יצירת אפליקציות מונפשות

פרק זה דן בשיטות ליצירת אפליקציות עם אנימציות פשוטות (אובייקטים זזים).


איור 17-1

תלמד את היסודות של יצירת משחקים דו-מימדיים עם App Inventor ותהיה נוח עם ספרייטים של תמונה וטיפול באירועים כגון שני אובייקטים מתנגשים.

כאשר אתה רואה אובייקט נע בצורה חלקה לאורך מסך המחשב, מה שאתה באמת רואה הוא רצף מהיר של תמונות עם האובייקט במקום מעט שונה בכל פעם. זוהי אשליה שאינה שונה ברמה מ-skoobpif, שבהם אתה רואה תמונה נעה על ידי דפדוף מהיר בין הדפים. זה הקונספט מאחורי האופן שבו מיוצרים פלמס מונפש!

עם App Inventor, תתכנת אנימציה על ידי הצבת רכיבי Ball ו- ImageSprite בתוך רכיב Canvas ולאחר מכן הזזה ושינוי של אובייקטים אלה בכל שבריר שניה עוקב. בפרק זה תלמדו כיצד פועלת מערכת הקואורדינטות של Canvas, כיצד ניתן להשתמש באירוע Clock.Timer כדי להפעיל תנועה, כיצד לשלוט במהירות של אובייקטים וכיצד להגיב לאירועים כגון שני אובייקטים המתרסקים זה בזה. .

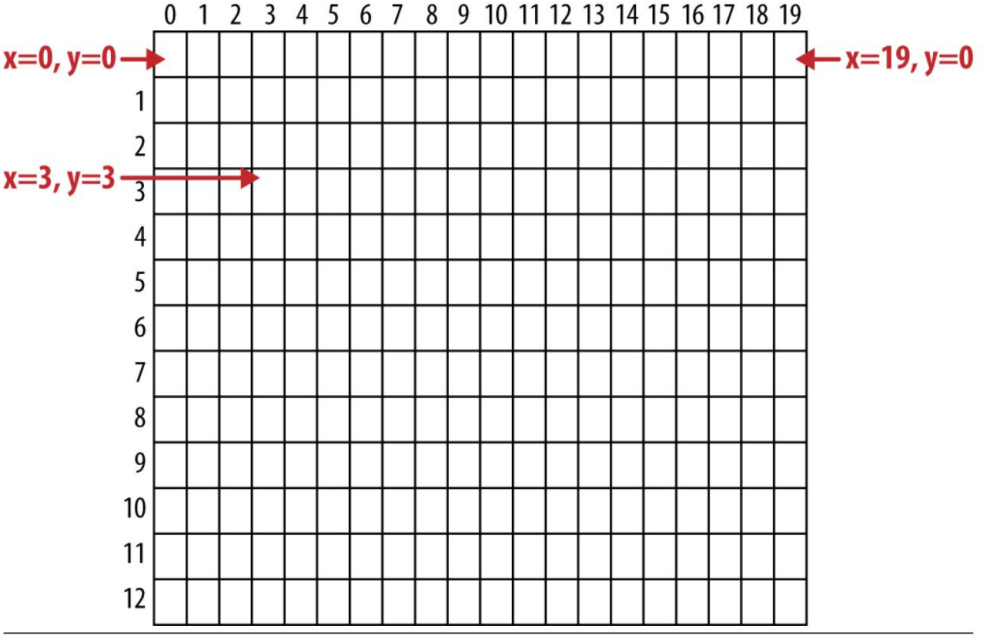
הוספת רכיב קנבס לאפליקציה שלך

מלוח הציור וההנפשה, גרור רכיב Canvas לתוך האפליקציה שלך.
לאחר שתמקם אותו, ציין את הרוחב והגובה שלו . לעתים קרובות, תרצה שה- Canvas יתפרש על רוחב מסך המכשיר. כדי לעשות זאת, בחר "מלא הורה" בעת ציון הרוחב.

אתה יכול לעשות את אותו הדבר עבור הגובה, אבל בדרך כלל תגדיר את זה למספר מסוים (למשל, 300 פיקסלים) כדי להשאיר מקום לרכיבים אחרים מעל ומתחת לקנבס.

מערכת קואורדינטות הקנבס

ציור על קנבס הוא באמת רשת של פיקסלים, כאשר פיקסל הוא נקודת הצבע הקטנה ביותר שיכולה להופיע על המסך. מיקום כל פיקסל מוגדר על ידי קואורדינטות xyבמערכת רשת, כפי שמוצג באיור .17-1במערכת קואורדינטות זו, xמגדיר מיקום במישור האופקי (מתחיל ב-0 בקצה השמאלי והולך וגדל ככל שעוברים ימינה על פני המסך), y-ומגדיר מיקום במישור האנכי (החל מ-0 בחלק העליון וגדל ככל שאתה מטה את המסך).

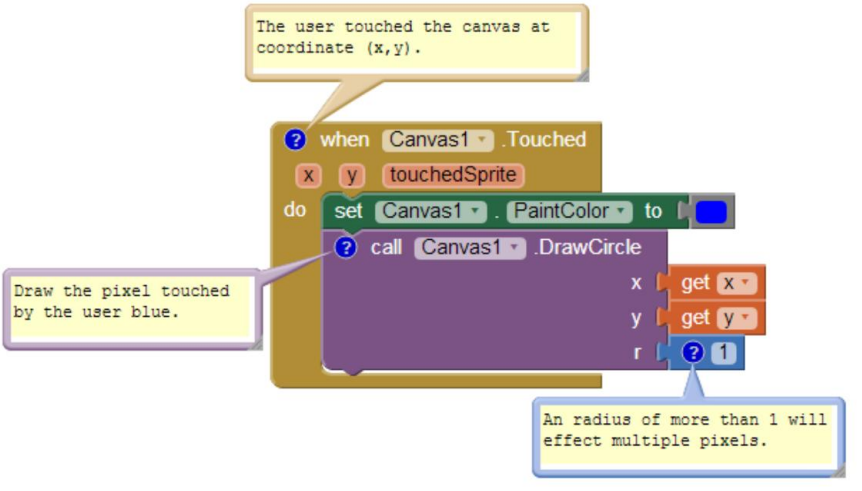


איור .17-2מערכת הקואורדינטות של קנבס

התא השמאלי העליון ב- Canvasמתחיל ב-0 עבור שתי הקואורדינטות, כך שהמיקום הזה מיוצג .(x=0,y=0)-כככל שאתה זז ימינה, קואורדינטת ה-x גדלה; ככל שאתה זז למטה, קואורדינטת y-הגדלה. התא מימין מיד לפינה השמאלית העליונה הוא .(x=1,y=0)לפינה הימנית העליונה יש קואורדינטת xהשווה לרוחב ה- Canvasמינוס .1לרוב מסכי הטלפון יש רוחב בסביבות 300פיקסלים, אך עבור המדגם המוצג כאן, הרוחב הוא ,20כך שהפינה הימנית העליונה היא הקואורדינטה .(x=19,y=0)

ניתן לשנות את מראה הקנבס בשתי דרכים: על ידי ציור עליו, או על ידי הצבה והזזה של אובייקטים בתוכו. פרק זה מתמקד בעיקר באחרון, אך בוא נדון קודם באיך אתה "מצייר" וכיצד ליצור אנימציה על ידי ציור (זהו גם הנושא של אפליקציית PaintPotבפרק .(2

כל תא של ה- Canvasמכיל פיקסל המגדיר את הצבע שאמור להופיע שם.
רכיב ה- Canvasמספק את בלוקים Canvas.DrawLine -וCanvas.DrawCircle לצביעת פיקסלים. תחילה עליך להגדיר את המאפיין Canvas.PaintColor לצבע הרצוי ולאחר מכן לקרוא לאחד מבלוקים Drawכדי לצייר בצבע זה. עם DrawCircle, אתה יכול לצבוע עיגולים בכל רדיוס, אבל אם תגדיר את הרדיוס ל-1, כפי שמוצג באיור 17-2, תצבע פיקסל בודד.



איור 17-3. DrawCircle עם רדיוס של 1 צובע פיקסל בודד בכל מגע

App Inventorמספק פלטת צבעים בסיסיים שבהם אתה יכול להשתמש כדי לצבוע פיקסלים (או רכיבי ממשק משתמש אחרים). אתה יכול לגשת למגוון רחב יותר של צבעים באמצעות ערכת מספור הצבעים המוסברת בתיעוד App Inventorבכתובת colors.html. http:// appinventor.mit.edu/explore/ai2/support/blocks/
מלבד צביעת פיקסלים בודדים, אתה יכול גם למקם רכיבי Ball -וImageSprite על קנבס. ספרייט הוא אובייקט גרפי הממוקם בתוך סצנה גדולה יותר, App Inventor-ב)ה"סצנה" היא רכיב Canvas(. שני הרכיבים Ball -וImageSprite הם ספרייטים; הם שונים רק במראה החיצוני. כדור הוא עיגול בעל מראה שניתן לשנות רק על ידי שינוי הצבע או הרדיוס שלו, בעוד ש- ImageSpriteיכול לקבל כל מראה, כפי שמוגדר על ידי דמות תמונה. ניתן להוסיף כדורים ו- ImageSpritesרק בתוך קנבס; אתה לא יכול לגרור אותם לממשק המשתמש מחוץ לממשק המשתמש.

הנפשת אובייקטים עם אירועי טיימר

אחת הדרכים לציין אנימציה בהיא לשנות אובייקט בתגובה לאירוע טיימר. לרוב, תעביר App Inventor- ספרייטים למיקומים שונים על הבד במרווחי זמן מוגדרים. שימוש באירועי טיימר היא השיטה הנפוצה ביותר להגדרת מרווחי הזמן המוגדרים. מאוחר יותר, נדון גם בשיטה חלופית לתכנות אנימציה באמצעות מאפייני המהירות והכותרת של רכיבי ImageSprite- וBall .

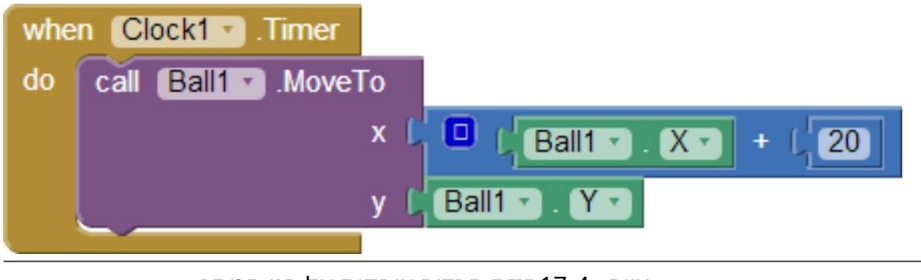
לחיצות על כפתורים ואירועים אחרים ביוזמת המשתמש הם פשוטים להבנה: המשתמש עושה משהו, והאפליקציה מגיבה בביצוע פעולות מסוימות. עם זאת, אירועי טיימר שונים מכיוון שהם אינם מופעלים על ידי משתמש הקצה אלא על ידי הזמן החולף. אתה צריך להמשיג את השעון של הטלפון המפעיל אירועים באפליקציה במקום שמשתמש יעשה משהו.

כדי להגדיר אירוע טיימר, תחילה תגרור רכיב שעון לתוך האפליקציה שלך בתוך מעצב הרכיבים. לרכיב השעון יש מאפיין TimerInterval המשויך אליו. המרווח מוגדר באלפיות שניות (1/1,000 של שנייה). אם תגדיר את TimerInterval ל-005, זה אומר שאירוע טיימר יופעל כל חצי שנייה.

ככל TimerInterval- שקטן יותר, כך קצב הפריימים של האנימציה מהיר יותר.
לאחר הוספת שעון והגדרת TimerInterval ב-Designer, תוכל לגרור אירוע Clock.Timer בעורך הבלוקים. אתה יכול לשים כל בלוקים שאתה אוהב באירוע הזה, והם יבוצעו בכל מרווח.

יצירת תנועה

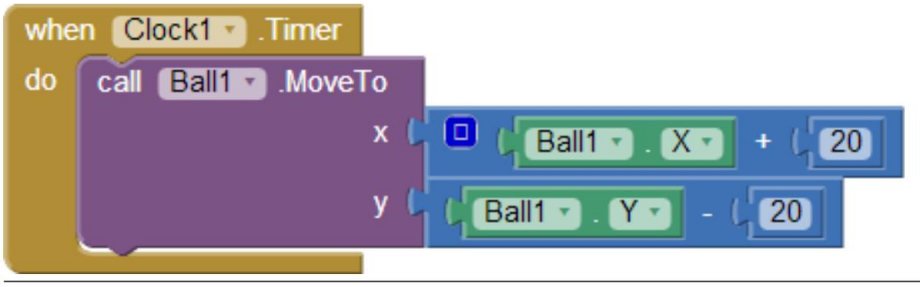
כדי להציג ספרייט שזז לאורך זמן, תשתמש בפונקציה MoveTo שנמצאת ברכיבים ImageSprite- וBall כאחד . לדוגמה, כדי להזיז כדור אופקית על פני המסך, תשתמש בלוקים כמו אלה באיור 17-3.



איור 17-4. הזזת הכדור אופקית על פני המסך

MoveTo מעביר אובייקט למיקום מוחלט על הבד, לא לכמות יחסית. אז כדי להזיז אובייקט מסוים, אתה מגדיר את הארגומנטים MoveTo- ל

המיקום הנוכחי של האובייקט בתוספת היסט. מכיוון שאנו נעים אופקית, הארגומנט x מוגדר למיקום xהנוכחי (Ball1.X)בתוספת הסט ,20ואילו הארגומנט yמוגדר להישאר בהגדרה הנוכחית שלו .(Ball1.Y)
כדי להזיז אובייקט למטה, תשנה רק את קואורדינטת Ball1.Yולהשאיר את Ball1.X זהה. אם תרצו להזיז את הכדור באלכסון, הייתם מוסיפים היסט לקואורדינטות ה-x וה-y, כפי שמוצג באיור .17-4



איור .17-5הקיזת קואורדינטות xו- yכדי להזיז את הכדור באלכסון

מְהִירוּת

כמה מהר הכדור זז בדוגמה הקודמת? המהירות תלויה הן בהגדרות שאתה מספק עבור המאפיין TimerIntervalשל Clock1והן בפרמטרים שאתה מציין בפונקציה . MoveToאם ה- Clock.TimerIntervalמוגדר ל-000,1 אלפיות השנייה, זה אומר שאירוע Clock1.Timer יופעל בכל שנייה. עבור הדוגמה האופקית המוצגת באיור ,17-3הכדור יזוז 20פיקסלים לשנייה.

אבל TimerIntervalשל 1,000אלפיות השנייה לא מספק אנימציה חלקה במיוחד; הכדור יזוז רק פעם בשנייה, מה שייראה קופצני. כדי להשיג תנועה חלקה יותר, אתה צריך מרווח קצר יותר. אם ה- TimerIntervalהיה מוגדר במקום זאת ל-001 מילישניות, הכדור היה זז 20 פיקסלים כל עשירית שנייה, או 200פיקסלים לשנייה -קצב שייראה חלק הרבה יותר.

זיהוי התנגשות

כדי ליצור משחקים ואפליקציות מונפשות אחרות, אתה צריך פונקציונליות מורכבת יותר מאשר רק תנועה. למרבה המזל, App Inventorמספק כמה בלוקים ברמה גבוהה להתמודדות עם אירועי אנימציה כמו אובייקט שמגיע לקצה המסך או שני אובייקטים מתנגשים.

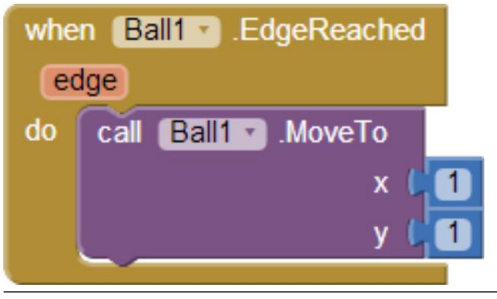
בהקשר זה, חסימה ברמה גבוהה פירושה ש-ppA Inventorדואג לפרטים ברמה נמוכה יותר של קביעה מתי מתרחש אירוע כגון התנגשות. אתה יכול לבדוק את ההתרחשויות האלה באופן ידני על ידי בדיקת מאפייני ספרייט וקנבס בתוך אירועי . Clock.Timerרמה זו של תכנות דורשת היגיון מורכב למדי,

למרות זאת. למרבה המזל, App Inventorמספק אותם עבורכם, וטוב שכך כי האירועים הללו משותפים להרבה משחקים ואפליקציות אחרות.

EdgeReached

שקול שוב את האנימציה באיור 17-4, שבה האובייקט נע באלכסון מהצד השמאלי העליון לימין התחתון של הבד. כפי שתוכנת, הכדור יזוז באלכסון ואז נעצר כשהגיע לקצה הימני או התחתון של הבד (המערכת לא תזיז אובייקט מעבר לגבולות הבד).

אם במקום זאת רצית שהאובייקט יופיע שוב בפינה השמאלית העליונה בכל פעם שהוא הגיע לקצה התחתון או הימני, תוכל להגדיר תגובה לאירוע Ball.EdgeReachedבדומה לזה שמוצג באיור 17-5.



איור 17-6. גורם לכדור להופיע שוב בפינה השמאלית העליונה כאשר הוא מגיע לקצה

EdgeReachedמופעל כאשר כדור או ספרייט תמונה פוגעים בכל קצה של קנבס. מטפל באירועים זה, בשילוב עם התנועה האלכסונית שצוינה עם אירוע הטיימר שתואר קודם לכן (איור 17-4), גורם לכדור לנוע באלכסון משמאל למעלה לימין למטה, לקפוץ חזרה למעלה לשמאל העליון כשהוא מגיע לקצה, ו ואז לעשות הכל שוב, לנצח (או עד שהאפליקציה תגיד לה אחרת).

בדוגמה זו, לא הבחנו בין איזה קצה נפגע. לאירוע EdgeReachedיש פרמטר, edge, המציין את הקצה המסוים באמצעות הקוד הבא:

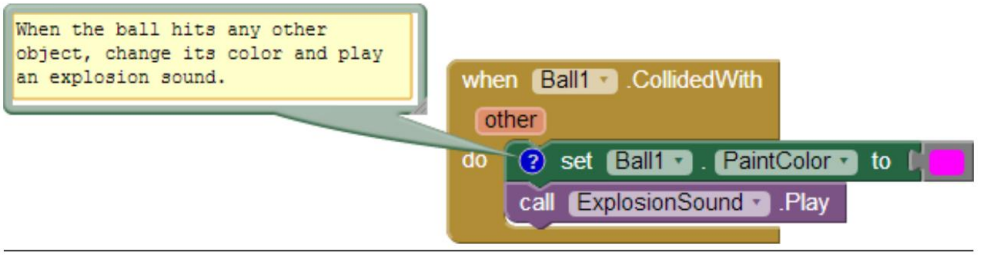
- צפון 1 =
- צפון מזרח 2 =
- מזרח 3 =
- דרום מזרח 4 =
- דרום 1– =
- דרום מערב 2– =

•מערב 3– =

•צפון מערב 4– =

מתנגש עם No LongerCollidingWith-ו

משחקים ואפליקציות מונפשות אחרות מסתמכות לרוב על פעילות המתרחשת כאשר שני עצמים או יותר מתנגשים (למשל, מחבט שפוגע בכדור).
חשבו על משחק, למשל, שבו אובייקט משנה צבעים ומשמיע צליל פיצוץ כשהוא פוגע באובייקט אחר. איור 17-6מציג את הבלוקים עבור מטפל באירוע כזה.

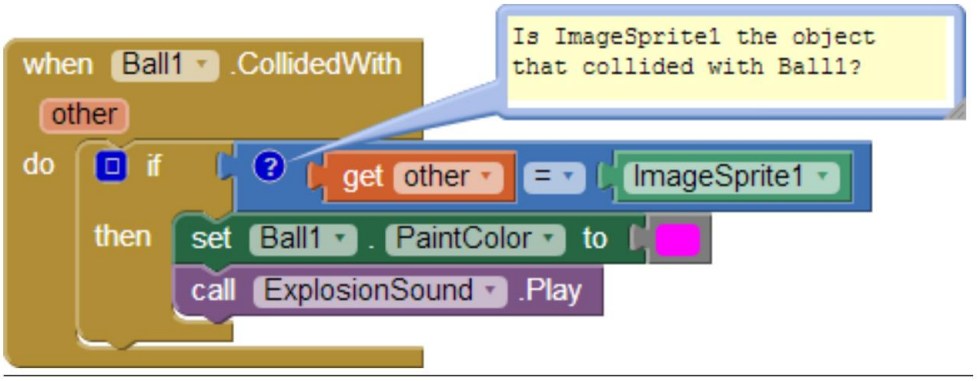


איור .17-7לגרום לכדור לשנות צבע ולהשמיע צליל פיצוץ כשהוא פוגע בחפץ אחר

NoLongerCollidingWithמספק את האירוע ההפוך מ- .CollidedWithזה מופעל רק כאשר שני עצמים התאחדו ואז נפרדו. אז, עבור המשחק שלך, אתה עשוי לכלול את הבלוקים המתוארים באיור .17-7

איור .17-8שינוי הצבע בחזרה ועצירת רעש הפיצוץ כאשר החפצים נפרדים

שימו לב שגם CollidedWith-לוגם ל- NoLongerCollidingWithיש ארגומנט, אחר, המציין את האובייקט המסוים איתו התנגשת (או ממנו נפרדת). זה מאפשר לך לבצע פעולות רק כאשר האובייקט (למשל, Ball1)מקיים אינטראקציה עם אובייקט אחר מסוים, כפי שמוצג באיור .17-8

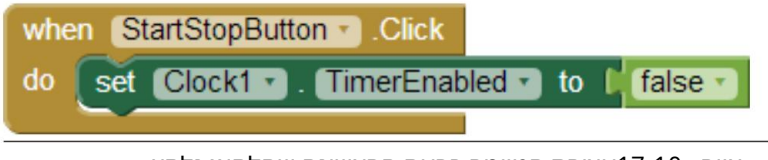


איור 17-9. בצע את התגובה רק אם Ball1פגע ImageSprite1-ב

בלוק ImageSprite1הוא אחד שעדיין לא דנו בו. בלוק זה מתייחס לרכיב בכללותו, לא למאפיין מסוים של הרכיב. כאשר אתה צריך להשוות רכיבים (למשל, כדי לדעת אילו מהם התנגשו), אתה משתמש בבלוק זה.
לכל רכיב יש בלוק כזה במגירה שלו, ולגוש יש שם זהה לזה של הרכיב.

אנימציה אינטראקטיבית

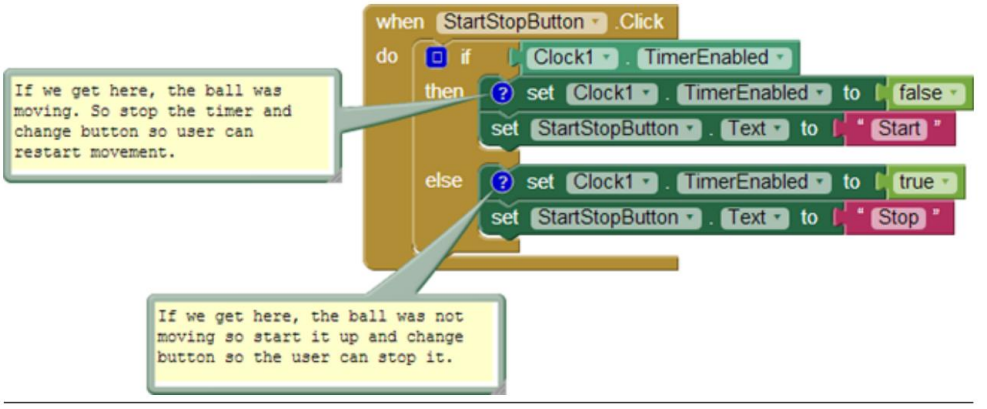
בהתנהגויות המונפשות שדנו בהן עד כה, משתמש הקצה אינו מעורב. המשחקים הם אינטראקטיביים, כמובן, כאשר משתמש הקצה משחק תפקיד מרכזי. לעתים קרובות, משתמש הקצה שולט במהירות או בכיוון של אובייקט באמצעות לחצנים או אובייקטים אחרים של ממשק משתמש.
כדוגמה, בואו נעדכן את האנימציה האלכסונית על ידי מתן אפשרות למשתמש לעצור ולהתחיל את התנועה האלכסונית. אתה יכול לעשות זאת על ידי תכנות מטפל באירועים Button.Clickכדי להשבית ולהפעיל מחדש את אירוע הטיימר של השעון רְכִיב.
כברירת מחדל, המאפיין timerEnabledשל רכיב השעון מסומן. אתה יכול להשבית אותו באופן דינמי על ידי הגדרתו ל-eslaf במטפל באירועים. המטפל באירועים באיור 17-9, למשל, יפסיק את הפעילות של טיימר שעון בלחיצה הראשונה.



איור 17-10. עצירת הטיימר בפעם הראשונה שהלחצן נלחץ

לאחר שהמאפיין Clock1.TimerEnabledמוגדר ,false-כהאירוע Clock1.Timerלא יופעל יותר, והכדור יפסיק לנוע.

כמובן, עצירת התנועה בלחיצה הראשונה אינה מעניינת מדי. במקום זאת,
אתה יכול "להחליף" את תנועת הכדור על ידי הוספת 'אם אחר 'במטפל האירוע שמאפשר
או משבית את הטיימר, כפי שמוצג באיור .17-10
מטפל באירועים זה עוצר את הטיימר בלחיצה הראשונה ומאפס את הכפתור כך שיציג
"התחל" במקום "עצור". בפעם השנייה שהמשתמש לוחץ על הכפתור, ה- TimerEnabled
הוא שקר, כך שהחלק "אחר" מבוצע. במקרה זה, הטיימר מופעל, מה שמניע את האובייקט
שוב, וטקסט הלחצן מועבר בחזרה ל"עצור".
למידע נוסף על בלוקים של ifalse , ראה פרק 18,ולדוגמאות של אנימציות אינטראקטיביות
המשתמשות בחיישן הכיוון, ראה פרק 5ופרק 23.

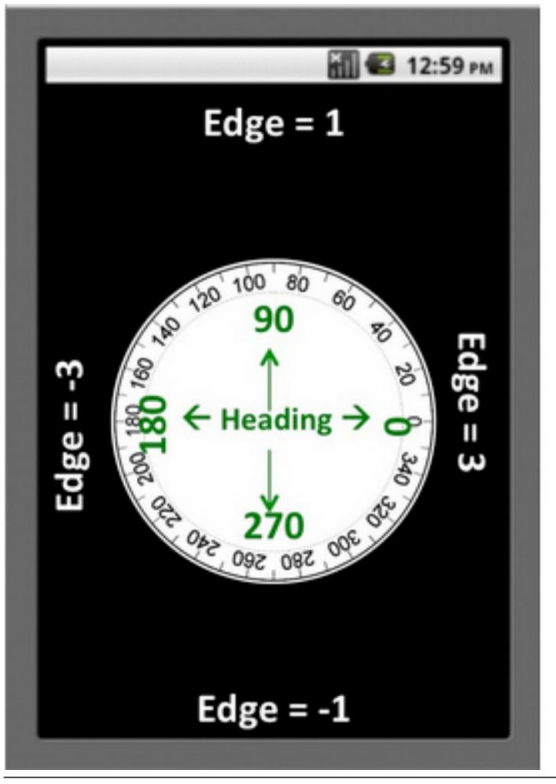


איור .17-11הוספת אם אחרת כדי שהלחיצה על הכפתור תתחיל ותעצור את תנועת הכדור

ציון אנימציית Spriteללא טיימר שעון

דוגמאות האנימציה שתוארו עד כה משתמשות ברכיב Clockומציינות שאובייקט צריך לזוז בכל
פעם שאירוע ה- Clock.Timerמופעל. סכימת האירועים Clock.Timerהיא השיטה הכללית
ביותר לציון הנפשה. מעבר להזזת אובייקט פשוט, תוכל גם לשנות את צבעו של אובייקט לאורך
זמן, לשנות טקסט כלשהו (שיראה כאילו האפליקציה מקלידה), או שהאפליקציה תדבר מילים
בקצב מסוים.

אם אתה רוצה רק להזיז אובייקטים, App Inventorמספק חלופה שלא
דורשים שימוש ברכיב שעון . כפי שאולי שמתם לב, לרכיבי ImageSpriteו- Ballיש מאפיינים
עבור Heading, Speedו- .Intervalבמקום להגדיר מטפל באירועים , Clock.Timerאתה יכול
להגדיר את המאפיינים האלה Component Designer-באו Blocks Editorכדי לשלוט על
האופן שבו ספרייט זז.
לשם המחשה, הבה נשקול מחדש את הדוגמה שהניעה כדור באלכסון. הכותרת _
לתכונה של ספרייט או כדור יש טווח של 360מעלות, כפי שמוצג באיור .17-11

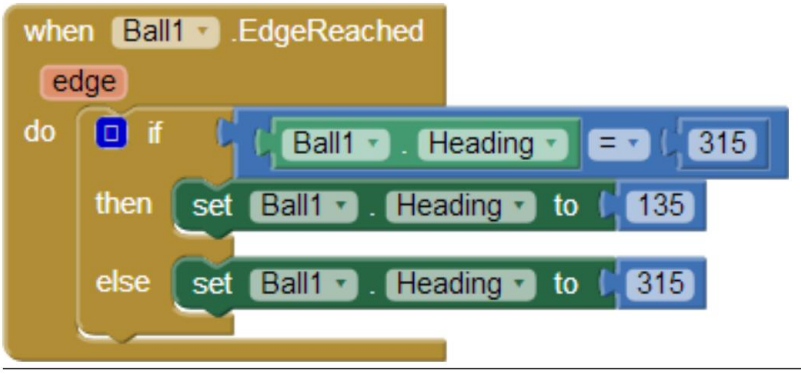


איור 17-12. למאפיין Heading יש טווח של 360 מעלות

אם תגדיר את הכותרת ל ,0- הכדור יזוז משמאל לימין. אם תגדיר אותו ל-09, הוא יזוז מלמטה למעלה. אם תגדיר אותו ל-081, הוא יעבור מימין לשמאל. אם תגדיר אותו ל-072, הוא יעבור מלמעלה למטה. ואם תגדיר אותו ל-513, הכדור ינוע משמאל למעלה לימין למטה.

כדי לגרום לאובייקט לזוז, עליך גם להגדיר את המאפיין Speed לערך שאינו 0. המהירות שהאובייקט מזיז נקבעת למעשה על ידי המאפיינים Speed ו- Interval ביחד. המאפיין Speed הוא המרחק, בפיקסלים, שהאובייקט יזוז בכל מרווח.

כדי לנסות את המאפיינים האלה, צור אפליקציית בדיקה עם קנבס וכדור והתחבר המכשיר או האמולטור שלך לבדיקה חיה. לאחר מכן, שנה את מאפייני הכיוון , המהירות והמרווח של הכדור כדי לראות כיצד הם פועלים.
לדוגמה, נניח שרצית להזיז כדור קדימה ואחורה מהצד השמאלי העליון לימין התחתון של הבד. במעצב, תוכל לאתחל את המהירות של הכדור ל-5 ו- Interval ל-001, ולאחר מכן להגדיר את המאפיין Heading ל-513. לאחר מכן תוסיף את המטפל באירועים Ball1.EdgeReached , אותו תוכל לראות באיור 17-12, כדי לשנות כיוון הכדור כשהוא מגיע לשני קצוות.



איור 17-13. שינוי כיוון הכדור כשהוא מגיע לשני הקצוות

סיכום

באמצעות רכיב ה- Canvas, ניתן להגדיר תת-אזור במסך המכשיר בו אובייקטים יכולים לנוע ולקיים אינטראקציה. אתה יכול לשים רק שני סוגים של רכיבים בתוך קנבס: ImageSprites ו- Balls.

אנימציה היא אובייקט שזז או משתנה לאורך זמן. אתה יכול לתכנת אנימציה, כולל תנועה ושינויים גרפיים אחרים, עם אירוע הטיימר של רכיב השעון. אם אתה רק רוצה להזיז אובייקטים, אתה יכול להשתמש בשיטה חלופית המבוססת על מאפייני Heading, Speed ו- Interval הפנימיים של רכיבי ImageSprite ו- Ball.

עם כל אחת מהשיטות, אתה יכול גם לנצל את הפונקציונליות ברמה גבוהה עבור טיפול באירועים המטפלים בהתנגשויות.