RESET BUTTON AND PLAYER CONTROLS

What is our GOAL for this MODULE?

In this class, we learned how to implement the reset button to reset the data in the database. We also created the leaderboard to show the score on the screen and added player controls as well.

What did we ACHIEVE in the class TODAY?

- Created the reset button to reset the player information and game state in the database.
- Added leader boards to show the player's score on the screen.
- The player controls the car to move left and right.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- p5.DOM.elements
- Reset functionality
- **mousePressed()**
- Arrow functions

How did we DO the activities?

1. Create a reset button and the text using **p5.DOM.elements** in **form.js.**

```
constructor() {
  this.resetTitle = createElement("h2");
  this.resetButton = createButton("");

}
```

2. Create **handleElements()** in **game.js** to display the reset button and its title.

```
handleElements() {
   form.hide();
   form.titleImg.position(40, 50);
   form.titleImg.class("gameTitleAfterEffect");


   this.resetTitle.html("Reset Game");
   this.resetTitle.class("resetText");
   this.resetTitle.position(width / 2 + 200, 40);

   this.resetButton.class("resetButton");
   this.resetButton.position(width / 2 + 230, 100);
```

3. Create text elements for the leaderboards in **constructor()** in **game.js.**

```
constructor() {
   this.resetTitle = createElement("h2");
   this.resetButton = createButton("");

   this.leadeboardTitle = createElement("h2");
   this.leader1 = createElement("h2");
   this.leader2 = createElement("h2");

 }
```

4. Display the leaderboard's properties using **handleElement()**.

```
handleElements() {
  form.hide();
  form.titleImg.position(40, 50);
  form.titleImg.class("gameTitleAfterEffect");

  this.resetTitle.html("Reset Game");
  this.resetTitle.class("resetText");
  this.resetTitle.position(width / 2 + 200, 40);

  this.resetButton.class("resetButton");
  this.resetButton.position(width / 2 + 230, 100);

  this.leadeboardTitle.html("Leaderboard");
  this.leadeboardTitle.class("resetText");
  this.leadeboardTitle.position(width / 3 - 60, 40);

  this.leader1.class("leadersText");
  this.leader1.position(width / 3 - 50, 80);

  this.leader2.class("leadersText");
  this.leader2.position(width / 3 - 50, 130);
}
```

5. Create a **showLeader()** function.
    - Set leader1 and leader2 based on the rank of the player.

```
showLeaderboard() {
   var leader1, leader2;
   var players = Object.values(allPlayers);
   if (
     (players[0].rank === 0 && players[1].rank === 0) ||
     players[0].rank === 1
   ) {
     //      This tag is used for displaying four spaces.
     leader1 =
       players[0].rank +
       " " +
       players[0].name +
       " " +
       players[0].score;

     leader2 =
       players[1].rank +
       " " +
       players[1].name +
       " " +
       players[1].score;
   }
```

- Use **.html()** to display the leader1 and leader2 variables.

```
  this.leader1.html(leader1);
  this.leader2.html(leader2);
}
```

6. Define the **handleResetButton()** function for defining action on the reset button.
    - Use **mousePressed()** to trigger an action when a mouse button is pressed.
    - Use **.set()** to update the value of **gameState**, **playerCount** and **carsAtEnd** to 0.
    - Remove all players from the database.

- Use **window.location.reload()** to reload the browser and reset the game.

```
handleResetButton() {
  this.resetButton.mousePressed(() => {
    database.ref("/").set({
      carsAtEnd: 0,
      playerCount: 0,
      gameState: 0,
      players: {}
    });
    window.location.reload();
  });
}
```

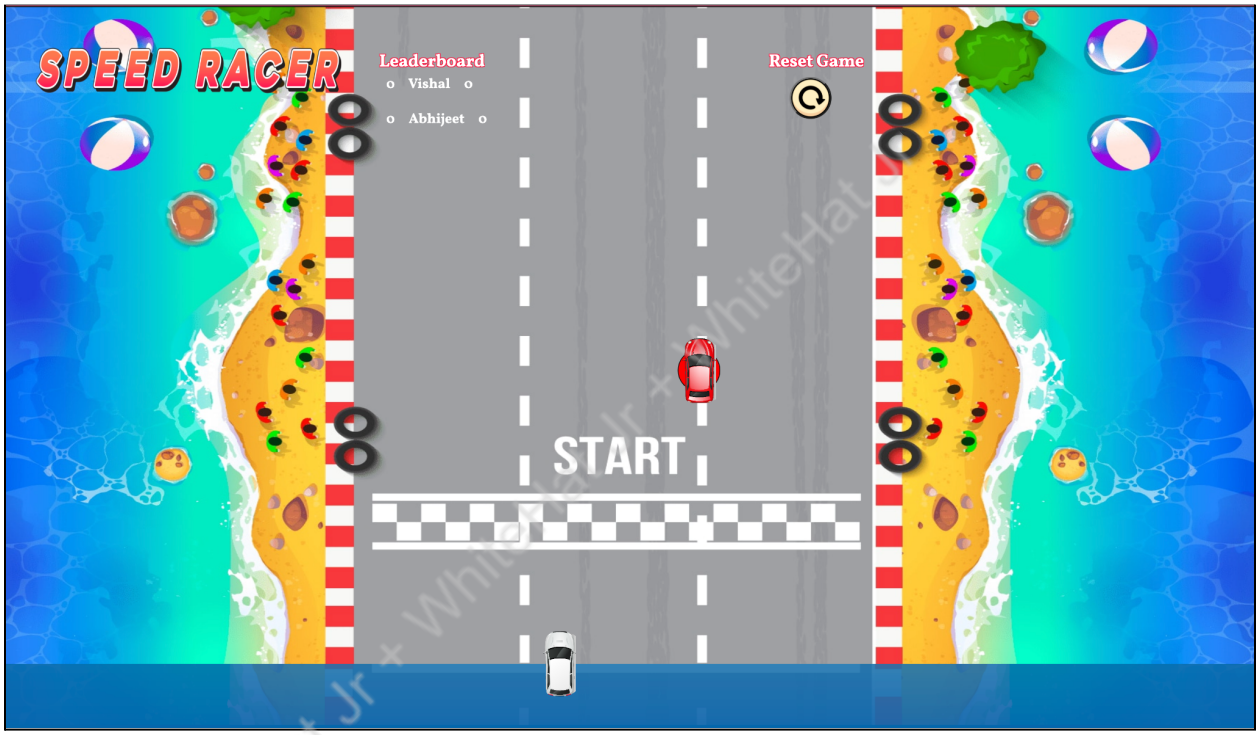
7. Add player controls to move the car in the left and right direction based on the arrow keypressed.

```
handlePlayerControls() {
  if (keyIsDown(UP_ARROW)) {
    player.positionY += 10;
    player.update();
  }

  if (keyIsDown(LEFT_ARROW) && player.positionX > width / 3 - 50) {
    player.positionX -= 5;
    player.update();
  }

  if (keyIsDown(RIGHT_ARROW) && player.positionX < width / 2 + 300) {
    player.positionX += 5;
    player.update();
  }
}
```

OUTPUT



What's next?

In the next class, we will make the game more fun and challenging by adding a few rewards and obstacles to the game.

EXTEND YOUR KNOWLEDGE

1. Bookmark the following link to know more about Firebase database queries: https://firebase.google.com/docs/database/web/read-and-write