

## STRUCTURING BEFORE CODING



### What is our GOAL for this MODULE?

We used our knowledge about OOPs programming style and created a form using p5 dom and log the players in the database.

### What did we ACHIEVE in the class TODAY?

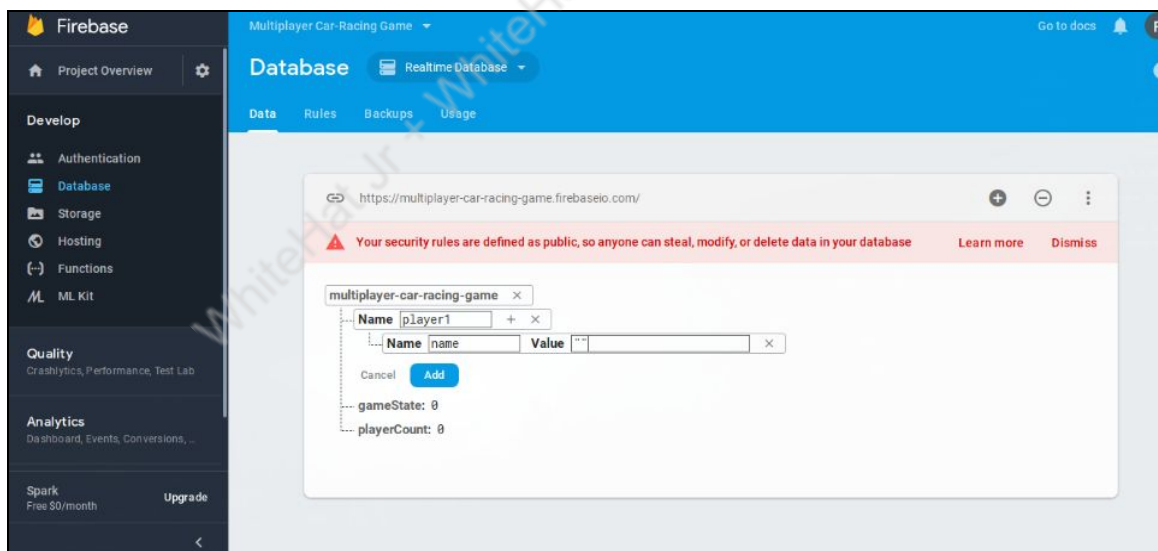
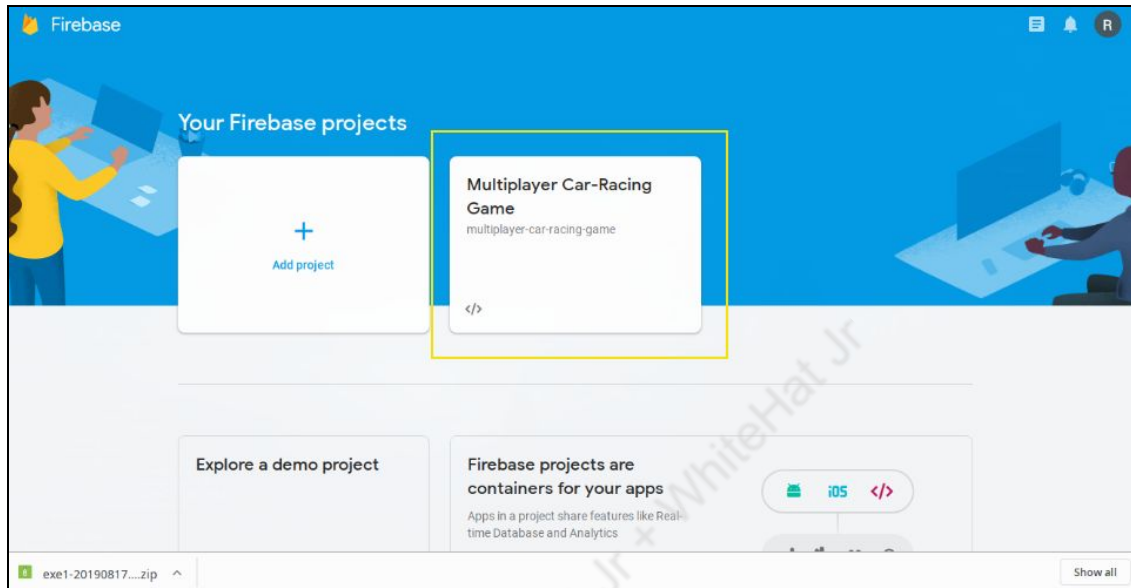
- Designed a form using p5 dom to allow players to login and log the player names to the database.
- Logged the gamestate and the playercount.
- Used the OOPs programming style to write the code.

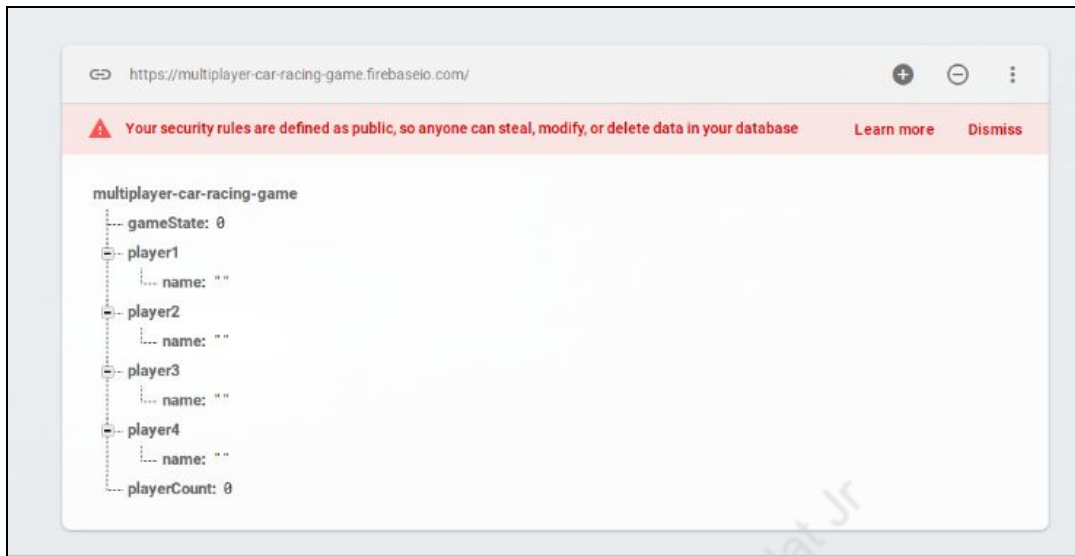
### Which CONCEPTS/CODING BLOCKS did we cover today?

- OOPs programming concept.
- Firebase database.
- Game states.
- p5 dom.

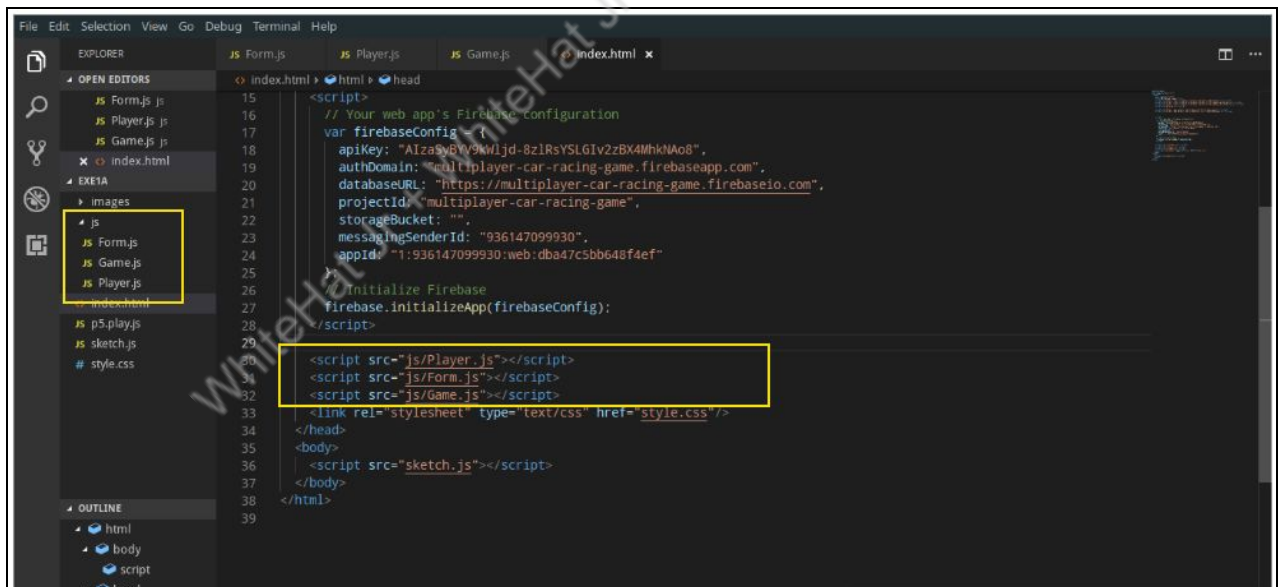
### How did we DO the activities?

1. Login to Firebase and Update the previous database as shown below.





2. Create a new folder in our directory called js which will contain the blueprint of the 2 objects in our game - Game and Form.



3. Include global variables in the sketch.js file.

```
JS sketch.js > ...
1  var canvas, backgroundImage;
2
3  var gameState = 0;
4  var playerCount;
5
6  var database;
7
8  var form, player, game;
9
10
11 function setup(){
12   canvas = createCanvas(400,400);
13   database = firebase.database();
14
15 }
16
17
18 function draw(){
19 }
20
21
```

4. Create a game class where we can listen to and update the database.

```
js ▶ JS Game.js ▶ Game ▶ getState
1  class Game {
2    constructor(){}
3
4    getState(){
5      var gameStateRef = database.ref('gameState');
6      gameStateRef.on("value",function(data){
7        gameState = data.val();
8      })
9    }
10
11
12    update(state){
13      database.ref('/').update({
14        gameState: state
15      });
16    }
17
18    start(){
19      if(gameState === 0){
20        player = new Player();
21        player.getCount();
22        form = new Form()
23        form.display();
24      }
25    }
26  }
27
```

5. Create Form class to register the players.

```
js ▶ JS Form.js ▶ Form ▶ display
1  class Form {
2    constructor() {
3
4    }
5
6    display(){
7      var title = createElement('h2')
8      title.html("Car Racing Game");
9      title.position(130, 0);
10
11      var input = createInput("Name");
12      var button = createButton('Play');
13      var greeting = createElement('h3');
14
15      input.position(130, 160);
16      button.position(250, 200);
17
18
19
20    }
21  }
22
```

6. Add mousePressed function to trigger the action when the mouse has been pressed.

```
js ▶ JS Form.js ▶ Form ▶ display
1  class Form {
2    constructor() {
3
4    }
5
6    display(){
7      var title = createElement('h2')
8      title.html("Car Racing Game");
9      title.position(130, 0);
10
11      var input = createInput("Name");
12      var button = createButton('Play');
13      var greeting = createElement('h3');
14
15      input.position(130, 160);
16      button.position(250, 200);
17
18      button.mousePressed();
19
20    }
21  }
22
```

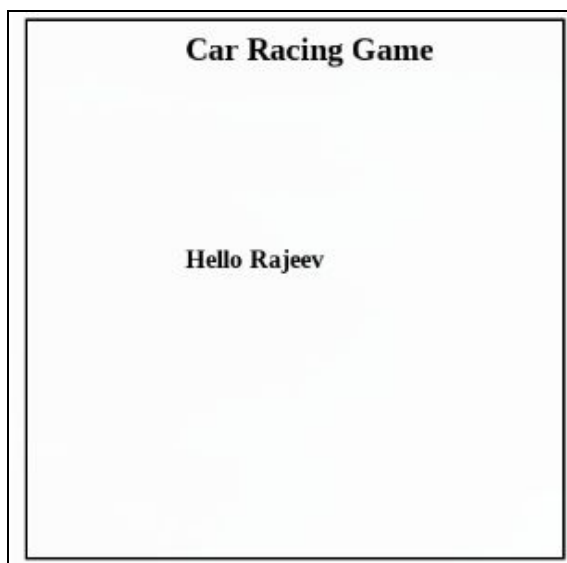
7. Add the code in our sketch.js file to create a new Game object, get the gameState and start the game.

```
JS sketch.js > setup
1  var canvas, backgroundImage;
2
3  var gameState = 0;
4  var playerCount;
5
6  var database;
7
8  var form, player, game;
9
10
11 function setup(){
12   canvas = createCanvas(400,400);
13   database = firebase.database();
14   game = new Game();
15   game.getState();
16   game.start();
17 }
18
19
20 function draw(){
21 }
22
```

8. Run the code and check for bugs and debug them.



The screenshot shows a web browser window with the title "Car Racing Game". Inside the browser, there is a text input field labeled "Name" and a button labeled "Play". The entire browser window is enclosed in a black rectangular border.

**What's next?**

In the next class, you will be creating a multiplayer car racing game.

**EXTEND YOUR KNOWLEDGE:**

1. Read this document for creating forms in p5:  
<https://p5js.org/examples/dom-input-and-button.html>