

Game in the Database



What is our GOAL for this MODULE?

Using the knowledge of OOPs programming we create a multiplayer car racing game.

What did we ACHIEVE in the class TODAY?

- We made a multiplayer car racing game which happens only in the database.
- We enabled press of the up arrow key to change the distance covered between the properties of each player in the database.
- The distance covered by all players is shown on the screen with the player playing the game highlighted.

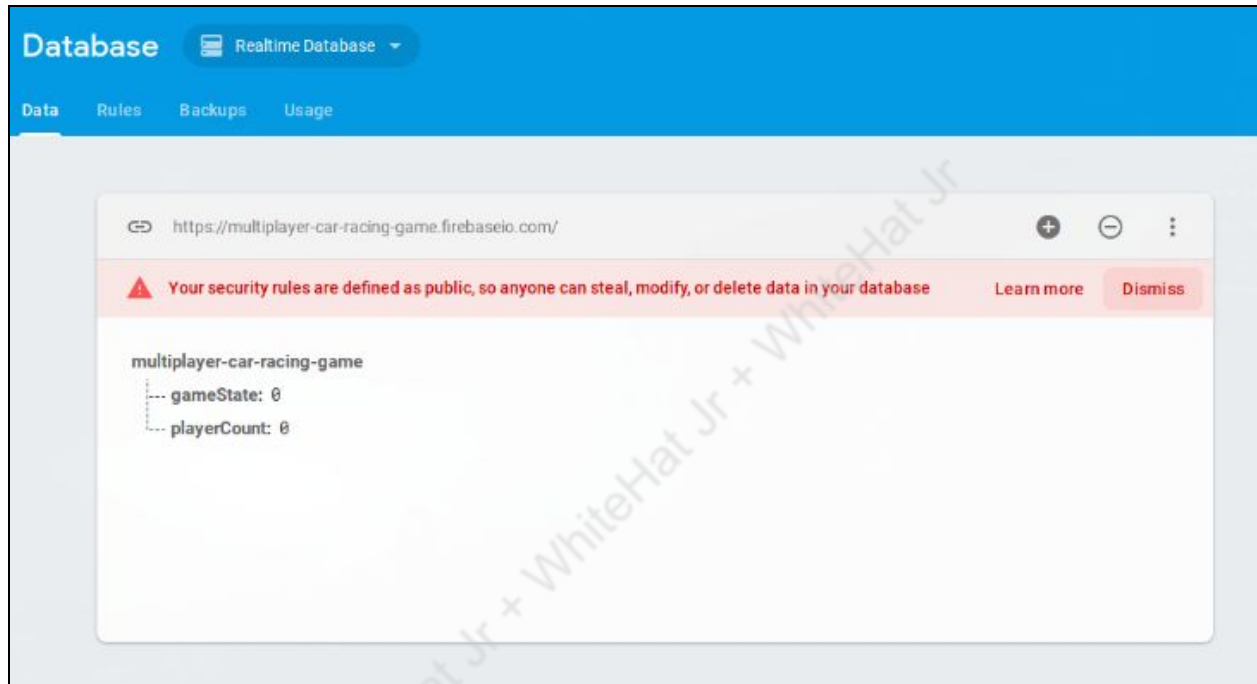
Which CONCEPTS/CODING BLOCKS did we cover today?

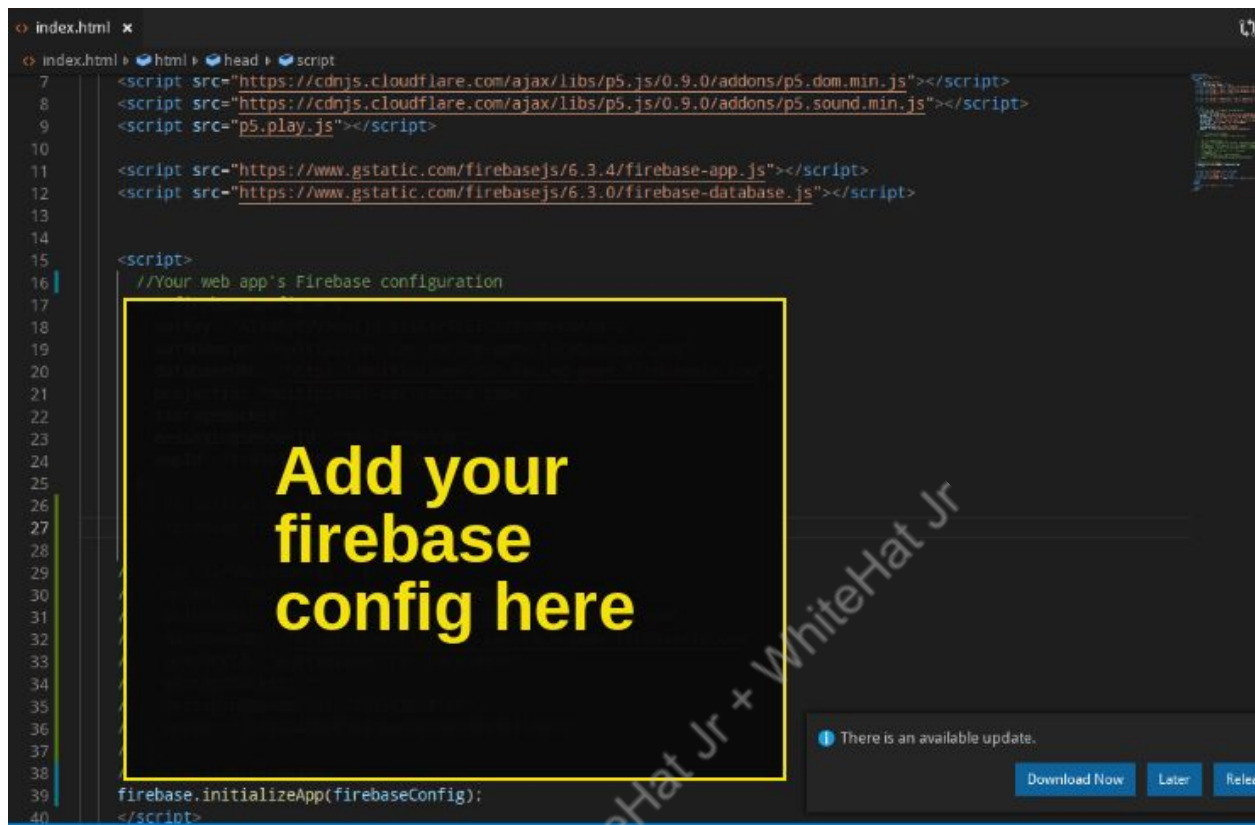
- The concept of OOPs programming
- Firebase database
- Game states

How did we DO the activities?

- Re-structured code to add more desired properties and functions for the code
- Built the car racing game on the database

Cloned the code from the reference link, added the firebase configuration and made the necessary changes in the firebase database structure. We made sure gameState and playerCount were 0.





```

index.html x
index.html | html | head | script
7 <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
8 <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.sound.min.js"></script>
9 <script src="p5.play.js"></script>
10
11 <script src="https://www.gstatic.com/firebasejs/6.3.4/firebase-app.js"></script>
12 <script src="https://www.gstatic.com/firebasejs/6.3.0/firebase-database.js"></script>
13
14
15 <script>
16   //Your web app's Firebase configuration
17   [
18     "apiKey": "AIzaSyB...",
19     "authDomain": "multiplayer-car-racing-game.firebaseapp.com",
20     "databaseURL": "https://multiplayer-car-racing-game.firebaseio.com",
21     "projectId": "multiplayer-car-racing-game",
22     "storageBucket": "multiplayer-car-racing-game.appspot.com",
23     "messagingSenderId": "1018912345678"
24   ]
25   const firebaseConfig = {
26     apiKey: "AIzaSyB...",
27     authDomain: "multiplayer-car-racing-game.firebaseapp.com",
28     databaseURL: "https://multiplayer-car-racing-game.firebaseio.com",
29     projectId: "multiplayer-car-racing-game",
30     storageBucket: "multiplayer-car-racing-game.appspot.com",
31     messagingSenderId: "1018912345678"
32   };
33   firebase.initializeApp(firebaseConfig);
34   // ...
35   // ...
36   // ...
37   // ...
38   // ...
39   // ...
40   // ...
  
```

Thus, as soon as 4 players logged in, the form and the greeting vanished.

As the game started, all the players and the distance covered by them was displayed on the screen.

The player who logged in was highlighted. The current distance covered by the player increased when the up arrow key was pressed.

- We wrote code for behavior in play state for the game when all the players were logged in
- Displayed all the players with their distance scores

Changed the firebase database structure:



```

{
  "gameState": 0,
  "playerCount": 0
}
  
```

Added new properties and functions to our Player class:

```
js Player.js ▶ Player ▶ getPlayerInfo
class Player {
  constructor(){
    this.index = null;
    this.distance = 0;
    this.name = null;
  }

  getCount(){
    var playerCountRef = database.ref('playerCount');
    playerCountRef.on("value",(data)->{
      playerCount = data.val();
    })
  }
}
```

getCount() and updateCount() functions remain the same.
Used the update() function to update both name and distance of the player.

```
js ▶ JS Player.js ▶ Player ▶ getPlayerInfo
7
8   getCount(){
9     var playerCountRef = database.ref('playerCount');
10    playerCountRef.on("value",(data)->{
11      playerCount = data.val();
12    })
13  }
14
15  updateCount(count){
16    database.ref('/').update({
17      playerCount: count
18    });
19  }
20
21  update(){
22    var playerIndex = "players/player" + this.index;
23    database.ref(playerIndex).set({
24      name: this.name,
25      distance: this.distance
26    });
27  }
28
```

Used the property 'index' of the player to update that particular player's values in the database.

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

The player's data was stored as JSON - since the firebase database structure is of JSON type.

```
js ▶ JS Player.js ▶ Player ▶ getPlayerInfo
7
8   getCount(){
9     var playerCountRef = database.ref('playerCount');
10    playerCountRef.on("value",(data)->{
11      playerCount = data.val();
12    })
13  }
14
15  updateCount(count){
16    database.ref('/').update({
17      playerCount: count
18    });
19  }
20
21  update(){
22    var playerIndex = "players/player" + this.index;
23    database.ref(playerIndex).set({
24      name:this.name,
25      distance:this.distance
26    });
27  }
28
29  static getPlayerInfo(){
30    var playerInfoRef = database.ref('players');
31    playerInfoRef.on("value",(data)->{
32      allPlayers = data.val();
33    })
34  }
35
36
```

Wrote a `hide()` function which was called when the game state changed and when we wanted to hide the form.

```
js > JS Form.js > Form > hide
1  class Form {
2
3    constructor() {
4      this.input = createInput("Name");
5      this.button = createButton('Play');
6      this.greeting = createElement('h2');
7    }
8    hide(){
9      this.greeting.hide();
10     this.button.hide();
11     this.input.hide();
12   }
13
14   display(){
15     var title = createElement('h2');
16     title.html("Car Racing Game");
17     title.position(130, 0);
18
19     this.input.position(130, 160);
20     this.button.position(250, 200);
21
22     this.button.mousePressed(()=>{
23       this.input.hide();
24       this.button.hide();
25       player.name = this.input.value();
26       playerCount+=1;
27       player.index = playerCount;
28       player.update();
29       player.updateCount(playerCount);
30       this.greeting.html("Hello " + player.name)
31       this.greeting.position(130, 100);
32     });
33
34 }
```

To write a play() function which will be called when the game state reaches play state or 1.
 Inside the play() function, hide the form.

Get all the players' data and display them on the screen.

Change the distance and update it in the database when an 'UP' arrow key is pressed.

Note: we used 'plr' because the player is already defined.

```

6   form = new Form()
7   form.display();
8   }
9   }
10  }

11  play(){
12      form.hide();
13      textSize(30);
14      text("Game Start", 120, 100)
15      Player.getPlayerInfo();

16      if(allPlayers !== undefined){
17          var display_position = 130;
18          display_position+=20;
19          textSize(15);
20          text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_position)
21      }
22  }

23  if(keyIsDown(UP_ARROW) && player.index !== null){
24      player.distance +=50
25      player.update();
26  }
27  }

```

To write the code to make the current player 'red':

```

11  play(){
12      form.hide();
13      textSize(30);
14      text("Game Start", 120, 100)
15      Player.getPlayerInfo();

16      if(allPlayers !== undefined){
17          var display_position = 130;
18          for(var plr in allPlayers){
19              if (plr === "player" + player.index)
20                  fill("red")
21              else
22                  fill("black");
23              display_position+=20;
24              textSize(15);
25              text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_position)
26          }
27      }
28  }

```


To modify the sketch file to write conditions to change the game state and call play() function:

```
8
9  var form, player, game;
10
11
12  function setup(){
13    canvas = createCanvas(400,400);
14    database = firebase.database();
15    game = new Game();
16    game.getState();
17    game.start();
18  }
19
20
21  function draw(){
22    if(playerCount --- 4){
23      game.update(1);
24    }
25    if(gameState === 1){
26      clear();
27      game.play();
28    }
29  }
30
```

What's NEXT?

In the next class, you will be learning about game camera and display size.

EXTEND YOUR KNOWLEDGE:

You can learn about firebase from their official docs:

<https://firebase.google.com/docs/firestore>.