Marking the player car

What is our GOAL for this MODULE?

The goal of this module is to add a reset button to reset the game.

What did we ACHIEVE in the class TODAY?

- Create a reset button at the top of the game which will reset the playerCount and gameState in our game.
- Add some identification to help the player identify which is their car.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Writing a reset function.
- Player identification.

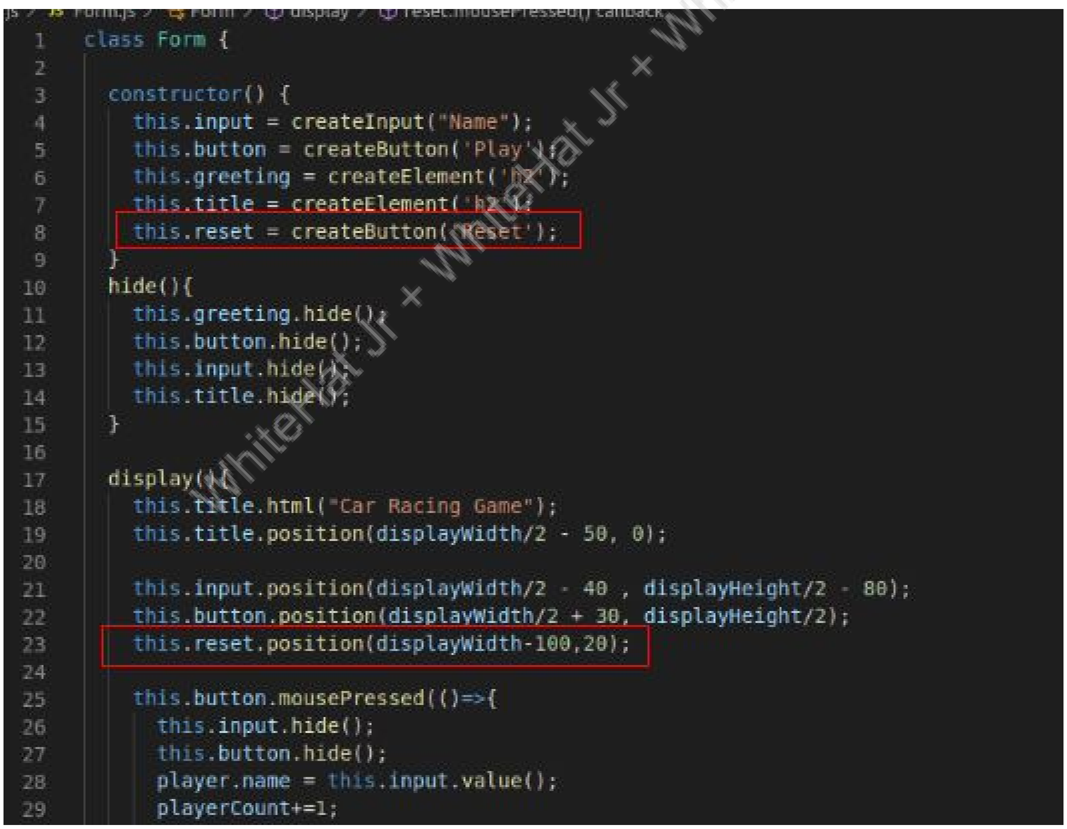
How did we DO the activities?

Introduced the problem of repeatedly having to open the firebase console to reset the gameState and playerCount.

In the last class, we learned how a good code structure - properly defined classes, objects, functions etc., keeping our code readable - allows us to write new functionality very easily. In this class, we wrote new functionality.

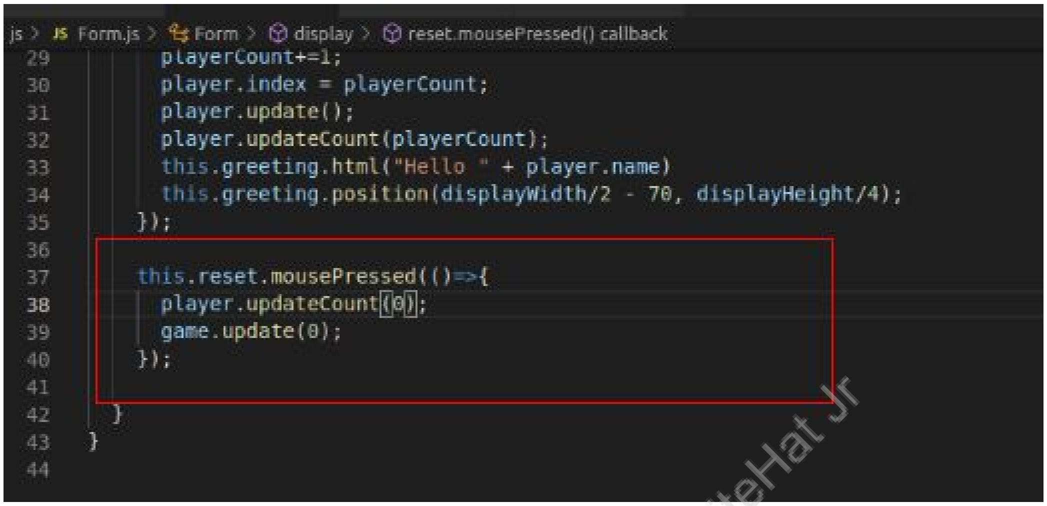
We wrote code to reset the button on the game screen in order to reset the player count. We have to always open the firebase console to manually reset the gameState and playerCount - everytime we want to test our game. Thus, to avoid this repetitive task, we build a reset button at the top which will reset the gameState and playerCount when we hit the button.

Adding the reset button and position it:


js > Form.js > Form > display > reset.mousePressed() callback

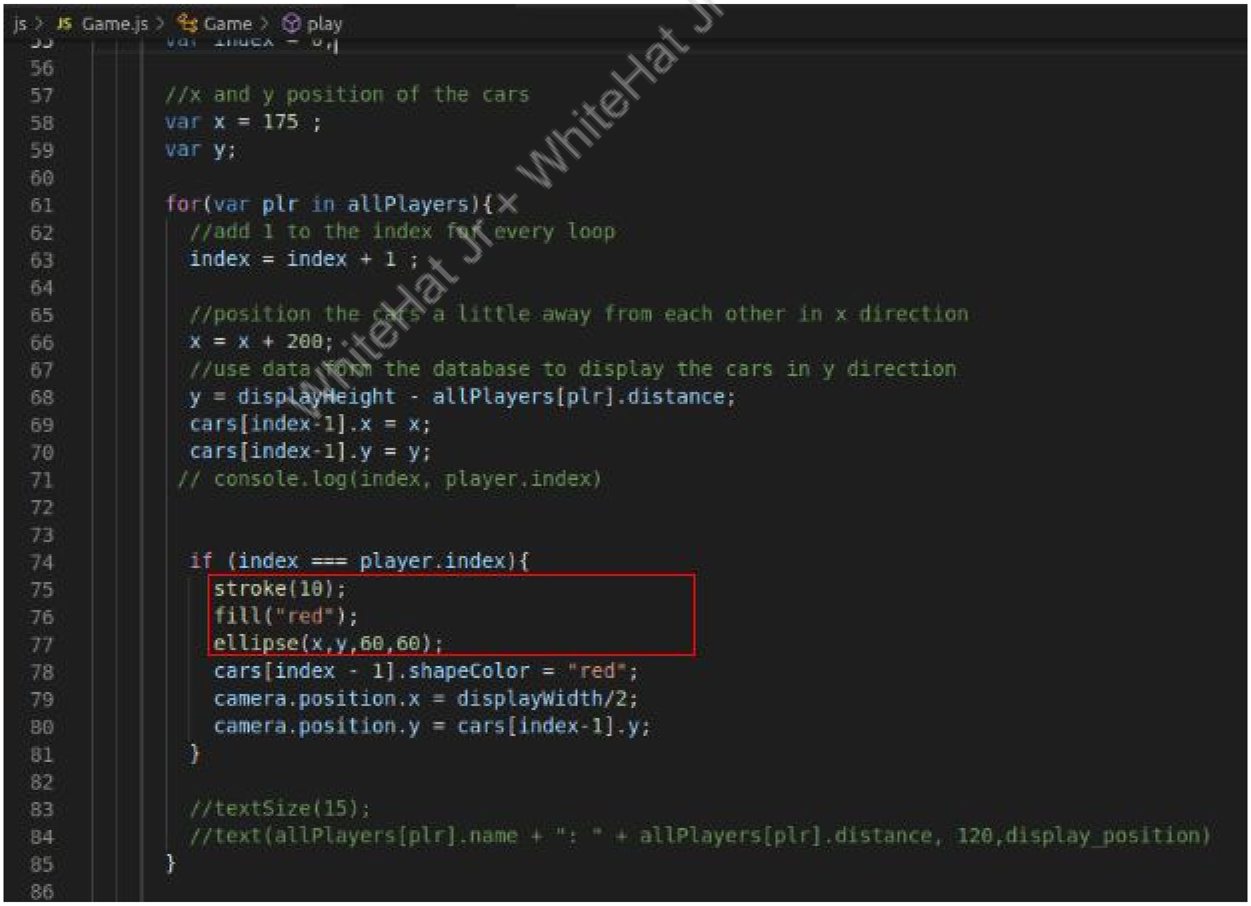
```
class Form {

  constructor() {
    this.input = createInput("Name");
    this.button = createButton('Play');
    this.greeting = createElement('h2');
    this.title = createElement('h2');
    this.reset = createButton('Reset');
  }
  hide(){
    this.greeting.hide();
    this.button.hide();
    this.input.hide();
    this.title.hide();
  }

  display(){
    this.title.html("Car Racing Game");
    this.title.position(displayWidth/2 - 50, 0);

    this.input.position(displayWidth/2 - 40 , displayHeight/2 - 80);
    this.button.position(displayWidth/2 + 30, displayHeight/2);
    this.reset.position(displayWidth-100,20);

    this.button.mousePressed(()=>{
      this.input.hide();
      this.button.hide();
      player.name = this.input.value();
      playerCount+=1;
```

To write a function to update the gameState and playerCount when the mouseButton is pressed:

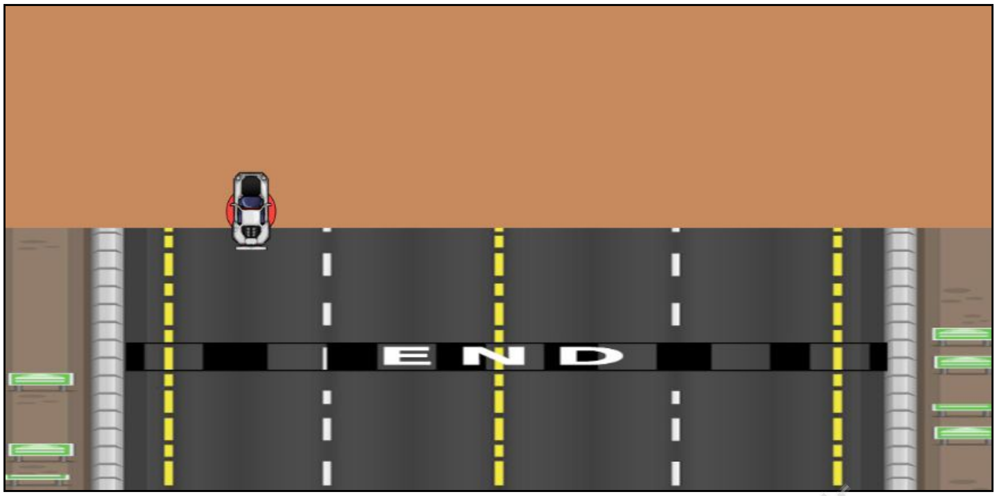

js > Form.js > Form > display > reset.mousePressed() callback

```
        playerCount+=1;
        player.index = playerCount;
        player.update();
        player.updateCount(playerCount);
        this.greeting.html("Hello " + player.name)
        this.greeting.position(displayWidth/2 - 70, displayHeight/4);
      });

      this.reset.mousePressed(()=>{
        player.updateCount(0);
        game.update(0);
      });

    }
  }
```

To give instructions to add the circle there at the x and y position of the car:



```
      var index = 0;

      //x and y position of the cars
      var x = 175 ;
      var y;

      for(var plr in allPlayers){
        //add 1 to the index for every loop
        index = index + 1 ;

        //position the cars a little away from each other in x direction
        x = x + 200;
        //use data form the database to display the cars in y direction
        y = displayHeight - allPlayers[plr].distance;
        cars[index-1].x = x;
        cars[index-1].y = y;
       // console.log(index, player.index)


        if (index === player.index){
          stroke(10);
          fill("red");
          ellipse(x,y,60,60);
          cars[index - 1].shapeColor = "red";
          camera.position.x = displayWidth/2;
          camera.position.y = cars[index-1].y;
        }

        //textSize(15);
        //text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_position)
      }
```



What’s NEXT?

Our next class will be a capstone class , you will be learning to create a new property for each player called “rank”.