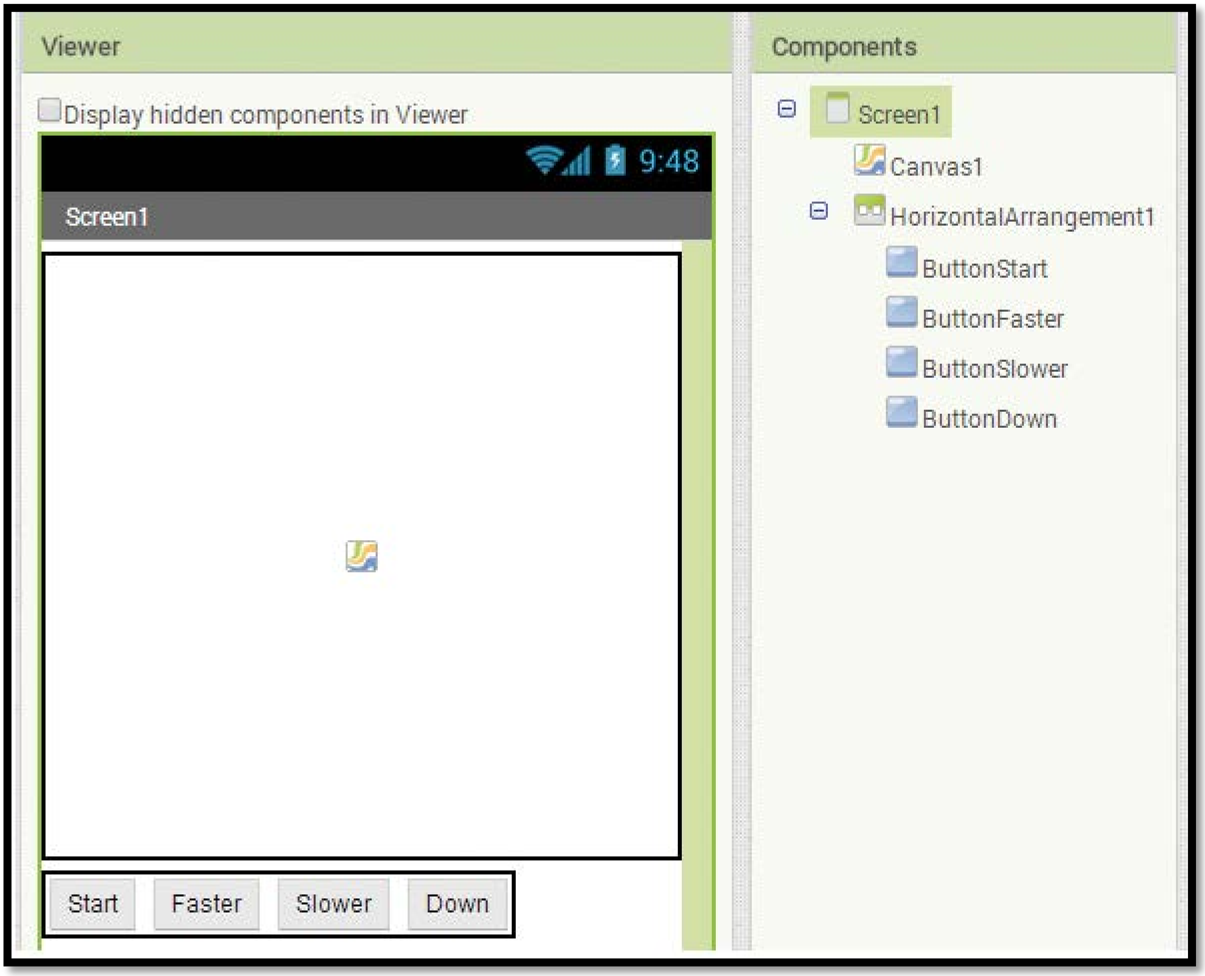
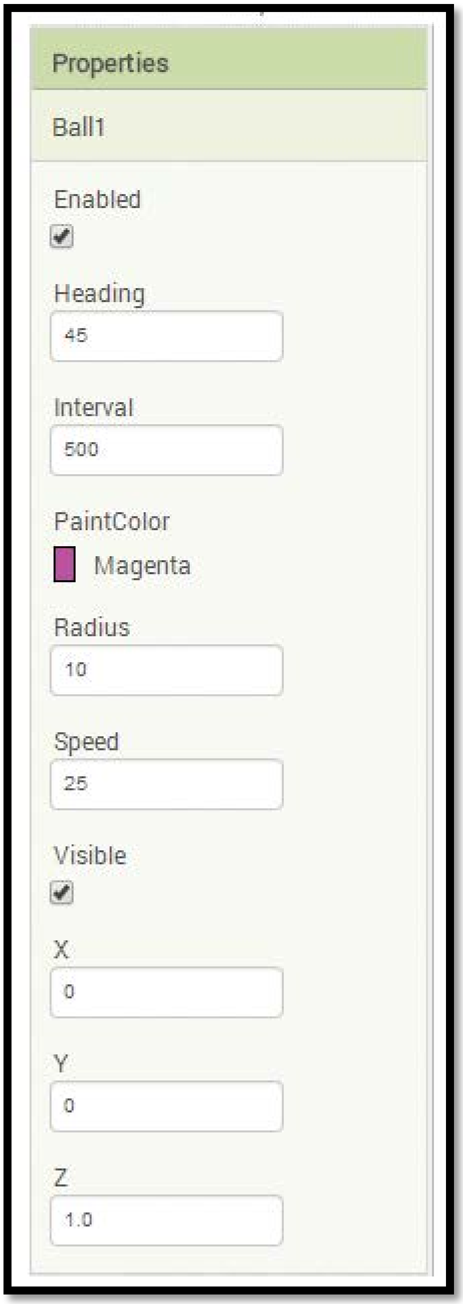
Chapter 9: Bouncing Ball

Let’s create an app that will make a ball bounce around a Canvas. We will also let the user control when to start over, speed up, and slow down. We will give the user a Button that will bounce the ball down from wherever it is at the time.

Step 1: Design the user interface. Start a new project and add a Canvas to the screen
Step 2: Add a HorizontalArrangement under the Canvas and then add four Buttons to it.
Step 3: Rename your Buttons and set their Text properties to match the image.



Step 4: Add a Ball component (from the Drawing and Animation Palette) to the Canvas and set the properties as shown below. Set the Heading to 45 degrees, the Interval to 500 (half second), the color to Magenta (or whatever you wish), the Radius to 10 and the X, Y coordinates to 0, 0



Properties

Ball1

Enabled

☑

Heading

45

Interval

500

PaintColor

🟪 Magenta

Radius

10

Speed

25

Visible

☑

X

0

Y

0

Z

1.0

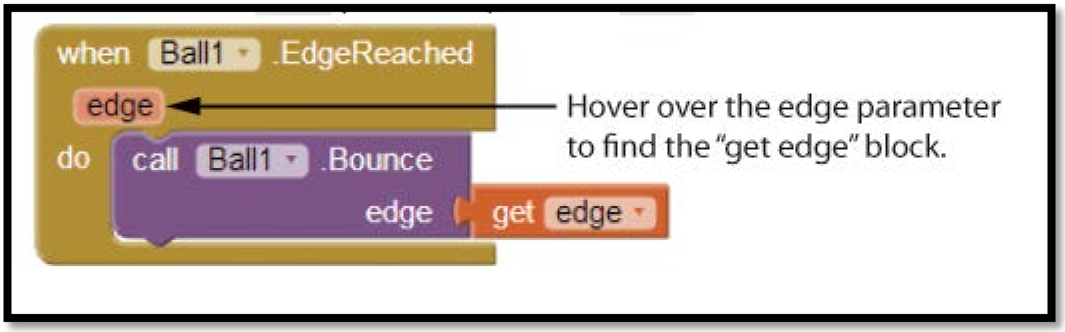
Step 5: Now Let's program the blocks, starting with the when ButtonStart.Click do event handler. Open the ButtonStart drawer and find the when ButtonStart.Click do block. Open to the Ball1 drawer and select three method blocks: set Ball1.Interval to, set Ball1.X to and set Ball1.Y to. Place all three in the when ButtonStart.Click do event handler. From the Math drawer, select three number blocks and set the first two to zero and the third to 500. Plug the zeros into the X and Y slots, and the 500 into the Interval slot.

Step 6: Now you will program Ball1 to move slower. Remember that the higher the interval the slower Ball1 will move. So, we will simply increase the interval by two times each time the user clicks the Slower Button. (You can also decrease the Speed to accomplish this step, but we are going to manipulate the Interval in this example.)

Open the Ball1 drawer and find the set Ball1.Interval and the value of Ball1.Interval to blocks. Multiply the Interval by 2 by using a multiplication block and a number block found in the Math drawer. Plug the Ball1.Interval block into the first slot of the multiplication block, and the number 2 block into the second slot. Plug the multiplication block into the set Ball1.Interval to block.

Step 7: Now you will program Ball1 to move faster. This step is almost identical to Step 6, but it uses the division math operation instead. We want to cut the interval in half to make the Ball travel faster.

Step 8: Now let's program Ball1 to bounce on any of the edges it reaches. Find the Ball1.EdgeReached event handler in the Ball1 drawer. Inside that event, place a Ball1.Bounce method block, also found in the Ball1 drawer. Plug a get edge block (found by hovering the mouse cursor over the edge parameter) into the edge socket of the Bounce method block.



Hover over the edge parameter to find the "get edge" block.

Step 9: The last action we need to program is the bounce down which happens when the user presses the ButtonDown Button. In the when ButtonDown.Click do event handler, call the Ball1.Bounce method. The call is similar to what we did in Step 8, but instead of using the value of the edge argument we will just put in a number 1 block which represents the top edge.

Step 10: Run and test your application