



Topic	Realtime Database	
Class Description	Students learn the importance of using a realtime database to create multiplayer games. Students learn how to connect, read and write data into a remote real time database.	
Class	C35	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> • Create a remote real time database. • Read and write data to a real time database. • Create a ball which moves synchronously in different browsers. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources <ul style="list-style-type: none"> ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen 	
Class structure	Warm Up Teacher-led Activity Student-led Activity Wrap up	5 mins 15 min 15 min 5 mins
CONTEXT <ul style="list-style-type: none"> • Role of a remote database in a multiplayer game. 		
Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Hey! Welcome back to the class! How are you feeling after the last class?	ESR: <i>varied</i>

	<p>So far, we have built many games - both in-class and after-class projects.</p> <p>We have games as simple and old as Pong. We even built games which use Physics engines like Angry Birds.</p> <p>But all these games that we have created are single-player games.</p> <p>Wouldn't it be nice if we could build multiplayer games which you and your friends could play on different computers while sitting online!</p>	<p>ESR: Yes! (excited)</p>
	<p>We are going to start doing exactly that!!</p> <p>We are going to start building a multiplayer car-racing game in the next few classes!</p> <p>Let's start on this exciting journey.</p>	-
	<p><i>Before that I have an exciting quiz question for you! Are you ready to answer this question?</i></p> <p>Teacher click on the  button on the bottom right corner of your screen to start the In-Class Quiz.</p> <p><i>A quiz will be visible to both you and the student.</i></p> <p><i>Encourage the student to answer the quiz question.</i></p>	<p>ESR: Yes</p>

	<p><i>The student may choose the wrong option, help the student to think correctly about the question and then answer again.</i></p> <p><i>After the student selects the correct option,</i></p> <p>the  button will start appearing on your screen.</p> <p><i>Click the End quiz to close the quiz pop-up and continue the class.</i></p>	
Teacher Initiates Screen Share		
<p style="text-align: center;"><u>CHALLENGE</u></p> <ul style="list-style-type: none"> • Create an arrow-key controlled ball. • Show how the ball is asynchronous in different browser windows. • Create and connect to a remote database. 		
<p>Step 2: Teacher-led Activity (15 min)</p>	<p>We already know how to design a game with multiple characters. We can also host the game online and then open the game on different browsers on different computers.</p> <p>What is stopping us from designing a multiplayer game then?</p>	<p><i>The student thinks about it.</i></p> <p>ESR: The game is in different states in the two browsers. For a multi-player game, we need the two browsers to have the game at the same state at the same time. Everything in the two browsers should be synchronous.</p> <p>Currently the games are asynchronous and independent from each other.</p>

	<p>Exactly! Let's look at this problem with an example.</p> <p><i>Teacher writes the code (same as Teacher Activity 1) to create a simple ball which can move using arrow keys.</i></p> <p>We can also add margin to our canvas to place it in the centre.</p>	<p><i>The student observes and learns.</i></p>
--	--	--

Sketch file code:

```

js sketch2.js > draw
1  var ball;
2
3  function setup(){
4      createCanvas(500,500);
5      ball = createSprite(250,250,10,10);
6      ball.shapeColor = "red";
7  }
8
9  function draw(){
10     background("white");
11     if(keyDown(LEFT_ARROW)){
12         changePosition(-1,0);
13     }
14     else if(keyDown(RIGHT_ARROW)){
15         changePosition(1,0);
16     }
17     else if(keyDown(UP_ARROW)){
18         changePosition(0,-1);
19     }
20     else if(keyDown(DOWN_ARROW)){
21         changePosition(0,+1);
22     }
23     drawSprites();
24 }
25
26 function changePosition(x,y){
27     ball.x = ball.x + x;
28     ball.y = ball.y + y;
29 }

```

Css file code:

```

# style.css | html
1  html, body {
2    margin: 0;
3    padding: 0;
4  }
5  canvas{
6    margin-left: 25%;
7    margin-right: 25%;
8    margin-top: 100px;
9    border: 2px solid black;
10 }
11
12
  
```

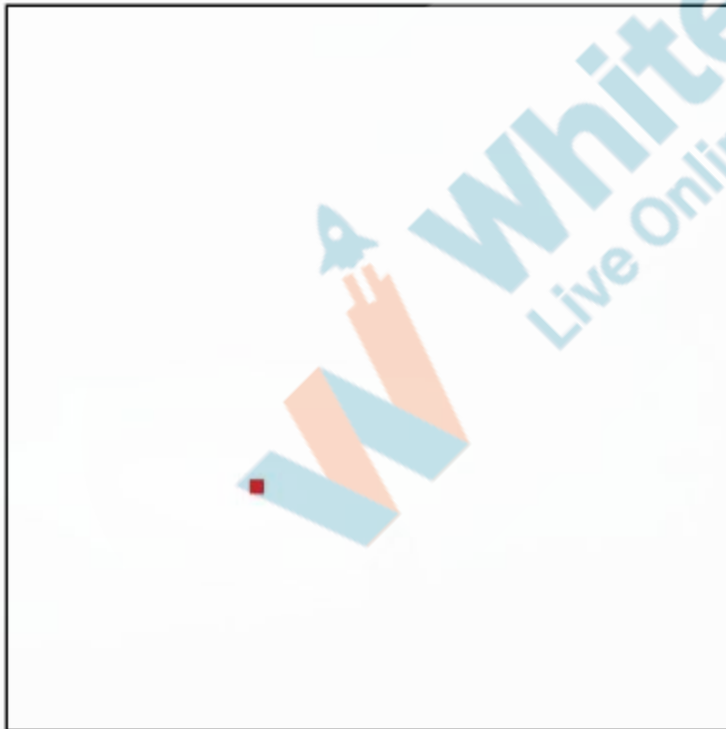
Let's open the application in two different browsers. Let's move the ball and see what happens.

What do you see?

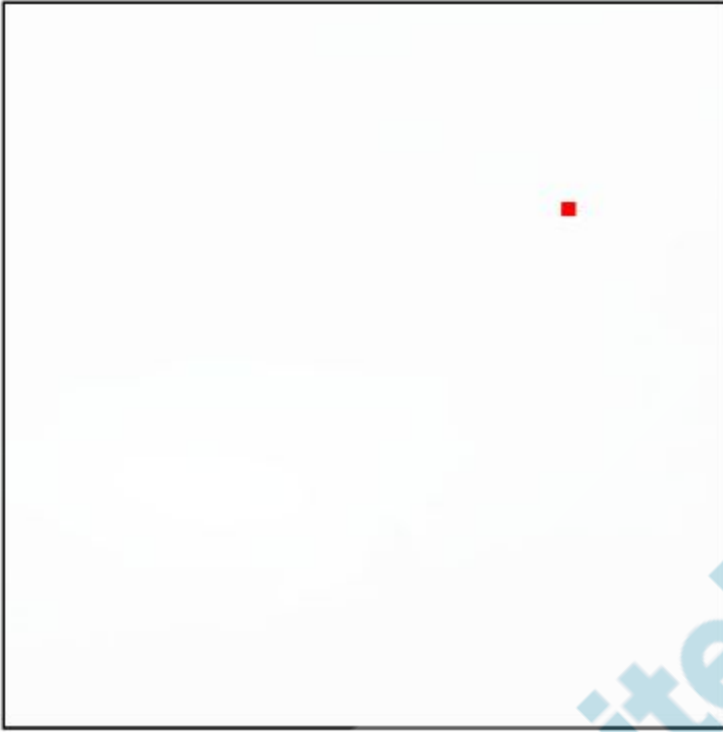
ESR:

The ball in the two browsers move independently. Their movements are asynchronous.

Browser 1:



Browser 2:



This happens because the ball's position in each browser is independent of the other's position.

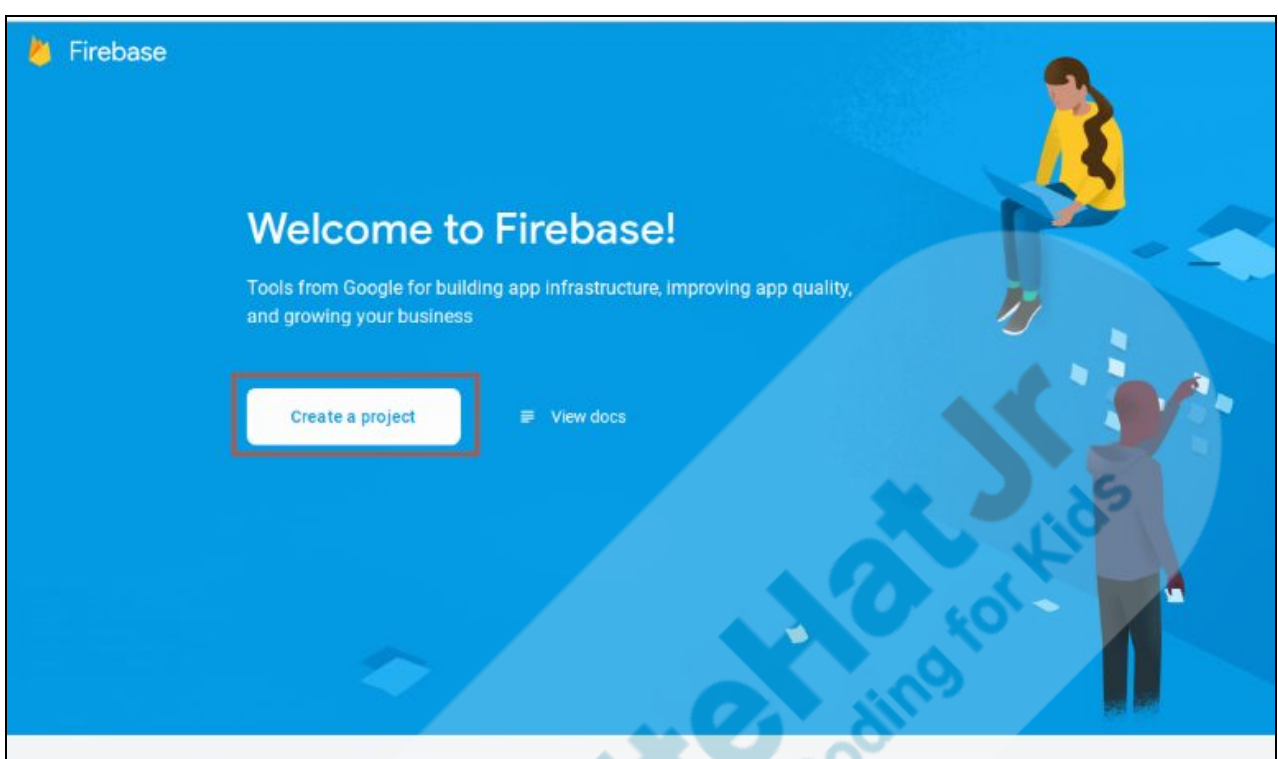
But what if we could store the ball's position in a remote common database and our application reads the ball's position from the database and updates it when it changes.

Database servers are computers which are remotely connected through the internet and maintain your data which you can use in your applications.

The two browsers will read the position of the ball from the common remote database and they will always be synchronized.

The student listens and observes.

	<p>This is how multiplayer games work. They store the position of the state of the game at all times in a remote database.</p> <p>All the players' console/browsers read the game from this remote database and write to it when they make any change in the game.</p>	
	<p>Let's make a remote database for our simple application on the cloud internet. This remote database will store the state (positions) of the ball and will allow us to read or write to it at any time.</p> <p>We will be using Google Firebase's Real Time Database for this purpose.</p> <p><i>Teacher guides the student on how to create a Real Time Database and create a variable call ball which stores two values x and y.</i></p> <p>The database can be compared to a JSON data structure format.</p> <pre>ball = { x: 250, y: 250 }</pre>	<p><i>Student observes on how to create a Real Time Database in Firebase.</i></p>
<p>Step1: Go to https://console.firebase.google.com and login with your gmail id.</p> <p>Step2: Click on 'Create a New Project'.</p>		




Step 3: Enter the name of your project. Accept terms and continue.

× Create a project (Step 1 of 3)

Let's start with a name for your project

Project name
Multiplayer Car-Racing Game

 multiplayer-car-racing-game

☐ I accept the [Firebase terms](#)

Continue

Step 4: Deny Google Analytics use. (We don't need it)

X
Create a project (Step 2 of 2)

and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing and Predictions.

Google Analytics enables:

- X A/B testing ?
- X User segmentation & targeting across Firebase products ?
- X Predicting user behavior ?
- X Crash-free users ?
- X Free unlimited reporting ?

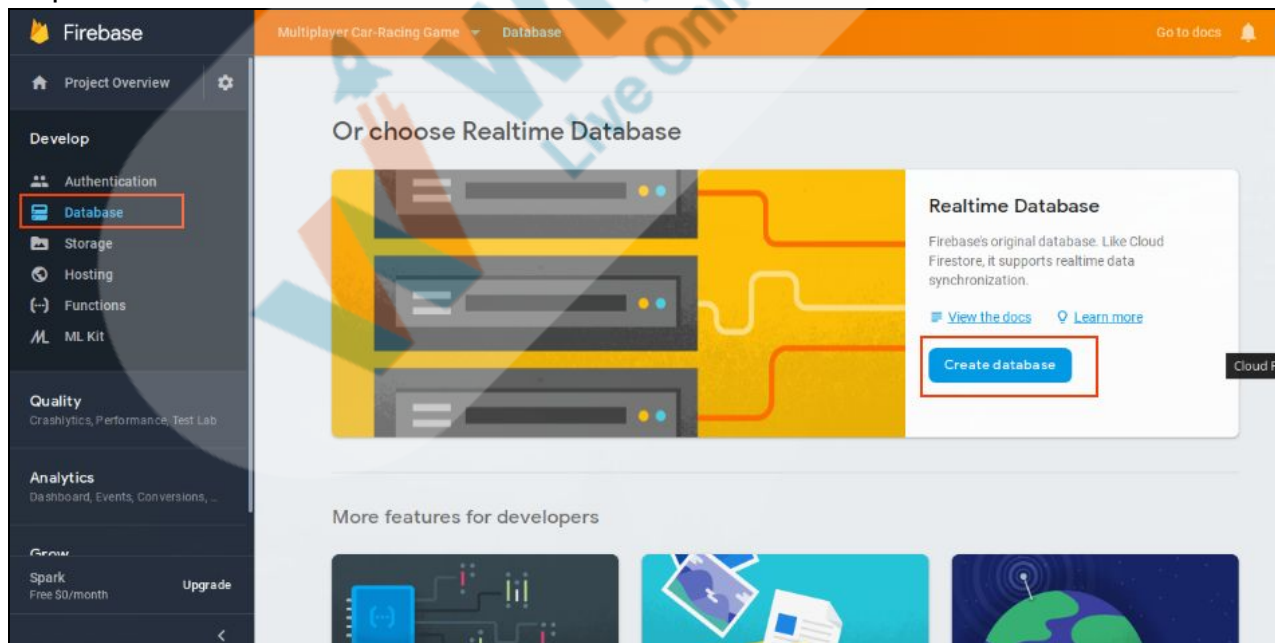
☐ Set up Google Analytics for my project
Configure in the next step

☒ Not right now
You can change this later

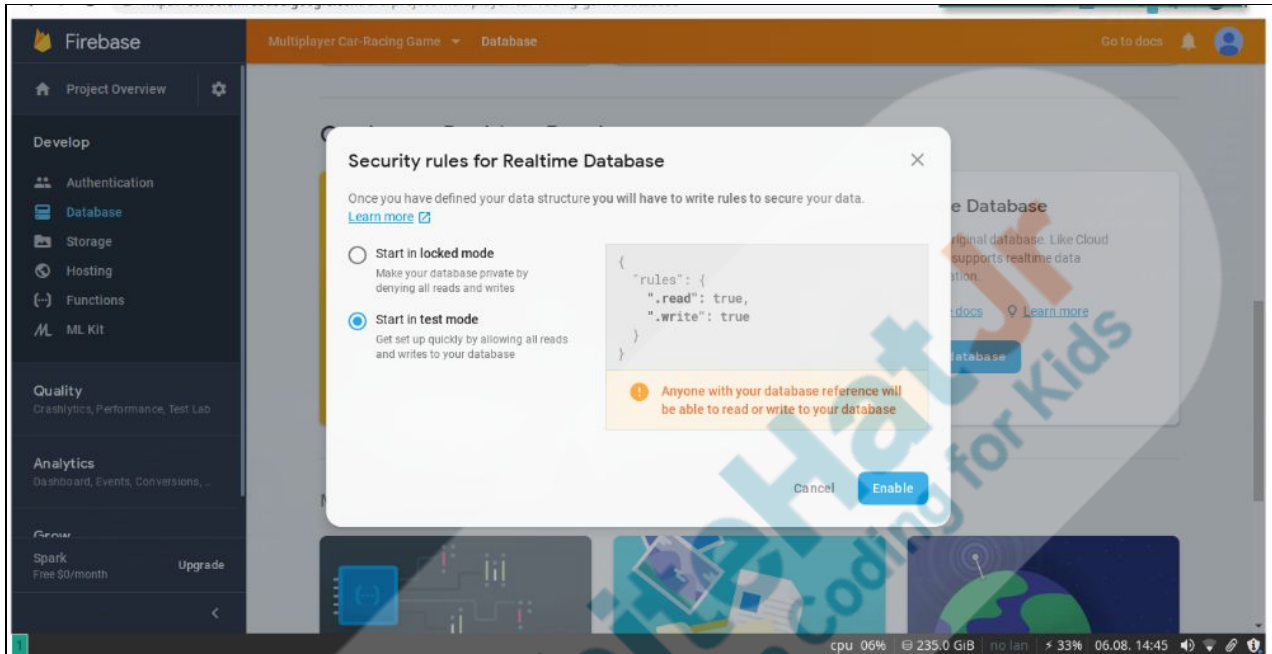
Previous

Create project

Step 5: Visit the Database section of the dashboard and click on 'Create Database'.

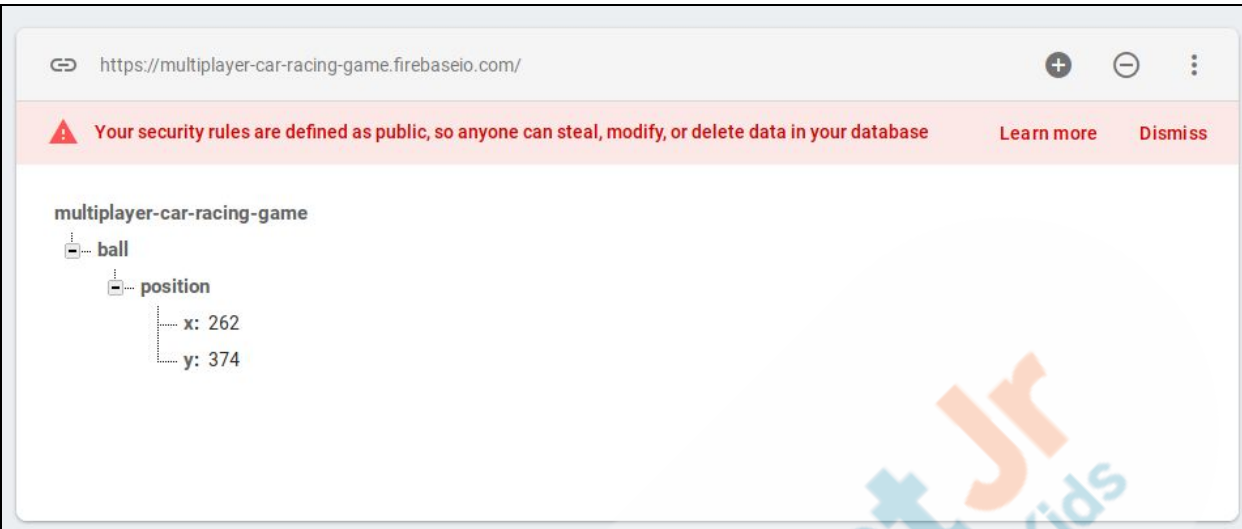


Step 6: Create the database in test mode for now. (Test mode will help us quickly get started. It will be less secure in this mode though later we will learn how to make the database more secure)

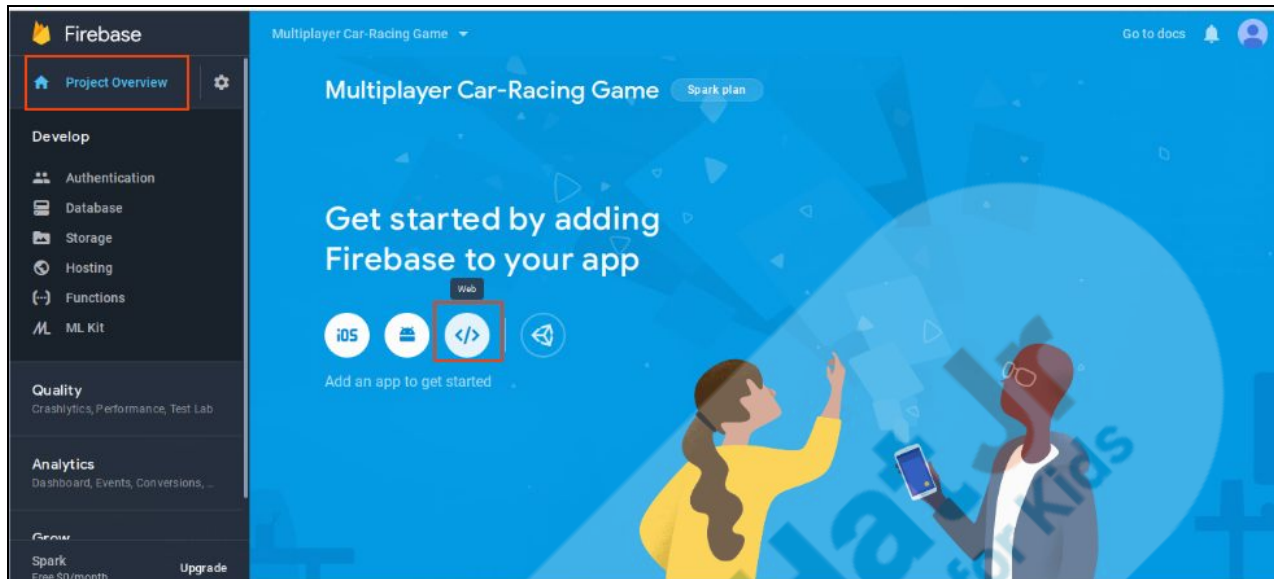


Step 7: Add a child to create nodes which can hold ball's x and y positions.

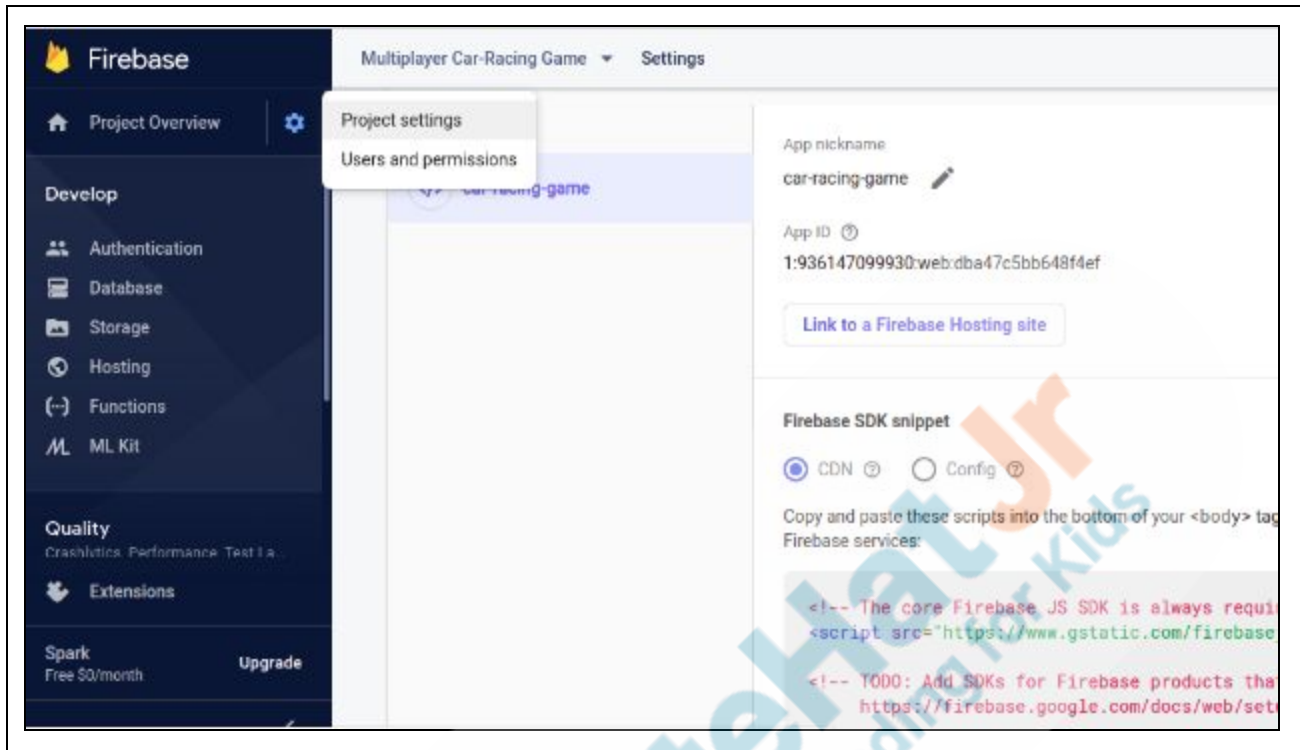


		
	<p>Let's connect the firebase database to our application.</p> <p>We will have to import a few firebase libraries which will allow us to read and write to our database.</p> <p>We will also need to supply some configuration settings like - api key, databaseURL etc. which can authenticate our application and allow it to write to the database.</p> <p>It is not wise to make this public - because then anyone will be able to write to your application. But our application does not contain any critical information so we can add this directly in our html file.</p> <p><i>Teacher shows how to connect the application to the firebase database and how to initialize the firebase App with the firebase config.</i></p>	<p><i>Student observes and learns how to initialize the firebase app.</i></p>

Step 1: Click on the “Project Overview”. Choose Add to web to get started.



Step 2: Get the firebase config key.



2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/6.3.4/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#config-web-app -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyBYV9kWljd[REDACTED]",
    authDomain: "multiplayer-car-racing-game.firebaseio.com",
    databaseURL: "https://multiplayer-car-racing-game.firebaseio.com",
    projectId: "multiplayer-car-racing-game",
    storageBucket: "",
    messagingSenderId: "936147099930",
    appId: "1:936147099930:web:dba47c5bb648f4ef"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

[Continue to console](#)

Step 3: Add this to the index.html file along with the src library for firebase database.
(Check the reference link)

```

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    var firebaseConfig = {
18      apiKey: "AIzaSyBYV9kwljd-8zlrSvSLGiv2zBX4MhkNAo8",
19      authDomain: "multiplayer-car-racing-game.firebaseio.com",
20      databaseURL: "https://multiplayer-car-racing-game.firebaseio.com",
21      projectId: "multiplayer-car-racing-game",
22      storageBucket: "",
23      messagingSenderId: "936147099930",
24      appId: "1:936147099930:web:dba47c5bb648f4ef"
25    };
26    // Initialize Firebase
27    firebase.initializeApp(firebaseConfig);
28  </script>
29  <link rel="stylesheet" type="text/css" href="style.css"/>
30 </head>
31 <body>
32   <script src="sketch.js"></script>
33 </body>
34 </html>
35

```

Teacher shows how to get a reference to the position of the ball in the database.

.ref() is used to refer to the location of the database value we care about.

.on() creates a listener which keeps listening to the changes in the database.

Everytime a change in the database values of position (reference) happens, the readPosition function is called.

If there is any error in reading the values in the database, the showError function is called.

Student observes, asks questions and learns.

<pre> 1 var hypnoticBall, database; 2 var position; 3 4 5 function setup(){ 6 database = firebase.database(); 7 console.log(database); 8 createCanvas(500,500); 9 10 hypnoticBall = createSprite(250,250,10,10); 11 hypnoticBall.shapeColor = "red"; 12 13 14 var hypnoticBallPosition = database.ref('ball/position'); 15 hypnoticBallPosition.on("value", readPosition, showError); 16 } 17 18 function draw(){ 19 background("white"); </pre>	<p>In the readPosition function we can read the position of the value in the database.</p> <p>We assign the x and y values of the ball position in the database to the ball sprite.</p>	<p><i>The student learns how to read data from the remote real time database.</i></p>
<pre> 37 38 function readPosition(data){ 39 position = data.val(); 40 console.log(position.x); 41 hypnoticBall.x = position.x; 42 hypnoticBall.y = position.y; 43 } </pre>	<p>We also need to write the change in position when arrow keys are pressed. Teacher shows how to write to the realtime database.</p> <p>.set() is used to set the value in the database.</p> <p>Essentially the arrow presses are changing the position values of the ball only in the database. In our application, we are</p>	<p><i>The student learns how to write data to the database.</i></p>

	reading those values and displaying the ball at that position.	
	<p>Let's see what happens if we now run the application in two different browsers.</p> <p><i>Teacher shows the application in two different browsers.</i></p> <p>The ball in both the browsers seem to move in synchronized positions when the arrow keys are pressed.</p> <p>There are a few bugs though - can you notice them.</p>	<p>ESR: If the arrow key is pressed immediately when the app starts, the app shows an error.</p> <p>Also, when the app is started, the initial position is 250, 250 before it synchronizes to the database values of the ball's position.</p>
 <pre> 7 8 function draw(){ 9 background("white"); 10 11 if(keyDown(LEFT_ARROW)){ 12 writePosition(-1,0); 13 } 14 else if(keyDown(RIGHT_ARROW)){ 15 writePosition(1,0); 16 } 17 else if(keyDown(UP_ARROW)){ 18 writePosition(0,-1); 19 } 20 else if(keyDown(DOWN_ARROW)){ 21 writePosition(0,1); 22 } 23 drawSprites(); 24 } 25 26 function writePosition(x,y){ 27 database.ref('ball/position').set({ 28 'x': position.x + x , 29 'y': position.y + y 30 }) 31 } 32 </pre>		
	Can you think of why these bugs are present and how we can fix them?	<p>ESR: varied</p>

	<p>Why don't you try and see if you can fix them on your own.</p> <p>Before that let's write this application on your system so that you know how to set up a realtime database, connect to it, write and read to it.</p>	-
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
<ul style="list-style-type: none"> • Ask Student to press ESC key to come back to panel • Guide Student to start Screen Share • Teacher gets into Fullscreen 		
<p style="text-align: center;">ACTIVITY</p> <ul style="list-style-type: none"> • Write and read from the remote database. • Debug and fix the errors. 		
Step 3: Student-Led Activity (15 min)	<i>Guide the student to clone the boiler plate code for the application.</i>	<i>Student clones the boiler plate code for the application. (Student Activity 1)</i>
	<i>Guide the student to create a new real time database in firebase.</i>	<i>Student creates a new realtime database in Google Firebase and creates a new entry to store the ball's x and y position.</i>
	<i>Guide the student to connect the application to the database.</i>	<i>Student connects the database to the application.</i>
	<i>Guide the student to write the code to read the position of the ball from the database and assign the position to the ball sprite.</i>	<i>The student writes code to read the ball's position data from the database and display the ball.</i>

	<i>Guide the student to write to the database the new position of the ball when the arrow keys are pressed.</i>	<i>The student writes code to write to the database the new position of the ball when the arrow keys are pressed.</i>
	<i>Allow the student to run the application in two different browsers and check the movement of the ball - if they are synchronized.</i>	<i>Student runs and tests the application to see synchronized ball movements.</i>
	What is the error that you are seeing?	<p>ESR: If the arrow key is pressed immediately when the app starts, the app shows an error.</p> <p>Also, when the app is started, the initial position is 250, 250 before it synchronizes to the database values of the ball's position.</p>
	Why do you think there is this error?	<i>The student spends some time thinking about fixing the bug.</i>
	<p>We are declaring the 'position' variable in line 2 but it takes some time for the application to read the values from the database and assign it to this value.</p> <p>Till then the position value is undefined and the ball sprite is displayed at the default value of 250, 250 we used to create it.</p>	-

When we press the arrow key immediately on the start of the application we are trying to write the 'undefined' position values into the database.

We can fix this by drawing the ball or writing to the database only when 'position' is NOT EQUAL to undefined.

```

12
13
14 var hypnoticBallPosition = database.ref('ball/position');
15 hypnoticBallPosition.on("value", readPosition, showError);
16 }
17
18 function draw(){
19   background("white");
20   if(position !== undefined){
21     if(keyDown(LEFT_ARROW)){
22       writePosition(-1,0);
23     }
24     else if(keyDown(RIGHT_ARROW)){
25       writePosition(1,0);
26     }
27     else if(keyDown(UP_ARROW)){
28       writePosition(0,-1);
29     }
30     else if(keyDown(DOWN_ARROW)){
31       writePosition(0,+1);
32     }
33     drawSprites();
34   }
35 }
36
37
38 function writePosition(x,y){
39   database.ref('ball/position').set({
40     'x': position.x + x ,
41     'y': position.y + y
42   })
43 }

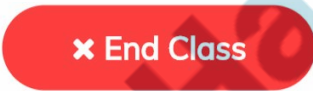
```

Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Encourage the student to make reflection notes in markdown format.**
- **Complement the student for her/his effort in the class.**
- **Review the content of the lesson.**

Step 4: Wrap-Up (5 mins)	<p>So you learned how to use a remote real time database in your application and you also did some debugging.</p> <p>How are you feeling about today's class?</p>	<p>ESR: Varied</p>
	<p>You get a hats off.</p> <p>We have just started with the multiplayer car racing game and connecting to the database is the first concept that we have uncovered.</p> <p>We will be learning more to make our first multiplayer car racing game.</p> <p>Keep practicing the skills you have learned till then.</p>	<p>Make sure you have given at least 2 Hats Off during the class for:</p> <div data-bbox="1019 726 1312 827">Creatively Solved Activities +10</div> <div data-bbox="1019 877 1312 978">Great Question +10</div> <div data-bbox="1019 1024 1312 1125">Strong Concentration +10</div>
Project Overview	<p>VIRTUAL PET - 1</p> <p>Goal of the Project:</p> <p>Today you learned how to create a remote real-time database, how to read and write and connect to a remote real-time database.</p> <p>In this project you will have to apply what you have learnt in the class and create a virtual pet app.</p> <p>Story:</p> <p>Shreya really wants a pet. But nobody else in her family wants to bring a pet in the</p>	

	<p>home. They believe Shreya is not old enough to take care of a pet.</p> <p>Can you create a virtual pet for Shreya? Create a pet in such a way that Shreya should be able to take care of it and make sure it does not die.</p> <p>I am very excited to see your project solution and I know you will do really well.</p> <p>Bye Bye!</p>	
<p style="text-align: center;">Teacher Clicks</p> <div style="text-align: center;">  </div>		
Additional Activities	<p>Encourage the student to write reflection notes in their reflection journal using markdown.</p> <p>Use these as guiding questions:</p> <ul style="list-style-type: none"> • What happened today? <ul style="list-style-type: none"> - Describe what happened - Code I wrote • How did I feel after the class? • What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult? 	<p>Student uses the markdown editor to write her/his reflection as a reflection journal.</p>

Activity	Activity Name	Links
Teacher Activity 1	Boilerplate Code	https://github.com/whitehatjr/asynchronousBallMovement

Student Activity 1	Boilerplate Code	https://github.com/whitehatjr/asynchronousBallMovement
Teacher Activity 2	Reference	https://github.com/whitehatjr/synchronousBallMovement

