

### What is our GOAL for this MODULE?

We used our knowledge algorithms to write a collision detection algorithm and used boilerplate code.

# What did we ACHIEVE in the class TODAY?

- Downloaded the Boilerplate code.
- Designed an algorithm for collision detection, wrote code and tested the program.

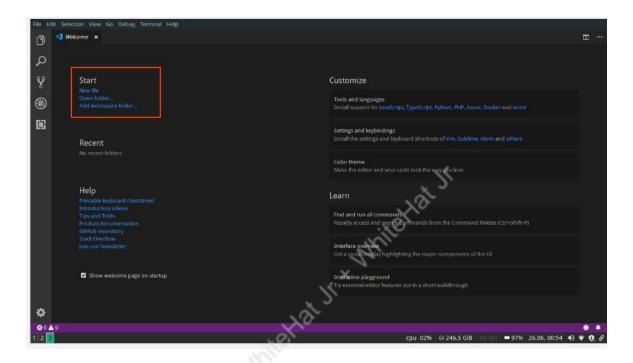
# Which CONCEPTS/ CODING BLOCKS did we cover today?

- Usage of boilerplate code
- Usage of touches

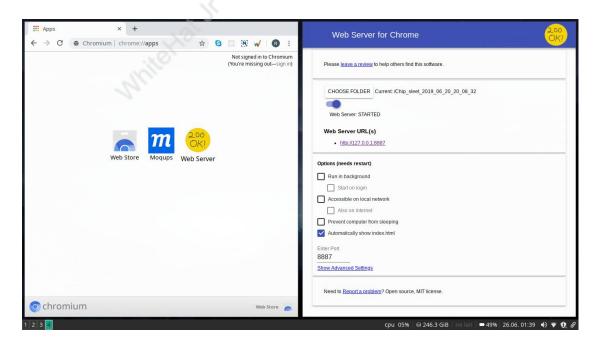


## How did we DO the activities?

1. Download the boilerplate code and open it in the VS code.



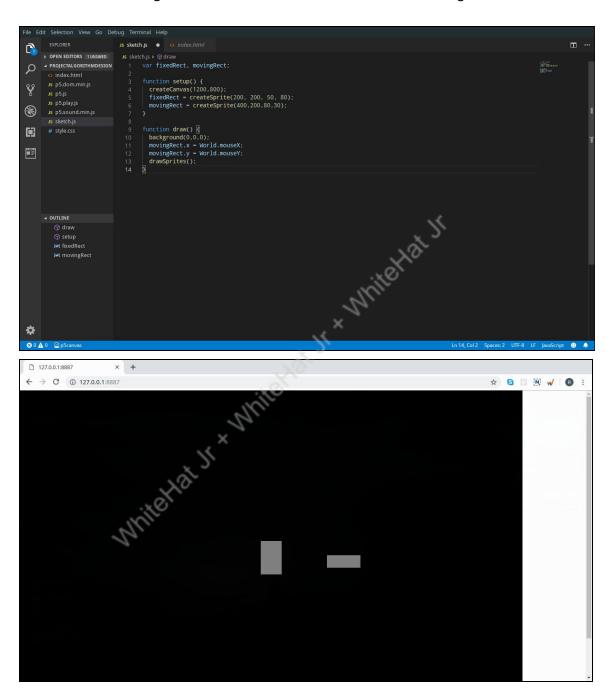
• We ran the server and pointed it to the same folder.



© 2019 The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

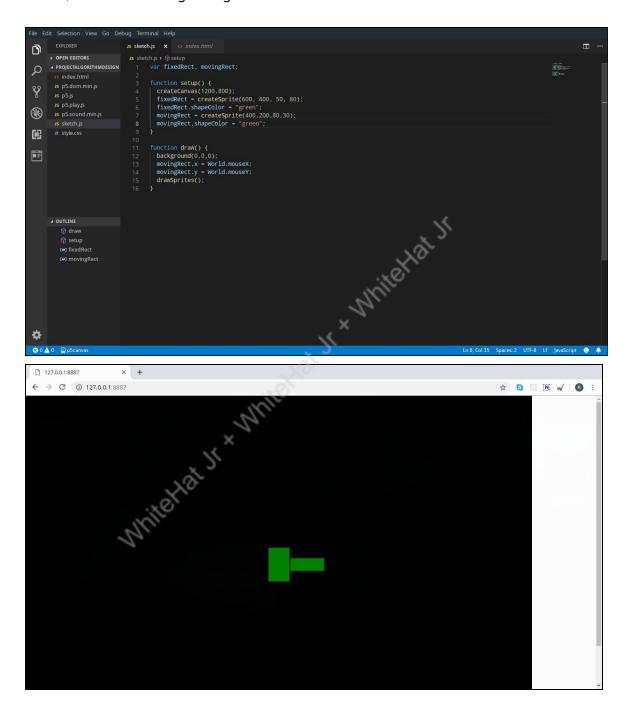


2. We created two rectangles with different widths and heights- one rectangle fixed, and the other moving, we also added controls to move the rectangle.





3. We wrote a collision detection algorithm so that when the two objects/rectangles collide, their color changes to green.





4. Then we added a condition to our code and changed the color of rectangles to red when the object touched each other, else kept it green.

```
Js sketch.js × 🕠 index.htm
0
          DOPEN EDITORS
                                                       Js sketch.js ♭ ⊕ draw

1 var fixedRect, movingRect;
                                                                   function setup() {
    createCanvas(1200.800);
    fixedRect = createSprite(600, 400, 50, 80);
    fixedRect.shapeColor = "green";
    fixedRect.debug = true;
    movingRect = createSprite(400.200.80.30);
    movingRect.shapeColor = "green";
    movingRect.debug = true;
}
function draw() {
  background(0,0,0);
                                                                        movingRect.x = World.mouseX;
movingRect.y = World.mouseY;
                                                                                                                                                                               r WildHal V
                                                                           | MovingRect.x - involved.x | Neuroct.Midth/2 + movingRect.width/2 | movingRect.x - movingRect.x - fixedRect.width/2 + movingRect.width/2 | movingRect.shapeColor = "red";
                                                                        else {
   movingRect.shapeColor = "green";
   fixedRect.shapeColor = "green";
                 ( fixedRect
                                                                         drawSprites():
*
                                                                                                                                                                                                                                                           n 24, Col 37 Spaces: 2 UTF-8 LF JavaScript 🙂 🜲
```

```
File Edit Selection View Go Debug Terminal Help

DOPLORE

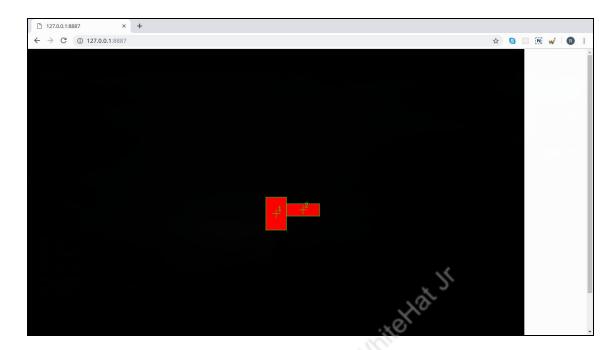
As Selectips * O Index.html

As Selectips * O Index.html

As Selectips * O Index.html

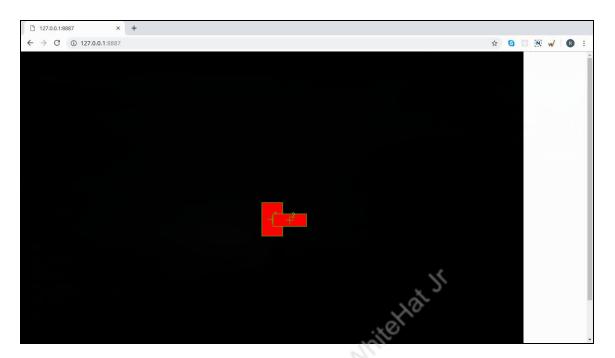
The Proceedings of Tax of
```





5. We saw that the rectangles turn red exactly at the point of touch when moving the rectangle to the LEFT. So we change the "===" sign in our condition to "<", we can detect when the moving rectangle has crossed over to the fixed rectangle.

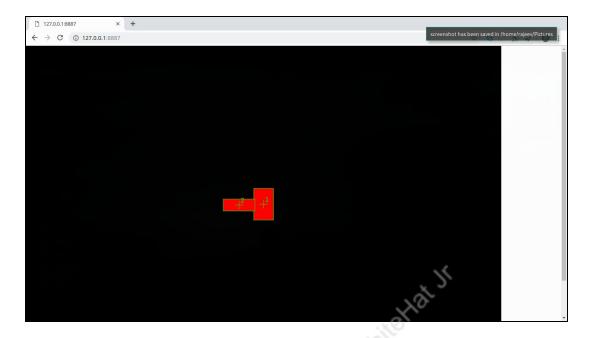




- 6. We used && operator to add two conditions.
- 7. We saw that when the rectangle is moving from left to right it is green and as it touches each other it becomes red.

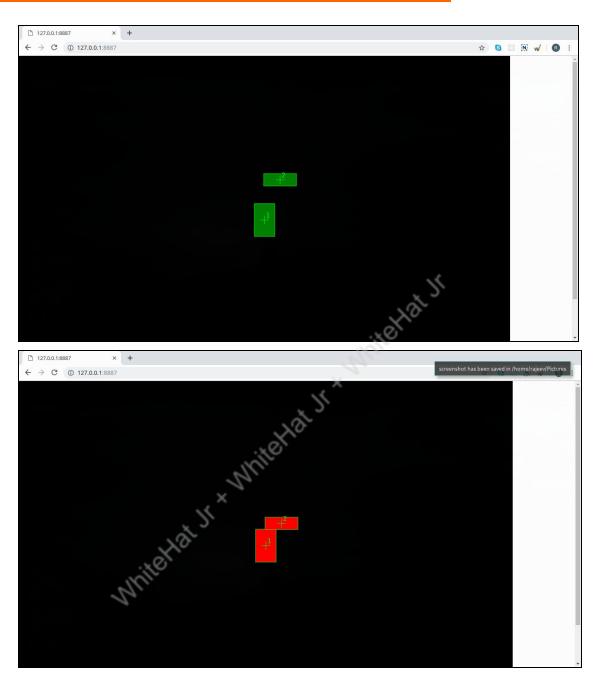






8. We wrote code to check if the vertical distance between the two rectangles is greater. We then set the rectangle color to green and if the distance is 0 then set the rectangle colour to red.





## What's next?

We will start working on the angry birds game.