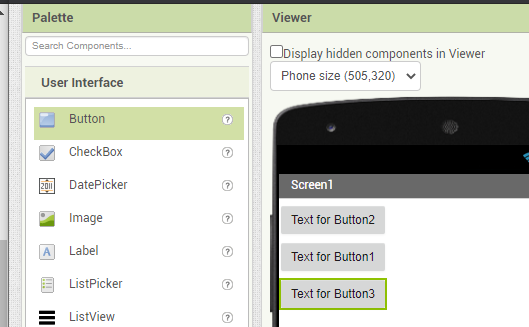
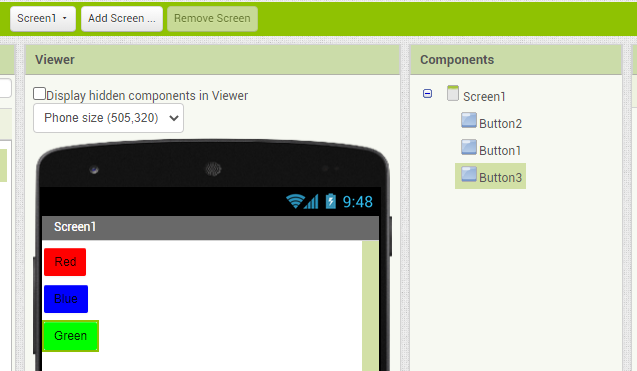
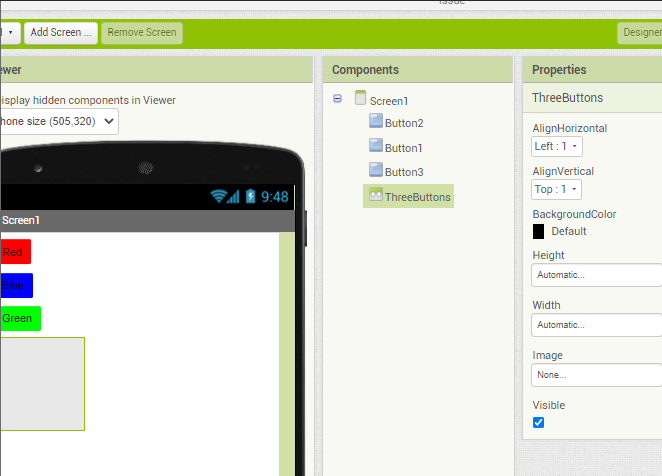
1. **PaintPic**
2. Create a project name with PaintPic
3. Make 3 buttons and drag to the viewer



1. Rename the buttons and give a colour in properties pane and set background colour to red, the 2nd button to blue and the last one to green.



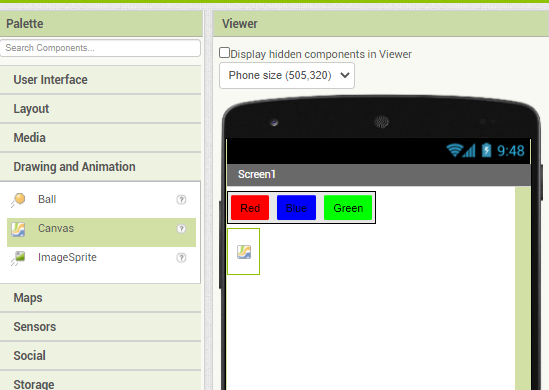
1. Add horizontal arrangement container from pallet, set it with Three Buttons named and place under the buttons. Set width to fill parent



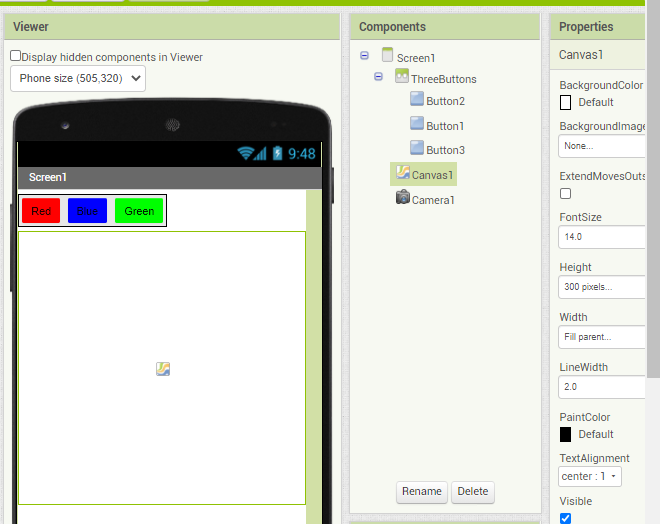
1. Arrange threebuttons horizontally with drag the red, blue, and green colour as one by one to inside the ThreeButtons



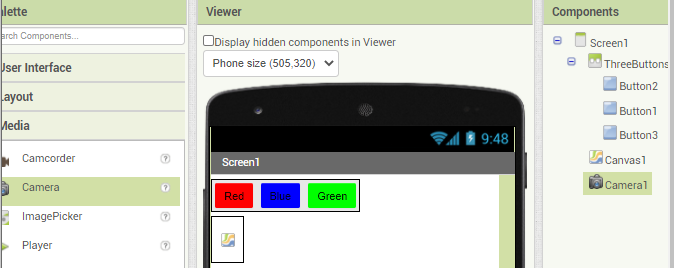
1. Add a Canvas component from pallet and change the name to DrawingCanvas, set width to fill parent



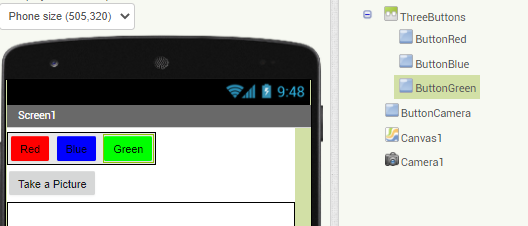
Set the width with fill parent and height with 300px



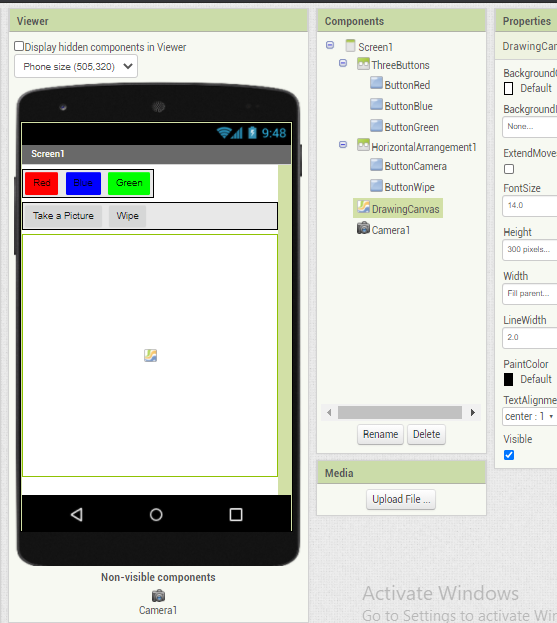
1. Add camera component in media and put it into invisible screen in the bottom



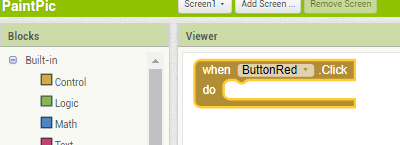
1. Add a button to activate the camera and change the text with Take a Picture and rename the button to CameraButton



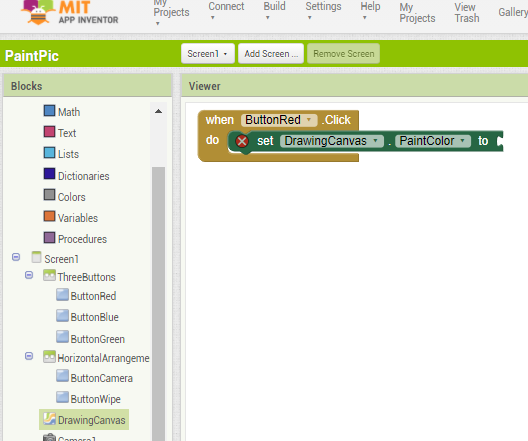
1. Add Wipe button and arrange it horizontally with ButtonCamera. Change the name with ButtonWipe and the text is Wipe. Drag out Horizontal Arrangement under the ThreeButtons



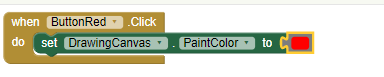
1. Now we will build the Blocks Editor.
2. First, open the ButtonRed drawer and drag over the whenButtonRed.Click block



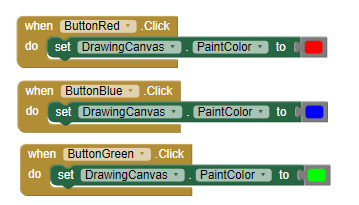
1. Drag out the set DrawingCanvas.PaintColor block and place inside do section in ButtonRed.Click



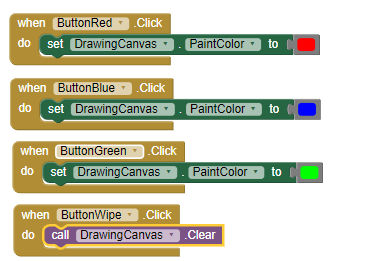
1. Set the color to red with drag out the block for colour Red inside to section in PaintColor



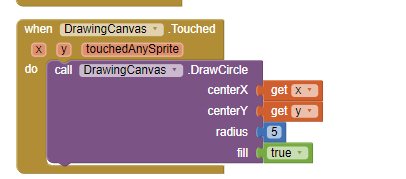
1. Do the same steps for ButtonBlue and Green



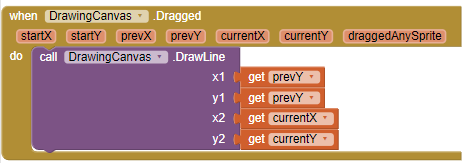
1. Drag ButtonWipe.Click and drag DrawingCanvas.Clear block inside do area of ButtonWipe.Click



1. Drag DrawingCanvas.Touched block and place DrawingCanvas.DrawCircle block under do section. Drag the number and set it into 5, place it to radius.



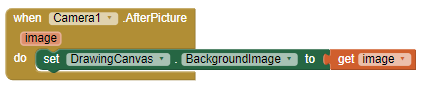
1. Drag DrawingCanvas.Dragged block and place do area with DrawingCanvas.DrawLine. fill the x1 and x2 parameter with prevX and prevY, and fill x2 y2 parameter with current and currentY



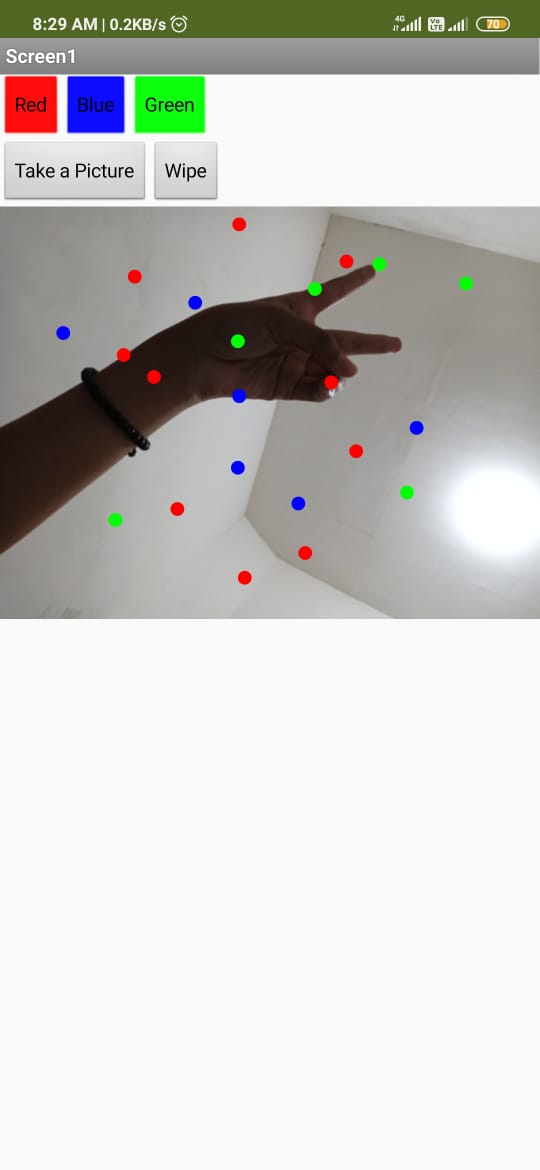
1. Drag ButtonCamera.Click block and call Camera1.TakePicture block and place it inside ButtonCamera.Click block



1. Change canvas background to the image that captured by the camera. Drag Camera1.AfterPicture and drag SetDrawingCanvas.BackgroundImage block in do section of Camera1.AfterPicture block. Fill to section in DrawingCanvas.BackgroundImage with getImage



1. Test with phone



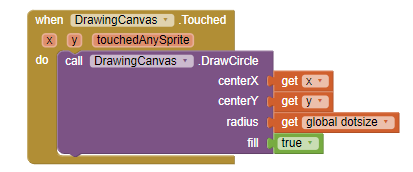
1. Next we will Create a Variables
2. Drag initialize “small” variable and fill the socket with “2” number



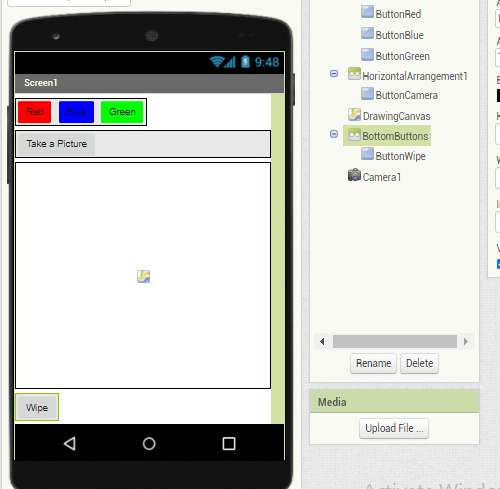
1. Do the same step with “big” variable and initial value with 8 and “dotsize” variable with initial value “2”



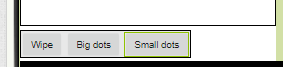
1. Go back to DrawCircle block and change 5 value inside radius to global dotsize variable. It will get the value of dotsize variable that we initialize before



1. Now, we will set up or change dotsize to be small or big size
2. Go back to designer and drag HorizontalArragement component below DrawingCanvas component. Name it with BottomButtons. Drag ButtonWipe into BottomButtons



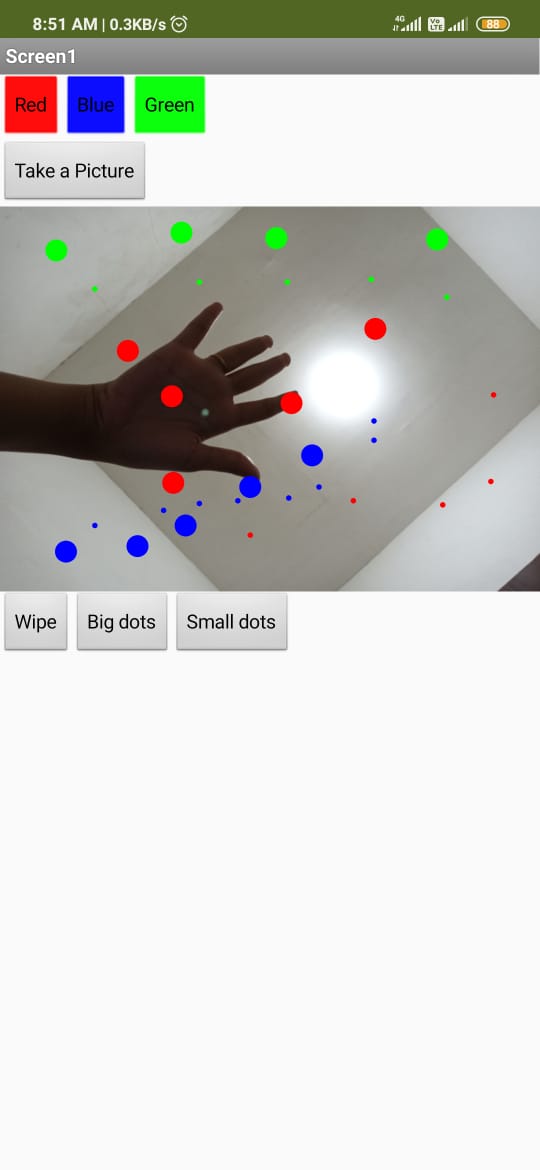
1. Drag 2 more buttons next to ButtonWipe and name it with ButtonBig and ButtonSmall, and set the text with Big dots and Small dots



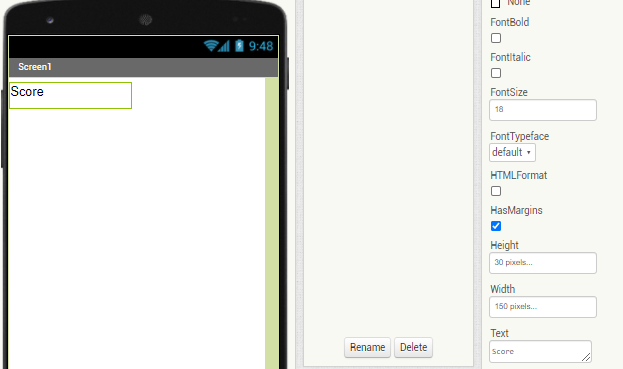
1. Go to Blocks Editor and drag ButtonBig.Click and set the global dotsize variable inside do statement. Fill to statement of dotsize with big variable. Do the same steps for small option



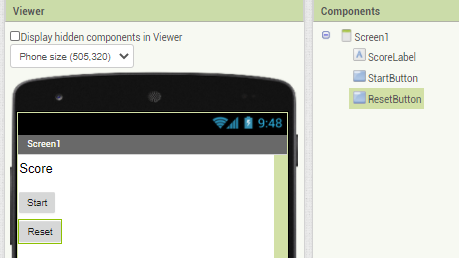
1. Check with the phone



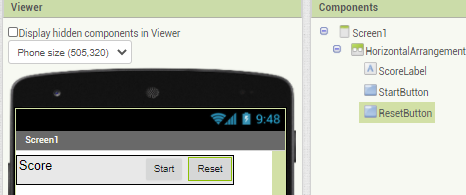
1. **PONG**
2. Drag a Label and set the name to Score and set the text with Score. Set the width with 150px and heigh with 30px



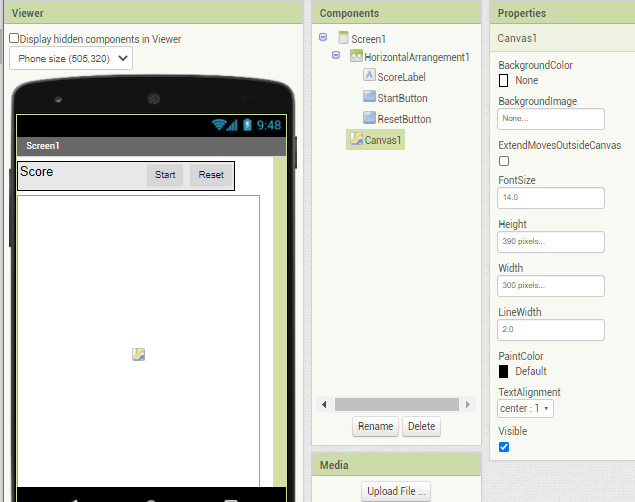
1. Add 2 buttons under the label and set the name with StartButton and ResetButton



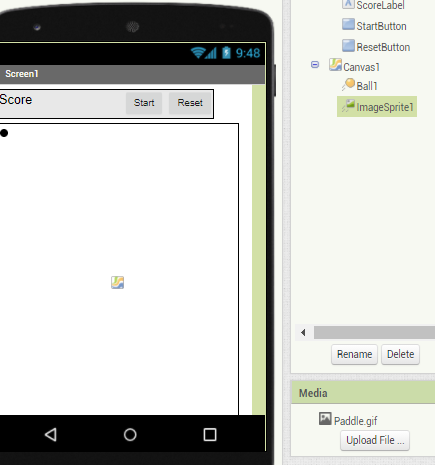
1. Drag HorizontalArrangement component and drag ScoreLabel inside it, drag two botton next to ScoreLabel



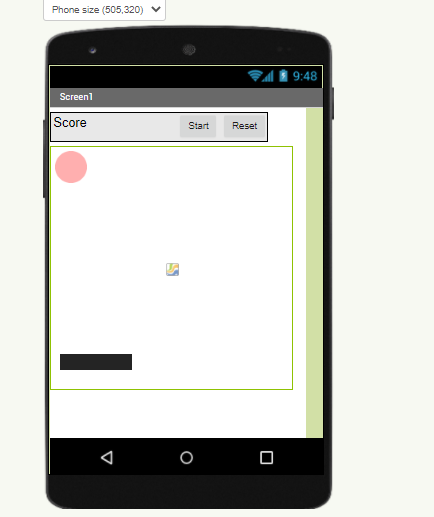
1. Drag a canvas component and set the width with 300x and height 390px, change the colour



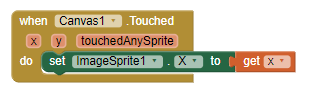
1. Drag Ball and Image Sprite into Canvas component, highlight the image and change it with paddle.gif



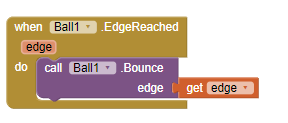
1. Set the heading of Ball to 30, interval to 50 and radius to 20, speed to 5. Change the colour of ball



1. Open Blocks Editor and drag Canvas1.Touched block. Fill the socket with ImageSprite1.block



1. Drag Ball1.EdgeReached block and fill the socket with Ball1.Bounce block



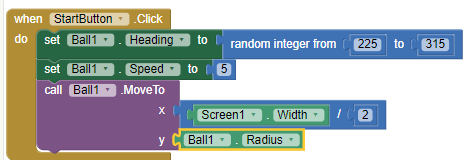
1. Drag StartButton.Click block and fill the socket with set StartButton.Heading. In Math drawer, drag call random integer block and added it inside socket of StartButton.Heading



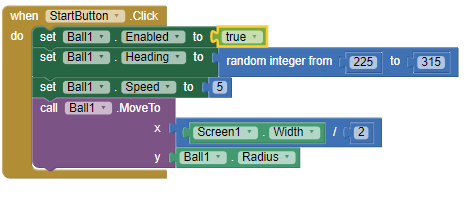
1. Drag Ball1.Speed and blace it under set Ball1.Heading block. Put a 5 number in the socket



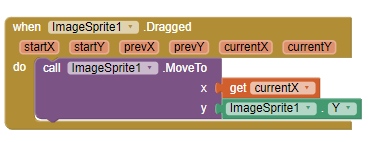
1. Drag Ball1.MoveTo block and put it under Ball1.Speed block. In x parameter, fill it with / block from math, in the first socket fill it with Screen1.Width and the last with 2 number



1. Drag set Ball1.Enabled and click Logic from small colored menu and click true



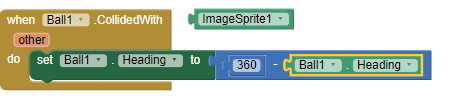
1. Drag ImageSprite1.Dragged block and ImageSprite1.Move to and drop it into the first block. Fill the x with CurrentX and the y with ImageSprit1.Y



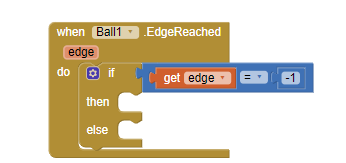
1. Drag when Ball1.CollidedWith block and type ImageSprite1 in other



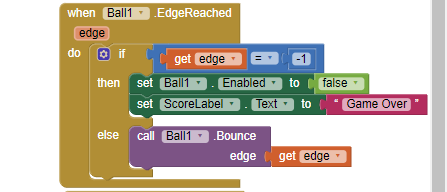
1. From the Ball1 drawer, drag out a set Ball1.Heading to block and drop it inside the Ball1.CollidedWith block. Click on the Built-In palette and open the Math drawer, and drag out a subtraction block. Click in the first blank area and type "360". Click on the My Blocks palette, open the Ball1 drawer, and then drag out a Ball1.Heading block and drop it in the second blank area. This will reverse the ball when it hits the paddle.



1. find the place where you put the when Ball1.EdgeReached block. We're going to add some new blocks to this event, so for now drag the Ball1.Bounce block away from Ball1.EdgeReached so that it becomes separated. Leave the Ball1.Bounce block loose in the work area for now. Under the Built-In palette, open the Control drawer and drag out an ifelse block. From the Math drawer, drag out an equal (=) block and drop it into the "test" socket. Open the My Definitions drawer and drag out a value edge block and drop it in the first blank area. Click on the second blank area and type "-1"



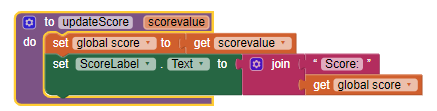
1. drag out a set Ball1.Enabled to block and drop it in the "then-do" area of the ifelse block. Click to the right of that to get the popup menu, then click on "Logic" and choose "false". This will stop the ball from moving when it gets past the paddle. In addition, drag out a set scoreLabel.Text block from the ScoreLabel drawer, and attach it underneath the set Ball1.Enabled block. From the Built-In palette, open the Text drawer, and drag out a text block and drop it after the "to" (you can also click the work area to get the popup menu and choose text). Click the text and change it to "Game Over!" This text will appear on the screen in the ScoreLabel when the ball gets past the paddle.



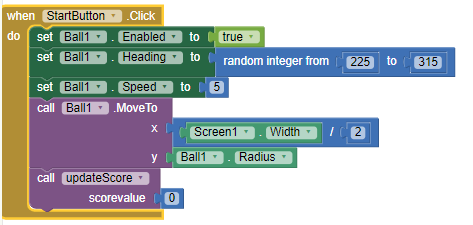
1. Create a global value and named it with score



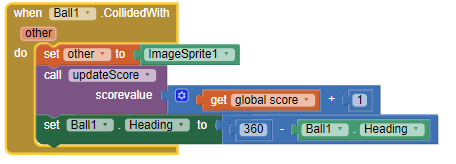
1. Under the Built-In palette, open the Definition drawer, and drag out a procedure block and drop it in an open area. Click on "procedure" and rename it to "updateScore". Then from the Definition drawer, drag out a name block and then click on the "name" and type "scorevalue". This creates a parameter for the procedure that is named "scorevalue". A parameter is a temporary variable that holds a value for a procedure. The value is specified when the procedure is called. From the My Blocks palette, open the My Definitions and drag out a set global score to block and drop it in "do" area of the updateScore procedure. From My Definitions drag out a value scorevalue block and drop it in the "to" area. This sets the score variable to the passed value.



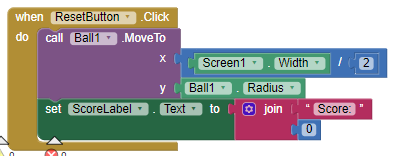
1. drag out a call updateScore block and drop it at the end of the StartButton.Click next to the block and type 0 to update the score to 0.



1. open the Math drawer drag out an addition (+) block and drop it at the end of the call updateScore block. From the My Definitions drawer, drag out a global score block and drop it in the first blank area. Click in the second blank area and type 1. This will increment the score by 1.



1. drag the ScoreLabel.Text to block and place it under the previous block. Type "0" and drag the "0" block to the empty socket of the ScoreLabel.Text to block. In addition, open the Text drawer under the Built-In palette and drag out a join block and drop it after the ScoreLabel.Text. Set the first blank area to the text "Score:" and set the second blank area to the number zero



1. Result

