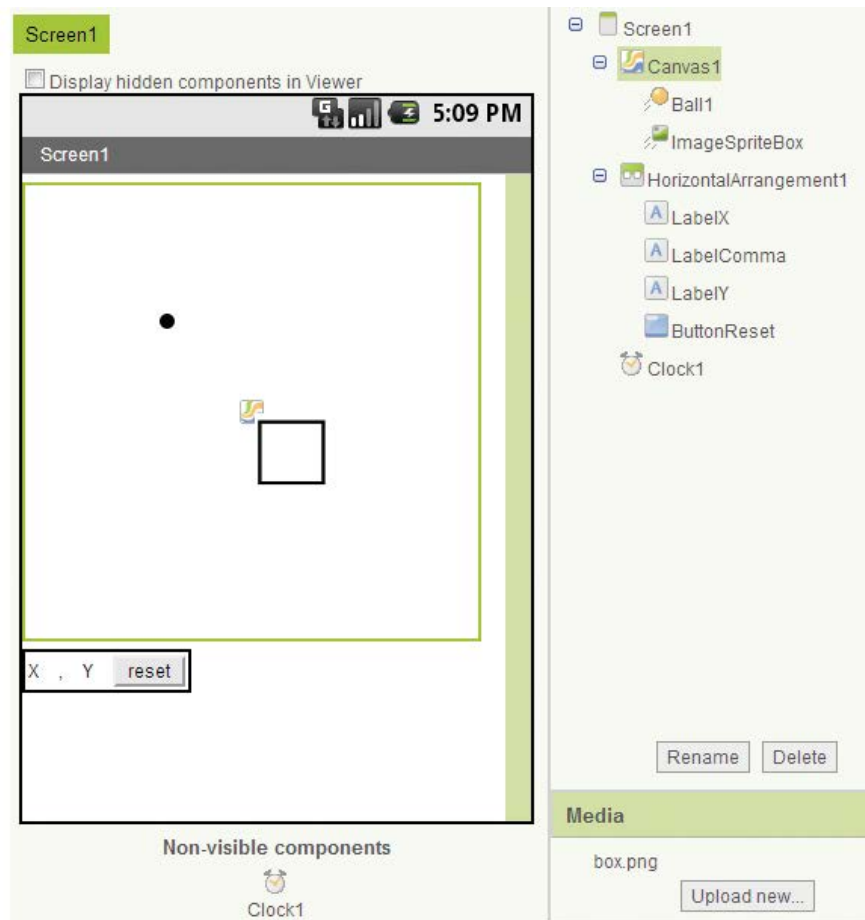


Chapter 9: Drag a Ball into a Box



Step 1: Drag a Canvas onto the screen and make both the Height and Width 300 pixels.

Step 2: Download box.png from the book's companion website. Place an **ImageSprite** on the Canvas and set the Height and Width to 50 pixels. Set the Picture property to box.png and set both the X and Y properties to 150. Rename the **ImageSprite** to **ImageSpriteBox**.

Step 3: Place a Ball sprite on the Canvas. Set the Heading to 45, the Interval to 0 and the Speed to 0 (because we only want this ball to move when we drag it). Set the X coordinate to 90 and the Y coordinate to 90.

Step 4: Add a **HorizontalArrangement** below the Canvas. Add three labels and rename them **LabelX**, **LabelComma** and **LabelY**. Set the Text property of **LabelX** to X, **LabelComma** to a comma, and **LabelY** to Y.

Step 5: Add a Button to the **HorizontalArrangement**, rename it **ButtonReset**, and set its Text property to reset.

Step 6: Add a Clock component to your screen. Make sure the `TimerEnabled` property is checked and set the `TimerInterval` to 40.

Step 7: In this step you will create the global variables. These global variables will hold the `currentX` and `currentY` values. We already have the `currentX` and `currentY` values in the Draggged event handler, but we also need to access these values elsewhere. To create the global variables, open the Variables drawer and click on the initialize global name block twice. Name the first one `gCurrentX` and the second `gCurrentY`. From the Math drawer, get two number blocks and set them each to zero. Plug the number blocks into the sockets of the `gCurrentX` and `gCurrentY` initialization blocks.

Step 8: Program the Ball1.Dragged event handler. You will find the Ball1.Dragged event handler and the Ball1.MoveTo method block in the Ball1 drawer. To create the get `currentX` and get `currentY` blocks, hover the mouse cursor over those parameters of the Ball1.Dragged event handler. To create the set global `gCurrentX` and set global `gCurrentY` blocks, hover the mouse cursor over those variables' global initialization blocks.

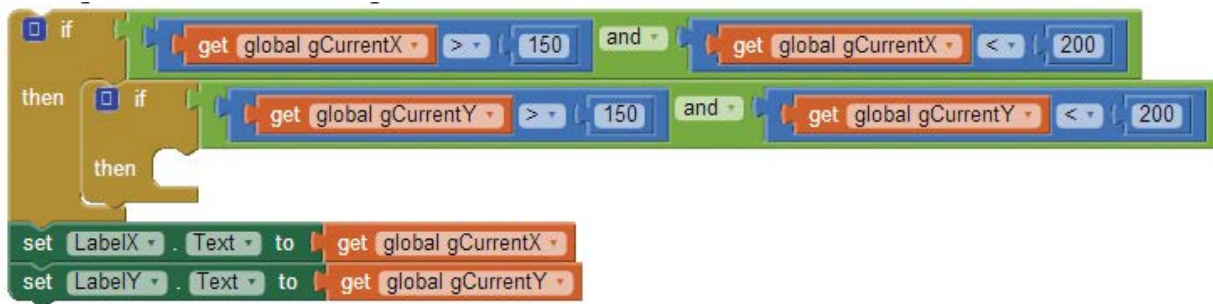
Step 9: Now you will program the Timer event handler. Find and select the when `Clock1.Timer` do event handler in the Clock1 drawer.

Step 10: Find and select the set `LabelX.Text` to and the set `LabelY.Text` to blocks from their respective drawers, the `LabelX` drawer and `LabelY` drawer. Assign the value of `gCurrentX` and `gCurrentY` to the `LabelX` and `LabelY` label's Text properties, respectively. Place these blocks in the `Clock1.Timer` event handler

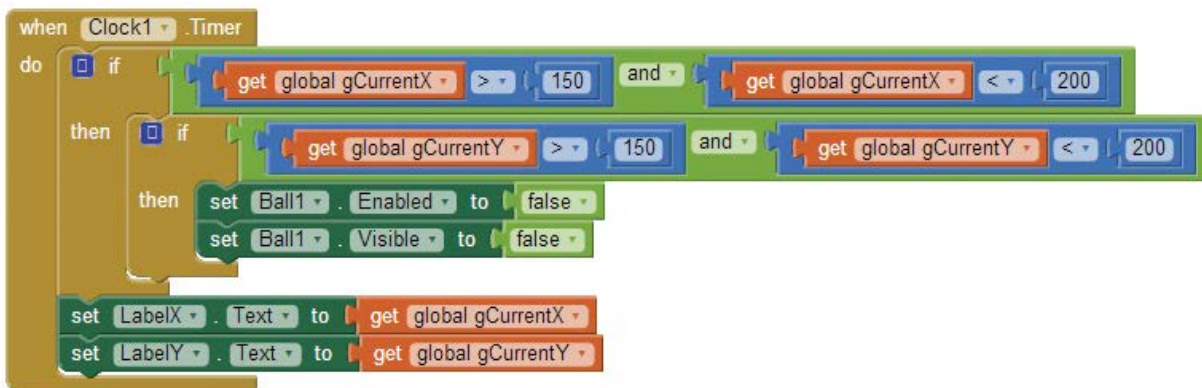
Step 11: You aren't done, but take a minute to run and test on your phone or emulator. Watch the X and Y Labels change as you drag the ball around.

Step 12: Now we are going to use a timer to see if the Ball sprite is in the same location as the box. Because the box's top left corner is at position 150, 150 on the Canvas and the box is 50 pixels tall and wide, we are going to check and do the following: If the x coordinate of the ball is greater than 150 and less than 200 and the y coordinate of the ball is greater than 150 and less than 200 then we want to make the ball disappear (because the ball will be over the box). Note: In this set of blocks, you want to use the global `gCurrentX` and `gcurrentY` values. Set up your if then block like the picture below.

The most efficient way to check the range of both the X and Y coordinate values is to use nested if then blocks. The first is the outer if then block checking the X coordinate range. If that passes, then we check for the Y coordinate range. If both conditions pass, the ball is over the box. When the ball is over the box, we want to simulate it "disappearing" by making it invisible and disabling it.



Step 13: Put the blocks shown below inside the Clock1.Timer event handler. From the Ball1 drawer, find and select the set Ball1.Enabled to and the set Ball1.Visible to blocks, and set them both to false. Place the two statements in the then portion of the inner if then block, as shown in below.



Step 14: Now you will program the ButtonReset.Click event handler. Find the ButtonReset.Click event handler in the ButtonReset drawer and place it in the editor. In the Ball1 drawer, find the following blocks and place them in the ButtonReset.Click event handler:

- set Ball1.Enabled to, plug a logic true block into its socket
- set Ball1.Visible to, plug a logic true block into its socket
- set Ball1.X to, plug a number 20 into its socket
- set Ball1.Y to, plug a number 20 into its socket
- set gCurrentX, plug a number 20 into its socket
- set gCurrentY, plug a number 20 into its socket

Step 15: Run and test your application