# Functions and Loops

## What is our GOAL for this MODULE?

We used our knowledge of functions to create custom functions to serve the ball, reset the ball, and draw the net.

## What did we ACHIEVE in the class TODAY?

- Wrote custom functions to serve the ball, reset the ball, and draw the net.
- Drew the net using line instruction and for-loop.

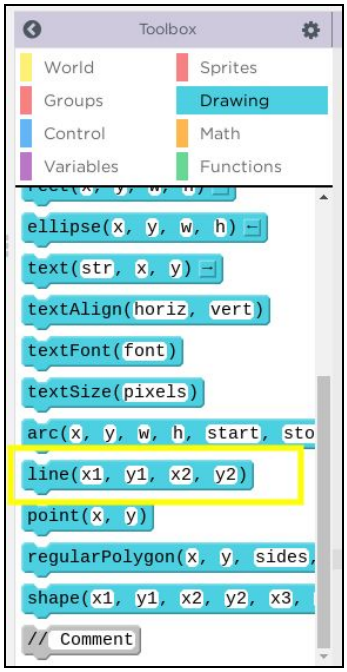
## Which CONCEPTS/ CODING BLOCKS did we cover today?

- Custom functions

## How did we DO the activities?

***Note:** In coding, we have a principle D-R-Y : Don't Repeat Yourself.
**Remember:** Good Programmers don't like to repeat themselves while writing code.

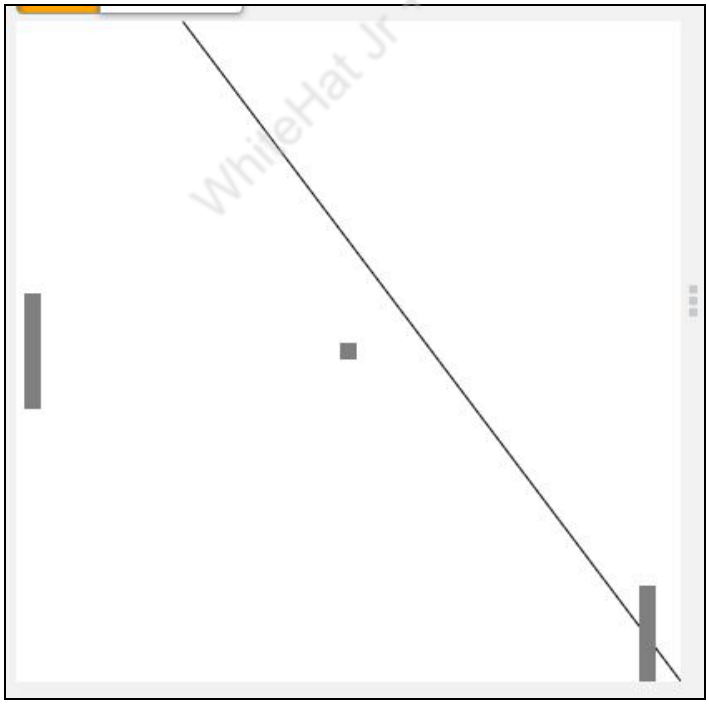
1. Use a pre-defined **line()** instruction(a function).



2. Draw a line by giving the start and the end coordinates.
    - Code:

```
var playerPaddle = createSprite(380,200,10,70);
var computerPaddle = createSprite(10,200,10,70);



function draw() {
  //clear the screen
  background("white");

  //make the player paddle move with the mouse's y position
  playerPaddle.y = World.mouseY;

  //AI for the computer paddle
  //make it move with the ball's y position
  computerPaddle.y = ball.y;


  line(100, 0, 400, 400);

  //create edge boundaries
  //make the ball bounce with the top and the bottom edges
  createEdgeSprites();
  ball.bounceOff(topEdge);
  ball.bounceOff(bottomEdge);

  //make the ball bounce off the paddles
  ball.bounceOff(playerPaddle);
  ball.bounceOff(computerPaddle);

```

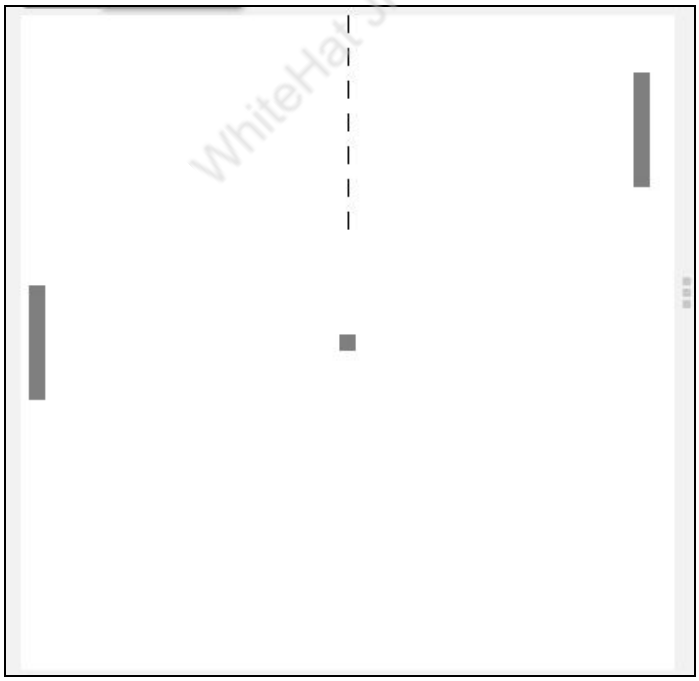
    - Output:



3. Make several small (dashed) lines with height 10 and leave a gap of 10 after every dash.
    - Code:

```
//make the player paddle move with the mouse's y position
playerPaddle.y = World.mouseY;

//AI for the computer paddle
//make it move with the ball's y position
computerPaddle.y = ball.y;


line(200,0,200,0+10);
line(200,0+20,200,0+20+10);
line(200,0+20+20,200,0+20+20+10);
line(200,0+20+20+20,200,0+20+20+20+10);
line(200,0+20+20+20+20,200,0+20+20+20+20+10);
line(200,0+20+20+20+20+20,200,0+20+20+20+20+20+10);
line(200,0+20+20+20+20+20+20,200,0+20+20+20+20+20+20+10);

//create edge boundaries
//make the ball bounce with the top and the bottom edges
createEdgeSprites();
bounceOff(topEdge, bottomEdge, playerPaddle,computerPaddle);


//serve the ball when space is pressed
if (keyDown("space")) {
  ball.velocityY = 3;
  ball.velocityX = 4;
}
```

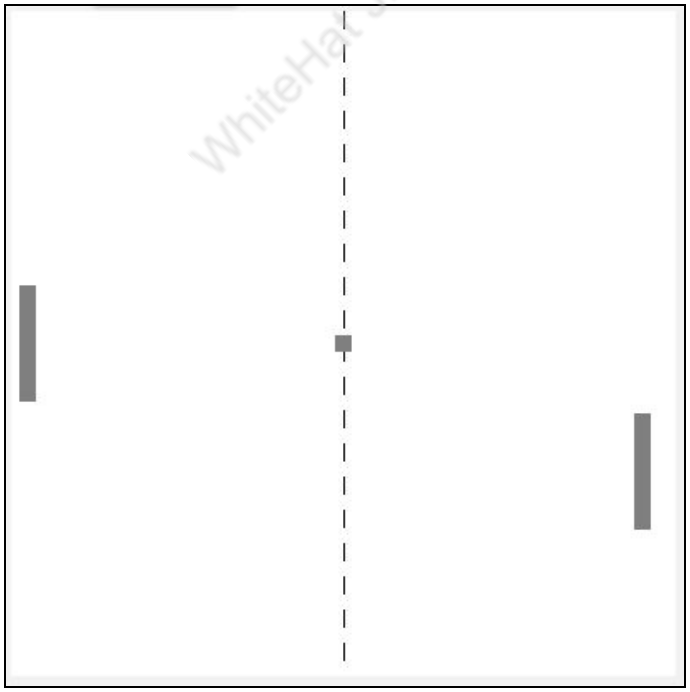
    - Output:



4. Use *For Loop* to run the same instructions without repeating the code.
    - Code:

```
12  //make the player paddle move with the mouse's y position
13  playerPaddle.y = World.mouseY;
14
15  //AI for the computer paddle
16  //make it move with the ball's y position
17  computerPaddle.y = ball.y;
18
19  for (var num = 0; num < 400; num = num +20) {
20    line(200,num, 200, num+10);
21  }
22
23
24
25  //create edge boundaries
26  //make the ball bounce with the top and the bottom edges
27  createEdgeSprites();
28  bounceOff(topEdge, bottomEdge, playerPaddle,computerPaddle);
29
30
31  //serve the ball when space is pressed
32  if (keyDown("space")) {
33    ball.velocityY = 3;
34    ball.velocityX = 4;
35  }
36
37
38  //reset the ball to the centre if it crosses the screen
39  if(ball.x > 400 || ball.x <0) {
40    ball.x = 200;
41    ball.y = 200;
42
```

    - Output:



5. Teach the computer to draw the net using a custom-defined function **drawnet()**.
    - Code:

```
function draw() {
  //clear the screen
  background("white");

  //make the player paddle move with the mouse's y position
  playerPaddle.y = World.mouseY;

  //AI for the computer paddle
  //make it move with the ball's y position
  computerPaddle.y = ball.y;

  drawnet();

  for (var num = 0; num < 400; num = num +20) {
    line(200,num, 200, num+10);
  }

  //create edge boundaries
  //make the ball bounce with the top and the bottom edges
  createEdgeSprites();

  //serve the ball when space is pressed
  if (keyDown("space")) {
    ball.velocityY = 3;
    ball.velocityX = 4;
```

6. Teach the computer to serve the ball and reset the ball by writing custom-defined functions.

```
34      resetball();
35    }
36
37    ball.bounceOff(topEdge);
38    ball.bounceOff(bottomEdge);
39    ball.bounceOff(playerPaddle);
40    ball.bounceOff(computerPaddle);
41
42    drawSprites();
43  }
44
45  function drawnet() {
46
47    for (var num = 0; num < 400; num = num +20) {
48     line(200,num, 200, num+10);
49   }
50
51  }
52
53  function serveball() {
54    ball.velocityY = 3;
55    ball.velocityX = 4;
56  }
57
58  function resetball() {
59      ball.x = 200;
60      ball.y = 200;
61      ball.velocityX = 0;
62      ball.velocityY = 0;
63  }
```

## What's next?

We will understand the different states of a game. We will also learn how to store information about the in-game states.