Author: Dauda Bilau

Project Title: Android Pong game

Language: Processing Programming Language

Date: 20th May, 2019

Introduction

This course introduces learners to computer programming in an interactive way using Processing, an open-source, Java-based software, and programming language. The goal of this course is to introduce learners with no programming background to programming in a manner that is fun as well as give them enough fundamental concepts for them to explore advanced concepts on their own. There is also the opportunity to write code on a smart-phone in Processing!

Overview of Processing:

Processing is an open-source, Java-based software and programming language for creating programs. Processing is geared towards creating visual and interactive media. Hence, it is useful for teaching programming in a visual context and makes it fun to use when learning programming. Processing has been used by various groups from beginners to research labs inside technology companies like Google and Intel.

Why Processing?

We choose to introduce programming with Processing for the following reasons:
Simple and fun to learn and use:

1. Tons of libraries to use
2. Create animations and simulation
3. Create standalone programs that can run on any computer: Windows, Mac, Linux
4. Create Web, Android apps
5. Interface to hardware like Arduino

Course Project: Pong Game

At the end of this course, you would have built a Pong game. This game has 2 paddles, one for each player and a ball. Once the ball starts moving, each player has one goal; to prevent the ball from exiting the vertical wall on his/her side. If that happens, the opponent's score increases. This course will introduce you to various programming concepts through the lesson notes, exercises and assignments. You will work on assignments that will build on each other and eventually result in a fully-fledged pong game, so get excited!

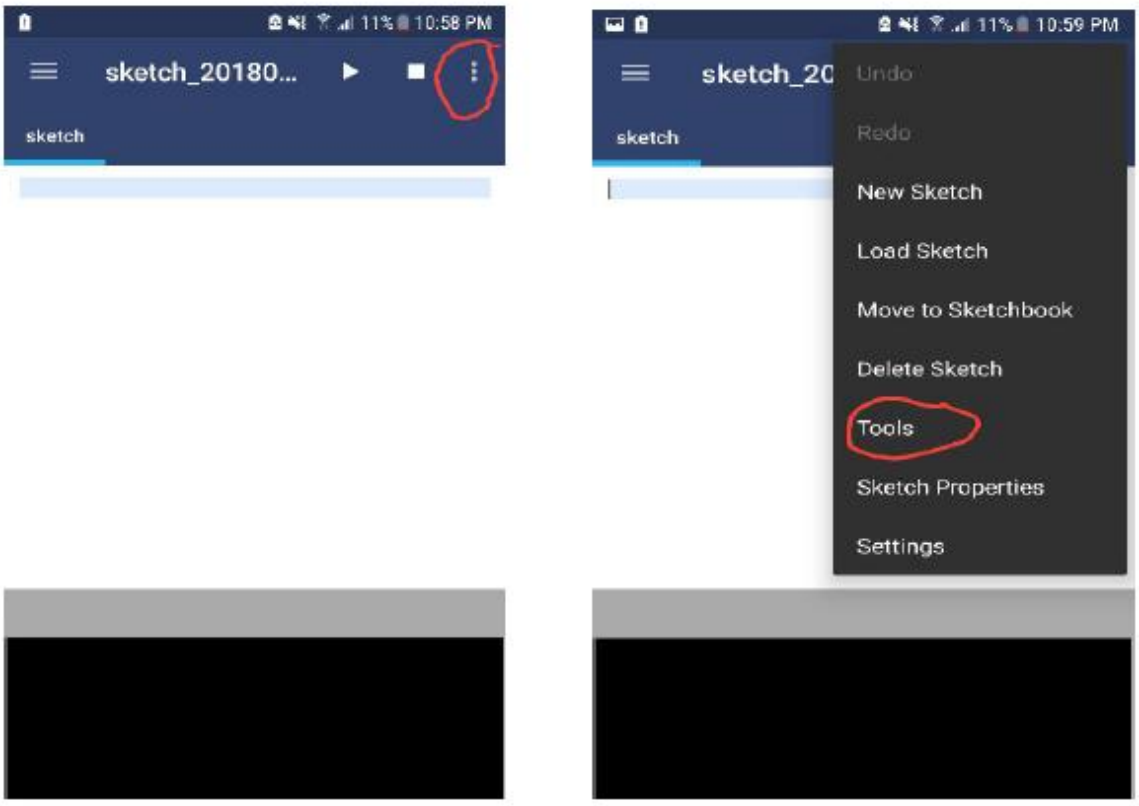
Expected Result

Figure 0.2: Pong Interface

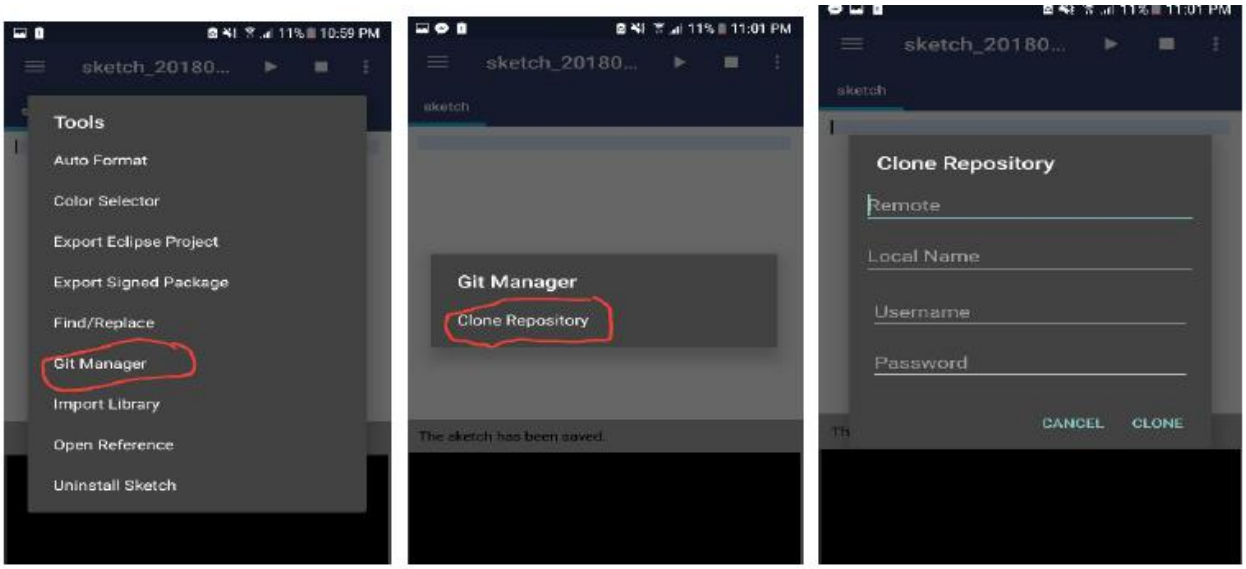
Tools and Set-ups

Android Processing Development Environment (APDE) Set Up

- Download APDE from Google PlayStore
- Click the "three dot" icon in the top right corner and choose Tools.

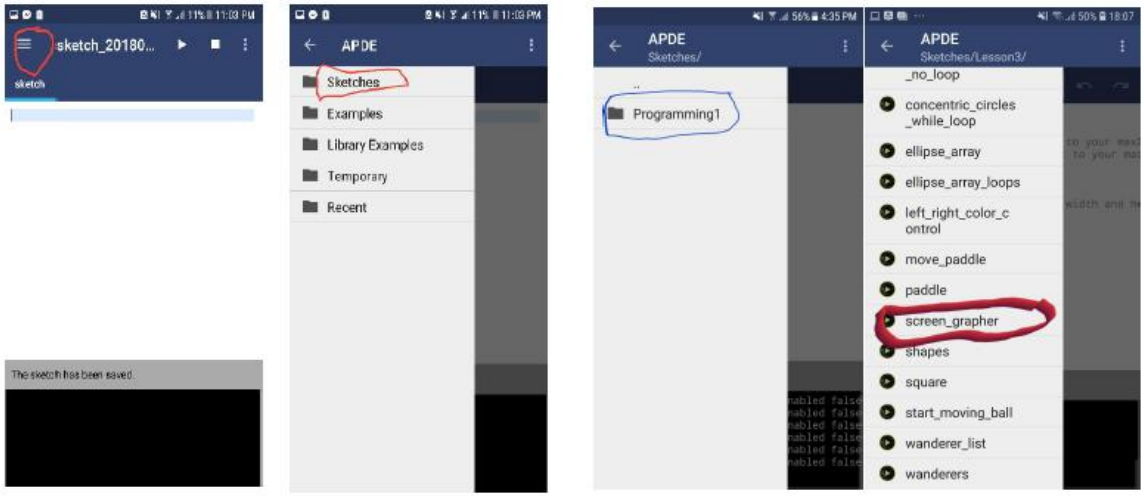


- Click Git Manager and choose Clone Repository
- In the Remote box paste the link below and make sure there're no spaces before or after the link git://github.com/Suacode-app/Suacode
- In the Local Name box type: Programming1. This is "Programming1" and **not** "Programming 1"
- Leave Username and Password **blank**
- Click **CLONE**

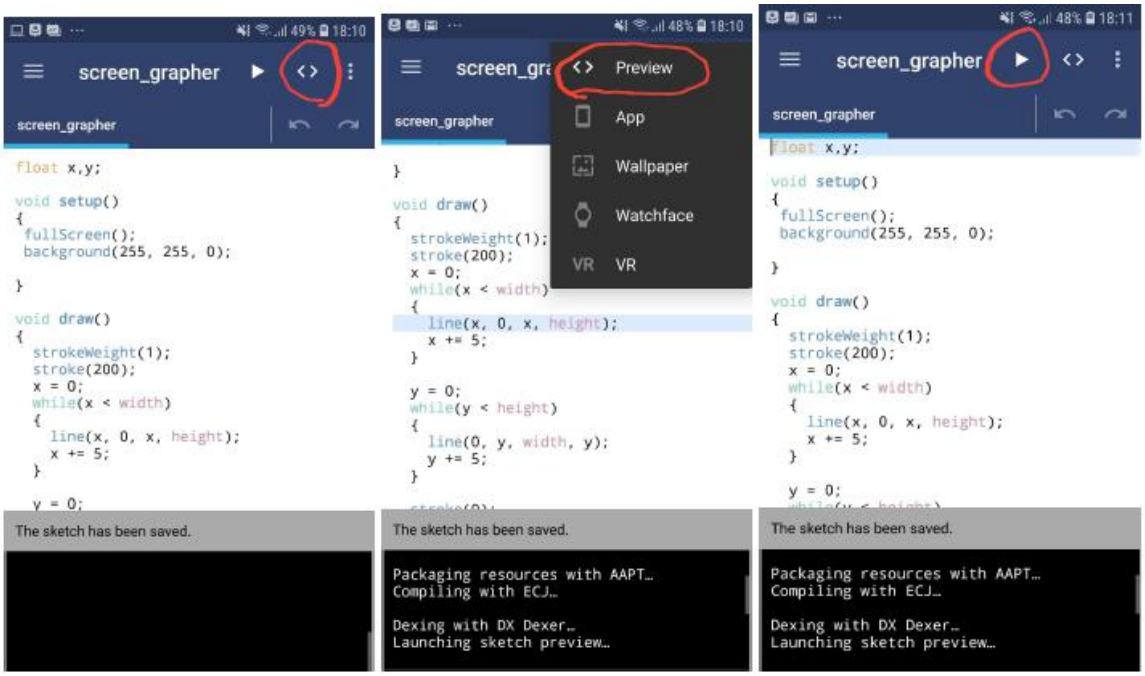


- Now click the "three bar" icon in the top left corner and choose Sketches

- Click Programming1, scroll down and click on the program called **screengrapher** (If you don't see the Programming1 folder after clicking sketches, you didn't clone successfully, so repeat the whole set up process, but this time use a different name for Local Name , don't use Programming1! Use another name like lesson1 or lesson2 etc. The folder you expect to see when you click sketches would have the same as the name you used for Local Name)



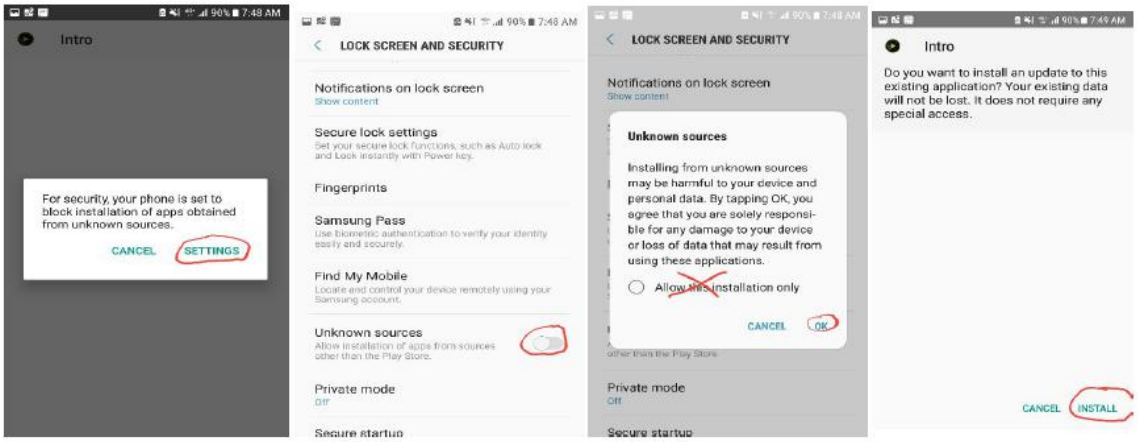
- Press the button to the right of the play button, it might be two angle brackets or a phone icon, either way it click.
- Choose preview (the angle brackets).
- Now press the play button



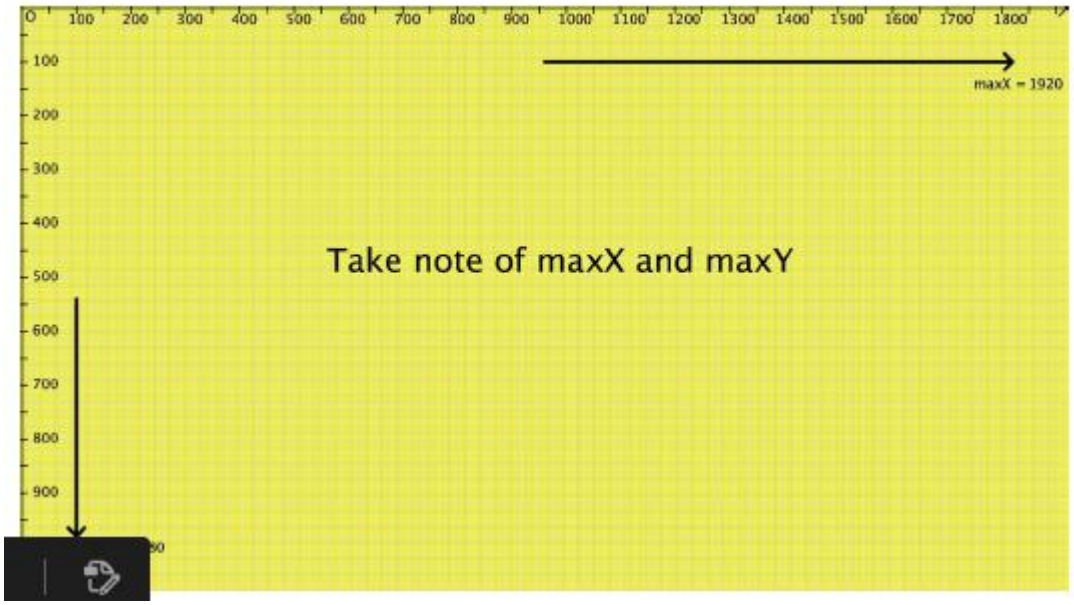
- Install Sketch viewer if you are prompted to install Sketch viewer

Security section

- If you get a security notification, then follow the steps below to fix it



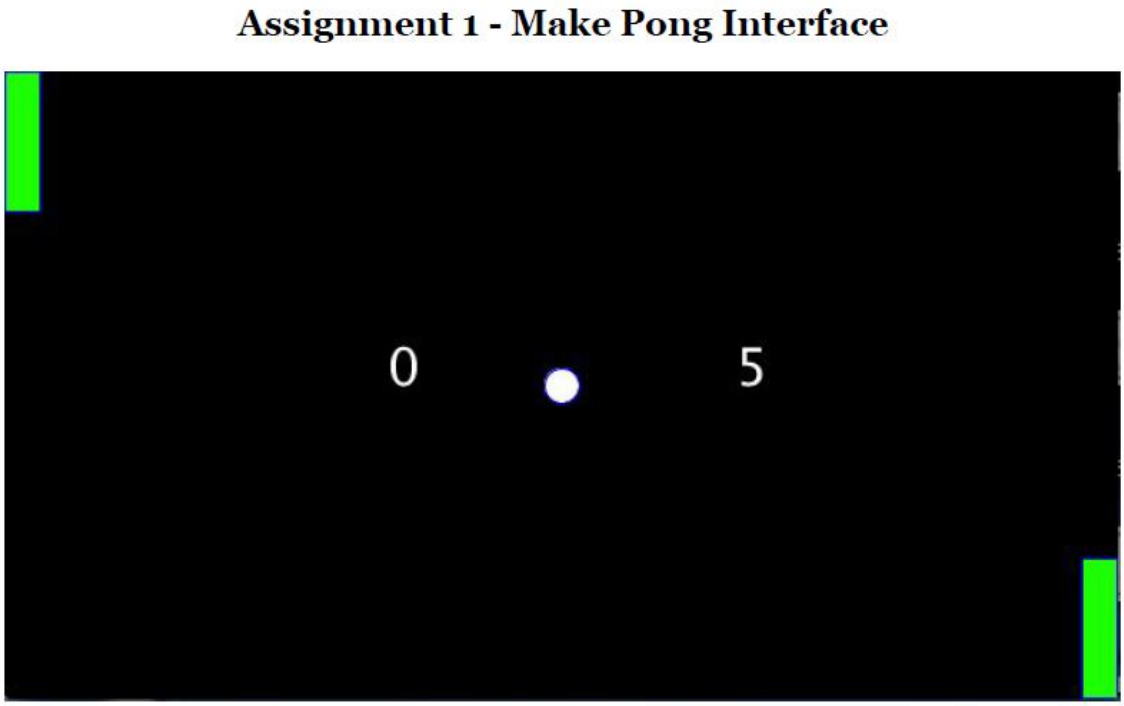
Take note of the values of the maxX and maxY values shown on your screen:



- The Processing Reference contains all the commands available to you in Processing. It can be accessed online at https://processing.org/reference/ .

Project Procedure

Assignment 1: Make Pong Interface



Instructions

Open the program Assignment1

Put your **maxX** and **maxY** values in the comment at the top of the code and use the **fullScreen()** function within your **setup()** function

Draw a Pong interface in the picture above with the following specs:

1. Two paddles, one at the top left end and one at the bottom right end
   a. pick your own width and height but they should be the same for both paddles
2. Ball at center
   a. Pick your circle's width and height
3. Write 2 arbitrary numbers, one at the left side and the other at right side of screen, representing the left and right players' scores respectively
   a. Set the size of the text
   4. Use different colors for the interior of the paddles, ball and window background
   a. Color should be the same for both paddles
   b. Set one color for the outlines all shapes including the line
5. Use ***setup()*** and ***draw()*** to organize your code
6. Add comments to your code
7. Your code looks cleaner when you group lines of code together like grouping all code of the ball together, and then the code for the paddles together, then code for the scores together, etc.

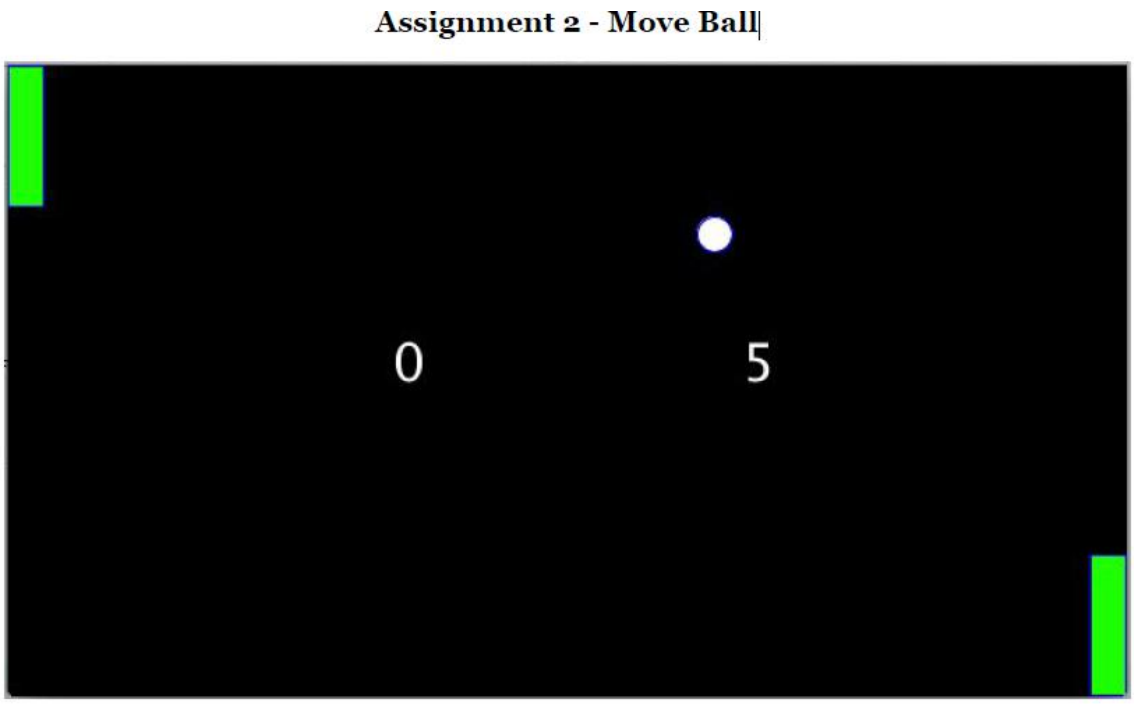
8. Indent code properly
9. Attach a screenshot of your program working ie, your pong game

Assignment 1 Solution:

```
//maxX = 1440, maxY = 720

float screenWidth = 1440; //set to your maxX
float screenHeight = 720; //set to your maxY


void setup()    //runs once

{

    fullScreen(); //Sets the program to run in full screen mode

}

void draw() //runs forever

{

    background(0);    // set the background black

    stroke(255, 0, 0);      // sets the color of the ouline of the paddles drawn below
red

    fill(0, 255, 0);      // set the color of the paddles drawn below this code green

    rect(0, 0, 50, 200); // draw the first rectangle

    rect(1390, 520, 50, 200);      // draw the second rectangle

    stroke(255, 0, 0); // sets the color of the ouline of the ball drawn below red

    fill(0, 0, 255); // set the color of the ball drawn below this code blue

    ellipse(720, 360, 40, 40); // draw a ball

    textSize(100);    // set the size of the text 100

    fill(255);    // set the color of the text white

    text("9", 360,360);    // write the first score on the screen
```

```
    text("6", 1080, 360); // write the second score on the screen

}
```

Assignment 2 - Move Ball



Instructions

Open the program Assignment2

Put your **maxX** and **maxY** values in the comment at the top of the code and use the **fullScreen()** function within your **setup()** function.

Write a program that makes the ball of your Pong game move with the following instructions:

- Modify the program to use variables to store
  - Ball's *x* and *y* positions, width, height.
  - Left paddle *x* and *y* positions,
  - Right paddle *x* and *y* positions
  - Paddle width and paddle height (remember that the left and right paddles should both have the same width and height)

- Left player score's number, x and y positions.
- Right player score's number, x and y positions.
- Text size

- Choose variable names that are descriptive and remember the camelCase naming style
- Make ball move at a certain speed and direction
  - Use variables for the ball's x speed and y speed
- Add comments to your code
- Attach a screenshot of your program working

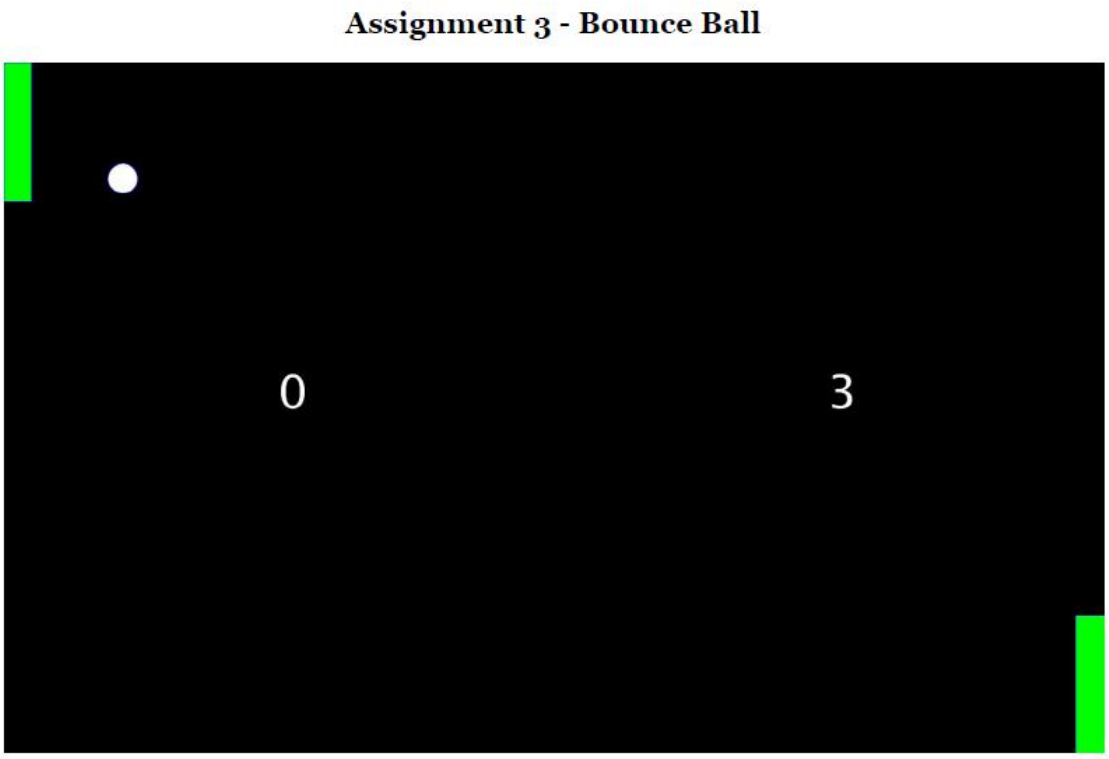
Assignment 2 Solution:

```
//maxX = 1440, maxY = 720

float screenWidth = 1440; //set to your maxX
float screenHeight = 720; //set to your maxY

int ballX = 720;    //Declear variable to hold x position of ball

int ballY = 360;    //Declear variable to hold y position of ball

int ballWidth = 40; //Declear variable to hold the ball width

int ballHeight = 40; //Declear variable to hold the ball height

int leftPaddleX = 0; //Declear variable to hold left paddle x position

int leftPaddleY = 0; //Declear variable to hold left paddle y position

int rightPaddleX = 1390; //Declear variable to hold right paddle x position

int rightPaddleY = 520; //Declear variable to hold right paddle y position

int paddleWidth = 50;    //Declear variable to hold paddle width

int paddleHeight = 200;    //Declear variable to hold paddle height

int leftScoreX = 360;    //Declear variable to hold left score x position

int leftScoreY = 360;    //Declear variable to hold left score y position

int rightScoreX = 1080;    //Declear variable to hold right score x position
```

```
int rightScoreY = 360;    //Declear variable to hold right score y position

int scoreSize = 100;    //Declear variable to hold the size of the scores

int xSpeed = 3; // ball's horizontal speed

int ySpeed = 3;    //ball's vertical speed


void setup()    //runs once
{
   fullScreen(); //Sets the program to run in full screen mode

}


void draw() //runs forever
{
   background(0);    // set the background black

   stroke(255, 0, 0);      // sets the color of the ouline of the paddles drawn below
red

   fill(0, 255, 0);      // set the color of the paddles drawn below this code green

   rect(leftPaddleX, leftPaddleY, paddleWidth, paddleHeight); // draw the first
rectangle

   rect(rightPaddleX, rightPaddleY, paddleWidth, paddleHeight);      // draw the
second rectangle

   stroke(255, 0, 0); // sets the color of the ouline of the ball drawn below red

   fill(0, 0, 255); // set the color of the ball drawn below this code blue

   ellipse(ballX, ballY, ballWidth, ballHeight); // draw a ball

   textSize(scoreSize);    // set the size of the text 100

   fill(255);    // set the color of the text white

   text("9", leftScoreX, leftScoreY);    // write the first score on the screen

   text("6", rightScoreX, rightScoreY); // write the second score on the screen

   //move the up and right the ball
```

```
    ballX = ballX + xSpeed;

    ballY = ballY - ySpeed;

}
```

Assignment 3 - Bounce Ball



Instructions

Open the program Assignment3

Put your ***maxX*** and ***maxY*** values in the comment at the top of the code and use the ***fullScreen()*** function within your ***setup()*** function. Write a program that makes the ball of your Pong game bounce off the top and bottom walls, but exits the left and right walls using the following instructions:

- If ball hits top or bottom wall, reverse the ball's direction
  - ***Hint:*** Look at ***bounce_ball***
  - Make sure to test thoroughly by making sure the ball goes in the direction of all four walls. You can do this by changing which direction the ball starts

moving towards

- If the screen is touched, set the game to be on
  - ***Hint 1:*** Use the ***mousePressed*** built-in variable and also use a variable you define (such as ***gameOn*** of type boolean) to store whether the game is on (mouse has been pressed) or the game is off (the ball has exited the left or right wall)
- If the ball hits right wall, increment left player's score and set the game to be off
- If the ball hits left wall, increment right player's score and set the game to be off
  - Make sure the scores of both players are initialized to 0
- Ball should move only if the ***gameOn*** is true, else it should be reset to the center
- As always, use descriptive variable names and add comments to your code
- Attach a screenshot of your program working.

Assignment 3 Solution:

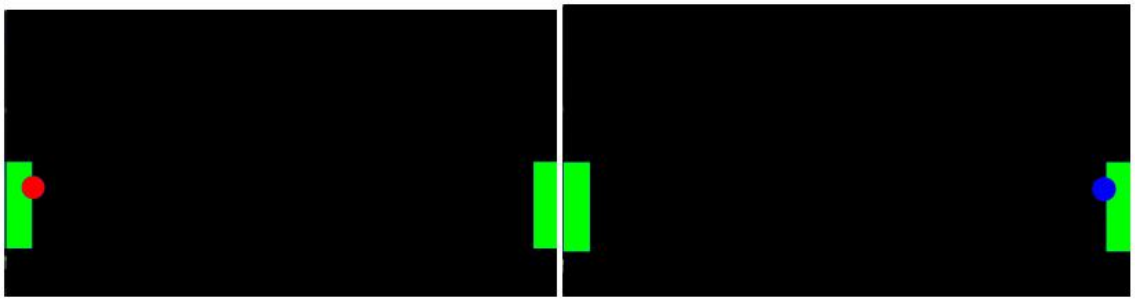
```
//maxX = 1440, maxY = 720

float screenWidth = 1440; //set to your maxX
float screenHeight = 720; //set to your maxY

float ballX = 720;    //Declear variable to hold x position of ball

float ballY = 360;    //Declear variable to hold y position of ball

int ballWidth = 60; //Declear variable to hold the ball width

int ballHeight = 60; //Declear variable to hold the ball height

int leftPaddleX = 0; //Declear variable to hold left paddle x position

int leftPaddleY = 0; //Declear variable to hold left paddle y position

int rightPaddleX = 1390; //Declear variable to hold right paddle x position

int rightPaddleY = 520; //Declear variable to hold right paddle y position

int paddleWidth = 50;    //Declear variable to hold paddle width

int paddleHeight = 200;    //Declear variable to hold paddle height
```

```
int leftScoreX = 360;    //Declear variable to hold left score x position

int leftScoreY = 360;    //Declear variable to hold left score y position

int rightScoreX = 1080;    //Declear variable to hold right score x position

int rightScoreY = 360;    //Declear variable to hold right score y position

int scoreSize = 100;    //Declear variable to hold the size of the scores

float xSpeed = 5; // ball's horizontal speed

float ySpeed = -5;    //ball's vertical speed

int radius = 30;    //since half the width or the height of the ball is the radius

boolean gameStart = false;    //new variable to start ball's movement

int firstScore = 0;    // variable to hold left score

int secondScore = 0;    // variable to hold right score


void setup()    //runs once
{
   fullScreen(); //Sets the program to run in full screen mode

}


void draw() //runs forever
{
   background(255, 0, 255);    // set the background black

   stroke(255, 0, 0);      // sets the color of the ouline of the paddles drawn below
red

   fill(0, 255, 0);      // set the color of the paddles drawn below this code green

   rect(leftPaddleX, leftPaddleY, paddleWidth, paddleHeight); // draw the first
rectangle

   rect(rightPaddleX, rightPaddleY, paddleWidth, paddleHeight);      // draw the
second rectangle

   stroke(0, 255, 0); // sets the color of the ouline of the ball drawn below red
```

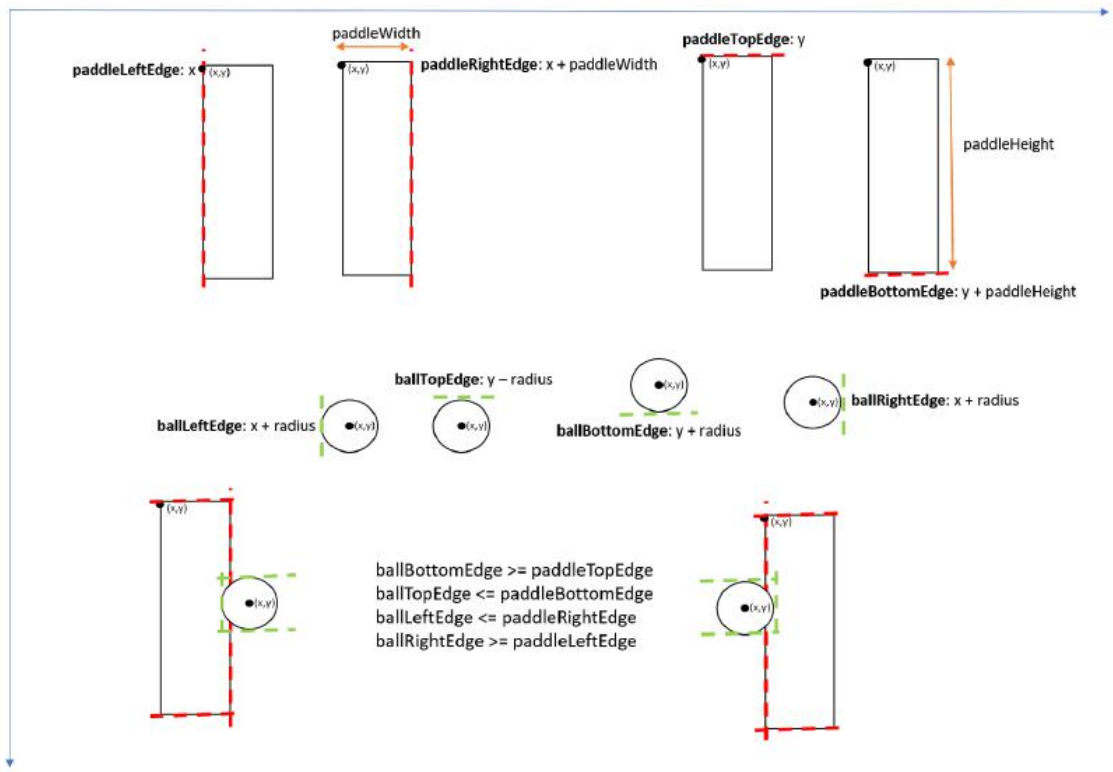
```
   fill(0, 150, 255); // set the color of the ball drawn below this code blue

   ellipse(ballX, ballY, ballWidth, ballHeight); // draw a ball

   textSize(scoreSize);    // set the size of the text 100

   fill(255);    // set the color of the text white

   text(firstScore, leftScoreX, leftScoreY);    // write the first score on the screen

   text(secondScore, rightScoreX, rightScoreY); // write the second score on the
screen

   //Check if mouse is pressed, set gameOn to true
   if (mousePressed) {
      gameStart = true;
   }

   //Move ball
   if (gameStart) {
      ballX = ballX+xSpeed;
      ballY = ballY+ySpeed;
   }

   //Check if ball hits top or bottom walls
   if ((ballX-radius < 0)|| (ballX+radius) > screenWidth) {

      xSpeed=xSpeed*-1;       //Reverse direction
   }


   if ((ballY-radius < 0) || (ballY+radius) > screenHeight) {

     ySpeed = ySpeed * -1;    //Reverse direction

   }

  // condition if the ball hit the left wall of the screen
  if ((ballX+radius) > screenWidth) {
     firstScore = firstScore + 1;    // if the condition is true, left score increase
     ballX = screenWidth / 2;    //maxX divided by 2
     ballY = screenHeight / 2; //maxY divided by 2
     gameStart = false;

  }
```

```
  // condition if the ball hit the left wall of the screen
 if ((ballX-radius) < 0){
    secondScore = secondScore + 1; // if the condition is true, right score increase
    ballX = screenWidth/2;      //maxX divided by 2
    ballY = screenHeight/2;      //maxY divided by 2
    gameStart = false;
 }


}
```

Snapshot ball turning red when overlapping left paddle and blue when overlapping right paddle.

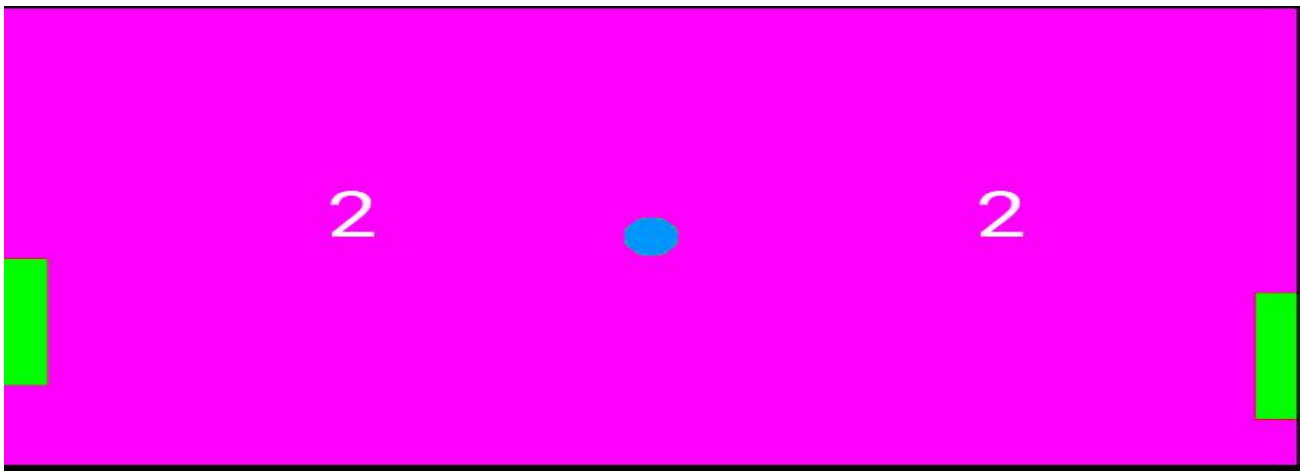


Edge overlap of ball and paddle



Assignment 4 - move the paddles and bounce the ball off the paddle

How will you use this function to check if your game ball has hit a paddle?



Assignment 4 Solution:

```
//maxX = 1440, maxY = 720
```

```
float screenWidth = 1440; //set to your maxX
float screenHeight = 720; //set to your maxY

float ballX = 720;    //Declear variable to hold x position of ball

float ballY = 360;    //Declear variable to hold y position of ball

float halfScreenWidth = 720;    // maxX/2 = 720

float ballWidth = 60; //Declear variable to hold the ball width

float ballHeight = 60; //Declear variable to hold the ball height

float leftPaddleX = 0; //Declear variable to hold left paddle x position

float leftPaddleY = 0; //Declear variable to hold left paddle y position

float rightPaddleX = 1390; //Declear variable to hold right paddle x position

float rightPaddleY = 520; //Declear variable to hold right paddle y position

float paddleWidth = 50;    //Declear variable to hold paddle width

float paddleHeight = 200;    //Declear variable to hold paddle height

float leftScoreX = 360;    //Declear variable to hold left score x position

float leftScoreY = 360;    //Declear variable to hold left score y position

float rightScoreX = 1080;    //Declear variable to hold right score x position

float rightScoreY = 360;    //Declear variable to hold right score y position

float scoreSize = 100;    //Declear variable to hold the size of the scores

float xSpeed = 10; // ball's horizontal speed

float ySpeed = -10;    //ball's vertical speed

float radius = 30;    //since half the width or the height of the ball is the radius

boolean gameStart = false;    //new variable to start ball's movement

boolean hasOverLappedLeftPaddle = false;

boolean hasOverLappedRightPaddle = false;
```

```
int firstScore = 0;    // variable to hold left score

int secondScore = 0;    // variable to hold right score


void setup()    //runs once
{
   fullScreen(); //Sets the program to run in full screen mode

}


void draw() //runs forever
{
   background(255, 0, 255);    // set the background black

   stroke(255, 0, 0);      // sets the color of the ouline of the paddles drawn below
red

   displayPaddles();    //function to display the paddles
   displayBall();          //function to display the ball
   displayScores();      //function to display the scores
   setGameMode();          //function to set the game on
   moveBall();             //function to move the ball
   checkWalls();        //function to check the walls
   movePaddles();      //function to move the paddles
   checkLeftPaddle(); //function to check the left paddle
   checkRightPaddle(); //function to check the right paddle
}


//To display the paddles
void displayPaddles(){
   fill(0, 255, 0);      // set the color of the paddles drawn below this code green
   rect(leftPaddleX, leftPaddleY, paddleWidth, paddleHeight); // draw the first
rectangle
   rect(rightPaddleX, rightPaddleY, paddleWidth, paddleHeight);      // draw the
second rectangle

}

//To display the ball
void displayBall(){
   stroke(0, 255, 0); // sets the color of the ouline of the ball drawn below red
```

```
    fill(0, 150, 255); // set the color of the ball drawn below this code blue
    ellipse(ballX, ballY, ballWidth, ballHeight); // draw a ball

}

//To display the scores
void displayScores(){
    textSize(scoreSize);    // set the size of the text 100
    fill(255);    // set the color of the text white
    text(firstScore, leftScoreX, leftScoreY);    // write the first score on the screen
    text(secondScore, rightScoreX, rightScoreY); // write the second score on the
screen

}

// To set the Game mode
void setGameMode(){
    //Check if mouse is pressed, set gameOn to true
    if (mousePressed) {
        gameStart = true;
    }

}

//To move the ball on the screen
void moveBall(){
    //Move ball
    if (gameStart) {
        ballX = ballX+xSpeed;
        ballY = ballY+ySpeed;
    }

}

//To check if the ball hits the top, down, right and left wall of the screen
void checkWalls(){
    //Check if ball hits top or bottom walls
    if ((ballX-radius < 0)|| (ballX+radius) > screenWidth) {

        xSpeed=xSpeed*-1;        //Reverse direction
    }


    if ((ballY-radius < 0) || (ballY+radius) > screenHeight) {

      ySpeed = ySpeed * -1;    //Reverse direction
```

```
   }

  // condition if the ball hit the left wall of the screen
  if ((ballX+radius) > screenWidth) {
     firstScore = firstScore + 1;    // if the condition is true, left score increase
     ballX = screenWidth / 2;    //maxX divided by 2
     ballY = screenHeight / 2; //maxY divided by 2
     gameStart = false;

  }


    // condition if the ball hit the left wall of the screen
  if ((ballX-radius) < 0){
     secondScore = secondScore + 1; // if the condition is true, right score increase
     ballX = screenWidth/2;      //maxX divided by 2
     ballY = screenHeight/2;      //maxY divided by 2
     gameStart = false;
  }

}

//To move the paddles
void movePaddles(){
   if(mouseX < halfScreenWidth) {
      leftPaddleY = constrain(mouseY, 0, screenHeight - paddleHeight);
   }else{
      rightPaddleY = constrain (mouseY, 0, screenHeight- paddleHeight);
   }

}

  //Check if there is an overlap between ball and left paddle
void checkLeftPaddle(){
      hasOverLappedLeftPaddle = doesOverlap(leftPaddleX, leftPaddleY,
paddleWidth, paddleHeight, ballX, ballY, radius);
     if(hasOverLappedLeftPaddle){
       xSpeed = xSpeed * -1;

     }
  }

//Check if there is an overlap between ball and right paddle

void checkRightPaddle() {
   hasOverLappedRightPaddle = doesOverlap(rightPaddleX, rightPaddleY,
paddleWidth, paddleHeight, ballX, ballY, radius);
```

```
    if(hasOverLappedRightPaddle) {
       xSpeed = xSpeed * -1;
          }
     }
      boolean doesOverlap(float xPaddle, float yPaddle, float widthPaddle, float
heightPaddle, float xBall, float yBall, float radius) {

    float ballLeftEdge = xBall-radius; //left edge of ball
    float ballBottomEdge = yBall+radius; //bottom edge of ball
    float ballRightEdge = xBall+radius; //right edge of ball
    float ballTopEdge = yBall-radius; //top edge of ball

    float paddleLeftEdge = xPaddle; //left edge of paddle
    float paddleBottomEdge = yPaddle+heightPaddle; //bottom edge of paddle
    float paddleRightEdge = xPaddle+widthPaddle; //right edge of paddle
    float paddleTopEdge = yPaddle; //top edge of paddle

    if (ballBottomEdge >= paddleTopEdge //Check if bottom edge of ball and top
edge of paddle overlap
        && ballTopEdge <= paddleBottomEdge //Check if top edge of ball and
bottom edge of paddle overlap
        && ballLeftEdge <= paddleRightEdge //Check if left edge of ball and right
edge of paddle overlap
        && ballRightEdge >= paddleLeftEdge ) //Check if right edge of ball and left
edge of paddle overlap
    {
        return true;
    } else {
        return false;
    }

}
```

Summary of assignment 4

In this assignment, I:

- Created and use functions
- Made the paddle move
- Check if the ball overlaps paddle

Finally:

I now move the paddles and bounce the ball off the paddle.

Extra aids: https://www.youtube.com/playlist?list=PLRqwX-V7Uu6ajGB2OI3hl5DZsD1Fw1WzR