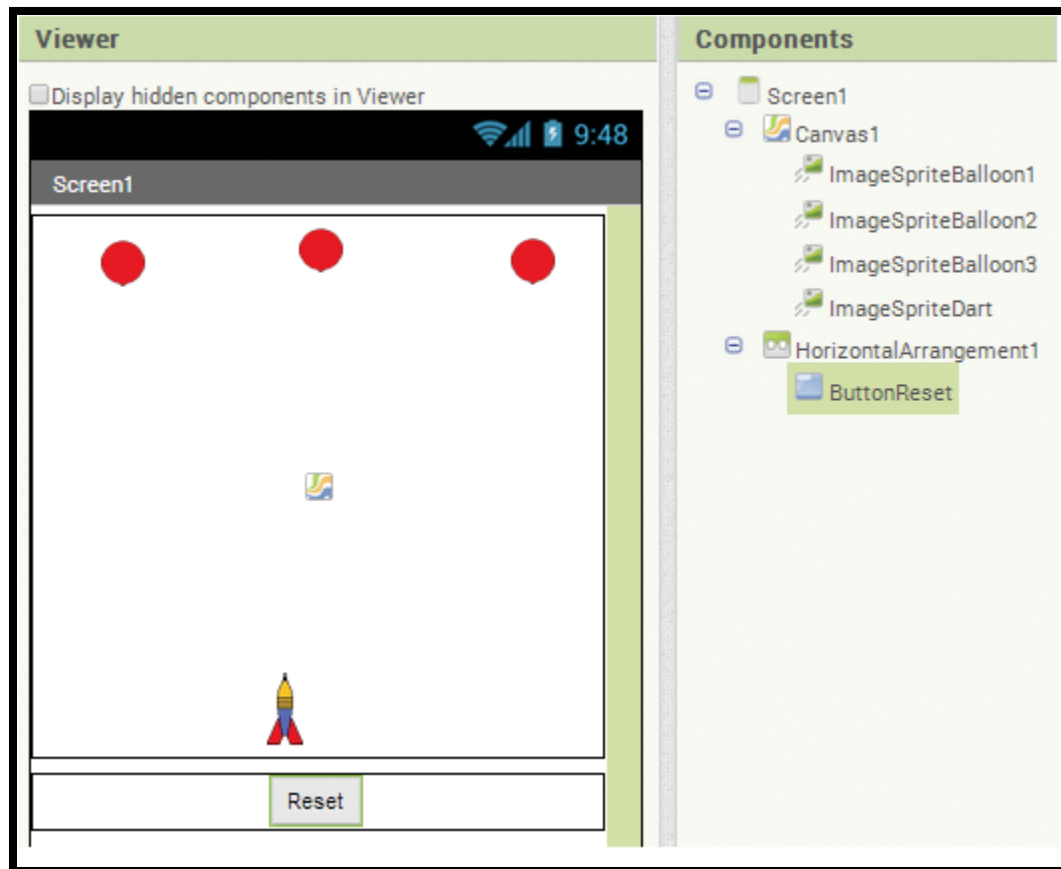


Chapter 9: Popping Balloons

In this tutorial, we are going to simulate three balloons falling from the sky. We are giving the user a dart to pop the balloons. The balloons will travel straight down at different speeds. The user can drag the dart from left to right and when the drag is stopped, the dart will travel straight up to pop the balloon.



Step 1: Add a Canvas component to the screen and set the Width to 300 and the Height to 300.

Step 2: Upload the three images. Balloon.png, Spiral.png, and Dart.png.

Step 3: Add four **ImageSprite** components from the Drawing and Animation Palette to the Canvas. Align them as shown in the picture. Rename the sprites **ImageSpriteDart**, ImageSpriteBalloon1, ImageSpriteBalloon2, and ImageSpriteBalloon3.

Step 4: Set the Picture property of the balloon sprites to Balloon.png and set the Picture property of the dart sprite to Dart.png.

Step 5: The Height and Width of the balloon sprites should be 25 pixels. The Height of the dart sprite should be 40 and the Width 20.

Step 6: The Heading of the balloon sprites should all be set at 270 (straight down). The Heading for the dart should be 90 (straight up).

Step 7: Set the Interval for the balloon sprites to 500. Set the Interval of the dart sprite to 100.

Step 8: Set the Speed properties of the balloon sprites to various values. For example, set one to 5, one to 7, and one to 10. Set the Speed of the dart to zero.

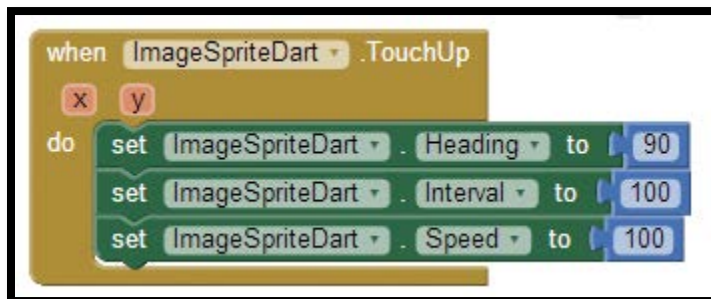
Step 9: Add a **HorizontalArrangement** (from the Layout Palette) under the Canvas, set its Width to fill parent, and then set the **AlignHorizontal** property to Center.

Step 10: Place a Button (from the User Interface Palette) in the **HorizontalArrangement**. Rename it to **ButtonReset** and change the Text property to Reset.

Step 11: Now you will program the Dragged event handler to allow the user to drag the dart sprite left and right. In the **ImageSpriteDart** drawer, find the Dragged event handler and the **MoveTo** method block. Place them in the editor and place the **MoveTo** method block inside the Dragged event handler.

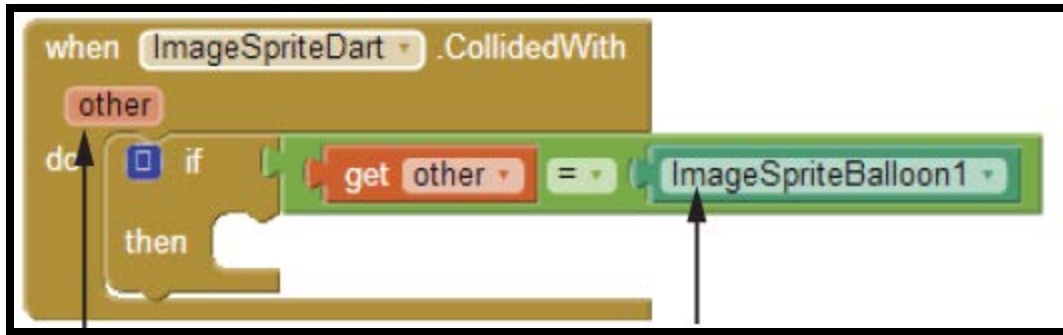
Step 12: For the x socket of the **MoveTo** method, hover the mouse cursor over the **currentX** parameter in the Dragged event handler, and then click the get **currentX** block. Use a number block set to the value 255 for the y socket. By using an unchanging number for the y argument, we allow the user to drag side to side but not upwards.

Step 13: In the **ImageSpriteDart** drawer, find the **TouchUp** event handler and the set blocks for the Heading, Interval and Speed properties. Set the properties according to the image below. The **TouchUp** event handler will execute when the user lets go of the drag. This is the point where we want to let the dart fly up the screen to pop a balloon.



Step 14: In the **ImageSpriteDart** drawer, find the **CollidedWith** event handler and the set **ImageSpriteDart.Picture** to method block.

Step 15: Place an if then block inside the event handler, and use the Math equals block to compare the other argument with the component blocks of each of the balloon sprites. Remember each component's component block is the last block in its drawer.



Step 16: Place two more if then blocks inside the event handler. Use the equals block (from the Logic drawer) to compare the other argument with the component blocks of the other balloon sprites. It may be easiest to copy and paste the first if then block and then change the value of the component blocks to ImageSpriteBalloon2 and ImageSpriteBalloon3

Step 17: For each balloon sprite, place their set ImageSprite.Picture property blocks in the proper if then condition. If a balloon has been hit, we are going to temporarily change its Picture property to a red spiral, helping indicate to the user that it has been hit. Step 18: Change the Visible property of ImageSpriteDart to false to temporarily hide it.

Step 18: Change the Visible property of ImageSpriteDart to false to temporarily hide it.

Step 19: Now you will program the ImageSpriteDart.EdgeReached event handler. When the dart hits the top edge of the Canvas (remember that the top edge is "1"), we want to put it back down to the bottom so the user can try again. In the ImageSpriteDart drawer, find and select the EdgeReached event handler, the set Speed to property block, and the MoveTo method block.

Step 20: In the Control drawer select an if then block and place it in the ImageSpriteDart.EdgeReached event handler. For the test socket use a Math equals block to determine if the value of the edge is equal to one. You can find the get edge block by hovering the mouse cursor over the edge parameter of the EdgeReached event handler.

Step 21: In the if then block's then socket, set the Speed property of the dart sprite to 0 to stop it, and move the dart to the x and y coordinates 100 and 255 respectively. This will stop the sprite and place it back to the bottom of the screen.

Step 22: Now you must program the application to take away the balloons and disable them once they have been popped. In the ImageSpriteDart drawer, find the NoLongerCollidingWith event handler, the MoveTo method, the set Visible to property block, and the set Speed to property block.

Step 23: Program the first if then block. This if then block determines whether a collision has occurred with ImageSpriteBalloon1. Use the equals block from the Logic drawer to compare the get other block (hover the mouse cursor over the other parameter to find that block) with the ImageSpriteBalloon1 component block (remember this is the last block in the ImageSpriteBalloon1 drawer). Find the ImageSpriteBalloon1 set Enabled to and set Visible to property blocks and set them both to false in the if then block. See image below.



Step 24: Grab two more if then blocks from the Control drawer, or use copy and paste by copying the if then block you have to program the conditions for ImageSpriteBalloon2 and ImageSpriteBalloon3.

Step 25: For each of the balloons, place their set Visible to and set Enabled to blocks in the proper if then block. Set each of these properties to false as shown above. Take care to make sure that each balloon block is in the proper spot. (It's easy at this point to mix the blocks up because they look similar, especially if you are using the copy and paste approach.)

Step 26: Use the dart sprite's **MoveTo** (X,Y) method block to place it toward the bottom of the screen at X, Y coordinates of 100, 255. Set the dart sprite's Visible property to true and set the Speed property to zero.

Step 27: The last event handler we need to program is the Click event handler for the reset Button. In the **ButtonReset** drawer, find the **ButtonReset.Click** event handler.

Step 28: For each of the balloon **ImageSprites**, find their Picture, Enabled and Visible property set blocks, and place them in the **ButtonReset.Click** event handler. Reset them as shown below by setting their Picture property back to a balloon and their Enabled and Visible properties to true.



Step 29: For each of the balloon **ImageSprites**, use their **MoveTo** methods to place them back at the top of the screen in their starting positions.

Step 30: Run and test your game on your device or emulator.