם
להעברת תגובה שהוזנה לאחרונה ל- ResponseLabel.

טבלה 4-3. בלוקים להצגת התגובה המותאמת אישית

סוג בלוק	מִגרָה	מַטָרָה
SubmitResponseButton.Click	SubmitResponseButton.Click	המשתמש לוחץ על כפתור זה כדי לשלוח הודעה חדשה הודעת תגובה.
ResponseLabel.Text	ResponseLabel.Text	ResponseLabel העבר (הגדר) את ערך הקלט ה
NewResponseTextbox.Text	NewResponseTextbox.Text	NewResponseTextbox המשתמש הזין כאן את
NewResponseTextbox.Text ל- NewResponseTextbox הגדר את		ריק את תיבת הטקסט לאחר ההעברה מִיָדַע
טֶקסט	טקסט ("")	הטקסט הריק.

איך הבלוקים עובדים

תחשוב על האופן שבו אתה מקיים אינטראקציה עם טופס קלט טיפוסי: אתה קודם כל מקליד משהו בטקסט
ולאחר מכן לחץ על לחצן שלח כדי לאותת למערכת לעבד אותו. טופס הקלט
שכן האפליקציה הזו אינה שונה. איור 4-4 מראה כיצד הבלוקים מתוכנתים כך
כאשר המשתמש לוחץ על כפתור SubmitResponse, על SubmitResponseButton.Click
האירוע מופעל.



איור 4-4. הגדרת התגובה לכניסת המשתמש

המטפל באירועים במקרה זה מעתיק (או, במונחי תכנות, מגדיר) את מה שהמשתמש הזין ב-NewResponseTextboxל- ResponseLabel. זכור ש- ResponseLabelמחזיקה את ההודעה שתישלח בתגובה האוטומטית, אז אתה רוצה להיות בטוח שתציב את ההודעה המותאמת החדשה שהוזנה שם.

בדוק את האפליקציה שלך הזן תגובה מותאמת אישית ושלח אותה, ולאחר מכן השתמש בטלפון השני כדי לשלוח טקסט נוסף לטלפון שמריץ את האפליקציה. האם התגובה המותאמת אישית נשלחה?

אחסון התגובה המותאמת אישית בהתמדה

המשתמש שלך יכול כעת להתאים אישית את התגובה האוטומטית, אבל יש מלכוד אחד: אם המשתמש יזין תגובה מותאמת אישית ולאחר מכן סוגר את האפליקציה ומפעיל אותה מחדש, התגובה המותאמת אישית לא תופיע (במקום זאת, תגובת ברירת המחדל תופיע). התנהגות זו אינה מה שהמשתמשים שלך יצפו; הם ירצו לראות את התגובה המותאמת אישית שהזינו כשהם יפעילו מחדש את האפליקציה. כדי לגרום לזה לקרות, עליך לאחסן את התגובה המותאמת אישית הזו בהתמדה.

הצבת נתונים במאפיין ResponseLabel.Textמאחסנת אותם מבחינה טכנית, אך הבעיה היא שהנתונים המאוחסנים במאפייני הרכיב הם נתונים חולפים. נתונים חולפים הם כמו הזיכרון לטווח קצר שלך; הטלפון "שוכח" אותו ברגע שאפליקציה נסגרת. אם אתה רוצה שהאפליקציה שלך תזכור משהו בהתמדה, אתה צריך להעביר אותו מזיכרון לטווח קצר (מאפיין רכיב או משתנה) לזיכרון לטווח ארוך (מסד נתונים או מאגר מידע).

כדי לאחסן נתונים באופן מתמיד ב-ppA, Inventorאתה משתמש ברכיב , TinyDBהמאחסן נתונים בפליסה במכשיר האנדרואיד. TinyDBמספקת שתי פונקציות: StoreValue- וGetValue. עם הראשון, האפליקציה יכולה לאחסן מידע במסד הנתונים של המכשיר, ואילו עם השניה, האפליקציה יכולה לאחזר מידע שכבר נשמר.
עבור אפליקציות רבות, תשתמש בסכימה הבאה:

1. אחסן נתונים במסד הנתונים בכל פעם שהמשתמש שולח ערך חדש.

2. כאשר האפליקציה מופעלת, טען את הנתונים ממסד הנתונים למשתנה או תכונה.

תתחיל בשינוי המטפל באירועים SubmitResponseButton.Clickכך שהוא מאחסן את הנתונים באופן קבוע, תוך שימוש בלוקים המפורטים בטבלה 4-4.

טבלה 4-4. בלוקים לאחסון התגובה המותאמת אישית עם TinyDB

מִטָרָה	שם בלוק	מראה
אחסן את ההודעה המותאמת אישית במסד הנתונים של הטלפון.	TinyDB1	TinyDB1.StoreValue

סוג בלוק	מִגרָה	מַטָרָה
ResponseLabel	טקסט("responseMessage")	טקסט responseMessage("responseMessage")
		ResponseLabel.Text הודעת התגובה כאן.

איך הבלוקים עובדים

האפליקציה הזו משתמשת ב- TinyDB כדי לקחת את הטקסט שזה עתה שמה ב- ResponseLabel ולאחסן אותו במסד הנתונים. כפי שמוצג באיור 4-5, כאשר אתה מאחסן משהו במסד הנתונים, אתה מספק איתו תג; במקרה זה, התג הוא "responseMessage". חשבו על התג כשם לנתונים במסד הנתונים; זה מזהה באופן ייחודי את הנתונים שאתה מאחסן. כפי שתראה בסעיף הבא, תשתמש באותו תג ("responseMessage") כאשר אתה טוען את הנתונים חזרה ממסד הנתונים.

איור 4-5. אחסון התגובה המותאמת אישית באופן מתמיד

אחזור התגובה המותאמת אישית כאשר האפליקציה נפתחת

הסיבה לאחסון התגובה המותאמת במסד הנתונים היא כדי שניתן יהיה לטעון אותה בחזרה לאפליקציה בפעם הבאה שהמשתמש יפתח אותה. App Inventor מספק בלוק אירועים מיוחד שמופעל כאשר האפליקציה נפתחת: מסך .1אתחול (אם השלמת את MoleMash בפרק 3, ראית את זה בעבר). אם תגרור את חסימת האירוע החוצה ותציב בו בלוקים, החסימות הללו יבוצעו מיד עם הפעלת האפליקציה.

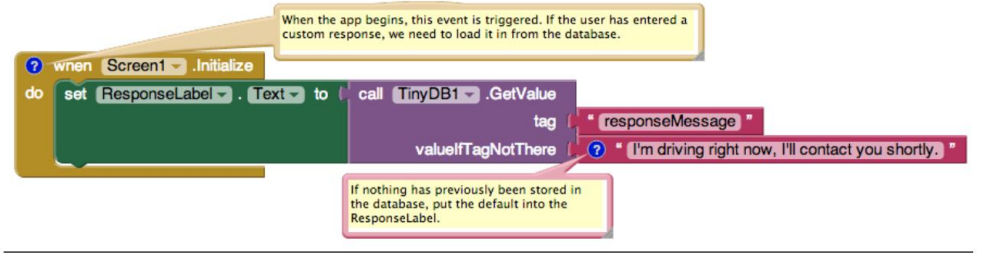
עבור אפליקציה זו, המטפל באירוע Screen1.Initialize שלך יטען את התגובה המותאמת אישית ממסד הנתונים באמצעות הפונקציה TinyDB.GetValue . הבלוקים שתזדקקו לכך מוצגים בטבלה 4-5.

טבלה 4-5. חסימות לטעינת הנתונים בחזרה בעת פתיחת האפליקציה

מַטָרָה	מְגִרָה	סוג בלוק
זה מופעל כאשר האפליקציה מתחילה.	מסך 1	מסך 1.אתחול
קבל את טקסט התגובה המאוחסן ממסד הנתונים.	TinyDB1	TinyDB1.GetValue
חבר את זה לשקע התג של TinyDB.GetValue, מה שהופך בטוח שהטקסט זהה לזה שבו נעשה שימוש ב TinyDB.StoreValue מוקדם יותר.	טֶקסט	טקסט ("responseMessage")
חבר את זה לחריץ valueIfTagNotThere של TinyDB.GetValue. זו הודעת ברירת המחדל שצריך לשמש אם המשתמש עדיין לא אחסן תגובה מותאמת אישית.	טֶקסט	טקסט ("אני נוהג עכשיו, אני אעשה צור איתך קשר בהקדם")
הצב lebaLesnopseR-ל set ResponseLabel.Text את הערך שאוחזר ב-lebaLesnopseR.		

איך הבלוקים עובדים

איור 4-6 מציג את הבלוקים. כדי להבין אותם, עליך לדמיין פתיחת משתמש האפליקציה בפעם הראשונה, הזנת תגובה מותאמת אישית ופתיחת האפליקציה פעמים שלאחר מכן. בפעם הראשונה שהמשתמש פותח את האפליקציה, לא יהיה שום התאמה אישית תגובה במסד הנתונים כדי לטעון, אז אתה רוצה להשאיר את תגובת ברירת המחדל ב- ResponseLabel. בהשקות עוקבות, אתה רוצה לטעון את המאוחסן קודם לכן תגובה מותאמת אישית ממסד הנתונים והצב אותה ב- ResponseLabel.



איור 4-6. טעינת התגובה המותאמת אישית ממסד הנתונים עם אתחול האפליקציה

כאשר האפליקציה מתחילה, האירוע Screen1.Initialize מופעל. האפליקציה קוראת ל- TinyDB1.GetValue עם תג של responseMessage, אותו תג שבו השתמשת כאשר אחסן את ערך התגובה המותאמת אישית של המשתמש קודם לכן. אם יש נתונים ב- TinyDB עם תג של responseMessage, הוא מוחזר וממוקם ב- ResponseLabel. עם זאת, לא יהיו נתונים בפעם הראשונה שבה האפליקציה תופעל; זה יהיה ה רישיות עד שהמשתמש מקליד תגובה מותאמת אישית. כדי לטפל במקרים כאלה, TinyDB1.GetValue יש פרמטר שני, valueIfTagNotThere. אם לא נמצא נתונים, הערך ב valueIfTagNotThere משמש במקום זאת. במקרה הזה, "אני נוהג עכשיו, אצור איתך קשר בקרוב," ערך ברירת המחדל, ממוקם ב- ResponseLabel.

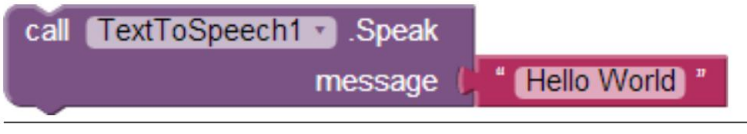
בדוק את האפליקציה שלך כדי לבדוק התנהגות זו, עליך להפעיל מחדש את האפליקציה שלך כדי לראות אם הנתונים באמת מאוחסנים באופן מתמשך ומאוחזרים כהלכה. בבדיקה חיה, אתה יכול להפעיל מחדש את האפליקציה על ידי שינוי מאפיין רכיב כלשהו במעצב, כגון גודל הגופן של תווית. זה יגרום לטעינה מחדש של האפליקציה ולהפעלת .Screen.Initializeכמובן, אתה יכול גם לבדוק את האפליקציה על ידי בנייתה בפועל והתקנת ה-.kpa fle בטלפון שלך. לאחר שהאפליקציה נמצאת בטלפון שלך, הפעל אותה, הקלד הודעה עבור התגובה המותאמת אישית, סגור את האפליקציה ולאחר מכן פתח אותה מחדש. אם ההודעה שהזנת עדיין שם, הדברים פועלים כהלכה.

דיבור הטקסטים הנכנסים בקול רם

בחלק זה, תשנה את האפליקציה כך שכאשר אתה מקבל הודעת טקסט, מספר הטלפון של השולח, יחד עם ההודעה, יושמע בקול. הרעיון כאן הוא שכאשר אתה נוהג ושומע טקסט נכנס, אתה עלול להתפתות לבדוק את הטקסט גם אם אתה יודע שהאפליקציה שולחת תגובה אוטומטית. עם טקסט לדיבור, אתה יכול לשמוע את הטקסטים הנכנסים ולהחזיק את הידיים על ההגה.

מכשירי אנדרואיד מספקים יכולות טקסט לדיבור, ו-ppA Inventorמספק רכיב, ,TextToSpeech שידבר כל טקסט שתיתן לו. שימו לב שה"טקסט" ב- TextToSpeechמתייחס לרצף של אותיות, ספרות וסימני פיסוק, לא לטקסט .SMS.

רכיב TextToSpeechהוא פשוט מאוד לשימוש. אתה פשוט קורא לפונקציית Speakשלה וחבר את הטקסט שברצונך להשמיע לתוך חריץ ההודעות שלו. לדוגמה, הבלוקים המוצגים באיור 7-4יגידו את המילים "Hello World."



איור 7-4בלוקים לדבר "שלום עולם" בקול.

עבור האפליקציה ללא הודעות טקסט בזמן נהיגה, תצטרך לספק אפליקציה מסובכת יותר הודעה לדיבור, כזו הכוללת גם את הטקסט שהתקבל וגם את מספר הטלפון של האדם ששלח אותה. במקום לחבר אובייקט טקסט סטטי כמו בלוק הטקסט ,"Hello World"תחבר בלוק הצטרפות . פונקציה נפוצה, ,joinמאפשרת לך לשלב פיסות טקסט נפרדות (או מספרים ותווים אחרים) לאובייקט טקסט אחד.

יהיה עליך לבצע את השיחה ל- TextToSpeech.Speakבתוך Texting.MessageReceivedמטפל באירועים שתכנת קודם לכן. הבלוקים שתכנתת בעבר מטפלים באירוע זה על ידי הגדרת מספר הטלפון וההודעה

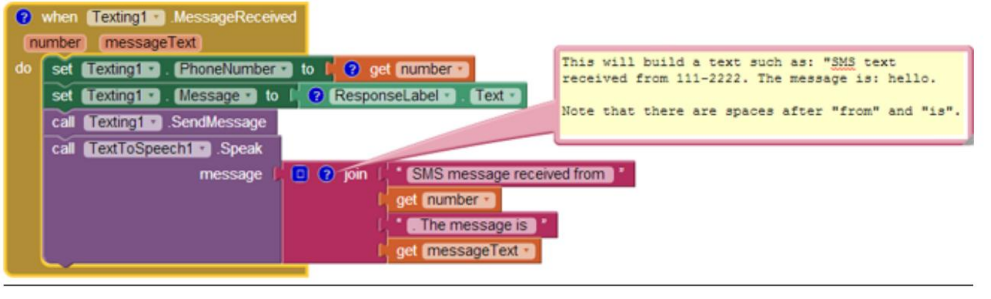
המאפיינים של רכיב הטקסטים בצורה מתאימה ולאחר מכן שליחת התגובה
טֶקסט. אתה תרחיב את המטפל באירועים על ידי הוספת הבלוקים המפורטים בטבלה 4-6.

טבלה 4-6. חסימות לקריאת הטקסט הנכנס בקול

סוג בלוק	מַטָרָה	מִגֵרָה	
TextToSpeech1.Speak	אמור (דבר יחד) את המילים שאמרו.		
שרשור	שרשר (חבר יחד) את המילים שיאמרו.		
טקסט	טקסט ("טקסט SMS שהתקבל מ")		
קבל מספר	קבל מספר מתי בלוק המספר שממנו התקבל הטקסט המקורי.		
טקסט	שים נקודה אחרי מספר הטלפון ואז אמור, "ה ההודעה היא."		
קבל הודעהטקסט	גרור פנימה מרגע החסימה ההודעה המקורית התקבלה.		

איך הבלוקים עובדים

לאחר שליחת התגובה, נקראת הפונקציה TextToSpeech1.Speak , כפי שמוצג ב-
התחתון של איור 4-8. אתה יכול לחבר כל טקסט לשקע ההודעות של
פונקציית TextToSpeech1.Speak . במקרה זה, הצטרף משמש לבניית המילים להיות
מדוברת -הוא משרשר (או מצטרף) יחד את הטקסט "טקסט SMS שהתקבל מ" וה-
מספר טלפון שממנו התקבלה ההודעה (קבל מספר), בתוספת הטקסט The."
ההודעה היא", ולבסוף ההודעה שהתקבלה (קבל הודעה טקסט). אז אם הטקסט "שלום"
נשלח מהמספר ,"111-2222" הטלפון היה אומר, "טקסט SMS התקבל מ
111-2222. ההודעה היא שלום."



איור 4-8. אמירת הטקסט הנכנס בקול רם

בדוק את האפליקציה שלך תצטרך טלפון שני כדי לבדוק את האפליקציה שלך.
מהטלפון השני, שלח הודעת טקסט לטלפון שמריץ את
אפליקציה. האם הטלפון שמריץ את האפליקציה מדבר את הטקסט בקול?
האם זה עדיין שולח תגובה אוטומטית?

הוספת פרטי מיקום לתגובה

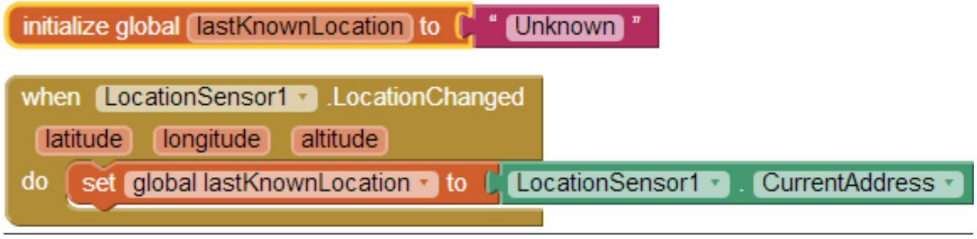
אפליקציות צ'ק-אין עוזרות לאנשים לעקוב אחר מיקומו של זה. יש פרטיות גדולה
חששות עם אפליקציות כאלה, אחת הסיבות היא שמעקב אחר מיקום מדליק את זה של אנשים
חשש ממנגנון "האח הגדול" שממשלה טוטליטרית עלולה להקים כדי לעקוב אחריו
מקום הימצאו של אזרחיה. אבל אפליקציות שמשתמשות במידע על מיקום יכולות להיות שימושיות למדי.
תחשוב על ילד שאבד, או מטיילים שעברו על השביל ביער.
באפליקציית ללא הודעות טקסט בזמן נהיגה, אתה יכול להשתמש במעקב אחר מיקום כדי להעביר קצת
מידע נוסף בתגובה האוטומטית להודעות טקסט נכנסות. במקום רק "אני נוהג",
הודעת התגובה יכולה להיות משהו כמו, "אני נוהג ואני כרגע ב-3143
שדרת הדובדבנים." עבור מישהו שמחכה לבואו של חבר או בן משפחה, זה
מידע נוסף יכול להועיל.

App Inventorמספק את רכיב LocationSensorלהתממשקות עם
ה-SPG של הטלפון (או מערכת המיקום הגלובלית). מלבד קו רוחב ואורך
מידע, חיישן המיקום יכול גם להתחבר למפות Googleכדי לספק את זה של הנהג
כתובת הרחוב הנוכחית.

חשוב לציין של- LocationSensorלא תמיד יש קריאה. לזה
סיבה, אתה צריך לדאוג להשתמש ברכיב כראוי. ליתר דיוק, האפליקציה שלך
צריך להגיב למטפל האירועים . LocationSensor.LocationChangedא
אירוע LocationChangedמתרחש כאשר חיישן המיקום של הטלפון מקבל תחילה קריאה,
וכאשר הטלפון מועבר ליצירת קריאה חדשה. שימוש בלוקים המפורטים ב
טבלה ,4-7הסכמה שלנו, המוצגת באיור ,4-9תגיב לאירוע LocationChanged
על ידי הצבת הכתובת הנוכחית במשתנה, נציין את שם .lastKnownLocationמאוחר יותר, נעשה
שנה את הודעת התגובה כך שתכלול את הכתובת שאנו מקבלים מהמשתנה הזה.

טבלה .4-7בלוקים להגדרת חיישן המיקום

מַטָרָה	מִגרָה	סוג בלוק
צור משתנה שיחזיק את הקריאה האחרונה כתובת.	משתנים	אתחול המשתנה הגלובלי ("מיקום אחרון ידוע")
הגדר את ערך ברירת המחדל למקרה של הטלפון החיישן לא עובד.	טֶקסט	טקסט ("לא ידוע")
זה מופעל במיקום הראשון קריאה וכל שינוי מיקום.	חיישן מיקום 1	LocationSensor1.LocationChanged
הגדר את המשתנה הזה לשימוש מאוחר יותר.	גרור מאתחול בלוק גלובלי.	הגדר את lastKnownLocationהגלובלי ל
זוהי כתובת רחוב כגון 2222" רחוב ווילארד, אטלנטה, ג'ורג'יה."	חיישן מיקום 1	LocationSensor1.CurrentAddress



איור 4-9. רישום מיקום הטלפון במשתנה בכל פעם שמיקום ה-SPG נמצא חש

איך הבלוקים עובדים

האירוע LocationSensor1.LocationChanged מופעל בפעם הראשונה שהחיישן מקבל קריאת מיקום ואז בכל פעם המכשיר מועבר כך שקריאה חדשה מופק. הפונקציה LocationSensor1.CurrentAddress נקראת כדי לקבל את כתובת הרחוב הנוכחית של המכשיר ואחסן אותה במשתנה lastKnownLocation .

שימו לב שעם הבלוקים האלה סיימתם רק חצי מהעבודה. האפליקציה עדיין צריכה כדי לשלב את פרטי המיקום בטקסט התגובה האוטומטית שיישלח בחזרה לשולח. אתה תעשה את זה בשלב הבא.

שליחת המיקום כחלק מהתגובה

באמצעות המשתנה lastKnownLocation, אתה יכול לשנות את ה- Texting1.MessageReceived מטפל באירועים כדי להוסיף מידע על מיקום לתגובה. טבלה 4-8 מפרטת את הבלוקים תצטרך בשביל זה.

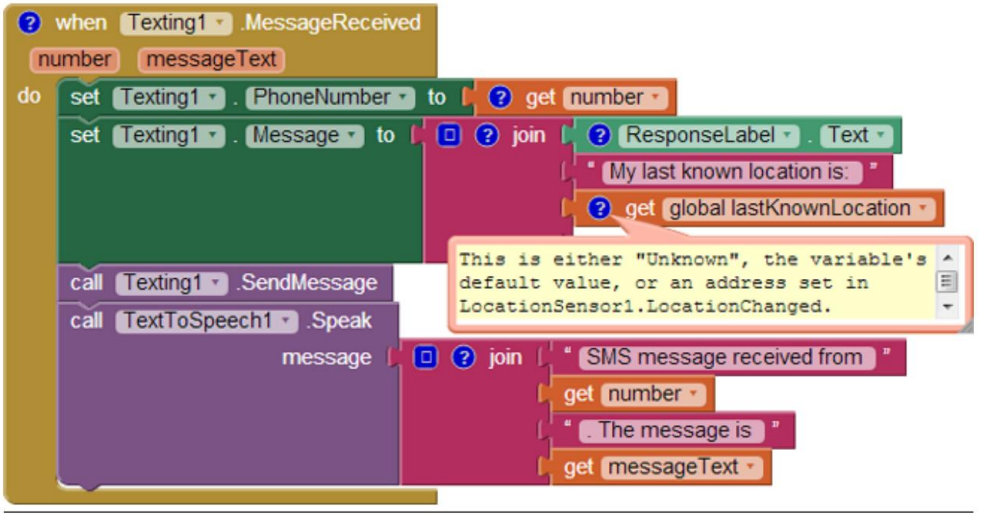
טבלה 4-8. חסימות להצגת מידע מיקום בתגובה האוטומטית

מַטָרָה	מְגִרָה	סוג בלוק
לשרשר קצת טקסט ביחד	טקסט	להצטרף
זוהי ההודעה (מותאמת אישית) בתיבת הטקסט. MessageTextBox		ResponseLabel.Text
זה ייאמר לאחר ההודעה המותאמת אישית (שים לב ל חלל מוביל).	טֶקסט	טקסט ("המיקום האחרון הידוע שלי הוא:")
זוהי כתובת כגון "1600 Pennsylvania Ave NW, וושינגטון די.סי. "20500.	קבל חיישן מיקום אחרון מיקום גלובלי	

איך הבלוקים עובדים

התנהגות זו פועלת בהתאם לאירוע LocationSensor1.LocationChanged ו המשתנה lastKnownLocation. כפי שניתן לראות באיור 4-10, במקום ישירות שליחת הודעה המכילה את הטקסט ב- ResponseLabel.Text, האפליקציה תחילה בונה א

הודעה באמצעות הצטרפות. הוא משלב את טקסט התגובה ב- ResponseLabel.Textעם הטקסט "המיקום האחרון הידוע שלי הוא:" ואחריו המשתנה lastKnownLocation.



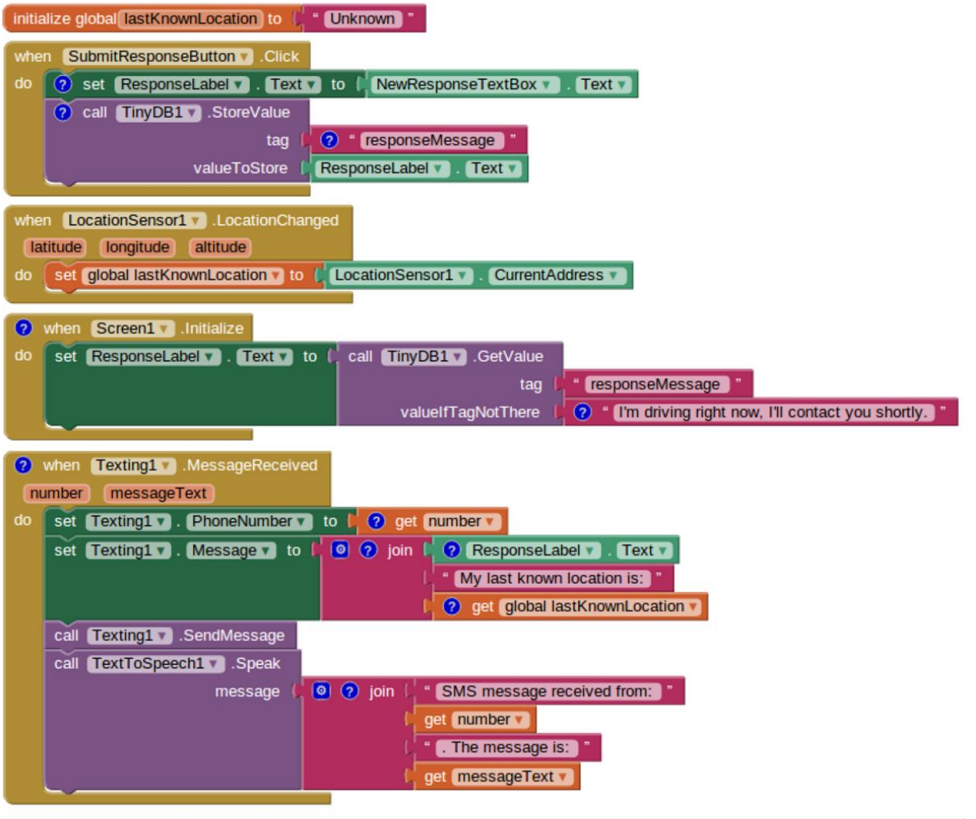
איור 4-10. כולל פרטי מיקום בטקסט התגובה

ערך ברירת המחדל של lastKnownLocationהוא "לא ידוע", אז אם חיישן המיקום עדיין לא יצר קריאה, החלק השני של הודעת התגובה יכיל את הטקסט, "המיקום האחרון הידוע שלי הוא: לא ידוע." אם הייתה קריאה, החלק השני של התגובה יהיה משהו כמו, "המיקום האחרון הידוע שלי הוא: 1600 Pennsylvania Ave NW, Washington, DC 20500."

בדוק את האפליקציה שלך מהטלפון השני, שלח הודעת טקסט לטלפון שמריץ את האפליקציה. האם הטלפון השני מקבל את טקסט התגובה עם פרטי המיקום? אם לא, ודא שהפעלת את ה-SPG בהגדרות המיקום של הטלפון שמריץ את האפליקציה.

האפליקציה השלמה: אין לשלוח הודעות טקסט בזמן נהיגה

איור 4-11מציג את תצורת הבלוק הסופי עבור ללא הודעות טקסט בזמן נהיגה.



איור 4-11. האפליקציה המלאה ללא הודעות טקסט בזמן נהיגה

וריאציות

לאחר שתפעיל את האפליקציה, אולי תרצה לחקור כמה וריאציות, כגון הבאות:

•כתוב גרסה המאפשרת למשתמש להגדיר תגובות מותאמות אישית לפרט מספרי טלפון נכנסים. תצטרך להוסיף בלוקים מותנים (אם) שבודקים את המספרים האלה. למידע נוסף על בלוקים מותנים, ראה פרק 18.

•כתוב גרסה ששולחת תגובות מותאמות אישית על סמך האם המשתמש כן בתוך גבולות קווי רוחב/קו אורך מסוימים. לכן, אם האפליקציה תקבע שאתה בחדר 222, היא תשלח בחזרה "בוב נמצא בחדר 222" ואינו יכול לשלוח הודעות טקסט עכשיו." למידע נוסף על חיישן המיקום וקביעת גבולות, ראה פרק 23.

•כתוב גרסה שמשמיעה אזעקה כאשר מתקבל טקסט ממספר ב
רשימת "להודיע". לעזרה בעבודה עם רשימות, ראה פרק 19.

סיכום

להלן כמה מהמושגים שכיסינו במדריך זה:

•אתה יכול להשתמש ברכיב SMSגם כדי לשלוח הודעות טקסט וגם לעבד את אלה שמתקבלות. לפני שתתקשר ל- ,Texting.SendMessageעליך להגדיר את המאפיינים PhoneNumber- ו Messageשל רכיב ה- . SMSכדי להגיב לטקסט נכנס, תכנת את המטפל . Texting.MessageReceived

•רכיב TinyDBמשמש לאחסון מידע מתמשך -ב-
מסד הנתונים של הטלפון -כך שניתן לטעון מחדש את הנתונים בכל פעם שהאפליקציה נפתחת.
למידע נוסף על ,TinyDBראה פרק 22.

•רכיב TextToSpeechלוקח כל אובייקט טקסט ומדבר אותו בקול רם.

•ניתן להשתמש ב- joinכדי לחבר (או לשרשר) פריטי טקסט נפרדים לאובייקט טקסט בודד.

•הרכיב LocationSensorיכול לדווח על קו הרוחב, קו האורך וכתובת הרחוב הנוכחית של הטלפון. כדי להבטיח שיש לו קריאה, עליך לגשת לנתונים שלו בתוך מטפל האירועים , LocationSensor.LocationChangedהמופעל בפעם הראשונה שקריאה מתבצעת ובכל שינוי לאחר מכן. למידע נוסף על חיישן המיקום, ראה פרק 23.

אם אתה מעוניין לחקור עוד יותר אפליקציות לעיבוד ,SMSבדוק את ה
אפליקציית Broadcast Hubבפרק 11.

מרדף פרת משה רבנו

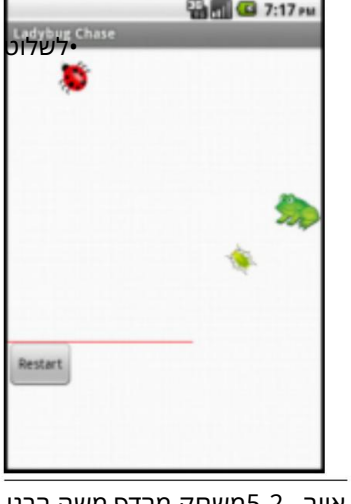
איור .5-1

משחקים הם בין האפליקציות המרגשות ביותר למכשירים ניידים, הן למשחק והן ליצירה. הלהיט האחרון Birds Angryהורד 50מיליון פעמים בשנה הראשונה שלו והוא מושמע יותר ממיליון שעות מדי יום, לפי ,Rovioהמפתחת שלו. (יש אפילו דיבורים על הפיכתו ל-mlF תכונה!) למרות שאיננו יכולים להבטיח הצלחה כזו, אנו יכולים לעזור לך ליצור משחקים משלך עם ,App Inventorכולל זה הכולל פרת משה רבנו אוכלת כנימות תוך הימנעות מצפרדע.

מה תבנה

במשחק "לעיסה בגוף ראשון", המשתמש יוצג על ידי פרת משה רבנו, שתנועתה תשלט על ידי הטיית המכשיר. זה מכניס את המשתמש למשחק בצורה שונה MoleMash-מ(פרק ,3שבו המשתמש היה מחוץ למכשיר והגיע פנימה.

אפליקציית Ladybug Chaseמוצגת באיור .5-1המשתמש יכול:



•לשלוט בפרת משה רבנו על ידי הטיית המכשיר.

•צפו בסרגל ברמת אנרגיה על המסך, שיורד עם הזמן, מה שמוביל לרעב של פרת משה רבנו.

•לגרום לפרת משה רבנו לרדוף ולאכול כנימות כדי לצבור אנרגיה ולמנוע הרעבה.

•עזרו לפרת משה רבנו להימנע מצפרדע שרוצה לאכול זה.

איור .5-2משחק מרדף משה רבנו ב-rengiseD

מה תלמד

עליך לעבוד דרך אפליקציית MoleMashבפרק 3לפני שתתעמק בפרק זה, מכיוון שהיא מניחה שאתה יודע על יצירת פרוצדורות, יצירת מספרים אקראיים, הבלוק אם-אז-אחר , ורכיבי ImageSprite, Canvas, Soundו- . Clock

בנוסף לסקירת חומר MoleMash-מומפרקים קודמים אחרים, זה הפרק מציג:

- שימוש במספר רכיבי ImageSpriteוזיהוי התנגשויות ביניהם אוֹתָם.
- זיהוי הטיות התקן עם רכיב OrientationSensorושימוש בו לשליטה ב- ImageSprite.
- שינוי התמונה המוצגת עבור .ImageSprite
- ציור קווים על רכיב . Canvas
- שליטה במספר אירועים עם רכיב שעון .
- שימוש במשתנים למעקב אחר מספרים (רמת האנרגיה של פרת משה רבנו).
- יצירה ושימוש בנהלים עם פרמטרים.
- שימוש ב- .and block

עיצוב הרכיבים

ליישום זה יהיה קנבס המספק שדה משחק עבור שלושה רכיבי : ImageSpriteאחד עבור פרת משה רבנו, אחד עבור הכנימה ואחד עבור הצפרדע, אשר ידרוש גם רכיב סאונד עבור ה"צלע" שלה. ה- OrientationSensorישמש למדידת ההטיה של המכשיר כדי להזיז את פרת משה רבנו, ושעון ישמש כדי לשנות את כיוון הכנימה.

יהיה קנבס שני שיציג את רמת האנרגיה של פרת משה רבנו. כפתור איפוס יפעיל מחדש את המשחק אם פרת משה רבנו תרעב או נאכלת. טבלה 5-1מספקת רשימה מלאה של הרכיבים באפליקציה זו.

טבלה .5-1כל הרכיבים למשחק Ladybug Chase

מַטָרָה	איך תקרא לזה	קבוצת פלטות	סוג רכיב
משחק בשדה.	FieldCanvas	ציור ו אנימציה	בַּד

מַטָרָה	איך תקרא לזה	קבוצת פלטות	סוג רכיב
נגן בשליטת משתמש.	פֶּרֶת מִשֶׁה רַבֵּנוּ	ציור ו אנימציה	ImageSprite
זהה את הטיית הטלפון כדי לשלוט על פֶּרֶת מִשֶׁה רַבֵּנוּ.	OrientationSensor1	חיישני OrientationSensor	
קובע מתי לשנות את הכותרות של .ImageSprites	שעון 1	ממשק משתמש	שָׁעוֹן
הטרף של פרת משה רבנו.	כְּנִימָה	ציור ו אנימציה	ImageSprite
הטורף של פרת משה רבנו.	צְפַרְדֵּעַ	ציור ו אנימציה	ImageSprite
הצג את רמת האנרגיה של פרת משה רבנו.	EnergyCanvas	ציור ו אנימציה	בַּד
הפעל מחדש את המשחק.	Restart Button	ממשק משתמש	לַחְצָן
"סרט" כאשר הצפרדע אוכלת את פרת משה רבנו.	צליל 1	כְּלֵי תִקְשׁוֹרֶת	נשמע

מתחילים

הורד את הפרקים הבאים:

- http://appinventor.org/bookFiles/LadybugChase/ladybug.png
- http://appinventor.org/bookFiles/LadybugChase/aphid.png
- http://appinventor.org/bookFiles/LadybugChase/dead_ladybug.png
- http://appinventor.org/bookFiles/LadybugChase/frog.png
- http://appinventor.org/bookFiles/LadybugChase/frog.wav

אלו תמונות של פרת משה רבנו, כנימה, פרת משה רבנו מתה וצפרדע, כמו גם צליל לברוח לצלע הצפרדע. לאחר הורדתם למחשב, הוסף אותם למחשב שלך אפליקציה בחלק המדיה של המעצב.
התחבר לאתר App Inventorוהתחל פרויקט חדש. תן שם "LadybugChase"וגם הגדר את כותרת המסך ל"gubydaL ."Chaseפתח את הבלוקים עורך והתחבר למכשיר.

הצבת הרכיבים הראשוניים

למרות שבפרקים הקודמים אתה יוצר את כל הרכיבים בבת אחת, זהו לא איך מפתחים עובדים בדרך כלל. במקום זאת, נפוץ יותר ליצור חלק אחד של a תוכנית בכל פעם, בדוק אותה ולאחר מכן עבור לחלק הבא של התוכנית. בזה בקטע, ניצור את פרת משה רבנו ונשלוט בתנועתה.

•ב-rengiseD, צור קנבס, שם לו ,FieldCanvasוהגדר את הרוחב שלו ל-"lliF "parentואת הגובה שלו ל-003 פיקסלים.

•הצב ImageSpriteעל הקנבס , שנה את שמו של Ladybugוקבע את תכונת התמונה שלו לתמונת פרת משה רבנו. אל תדאג לגבי הערכים של המאפיינים Xו- , Yכי אלה יהיו תלויים היכן על הקנבס הנחת את .ImageSprite

כפי שאולי שמת לב, ImageSprites-ליש גם מאפייני מרווח, כותרת ומהירות , שבהם תשתמש בתוכנית זו:

•המאפיין , Intervalשתוכל להגדיר ל-01 (מילישניות) עבור המשחק הזה, מציין באיזו תדירות ה-ImageSpriteצריך להזיז את עצמו (בניגוד להזזה על ידי הליך , MoveToשבו השתמשת עבור .MoleMash)

•המאפיין Headingמציין את הכיוון שאליו אמור ה- ImageSpriteלנוע, במעלות. לדוגמה, 0 פירושו עקב ימינה, 90פירושו ישר כלפי מעלה, 180פירושו עקב שמאלה, וכן הלאה. השאר את הכותרת כפי שהיא כרגע; נשנה את זה בעורך הבלוקים.

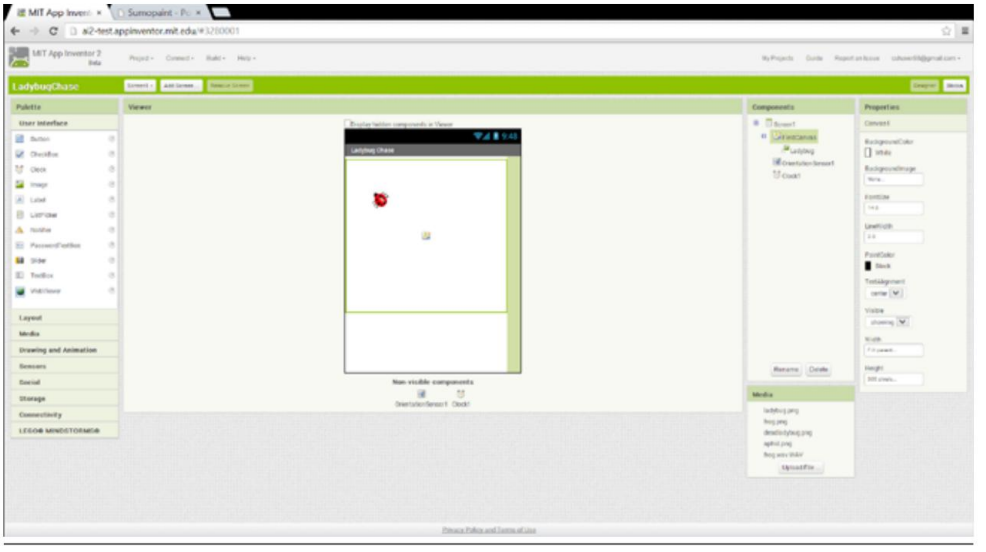
•המאפיין Speedמציין כמה פיקסלים ה- ImageSpriteצריך להזיז בכל פעם שחולף המרווח שלו 10)אלפיות שניות). אנחנו גם נגדיר את המאפיין Speedבעורך הבלוקים.

התנועה של פרת משה רבנו תישלט על ידי ,OrientationSensorהמזהה כיצד המכשיר מוטה. אנו רוצים להשתמש ברכיב Clockכדי לבדוק את כיוון המכשיר כל 10מילישניות 100)פעמים בשנייה) ולשנות את ה- Heading(כיוון) של פרת משה רבנו בהתאם. נגדיר זאת בעורך הבלוקים באופן הבא:

.1הוסף ,OrientationSensorשיופיע ב"רכיבים שאינם נראים" סָעִיף.

.2הוסף שעון, שיופיע גם בסעיף "רכיבים שאינם נראים", וקבע את ה- TimerIntervalשלו ל-01 מילישניות. בדוק מה הוספת מול איור .2-5

אם תשתמש במכשיר אחר מלבד האמולטור, תצטרך להשבית את האוטומטי סיבוב למסך, אשר משנה את כיוון התצוגה כאשר אתה מסובב את המכשיר. בחר Screen1והגדר את המאפיין ScreenOrientationשלו ל- .Portrait



איור .5-3הגדרת ממשק המשתמש Component Designer-בהנפשת פרת משה רבנו

הוספת התנהגויות לרכיבים

מזיזים את פרת משה רבנו

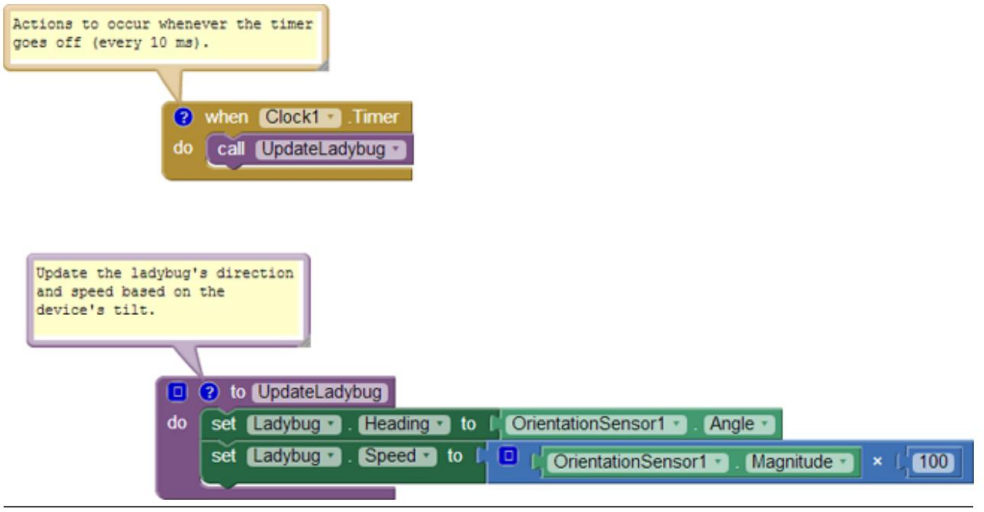
מעבר לעורך בלוקים, צור את ההליך UpdateLadybugובלוק , Clock1.Timerכפי שמוצג באיור .5-3נסה להקליד את השמות של חלק מהבלוקים (כגון "Clock1.Timer"במקום לגרור אותם מהמגירות. (שים לב שהפעולה המוחלת על המספר 100היא כפל, המסומנת בכוכבית, שאולי קשה לראות אותה בתמונה.) אינך צריך ליצור את תוספי ההערה הצהובים, למרות שאתה יכול על ידי לחיצה ימנית על בלוק ו בחירה בהוספת הערה.

הליך UpdateLadybugעושה שימוש בשניים מהרובים של OrientationSensor מאפיינים שימושיים:

•זווית, המציינת את כיוון הטיית המכשיר (במעלות).

•גודל, המציין את כמות ההטיה, נע בין 0(ללא הטיה) ל-1 (הטיה מקסימלית).

הכפלת ה- Magnitudeב-001 אומרת לפרת משה רבנו שהיא צריכה לנוע בין 0ל-001 פיקסלים ב- Heading(כיוון) שצוין בכל פעם שעובר TimerIntervalשלו, שהגדרתם בעבר ל-01 אלפיות שניות .Component Designer-ב
למרות שאתה יכול לנסות את זה במכשיר המחובר, ייתכן שתנועת פרת משה רבנו תהיה איטית יותר ומקומטת יותר מאשר אם תארז ותוריד את האפליקציה למכשיר. אם, לאחר שעשית זאת, אתה מוצא את התנועה של פרת משה רבנו איטית מדי, הגדל את מכפיל המהירות. אם פרת משה רבנו נראית קופצנית מדי, הקטינו אותה.



איור .5-4שינוי הכיוון והמהירות של פרת משה רבנו כל 10מילישניות

הצגת רמת האנרגיה

נציג את רמת האנרגיה של פרת משה רבנו באמצעות פס אדום בקנבס שני. גובה הקו יהיה פיקסל אחד, והרוחב שלו יהיה אותו מספר פיקסלים כמו האנרגיה של פרת משה רבנו, שנעה בין 200(מוזנת היטב) ל-0 (מתה).

הוספת רכיב

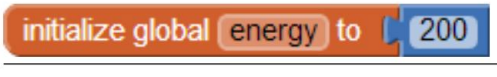
במעצב, צור קנבס חדש . מקם אותו מתחת FieldCanvas-לוקרא לו . EnergyCanvas הגדר את מאפיין הרוחב שלו ל-"lliF" "parentואת הגובה שלו ל-1 פיקסל.

יצירת משתנה: אנרגיה

בעורך בלוקים, תצטרך ליצור אנרגיה משתנה עם ערך התחלתי של 200כדי לעקוב אחר רמת האנרגיה של פרת משה רבנו, כפי שמוצג באיור .5-4(כפי שאתם אולי זוכרים, השתמשנו תחילה במשתנה, ,dotSizeבאפליקציית PaintPotבפרק (.2הנה איך לעשות זאת:

.1בעורך הבלוקים, גרור החוצה שם גלובלי לאתחל לחסימה. שנה את הטקסט "שם" ל"אנרגיה".

.2צור בלוק מספר 200(על ידי התחלת הקלדת המספר 200או גרירת גוש מספרים ממגירת המתמטיקה) וחבר אותו לאנרגיה גלובלית, כפי שמוצג באיור .5-4



איור .5-5אתחול האנרגיה המשתנה ל-002

איור 5-5ממחיש שכאשר אתה מגדיר משתנה, סט חדש וקבל בלוקים
נוצר עבורו שאליו תוכל לגשת על ידי העברת העכבר מעל שם המשתנה.

איור .5-6כאשר אתה מעביר את העכבר מעל משתנה בבלוק הגלובלי האתחול, אתה יכול לגרור את הסט ולקבל בלוקים עבור המשתנה

ציור חטיף האנרגיה

אנו רוצים לתקשר את רמת האנרגיה באמצעות פס אדום, שאורך זה בפיקסלים שווה לערך האנרגיה. לשם כך, נוכל ליצור שתי קבוצות דומות של בלוקים באופן הבא:

.1צייר קו אדום מ-(0, (0עד (אנרגיה, (0ב- FieldCanvasכדי להציג את הנוכחי רמת אנרגיה.

.2צייר קו לבן מ-(0, (0עד (EnergyCanvas.Width, 0)ב- FieldCanvasכדי למחוק רמת האנרגיה הנוכחית לפני ציור הרמה החדשה.

עם זאת, חלופה טובה יותר היא ליצור הליך שיכול לצייר קו בכל אורך ובכל צבע ב-FieldCanvas.לשם כך, עלינו לציין שני פרמטרים, אורך וצבע , כאשר הפרוצדורה שלנו נקראת, בדיוק כפי שהיינו צריכים לציין ערכי פרמטרים MoleMash-בכאשר קראנו לפרוצדורה האקראית המובנית . להלן השלבים ליצירת הליך : DrawEnergyLine

.1עבור אל מגירת הפרוצדורות וגרור החוצה את הבלוק להליך לעשות . בחר את הגרסה שמכילה "עשה" במקום "החזר" כי ההליך שלנו לא יחזיר ערך.

.2לחץ על שם הפרוצדורה ("פרוצדורה") ושנה אותו ל"eniLygrenEwarD".

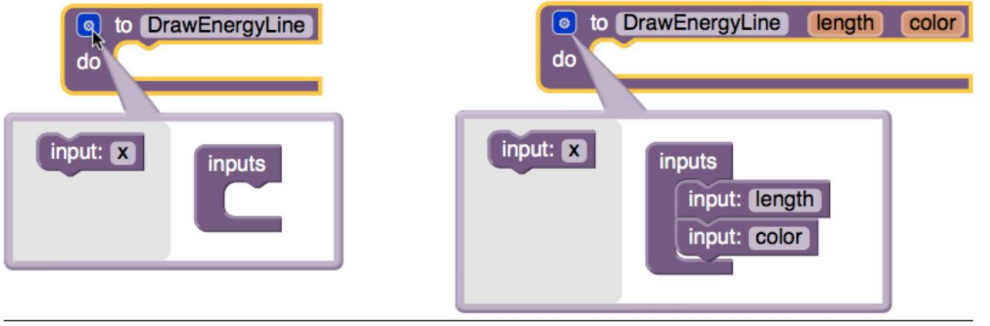
.3בפינה השמאלית העליונה של הבלוק החדש, לחץ על הריבוע הכחול הקטן. זה פותח את חלון המוצג בצד שמאל של איור .5-6

.4מהצד השמאלי של חלון זה, גרור קלט לצד ימין, ושנה את שמו מ-"x" ל-"אורך". זה מצביע על כך שלהליך יהיה פרמטר בשם "אורך".

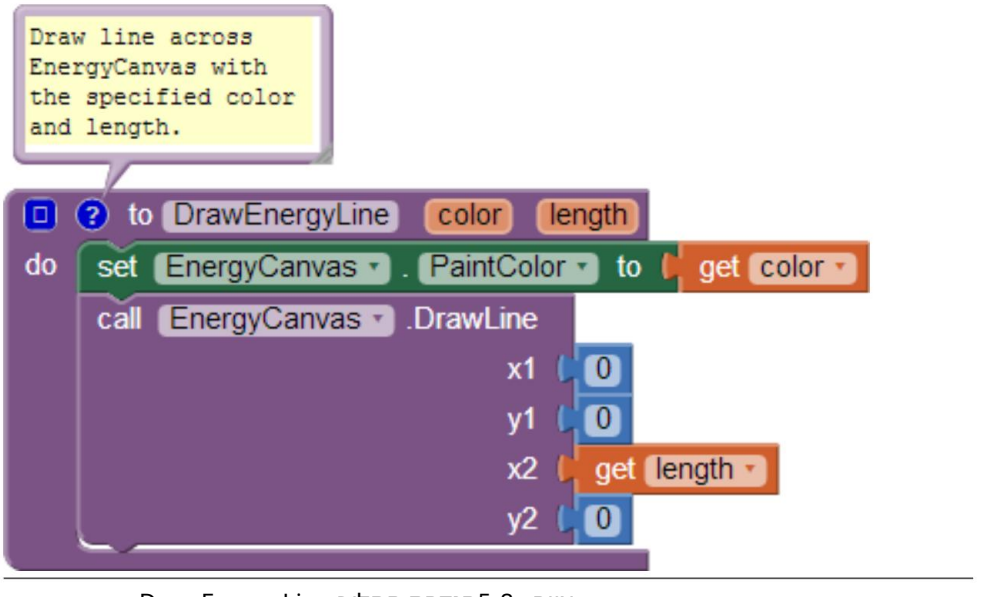
.5חזור על פרמטר שני בשם "צבע", שעליו לעבור מתחת לפרמטר ששמו "אורך". זה אמור להיראות כמו הצד הימני של איור .5-6

6. לחץ שוב על הסמל הכחול כדי לסגור את חלון הקלטות.

7. מלא את שאר ההליך כפי שמוצג באיור 5-7. אתה יכול למצוא צבע ולקבל אורך על ידי העברת העכבר מעל שמותיהם בהגדרת הנוהל.

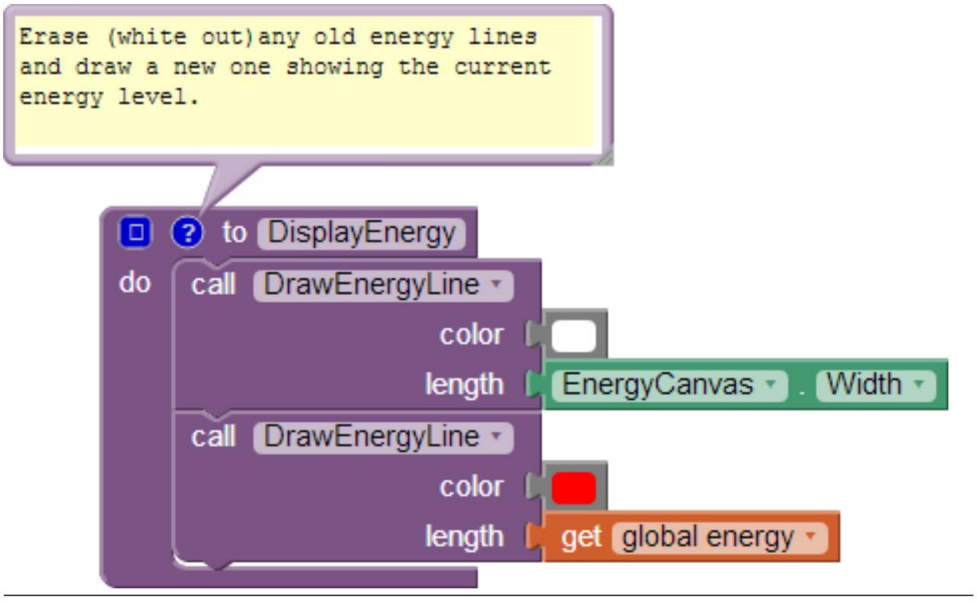


איור 5-7. הוספת כניסות (פרמטרים) להליך DrawEnergyLine



איור 5-8. הגדרת ההליך DrawEnergyLine

עכשיו, כשאתה מתחיל ליצור נהלים משלך, בוא נכתוב גם נוהל DisplayEnergy שקורא ל-DrawEnergyLine פעמיים: פעם אחת כדי למחוק את הקו הישן (על ידי ציור קו לבן לאורך כל הקנבס), ופעם אחת כדי להציג את קו חדש, כפי שמוצג באיור 5-8.



איור 5-9. הגדרת הנוהל DisplayEnergy

הליך DisplayEnergy מורכב מארבע שורות שעושות את הפעולות הבאות:

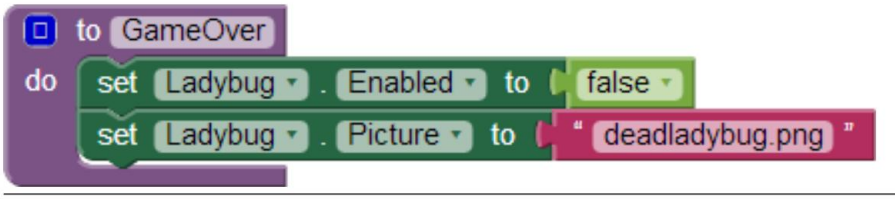
1. הגדר את צבע הצבע ללבן.
2. צייר קו לאורך כל הדרך של EnergyCanvas (שגובהו רק פיקסל אחד).
3. הגדר את צבע הצבע לאדום.
4. צייר קו שאורכו בפיקסלים זהה לערך האנרגיה.

הערה התהליך של החלפת קוד נפוץ בקריאות להליך חדש נקרא refactoring, קבוצה של טכניקות חזקות להפיכת תוכניות לניתנות לתחזוקה ואמינות יותר. במקרה זה, אם אי פעם רצינו לשנות את הגובה או המיקום של קו האנרגיה, היינו צריכים פשוט לבצע שינוי בודד ב- DrawEnergyLine, במקום לבצע שינויים בכל קריאה אליו.

רֶעֶב

בניגוד לאפליקציות בפרקים הקודמים, למשחק הזה יש דרך לסיים: זה נגמר אם פרת משה רבנו לא מצליחה לאכול מספיק כנימות או שהיא עצמה נאכלת על ידי הצפרדע. בכל אחד מהמקרים הללו, אנו רוצים שהפרת משה רבנו תפסיק לזוז (מה שאנו יכולים לעשות על ידי הגדרת Ladybug.Enabled ל-eslaf) ושהתמונה תשתנה מפרת משה רבנו למת

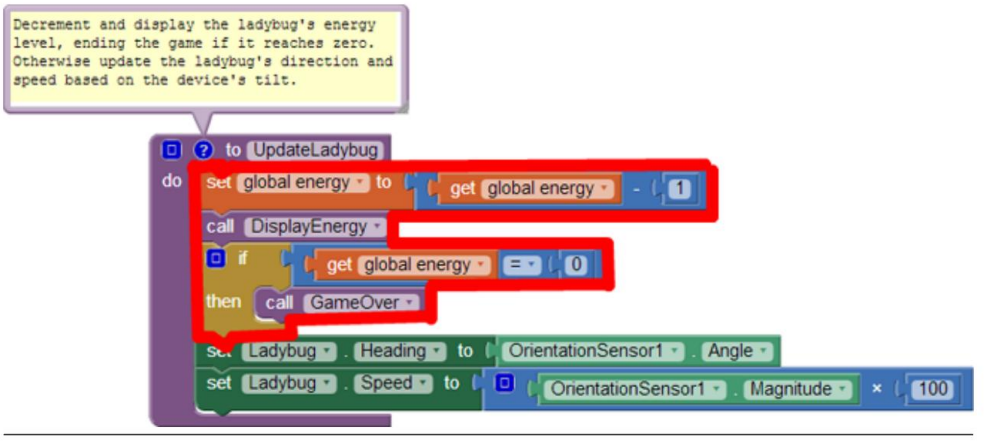
אחד (שנוכל לעשות על ידי שינוי Ladybug.Pictureלשם התמונה המתאימה שהועלתה). צור את הליך GameOverכפי שמוצג באיור 5-9.



איור 5-10. הגדרת הליך GameOver

לאחר מכן, הוסף את הקוד המתואר באדום באיור 5-10ל- UpdateLadybug(אשר, כפי שאתה עשוי להיזכר, נקרא על ידי Clock.Timerכל 10אלפיות שניות) לדברים הבאים:

- הורד את רמת האנרגיה שלו.
- הצג את הרמה החדשה.
- סיים את המשחק אם האנרגיה היא 0.



איור 5-11. גרסה שנייה של ההליך UpdateLadybug

בדוק את האפליקציה שלך במכשיר שלך, ודא שרמת האנרגיה יורדת עם הזמן, ובסופו של דבר גורמת למותו של פרת משה רבנו.

הוספת כנימה

השלב הבא הוא הוספת כנימה. באופן ספציפי, כנימה צריכה להתאים סביב FieldCanvas. אם פרת משה רבנו נתקלת בכנימה (ובכך "אוכלת" אותה), רמת האנרגיה של הפרת משה רבינו אמורה לעלות והכנימה תיעלם ותוחלף באחרת מעט.

יותר מאוחר. (מנקודת המבט של המשתמש, זו תהיה כנימה שונה, אבל זה באמת יהיה אותו רכיב ImageSprite .)

הוספת ImageSprite

הצעד הראשון שאתה צריך לעשות כדי להוסיף כנימה הוא לחזור למעצב וליצור ImageSpriteנוסף, תוך הקפדה לא למקם אותו על גבי פרת משה רבנו. יש לשנות את שמו של Aphidולהגדיר כדלקמן:

•הגדר את המאפיין Pictureשלו לקובץ תמונת הכנימה שהעלית.

•הגדר את מאפיין ה- Intervalשלו ל-01, אז, כמו פרת משה רבנו, הוא זז כל 10 אלפיות השנייה.

•הגדר את המהירות שלו ל-2, כדי שהוא לא יזוז מהר מדי כדי שהפרת משה רבנו תתפוס אותו.

אל תדאג לגבי מאפייני ה- xוה- yשלו (כל עוד זה לא על גבי פרת משה רבנו) או המאפיין Headingשלו , שיוגדר בעורך הבלוקים.

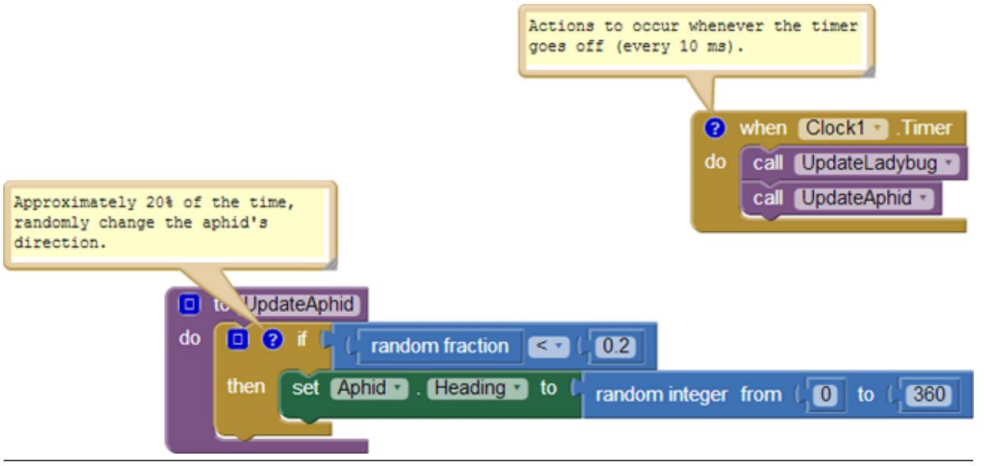
שליטה בכנימה

על ידי ניסויים, גילינו שזה עבד הכי טוב עבור הכנימה לשנות כיוון בערך אחת ל-05 אלפיות השנייה 5)"תקתוקים" של .(Clock1גישה אחת להפעלת התנהגות זו תהיה יצירת שעון שני עם מרווח טיימר של 50מילישניות. עם זאת, נרצה שתנסה טכניקה שונה כדי שתוכל ללמוד על גוש השבר האקראי , המחזיר מספר אקראי הגדול או שווה ל-0 וקטן מ-1 בכל פעם שהוא נקרא. צור את הליך UpdateAphidהמוצג באיור 11-5והוסף לו קריאה ב- .Clock1.Timer

איך הבלוקים עובדים

בכל פעם שהטיימר כבה 100)פעמים בשנייה), גם UpdateLadybug(כמו קודם) וגם UpdateAphidנקראים. הדבר הראשון שקורה ב- UpdateAphidהוא שנוצר שבר אקראי בין 0ל-1 -לדוגמה, .0.15אם מספר זה קטן מ-02.0 (מה שיקרה ב-02% מהמקרים), הכנימה תשנה את כיוונה למספר אקראי של מעלות בין 0ל-063. אם המספר לא קטן מ-02.0 (מה שיקרה שנותרו 80%מהזמן), הכנימה תישאר

קורס.



איור 5-12. הוספת הנוהל UpdateAphid

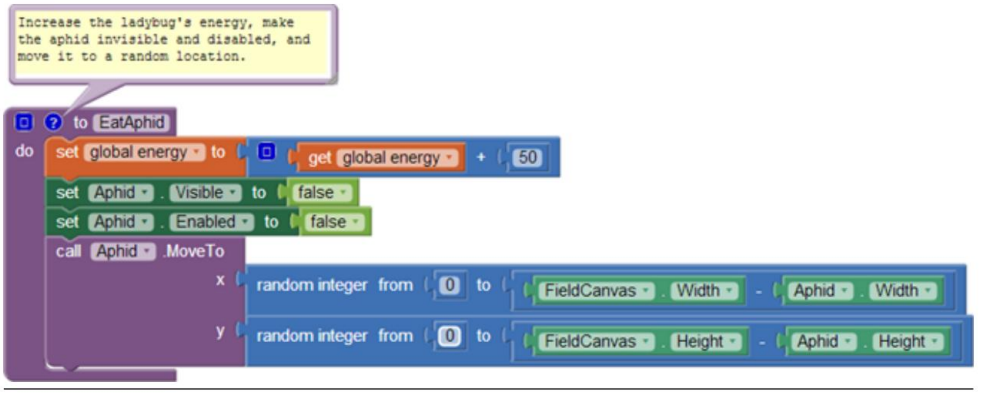
תכנות את פרת משה רבנו לאכול את הכנימה

השלב הבא הוא להגדיר את פרת משה רבנו "לאכול" את הכנימה כשהן מתנגשות. למרבה המזל, App Inventor מספק בלוקים לזיהוי התנגשויות בין רכיבי ImageSprite, מה שמעלה את השאלה: מה צריך לקרות כאשר פרת משה רבנו והכנימה מתנגשות? אולי כדאי לעצור ולחשוב על זה לפני שתמשיך לקרוא.

כדי להתמודד עם מה שקורה כאשר פרת משה רבנו וכנימה מתנגשות, בואו ניצור א נוהל, EatAphid, שעושה את הפעולות הבאות:

- מעלה את רמת האנרגיה ב-05 כדי לדמות אכילת הפינוק הטעים.
- גורם לכנימה להיעלם (על ידי הגדרת המאפיין Visible שלה ל-eslaf).
- גורם לכנימה להפסיק לנוע (על ידי הגדרת המאפיין Enabled שלה ל-eslaf).
- גורם לכנימה לעבור למיקום אקראי על המסך. (זה עוקב אחר אותו דפוס כמו הקוד להזזת השומה .(MoleMash-ב

בדוק שהבלוקים שלך תואמים לתמונה 5-12. אם היו לך רעיונות אחרים מה צריך להתרחש, כגון השפעות סאונד, אתה יכול להוסיף גם את אלה.



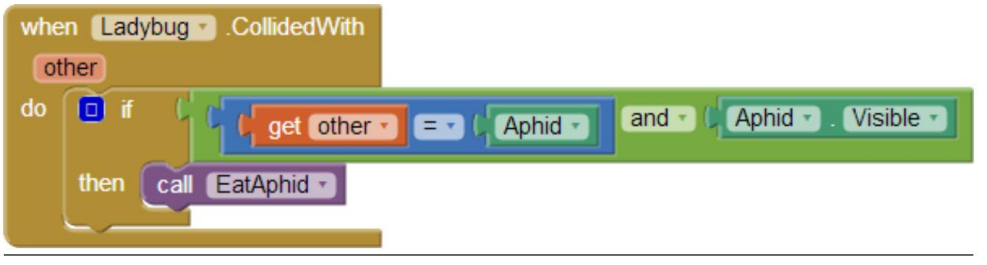
איור 5-13. הוספת הפרוצדורה EatAphid

איך הבלוקים עובדים

בכל פעם שקוראים ל-dihpAtaE , זה מוסיף 50לאנרגיה המשתנה, ומקל על הרעבה עבור פרת משה רבנו. לאחר מכן, המאפיינים Visibleו- Enabledשל הכנימה מוגדרים ל-eslaf כך שנראה שהיא נעלמת ומפסיקה לזוז, בהתאמה. לבסוף, קואורדינטות xו-y אקראיות נוצרות עבור קריאה ל- Aphid.MoveToכך שכאשר הכנימה תופיע שוב, היא נמצאת במיקום חדש (אחרת, היא תיאכל ברגע שהיא תופיע מחדש).

זיהוי התנגשות פרת משה רבנו-כנימה

איור 5-13מציג את הקוד לזיהוי התנגשויות בין פרת משה רבנו לכנימה.



איור 5-14. זיהוי ופעולה בהתנגשויות בין פרת משה רבנו לכנימה

איך הבלוקים עובדים

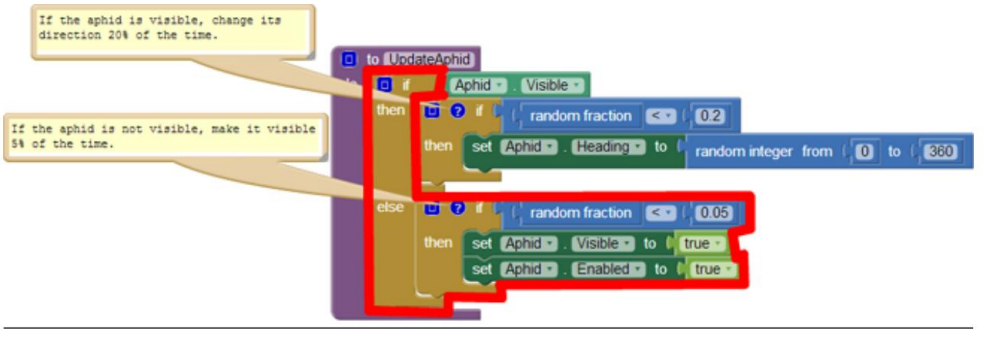
כאשר פרת משה רבנו מתנגשת עם ImageSpriteאחר , נקראת , Ladybug.CollidedWith כאשר הפרמטר "אחר" קשור לכל מה שהפרת משה רבנו התנגשה בו. כרגע, הדבר היחיד שהוא יכול להתנגש בו הוא הכנימה, אבל נוסיף צפרדע מאוחר יותר. נשתמש בתכנות הגנתי ונבדוק במפורש שההתנגשות הייתה עם הכנימה לפני שנתקשר ל-dihpAtaE. יש גם בדיקה כדי לאשר שהכנימה גלויה.
אחרת, לאחר אכילת כנימה אך לפני שהיא מופיעה שוב, היא עלולה להתנגש עם הכנימה

שוב פרת משה רבנו. ללא הצ'ק, הכנימה הבלתי נראית תיאכל שוב, וגורמת לקפיצה נוספת באנרגיה מבלי שהמשתמש יבין מדוע.

הערה תכנות הגנתי הוא התרגול של כתיבת קוד בצורה כזו שסביר להניח שהוא עדיין יעבוד גם אם התוכנית משתנה. באיור 5-13, הבדיקה "אחר =כנימה" אינה הכרחית לחלוטין מכיוון שהדבר היחיד בו פרת משה רבנו יכולה להתנגש כעת הוא הכנימה, אך ביצוע הבדיקה ימנע תקלה בתוכנית שלנו אם נוסיף עוד ImageSprite ונשכח לשנות פרת משה רבנו. התנגש. מתכנתים בדרך כלל מבלים יותר זמן בתיקון באגים (מגוון התוכנה, לא מסוג אכילת כנימות) מאשר בכתיבת קוד חדש, אז כדאי מאוד להקדיש זמן לכתיבת קוד באופן שמונע בעיות מלכתחילה.

שובה של הכנימה

כדי לגרום לכנימה להופיע בסופו של דבר, עליך לשנות את UpdateAphid כפי שמוצג באיור 5-14 כך שהוא ישנה את כיוון הכנימה רק אם היא נראית לעין. (החלפתו אם היא בלתי נראית היא בזבוז זמן.) אם הכנימה אינה נראית לעין (כמו ב, היא נאכלה לאחרונה), יש סיכוי של 1 ל-02 (5%) שהיא תופעל מחדש -ב במילים אחרות, זכאי לאכילה שוב.



איור 5-15. שינוי UpdateAphid כדי לגרום לכנימות בלתי נראות לחזור לחיים

איך הבלוקים עובדים

UpdateAphid נהיה די מורכב, אז בואו נעבור בזהירות על ההתנהגות שלו:

•אם הכנימה גלויה (מה שיקרה אלא אם זה עתה נאכלה),
UpdateAphid מתנהג כפי שכתבנו אותו לראשונה. באופן ספציפי, יש סיכוי של 20% לשנות את הכיוון שלו.

•אם הכנימה אינה נראית לעין (כלומר, היא נאכלה לאחרונה), אז החלק האחר של בלוק else if יפעל. לאחר מכן נוצר מספר אקראי. אם הוא נמוך מ-50.0 (שזה יהיה 5% מהמקרים), הכנימה הופכת שוב לגלויה והיא מופעלת, מה שהופך אותה כשירה לאכילה שוב.

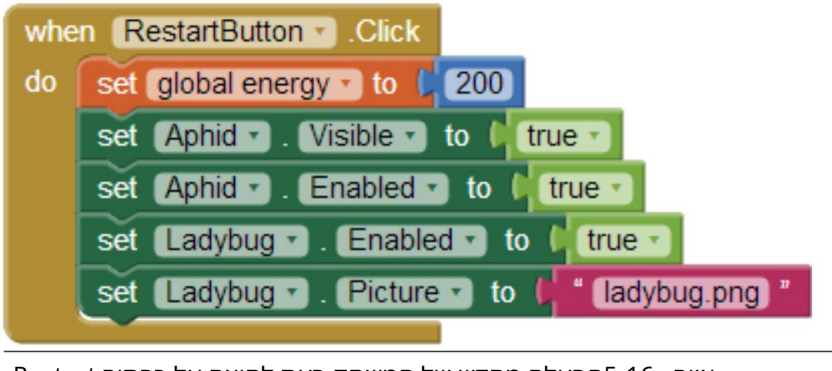
מכיוון ש- UpdateAphid נקרא על ידי Clock1.Timer, המתרחש כל 10 אלפיות שניות, ויש סיכוי של 1 ל-20 (5%) שהכנימה תחזור להיות גלויה, הכנימה ייקח בממוצע 200 אלפיות השנייה (חצי שנייה) להופיע שוב.

הוספת כפתור הפעלה מחדש

כפי שאולי שמתם לב מבדיקת האפליקציה עם הפונקציונליות החדשה של אכילת כנימות, המשחק באמת צריך כפתור Restart. (זו סיבה נוספת לכך שמועיל לעצב ולבנות את האפליקציה שלך בנתחים קטנים ואז לבדוק אותה - לעתים קרובות אתה מגלה דברים שהתעלמתם מהם, וקל יותר להוסיף אותם תוך כדי התקדמות מאשר לחזור ולשנות אותם לאחר האפליקציה הושלמה.) בהוספ Designer, רכיב Button מתחת ל- Component, שנה את שמו של EnergyCanvas, "RestartButton", והגדר את מאפיין ה- Text שלו ל-"tratseR".

בעורך הבלוקים, צור את הקוד המוצג באיור 5-15 כדי לבצע את הפעולות הבאות מתי לוחצים על כפתור ההפעלה מחדש :

1. החזר את רמת האנרגיה ל-200.
2. הפעל מחדש את הכנימה והפוך אותה לגלויה.
3. הפעל מחדש את פרת משה רבנו ושנה את התמונה שלה בחזרה לפרת משה רבנו (אלא אם כן אתה רוצה פרת משה רבנו זומבי!).



איור 5-16. הפעלה מחדש של המשחק בעת לחיצה על כפתור Restart

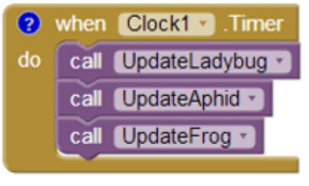
הוספת הצפרדע

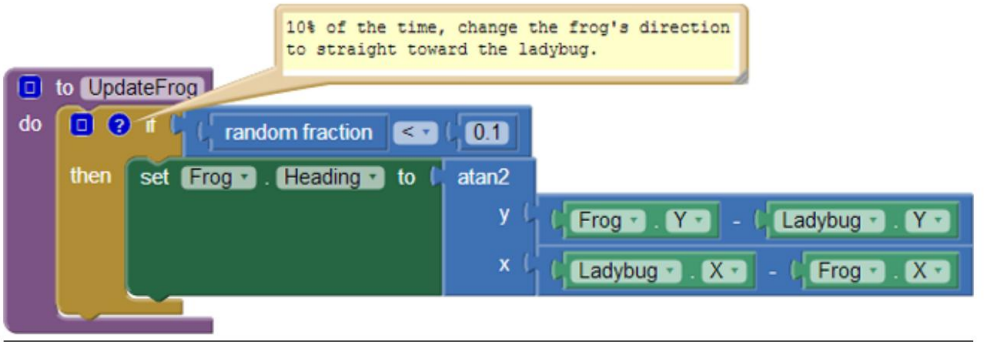
כרגע, לשמור על פרת משה רבנו בחיים זה לא קשה מדי. אנחנו צריכים טורף. באופן ספציפי, נוסיף צפרדע שנעה ישירות לכיוון פרת משה רבנו. אם הם מתנגשים, פרת משה רבנו הופכת לארוחת ערב, והמשחק מסתיים.

לגרום לצפרדע לרדוף אחרי פרת משה רבנו

הצעד הראשון להגדרת הצפרדע לרדוף אחרי פרת משה רבנו הוא החזרה למעצב הרכיבים והוספת ImageSprite שלישית - Frog -ל- .FieldCanvasהגדר את תכונת התמונה שלו לתמונה המתאימה, המרווח שלו ל-01, והמהירות שלו ל-1, כי הוא אמור לנוע לאט יותר מהיצורים האחרים.

איור 16-5מציג את ,UpdateFrogהליך חדש שאתה צריך ליצור ולהתקשר ממנו שעון1. טיימר.





איור .5-17לגרום לצפרדע לנוע לעבר פרת משה רבנו

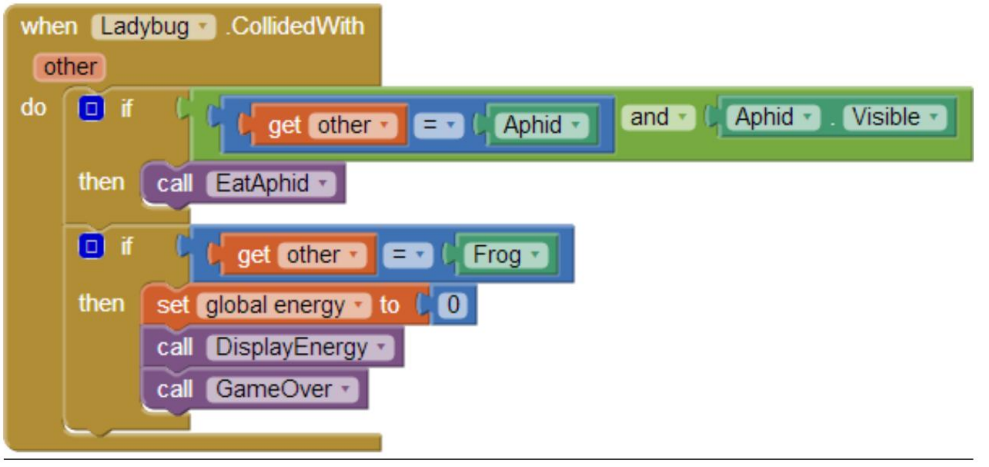
איך הבלוקים עובדים

עד עכשיו, אתה אמור להכיר את השימוש בגוש השבר האקראי כדי לגרום לאירוע להתרחש בהסתברות מסוימת. במקרה זה, יש סיכוי של 10%שהכיוון של הצפרדע ישתנה לכיוון ישר לכיוון פרת משה רבנו. זה דורש טריגונומטריה, אבל אל תיכנס לפאניקה -אתה לא צריך לברר את זה בעצמך. App Inventorמטפל בהרבה פונקציות מתמטיות עבורך, אפילו דברים כמו טריגונומטריה. במקרה זה, אתה רוצה להשתמש בבלוק ,(arctangent) atan2 אשר מחזיר את הזווית המתאימה לקבוצה נתונה של ערכי x-y. (לאלו מכם שמכירים טריגונומטריה, הסיבה

לארגומנט y של atan2 יש את הסימן ההפוך למה שהיית מצפה - הסדר ההפוך של הארגומנטים להחסיר - הוא שקואורדינטת ה-y גדלה בכיוון מטה ב-Canvas, ההפך ממה שיתרחש בקואורדינטת xy סטנדרטית. מערכת.)

מגדירים את הצפרדע לאכול את פרת משה רבנו

כעת עלינו לשנות את קוד ההתנגשות כך שאם פרת משה רבנו תתנגש בצפרדע, רמת האנרגיה והסרגל יעברו ל-0 והמשחק יסתיים, כפי שמוצג באיור 5-17.



איור 5-18. לגרום לצפרדע לאכול את פרת משה רבנו

איך הבלוקים עובדים

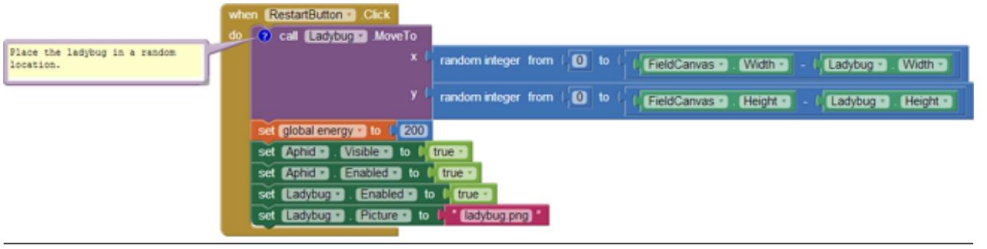
בנוסף ל-if, tsrf שבודק אם פרת משה רבנו התנגשה בכנימה, יש כעת if, שבודק אם פרת משה רבנו התנגשה בצפרדע. אם פרת משה רבנו והצפרדע מתנגשות, קורים שלושה דברים:

1. האנרגיה המשתנה יורדת ל-0, כי פרת משה רבנו איבדה את כוח החיים שלה.
2. נקראת DisplayEnergy למחוק את קו האנרגיה הקודם (ולצייר את החדש, אחד ריק).
3. הנוהל שכתבת קודם, GameOver, נקרא לעצור את תנועת פרת משה רבנו ולשנות את תמונתה לזו של פרת משה רבנו מת.

שובה של פרת משה רבנו

לכבר יש קוד להחלפת התמונה של פרת משה רבנו המתה בזו של RestartButton.Click- פרת משה רבנו. כעת, עליך להוסיף קוד כדי להעביר את פרת משה רבנו למיקום אקראי. (תחשוב על מה יקרה אם לא תזיז את

פרת משה רבנו בתחילת משחק חדש. איפה זה יהיה ביחס לצפרדע?)
איור 5-18מציג את הבלוקים להזזת פרת משה רבנו כאשר המשחק מתחיל מחדש.



איור .5-19הגרסה הסופית של RestartButton.Click

איך הבלוקים עובדים

ההבדל היחיד בין גרסה זו של RestartButton.Clickלגרסה הקודמת הוא בלוק Ladybug.MoveToוהטיעונים שלו. הפונקציה המובנית מספר שלם אקראי נקראת פעמיים: פעם אחת כדי ליצור קואורדינטת xחוקית ופעם כדי ליצור קואורדינטת yחוקית. אף על פי שאין דבר שימנע את הנחת פרת משה רבנו על גבי הכנימה או הצפרדע, הסיכויים נגדו.

בדוק את האפליקציה שלך הפעל מחדש את המשחק וודא שהפרת משה רבנו מופיעה במיקום אקראי חדש.

הוספת אפקטים קוליים

כשבדקת את המשחק, אולי שמת לב שאין משוב טוב במיוחד כשאוכלים משהו. כדי להוסיף השפעות קול ומשוב מישוש, בצע את הפעולות הבאות:

,Component Designer-ב .1הוסף רכיב סאונד . הגדר את המקור שלו ל-
קובץ קול שהעלית.

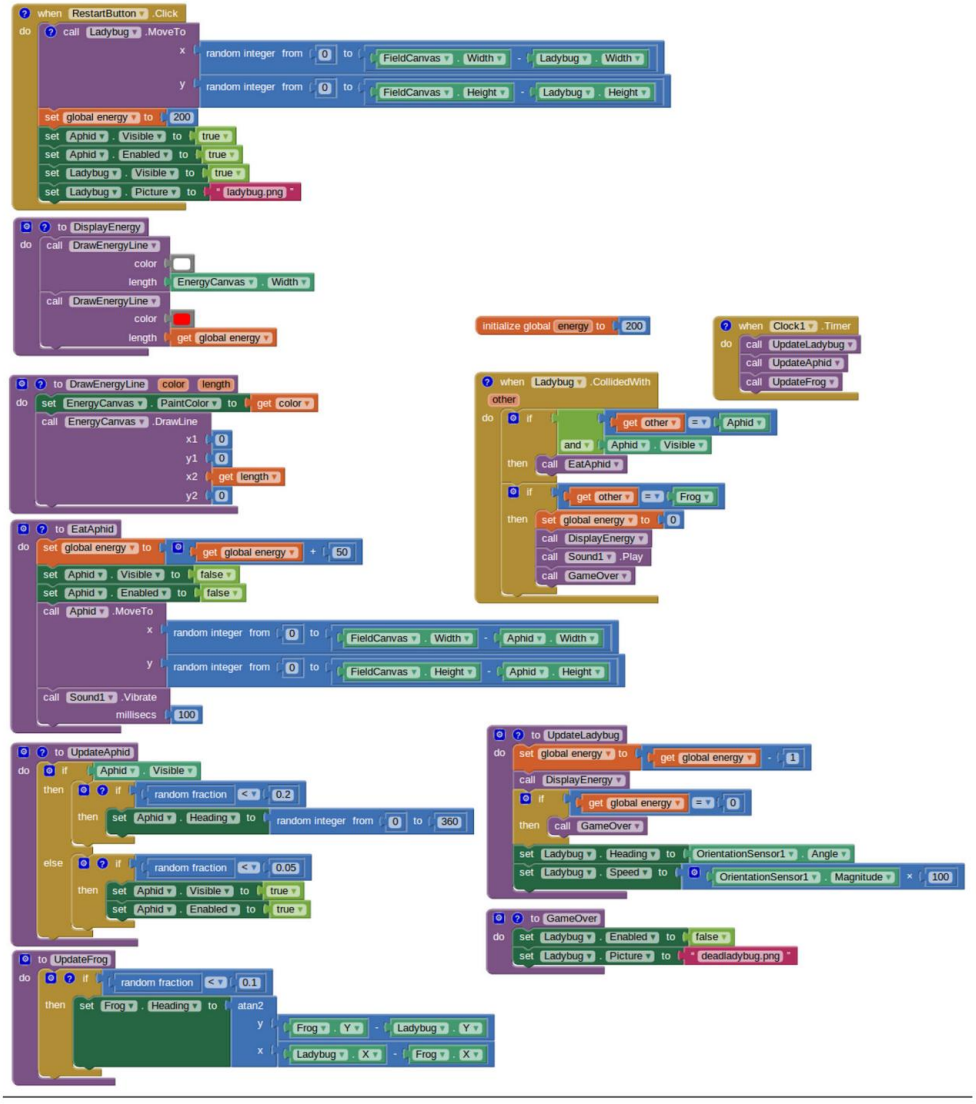
.2עבור אל עורך הבלוקים, שם תוכל:

גרמו למכשיר לרטוט כשאוכלים כנימה על ידי הוספת א
Sound1.Blockרטט עם ארגומנט של 100(מילישניות) ב- .EatAphid

צור את צפרדע הצפרדע כשהיא אוכלת את פרת משה רבנו על ידי הוספת קריאה אליו
Sound1.Playב Ladybug.CollidedWithממש לפני הקריאה ל- .GameOver

האפליקציה השלמה: Ladybug Chase

איור 5-19מציג את תצורת הבלוק הסופי עבור .Ladybug Chase



איור 5-20. אפליקציית Ladybug Chase המלאה

וריאציות

הנה כמה רעיונות כיצד לשפר או להתאים אישית את המשחק הזה:

- נכון לעכשיו, הצפרדע והכנימה ממשיכות לנוע לאחר שהמשחק הסתיים. מנע זאת על ידי הגדרת המאפיינים הפעילים שלהם ל- false ב- GameOver ו- in eurt- RestartButton.Click.

•הצג ניקוד המציין כמה זמן פרת משה רבנו נשארה בחיים. אתה יכול לעשות זאת על ידי יצירת תווית שתגדיל ב- Clock1.Timer.

•הפוך את סרגל האנרגיה לגלוי יותר על ידי הגדלת הגובה של EnergyCanvasל-2 וציור שני קווים, אחד מעל השני, ב- DrawEnergyLine.(זהו יתרון נוסף של קיום הליך במקום קוד משוכפל למחיקת קו האנרגיה ולציור מחדש: אתה רק צריך לבצע שינוי במקום אחד כדי לשנות את הגודל -או הצבע, או המיקום -של הקו.)

•הוסף אווירה עם תמונת רקע ועוד אפקטים קוליים, כגון צלילי טבע או אזהרה כאשר רמת האנרגיה של פרת משה רבנו הופכת נמוכה.

•האם המשחק נעשה קשה יותר עם הזמן, כמו על ידי הגברת המהירות של הצפרדע מאפיין או הקטנת מאפיין ה-lavretnI שלו.

•מבחינה טכנית, פרת משה רבנו אמורה להיעלם כאשר היא נאכלת על ידי הצפרדע. שנה את המשחק כך שהפרת משה רבנו תהפוך לבלתי נראית אם היא נאכלת על ידי הצפרדע אבל לא אם היא גוועה ברעב.

•החלף את תמונות פרת משה רבנו, הכנימה והצפרדע בתמונות יותר לטעמכם, כמו הוביטים, אורקים ומכשף מרושע או לוחם כוכבים מורד, תרמיל אנרגיה ולוחם כוכבים אימפריאלי.

סיכום

עם שני משחקים עכשיו תחת החגורה שלך (אם השלמת את המדריך של ,(MoleMash אתה יודע עכשיו איך ליצור משחקים משלך, שזו המטרה של הרבה מתכנתים חדשים או !wannabesבאופן ספציפי, למדת:

•אתה יכול לקבל מספר רכיבי ImageSprite(פרת משה רבנו, הכנימה והצפרדע) ויכולים לזהות התנגשויות ביניהם.

•יכול לזהות את הטיית המכשיר ואתה יכול להשתמש בערך זה כדי OrientationSensor לשלוט בתנועה של ספרייט (או כל דבר אחר שאתה יכול לדמיין).

•רכיב שעון בודד יכול לשלוט במספר אירועים המתרחשים באותו תדירות (שינויים בכיוונים של פרת משה רבנו וצפרדע), או בתדרים שונים, באמצעות בלוק השבר האקראי . לדוגמה, אם אתה רוצה שאירוע יתרחש כרבע 25)אחוז) מהזמן, הכנס אותו לגוף של בלוק ifשמתבצע רק כאשר התוצאה של שבר אקראי קטנה מ-52.0.

•אתה יכול לקבל רכיבי Canvasמרובים באפליקציה אחת, מה שעשינו כדי שיהיה לנו מגרש משחק וכדי להציג משתנה בצורה גרפית (במקום דרך תווית).

•ניתן להגדיר נהלים עם פרמטרים (כגון "צבע" ו"אורך" ב- DrawEnergyLine)
השולטים בהתנהגות, מה שמרחיב מאוד את כוח ההפשטה הפרוצדורלית.