Object-Oriented Programming

What is our GOAL for this MODULE?

We learned to think about programming in an object-oriented manner. We designed Paddle and Ball class and created objects using the class to use it in our program.

What did we ACHIEVE in the class TODAY?

- Designed a Paddle and Ball class.
- Created objects using Paddle and Ball class and used it in the program.
- Stored objects in variables.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Variables
- Class
- Object

How did we DO the activities?

Activity 1: Adding comments to codes.

Coding needs to be easily understandable and readable. A good programmer always adds comments for this reason.

Code: Comments added to the screen

```
function draw() {
  //clear the screen
   background("white");
   //draw the Player Paddle
   rect(390, World.mouseY, 10, 70);
   //draw the Computer Paddle
   rect(0, 150, 10, 70 );
   //draw the ball
   rect(200,200,10,10);
}
```

Activity 2: Object-Oriented Programming (OOP)

- In this programming style, we write code as if everything in the programming world were an object - just like in the real world.
- In the real world, everything around us is an object. Each object has some properties and functions.
    - PROPERTIES are the qualities/characteristics of the object.
    - FUNCTIONS are something which the object can do”

Activity 3: Creating Paddle and Ball game using OOP

1. A programmer creating a paddle object would first make the design and assign all the properties and functions of the object to it. Based on this design, the programmer will create as many paddle objects as he/she wants in the game.
    - The design is called a CLASS in programming.

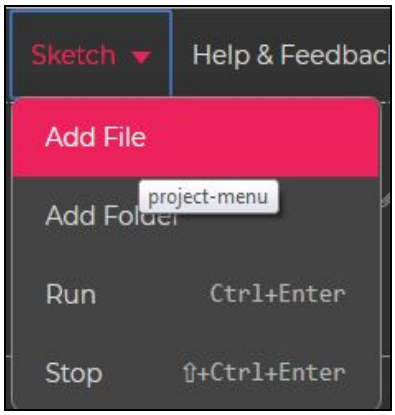
2. Create a blueprint of an object (class) and make an object using the blueprint.

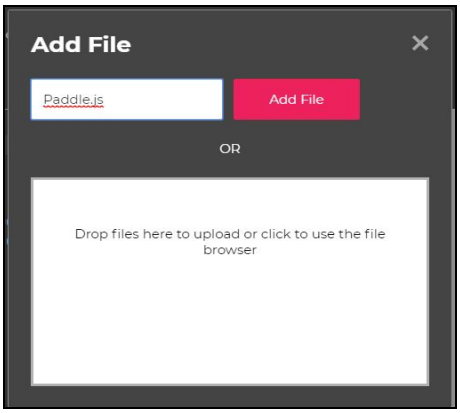
3. Define the properties.

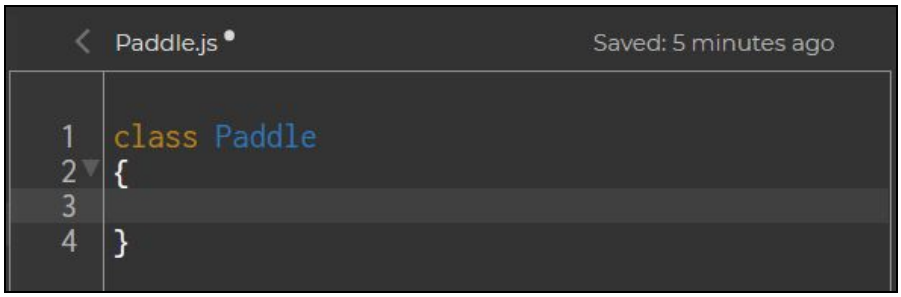
```
function Paddle(){
  this.width = 10;
  this.height = 70;
  this.xPosition = 0;
  this.yPosition= 0;
}

function draw() {
   //set background to white
  background("white");

  //draw the player Paddle
  rect(390,World.mouseY, 10,70);

  //draw the computer Paddle

  //draw the ball
  rect(200,200,10,10);
}
```

4. Create a PlayerPaddle object using the Paddle class.
5. Assign properties to the paddle design.

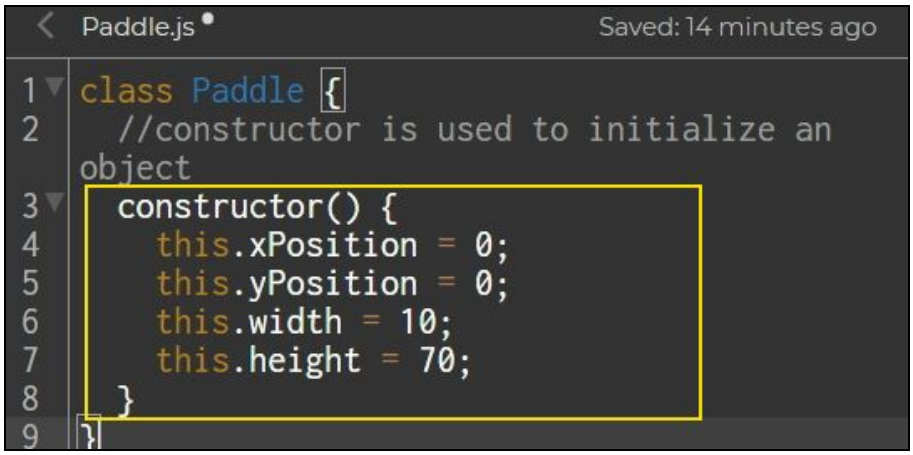
6. Add a new file:







7. Assign all the properties of the paddle inside our Paddle class/design.



8. Tell the computer where to find Paddle Class.

```
    src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5
    .js"></script>
5       <script
    src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/ad
    dons/p5.dom.min.js"></script>
6       <script
    src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/ad
    dons/p5.sound.min.js"></script>
7       <link rel="stylesheet" type="text/css"
    href="style.css">
8       <meta charset="utf-8" />
9       <script src="Paddle.js"></script>
10    </head>
11    <body>
12      <script src="sketch.js"></script>
13    </body>
14  </html>
15
```

9. Create a playerPaddle object using the Paddle class.

*__Note__: Variables are memory spaces where computers store objects.

10. Delete the old rect() instruction for the player paddle and let us try to create a new player paddle object based on the Paddle class/design we just defined
new Paddle();

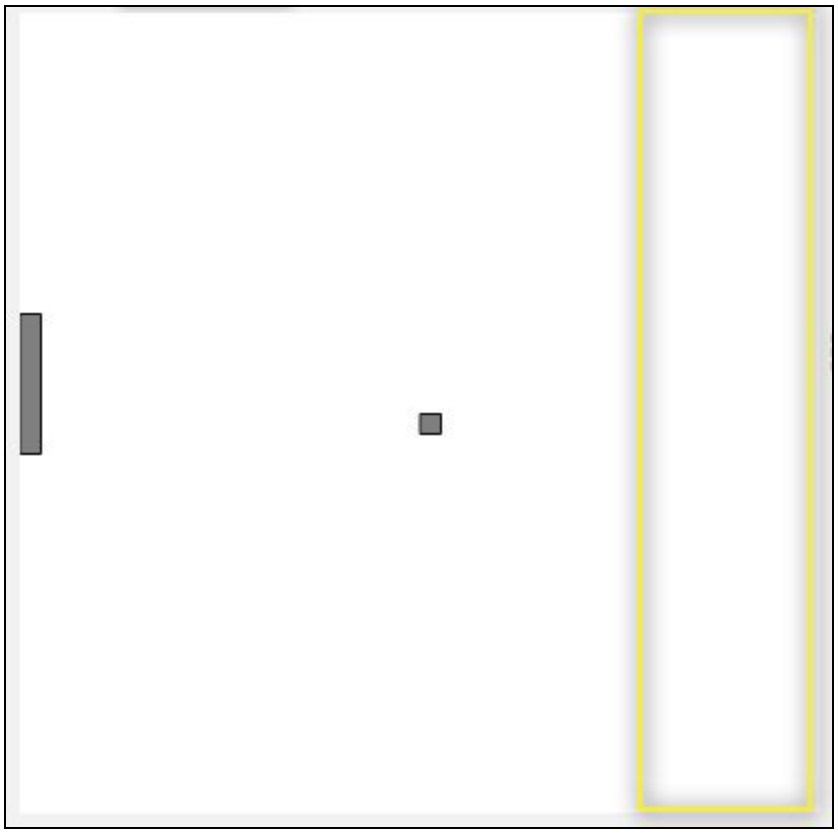
    Remember we use a semicolon to tell the computer that this is the end of the line. Just like a full-stop for English language.

sketch.js — Saved: 1 minute ago

```
1  var playerPaddle;
2
3  function setup() {
4    createCanvas(400, 400);
5    playerPaddle = new Paddle();
6  }
7
```

11. Change xPosition and yPosition properties for the playerPaddle object.
    - playerPaddle.xPosition = 390;
    - playerPaddle.yPosition = World.mouseY;

12. Run the code.



13. Write display function for the playerPaddle object to display the paddle.
    - Add this line inside the Paddle class
      display() {
      rect(this.xPosition,this.yPosition,this.width,this.height) ;
      }

Paddle.js — Saved: 19 minutes ago

```
class Paddle {
  //constructor is used to initialize an object
  constructor() {
    this.xPosition = 0;
    this.yPosition = 0;
    this.width = 10;
    this.height = 70;
  }
  display() {
    rect(this.xPosition, this.yPosition, this.width, this.height);
  }
}
```

14. Add a playerPaddle.display() inside the draw().

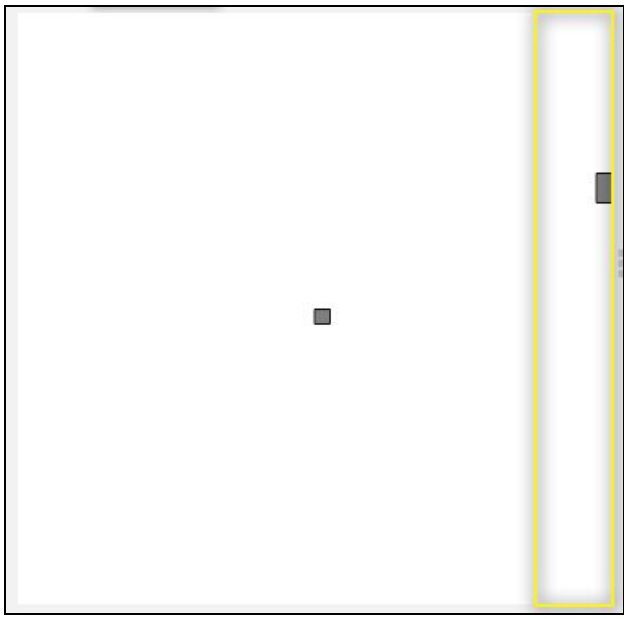
sketch.js — Saved: 5 minutes ago

```
var playerPaddle;

function setup() {
  createCanvas(400, 400);
  playerPaddle = new Paddle();
}

function draw() {
  background("white");
  playerPaddle.xPosition=390;
  playerPaddle.yPosition=mouseY;
  playerPaddle.display();
  rect(0,165,10,70);
  rect(200,200,10,10);
}
```

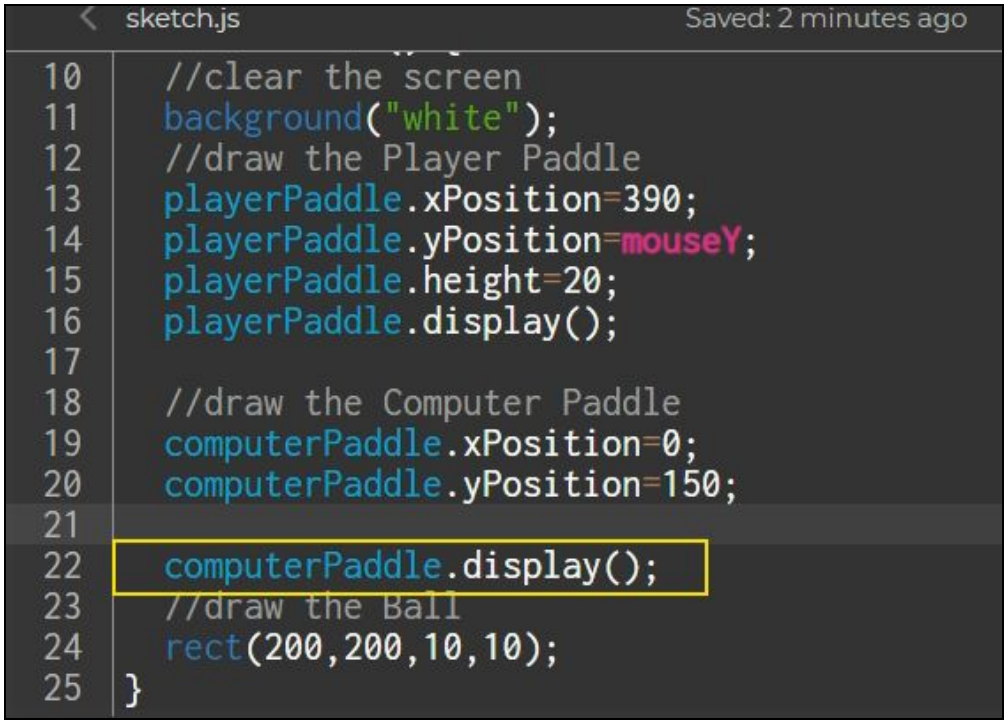
15. Change the width and height of the PlayerPaddle object.

```
sketch.js                                   Saved: 1 minute ago
 5    playerPaddle = new Paddle();
 6  }
 7
 8  function draw() {
 9    //clear the screen
10    background("white");
11    //draw the Player Paddle
12    playerPaddle.xPosition=390;
13    playerPaddle.yPosition=mouseY;
14    playerPaddle.height=20;
15    playerPaddle.display();
16
17    //draw the Computer Paddle
18
19    //draw the Ball
20    rect(200,200,10,10);
```

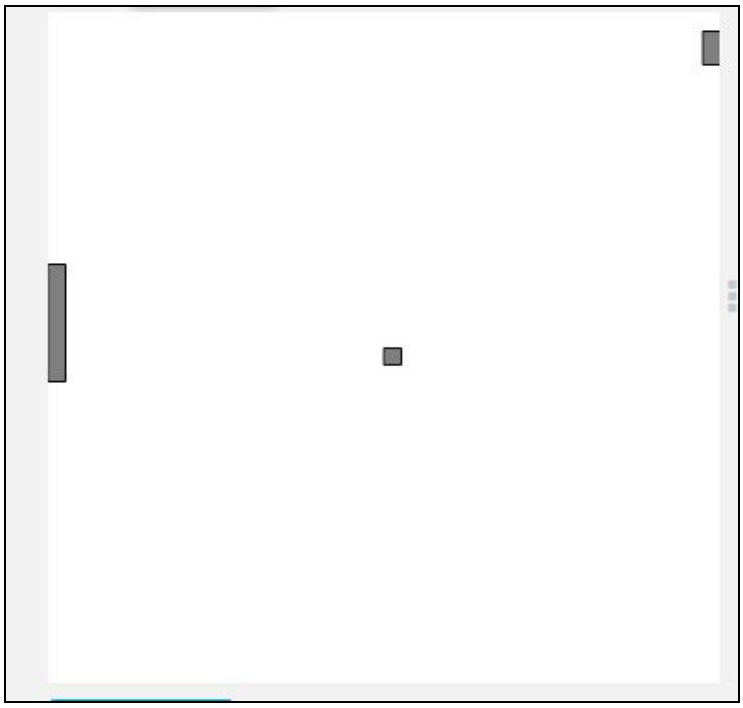
16. Run the code to see the output.



17. Create another paddle using the same steps.



18. Run and see the output.

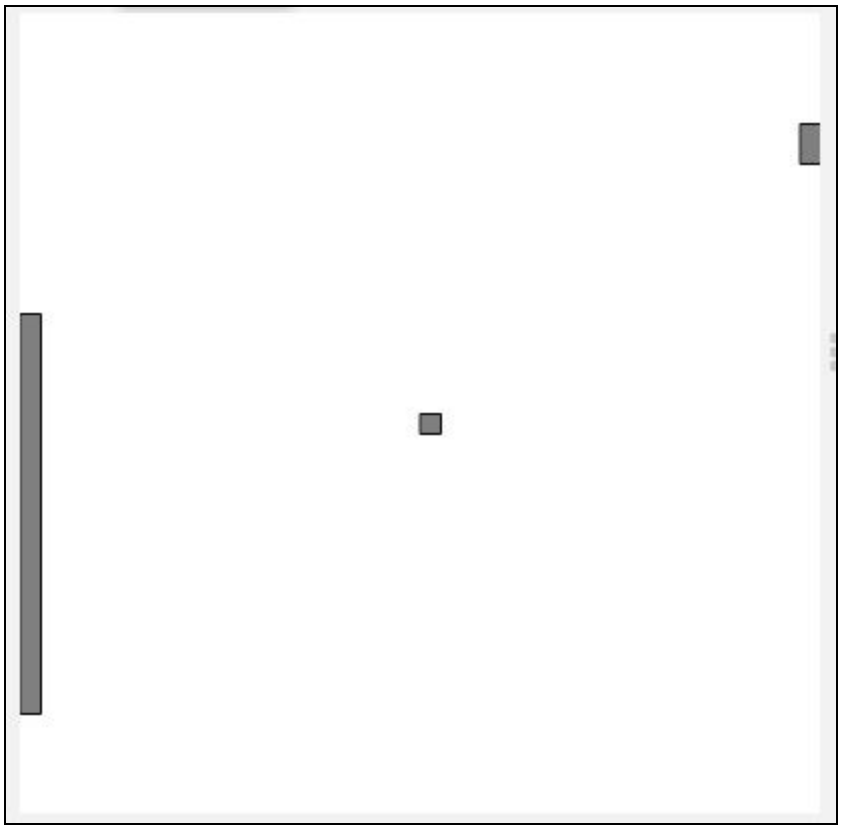


19. Modify the ComputerPaddle's width and height.

sketch.js • Saved: 4 minutes ago

```
10  //clear the screen
11  background("white");
12  //draw the Player Paddle
13  playerPaddle.xPosition=390;
14  playerPaddle.yPosition=mouseY;
15  playerPaddle.height=20;
16  playerPaddle.display();
17
18  //draw the Computer Paddle
19  computerPaddle.xPosition=0;
20  computerPaddle.yPosition=150;
21
22  computerPaddle.height=200;
23
24  computerPaddle.display();
25  //draw the Ball
```

20. Run and see the final output.



What's NEXT?

We will continue to create more objects and assign additional properties to it, like make the ball bounce!

EXTEND YOUR KNOWLEDGE

- OOPS Philosophy
  https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/