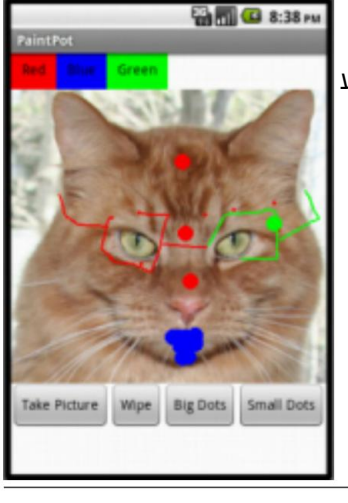
# PaintPot

איור 2-1.

מדריך זה מציג את רכיב Canvas-הליצירת 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, ליצור 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 על זה 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 לזכור דברים כמו גודל הנקודה שהמשתמש בחר עבורו צִיוּר.

# מתחילים

נווט לאתר App Inventor.התחל פרויקט חדש וקרא לו "PaintPot".לחץ על התחבר והגדר את המכשיר (או האמולטור) שלך לבדיקה חיה (ראה appinventor.mit.edu/explore/ai2/setup http://לעזרה בהגדרה).

לאחר מכן, בצד ימין של המעצב, עבור לחלונית המאפיינים ושנה את כותרת המסך "PaintPot"לאתה 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.

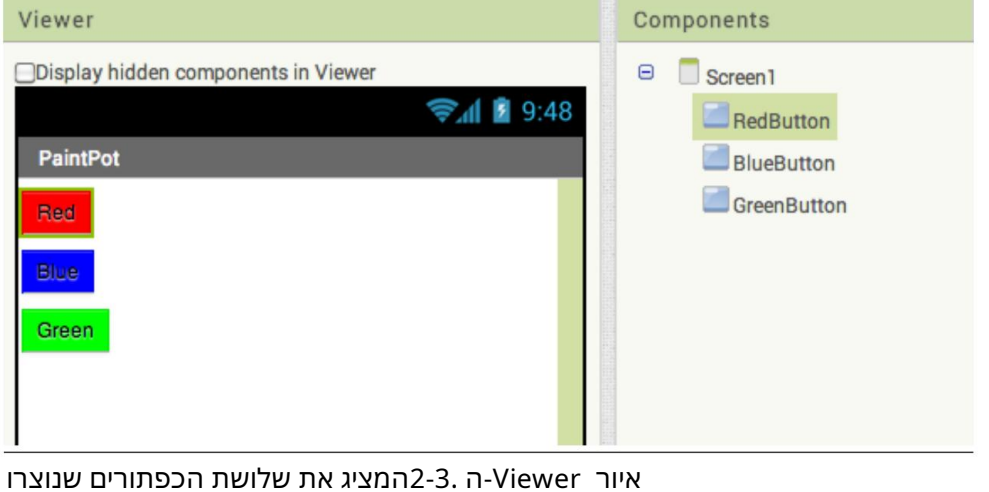
### יצירת כפתורי הצבע

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

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

,Viewer-ב .2ברשימת 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, לכן 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-Viewer

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

שמות משמעותיים הופכים את הפרויקטים שלך לקריאה יותר, וזה באמת יעזור כשאתה עובר לעורך הבלוקים וחייב להתייחס לרכיבים בשם. 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.

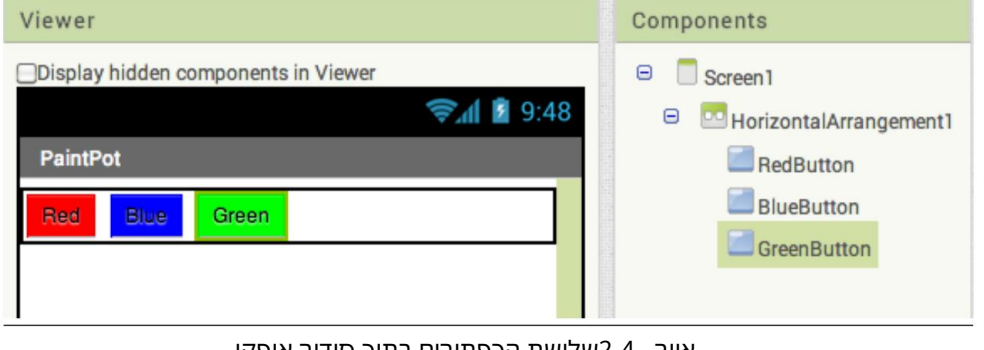
שימוש בהסדרים לפריסות טובות יותר

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 Layout-השל הפלטה, גרור החוצה רכיב HorizontalArrangement והנח אותו מתחת ללחצנים.

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

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 מבולטים תחת הרכיב .Horizontal Arrangementזה מציין שהלחצנים הם כעת רכיבי משנה של רכיב . HorizontalArrangementשים לב שכל הרכיבים מוכנסים מתחת למסך .1

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 .Designer-בלדוגמה, המתאר סביב HorizontalArrangementמופיע Viewer-באך 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

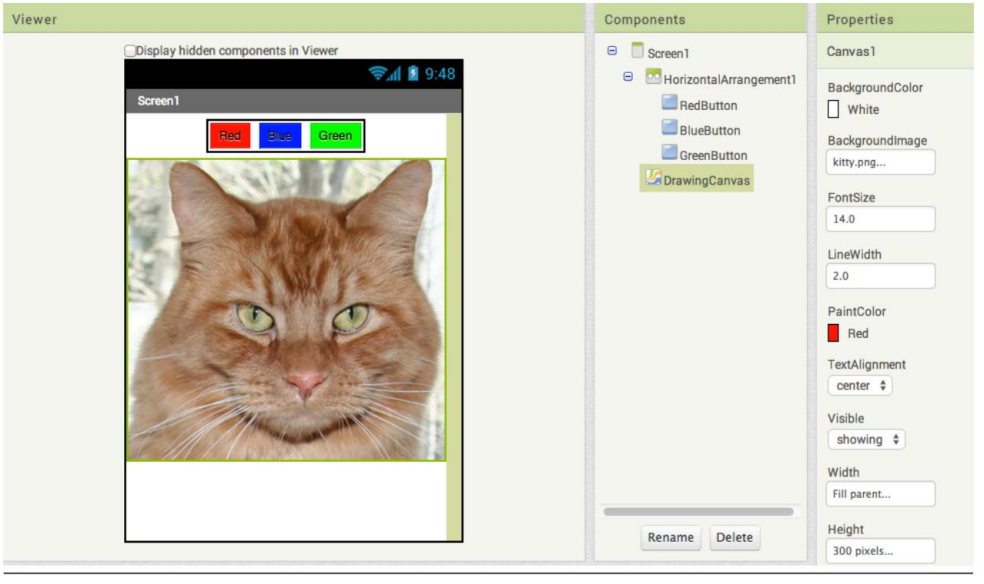
השלב הבא הוא להגדיר את הקנבס שבו יתרחש הציור:

1.ממגירת הציור וההנפשה של הפלטה, גרור רכיב Canvasאל .Viewer-השנה את שמו ל- .DrawingCanvasהגדר 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 ,1כבר הורד את הקובץ .kitty.pngאם לא עשית זאת, תוכל להוריד אותו בכתובת http://appinventor.org/bookFiles/HelloPurr/kitty.png .

3.הגדר את תמונת הרקע של DrawingCanvasלקובץ .kitty.pngבתוך ה בקטע מאפיינים של מעצב הרכיבים, תמונת הרקע תוגדר .None-ללחץ על השדה והעלה את הקובץ kitty.png .

4.הגדר את PaintColorשל DrawingCanvasלאדום 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 2-5. לרכיב DrawingCanvas יש תמונת רקע של תמונת הקיטי

סידור הכפתורים התחתונים ורכיב המצלמה

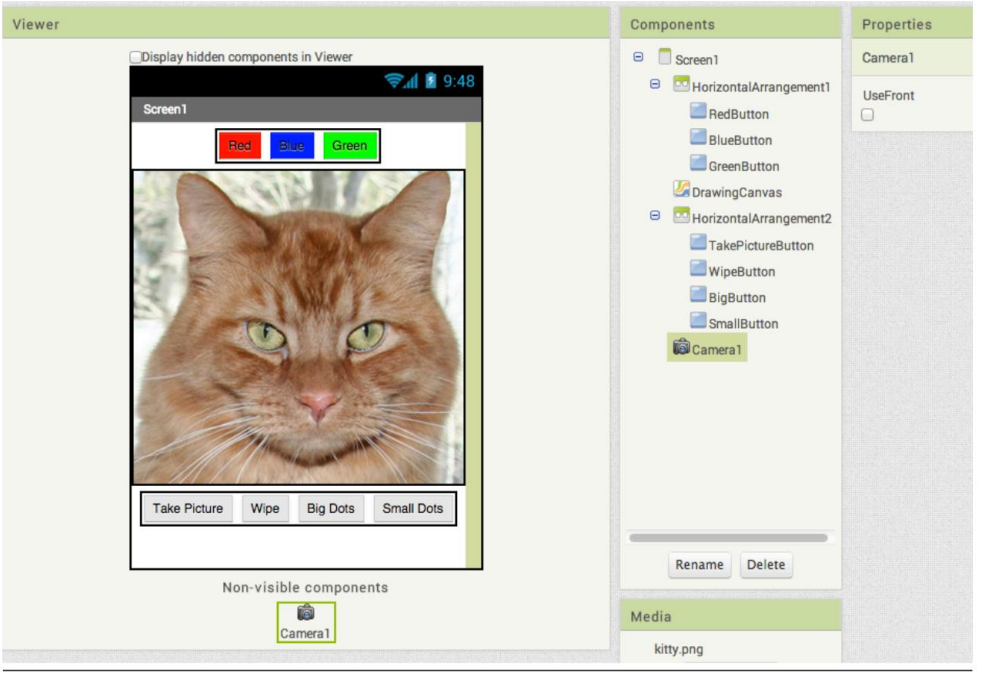
1. מהפלטה, גרור סידור אופקי שני והנח אותו מתחת לקנבס. לאחר מכן, גרור שני רכיבי כפתור נוספים על המסך והצב אותם בסידור האופקי התחתון הזה. שנה את שם הכפתור הראשון ל- TakePictureButton ואת תכונת הטקסט שלו ל- Take Picture. שנה את שם הכפתור השני ל- WipeButton ואת תכונת הטקסט שלו ל- "Wipe".

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

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

4. ממגירת המדיה, גרור רכיב מצלמה אל Viewer-. הזה יהיה מופיעים באזור הרכיבים הלא גלויים.

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




איור .2-6ממשק 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

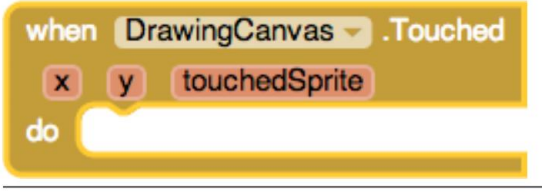
השלב הבא הוא להגדיר כיצד הרכיבים מתנהגים. יצירת תוכנית ציור עשויה להיראות מכריעה, אבל תהיו בטוחים ש-ppA Inventorעשה עבורכם הרבה מהעבודה הכבדה: ישנם בלוקים קלים לשימוש לטיפול בפעולות המגע והגרירה של המשתמש, ולציור וצילום תמונות.

בהוספת רכיב Canvas בשם DrawingCanvas. כמו כל רכיבי Designer, הל-Canvas, יש אירוע נגע ואירוע נגרר . תתכנת את אירוע DrawingCanvas.Touched כך שיצויר עיגול בתגובה לנגיעה של המשתמש fnger-בשלה על המסך. תתכנת את האירוע DrawingCanvas.Draggedכך שקו יצויר כשהמשתמש גורר את Fnger-השלה על פני הבד. לאחר מכן תתכנת את הלחצנים לשנות את צבע הציור, לנקות את הקנבס ולשנות את רקע הקנבס לתמונה שצולמה במצלמה.

הוספת אירוע מגע כדי לצייר נקודה

ראשית, תסדר את הדברים כך שכאשר אתה נוגע ,DrawingCanvas-בתצייר נקודה בנקודת המגע:

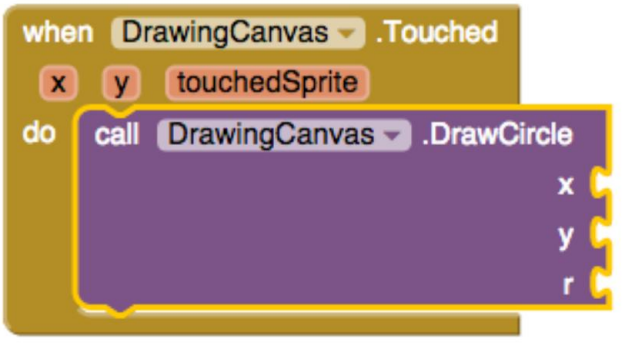
1. בעורך בלוקים, בחר את המגירה עבור DrawingCanvasולאחר מכן גרור את הבלוק DrawingCanvas.Touchedאל סביבת העבודה. לבלוק יש פרמטרים עבור y ,x-ו touchedSprite,כפי שמוצג באיור .2-6פרמטרים אלה מספקים מידע על מיקום המגע.




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

הערה אם השלמת את אפליקציית HelloPurrבפרק ,1אתה מכיר את אירועי , Button.Clickאך לא עם אירועי .Canvasאירועי 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 צורת הקנבס (ב-App Inventor)זה נקרא ספרייט) היה נגע, אבל לא נצטרך את זה עד פרק .3קואורדינטות x-ו y-הן הארגומנטים שבהם נשתמש כדי לזהות היכן המשתמש נגע במסך. אז נוכל לצייר את הנקודה על זה עמדה.

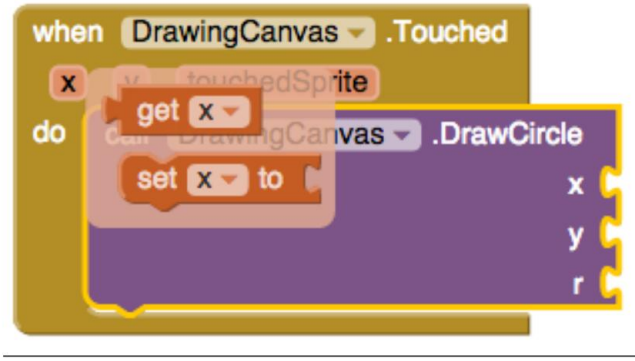
2. מהמגירה ,DrawingCanvasגרור החוצה פקודת DrawingCanvas.DrawCircle והצב אותה בתוך המטפל באירוע , DrawingCanvas.Touchedכפי שמוצג באיור .2-7




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

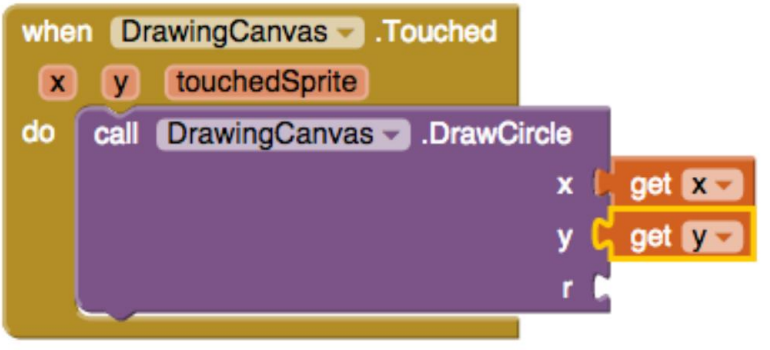
בצד ימין של בלוק DrawingCanvas.DrawCircle, תראה שלושה שקעים עבור הארגומנטים שאנו צריכים לחבר: x, y ו-r. הארגומנטים x ו-y מציינים את המיקום שבו יש לצייר את העיגול, ו-r קובע את הרדיוס (או הגודל) של המעגל. מטפל באירועים זה יכול להיות מעט מבלבל מכיוון שלאירוע DrawingCanvas.Touched יש גם x ו-y; ארק זכור שה-x ו-y מציינים היכן המשתמש נגע, ועבור האירוע DrawingCanvas.Touched בעוד שה-x ו-y ועבור 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 בתור הערכים לשימוש עבור הפרמטרים x ו-y בב- 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

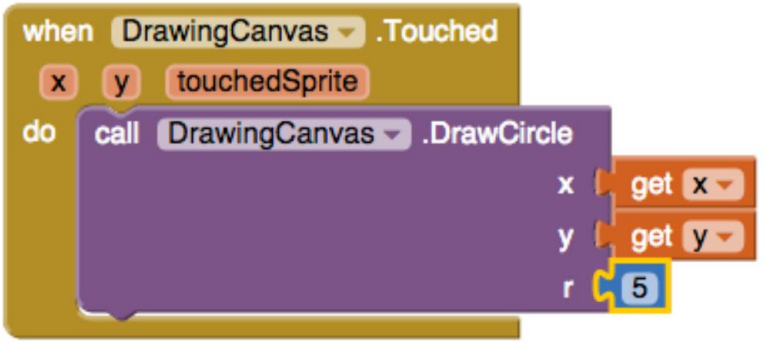
3. גרור הוצאת בלוקים עבור ערכי xו- yוחבר אותם לשקעים פנימה בלוק , DrawingCanvas.DrawCircleכפי שמוצג באיור 2-9.




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 ייצור 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 הבלוקים 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 ,DrawingCanvas-במעגל ברדיוס 5מצויר במיקום המגע (x,y)

בדוק את האפליקציה שלך נסה את מה שיש לך עד כה במכשיר.
כשאתה נוגע ה-Fnger ,DrawingCanvas-בשלך צריך להשאיר נקודה בכל מקום בו אתה נוגע. הנקודות יהיו אדומות אם תגדיר את המאפיין DrawingCanvas.PaintColor לאדום ב-Component Designer (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 DrawingCanvasולאחר מכן מרים אותו מבלי להזיז אותו.

•גרירה היא כאשר אתה מניח את האצבע שלך על DrawingCanvasומזיז אותו מסביב תוך שמירה על מגע עם המסך.

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

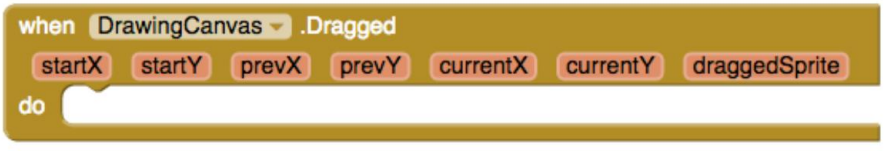
1. מהמגירה ,DrawingCanvasגרור את הבלוק DrawingCanvas.Draggedאל סביבת העבודה. אתה אמור לראות את המטפל באירועים כפי שהוא מוצג באיור 2-11.
האירוע DrawingCanvas.Draggedמגיע עם הטיעונים הבאים:

startX, startY: מיקום האצבע בנקודה שבה הגרירה התחיל.

currentX, currentY: המיקום הנוכחי של האצבע שלך

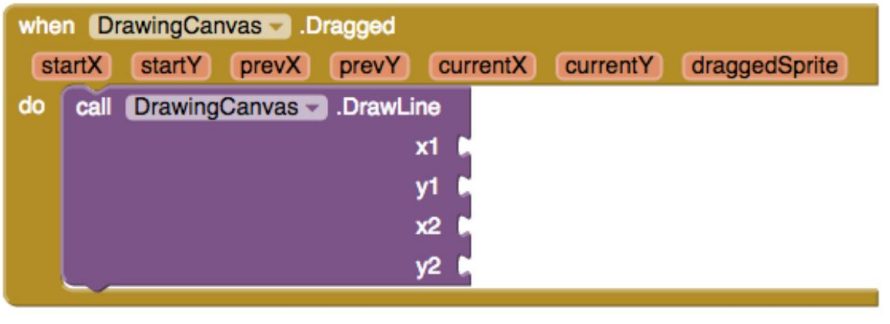
prevX, prevY: המיקום הקודם של האצבע שלך.

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 Touched-ב

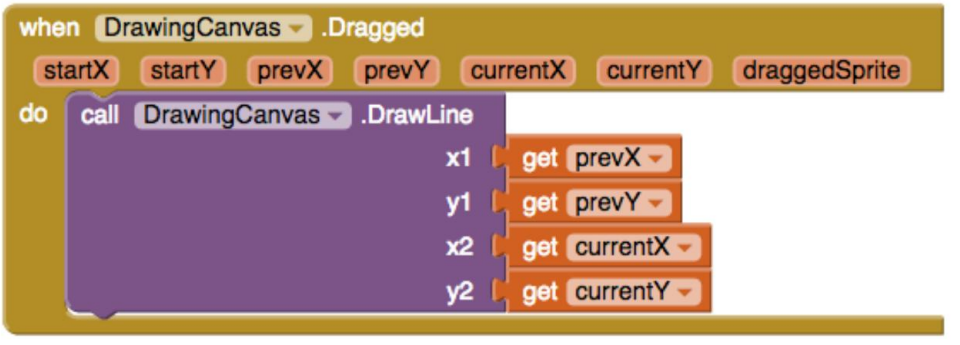
2. מהמגירה DrawingCanvas, גרור את הבלוק DrawingCanvas.DrawLine לתוך הבלוק DrawingCanvas.Dragged, כפי שמוצג באיור 2-12.




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 Fnger-השלך 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. A get prevX וקבל prevY צריך להיות מחובר לשקעי 1x-1y, בהתאמה. יש לחבר את get currentX-הו- get currentY לשקעי 2x-2y, בהתאמה, כפי שמוצג באיור 2-13.



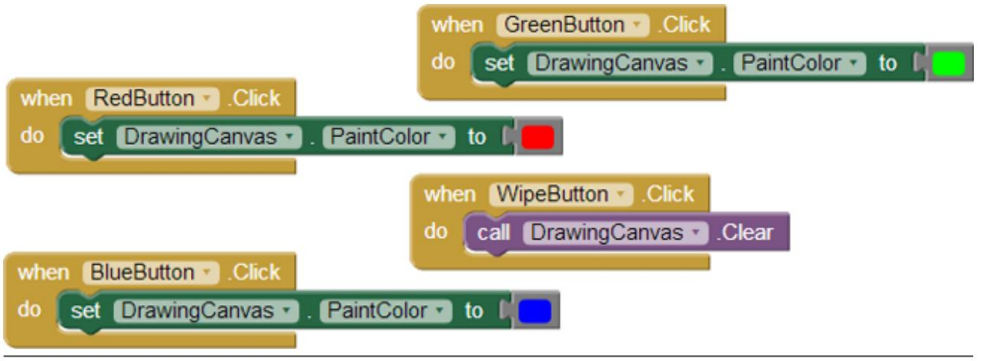

איור 2-14. כשהמשתמש גורר, האפליקציה תשרטט קו מהנקודה הקודמת לנקודה הנוכחית

בדוק את האפליקציה שלך נסה התנהגות זו במכשיר. גרור את Fnger-השלך 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 WipeButton-הכדי 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 של 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 WipeBepiWottuN. לחץ מהמגירה WipeButton, גרור החוצה DrawingCanvas מהמגירה והנח אותו בבלוק WipeButton.Click. ודא שהבלוקים שלך DrawingCanvas.Clear 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 יש שני בלוקים מרכזיים. בלוק Camera.TakePicture מפעיל את אפליקציית המצלמה במכשיר. האירוע Camera.AfterPicture מופעל כאשר המשתמש סיים לצלם את התמונה. תוסיף בלוקים במטפל האירוע Camera.AfterPicture כדי להגדיר את DrawingCanvas.BackgroundImage לתמונה שהמשתמש צילם זה עתה.

1. פתח את מגירת TakePictureButton וגרור את המטפל באירוע TakePictureButton.Click אל סביבת העבודה.

2. מתוך Camera1, גרור את Camera1.TakePicture והצב אותה ב- TakePictureButton.Click מטפל באירועים.

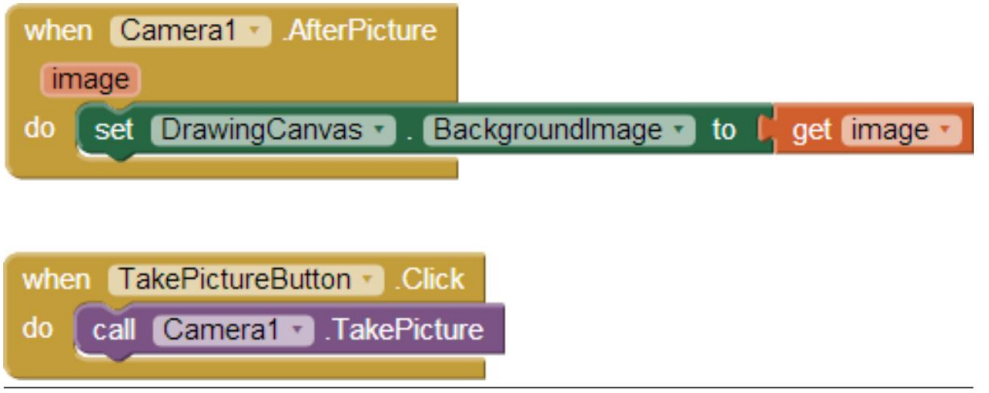
3. מתוך Camera1, גרור את המטפל באירועים Camera1.AfterPicture לתוך סביבת עבודה.

4. מתוך DrawingCanvas, גרור את הסט DrawingCanvas.BackgroundImage לחסימה והצב 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

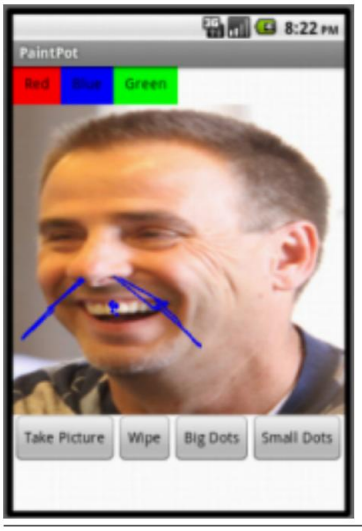
בדוק את האפליקציה שלך נסה התנהגות זו על ידי לחיצה על צלם תמונה במכשיר שלך וצילום תמונה. התמונה של החתול צריכה להשתנות לתמונה שזה עתה צילמת, ואז תוכל לצייר על התמונה הזו. (ציור על פרופסור וולבר הוא הבילוי המועדף על תלמידיו, כפי שמוצג באיור 2-16.)

שינוי גודל הנקודה

גודל הנקודות המצוירות על DrawingCanvasנקבע 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 שתגדירו 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-"dotSize".

2. שים לב שלתחול 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 לחץ על Return) ולאחר מכן חבר אותו לאתחל dotSize to, כפי שמוצג באיור 2-17.


איור 2-18. אתחול המשתנה dotSize בערך 2

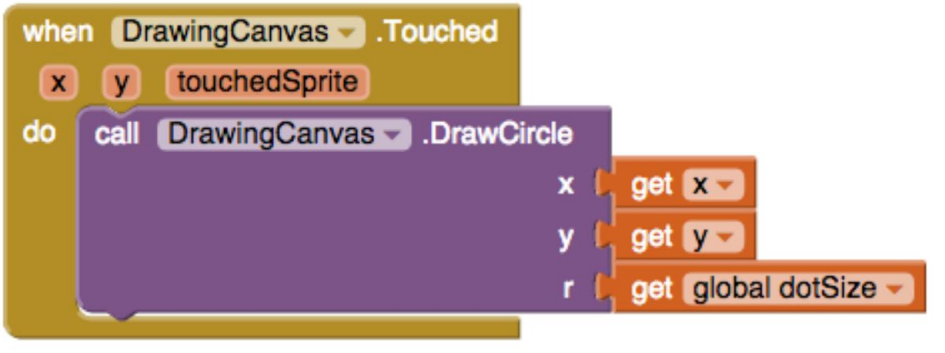
התייחסות למשתנה DOTSIZE- הבמעגל השרטוט

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 אולי נראה כאילו "שיפכנו" 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 5מחוץ לחריץ rולאחר מכן הנח אותו לפח. לאחר מכן, החלף אותו בגוש get global dotSize (ראה איור .(18-2כאשר המשתמש נוגע ,DrawingCanvas-בהאפליקציה תקבע כעת את הרדיוס מהמשתנה .dotSize




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

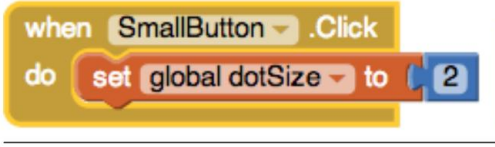
שינוי הערך של DOTSIZE

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

1.גרור החוצה מטפל באירוע SmallButton.Clickמהמגירה של .SmallButton לאחר מכן, העבר את העכבר מעל "גודל הנקודות" בתוך הבלוק הגלובלי של האתחול וגרור החוצה את ה- dotSizeהגלובלי המוגדר לחסימה. חבר אותו ל- SmallButton.Clickלבסוף, 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-App Inventor, אתה יכול גם להגדיר משתנים שהם "מקומיים" לחלק מסוים של התוכנית (ראה פרק 21לפרטים).

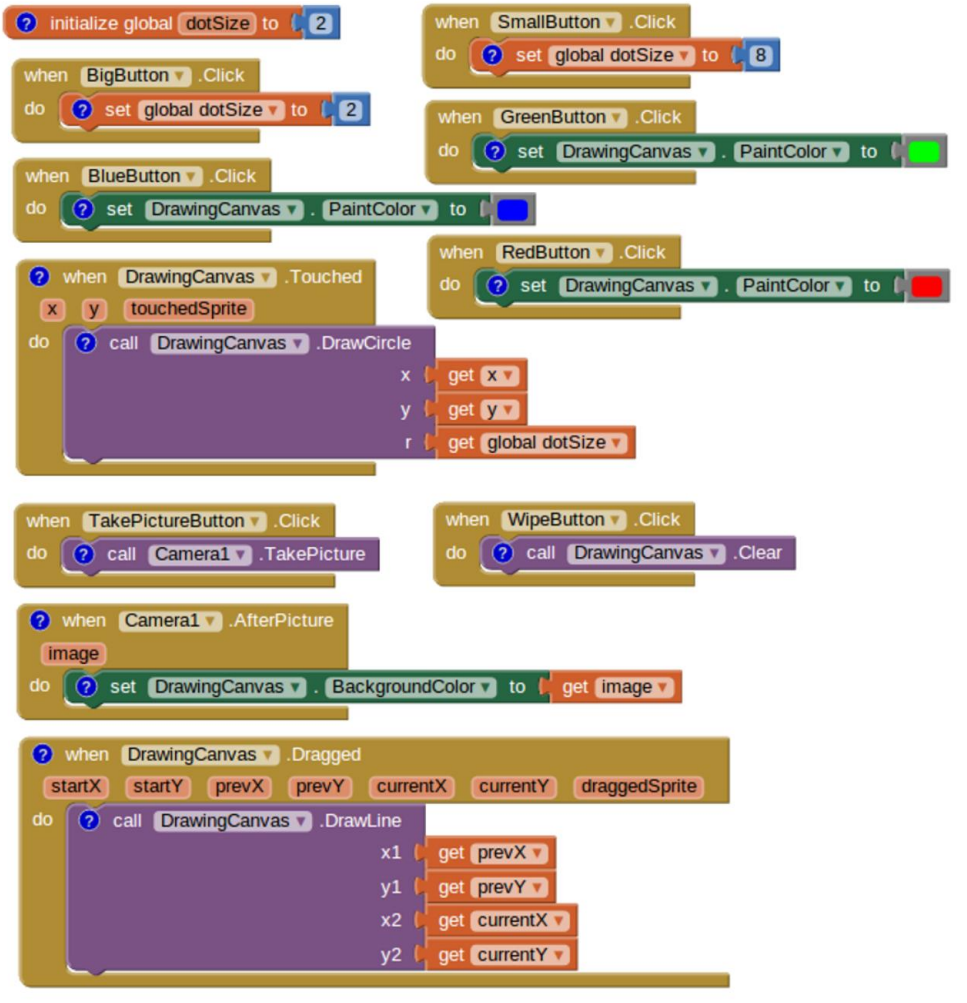



איור 2-20. לחיצה על הכפתורים משנה את גודל הנקודה; נגיעות לאחר מכן יציירו בגודל זה

בדוק את האפליקציה שלך נסה ללחוץ על לחצני הגודל ולאחר מכן לגעת בהאם העיגולים מצוירים בגדלים שונים? DrawingCanvas. 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 .21-2ערכת הבלוקים הסופית עבור PaintPot

# וריאציות

להלן 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 אלו יוצגו ב-

מִשׁתַמֵשׁ.

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

# סיכום

הנה כמה מהרעיונות שכיסינו בפרק זה:

•רכיב Canvas-המאפשר 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 רק למקם אותם אחד מתחת לשני.

•מטפלי אירועים מסוימים מגיעים עם מידע על האירוע, כגון
קואורדינטות של המקום בו נגעה 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 באופן אוטומטי הפניה גלובלית של getהמעניקה 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.