



Topic	REAL-TIME DATABASE	
Class Description	The student learns the importance of using Google's real-time Firebase database to create multiplayer games. The student learns how to connect, read and write data to 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 multiple browsers at the same time. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ VS Code Editor ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources <ul style="list-style-type: none"> ○ VS Code Editor ○ 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 20min 5 mins
WARM-UP SESSION - 5 mins		
<div>  </div> <p>Teacher starts slideshow from slide 1 to slide 18 Refer to the speaker notes and follow the instructions on each slide.</p>		

Teacher Action	Student Action
<p><i>Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?</i></p> <p>Run and show the presentation in a slideshow from slides 1 to 3</p> <p>Following are the objectives of the WARM-UP session:</p> <ul style="list-style-type: none"> • Greet the student • Revision of previous class activity • Warm-up Quiz 	<p>ESR: Hi, thanks, Yes I am excited about it!</p> <p>Click on the slideshow tab and present the slides</p>
Q&A Session	
Question	Answer
<p>What is Matter.js?</p> <p>A. 1D Physics engine built for the web. B. 2D Physics engine built for the web. C. 3D Physics engine built for the web. D. 4D Physics engine built for the web.</p>	B
<p>What property is used to set the bounciness of a body in the matter.js engine?</p> <p>A. friction B. density C. restitution D. velocityY</p>	C
Continue the WARM-UP Session	
Teacher Action	Student Action
<p>Run and show the presentation in a slideshow from slides 4 to 18 to set the problem statement.</p> <p>Following are the objectives of the WARM-UP session:</p>	<p>Narrate the story by using hand gestures and voice modulation methods to create more interest in</p>

<ul style="list-style-type: none"> • Appreciate the student on his/her performance in the quizzes. • Create an arrow key-controlled ball. • Show how the ball is asynchronous in the different browser windows. • Create and connect to a remote database. 	students.
<div>  Teacher ends slideshow </div>	
TEACHER-LED ACTIVITY - 15 mins	
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 the different browser windows. • Create and connect to a remote database. 	
Teacher Action	Student Action
<p>We already know how to design a game with multiple characters.</p> <p>We can host the game online and then open the game on different browsers on multiple computers.</p> <p>However, can you think about what is stopping us from designing a multiplayer game?</p>	<p><i>The student thinks about it.</i></p> <p>ESR:</p> <p>The game is in different states in the two browsers. For a multi-player game, we need the two browsers to have the game in 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>

Exactly! Let's look at this problem with an example.

The teacher writes the code (same as [Teacher Activity 1](#)) to create a simple ball that can move using arrow keys.

We can also add a margin to our canvas to place it in the center.

The student observes and learns.

In **sketch.js**:

```

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 }
  
```

In **.css** file:

```

# 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 now move the ball and see what happens.

What do you see?

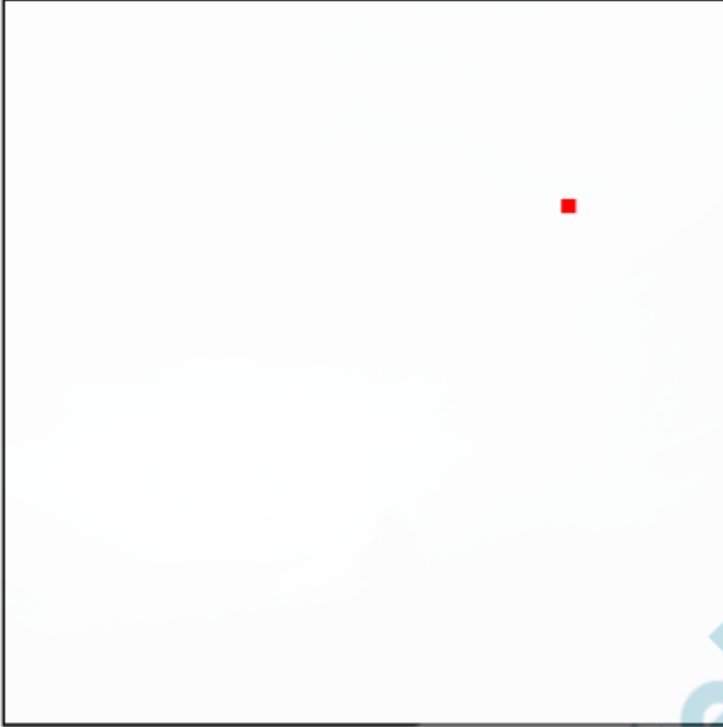
ESR:

The ball in the two browsers moves 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.

Multiplayer games use a remote database to work. Multiplayer games store the position of the state of the game at all times in a remote database.

ESR:

The student listens and observes.

All the players' consoles /browsers read the game from this remote database and write to it when they make any change in the game.

Therefore, in our game, the two browsers will read the position of the ball from the common remote database and the balls on two different browsers, on two different computers, will always be synchronized.

Let's create a remote database on the cloud for our simple application. This remote database will store the state (positions) of the ball and will allow us to read or write to it at any time.

We will be using **Google's Firebase Realtime Database** for this purpose.

The teacher guides the student on how to create a Realtime Database and create a variable named ball, which stores two values x and y.

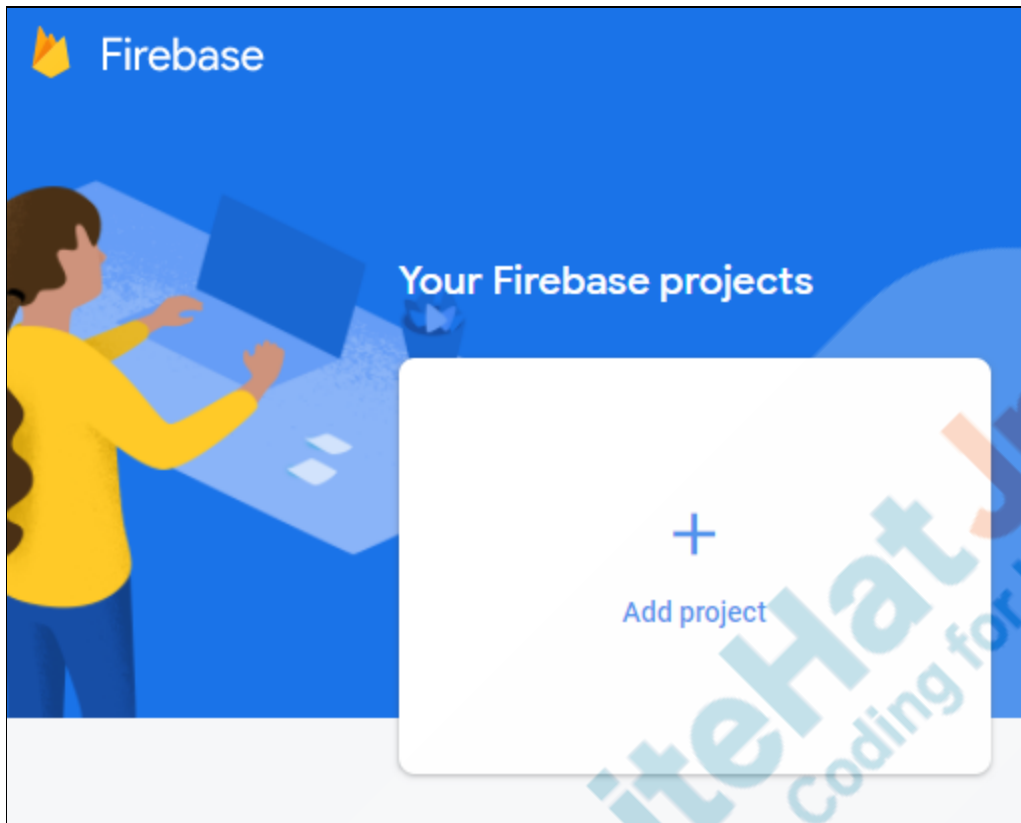
The database can be compared to a **JSON** data structure format.

```
ball = {  
  x: 250,  
  y: 250  
}
```

The student observes on how to create a realtime Database in Firebase.

Step 1: Go to <https://console.firebase.google.com> and log in with your Gmail ID.

Step 2: Click on 'Add Project'.



Step 3: Enter the name of your project. Accept the Firebase terms and click the **Continue** button.

× Create a project (Step 1 of 3)

Let's start with a name for
your project?

Project name

BallMovement

ballmovement-32892

Select parent resource

Continue

Step 4: Disable **Google Analytics** for this project. (We don't need it)

× 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:

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

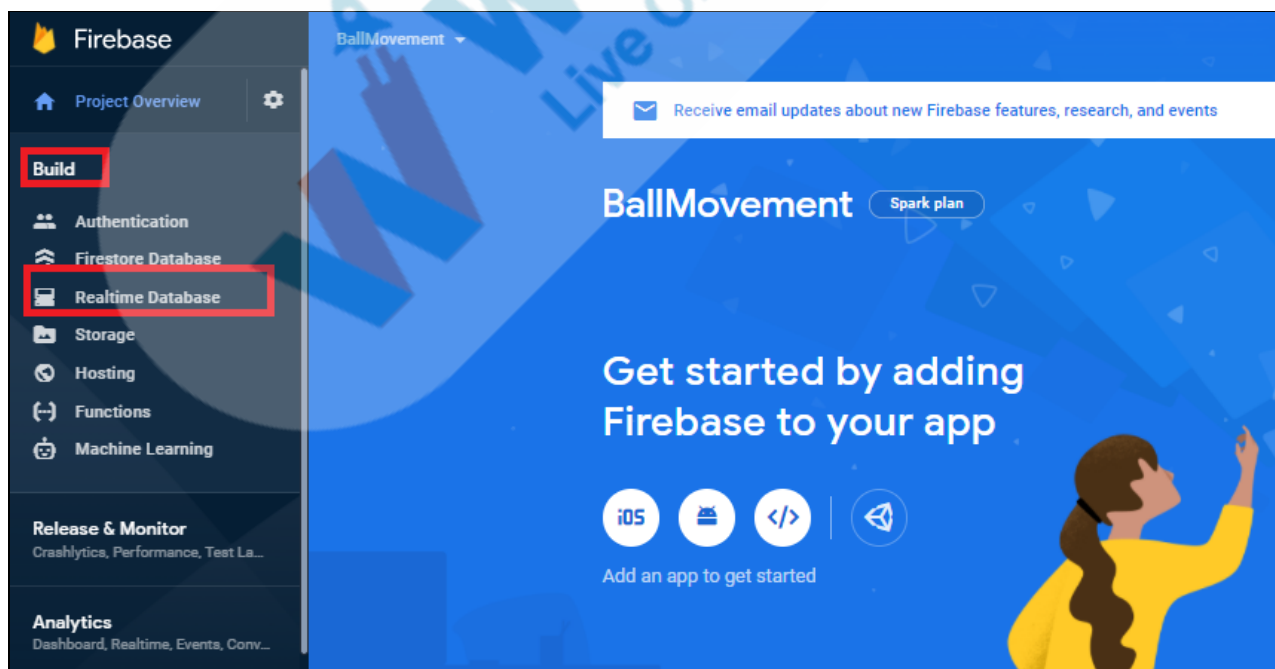
☐ Set up Google Analytics for my project
Configure in the next step

☒ Not right now
You can change this later

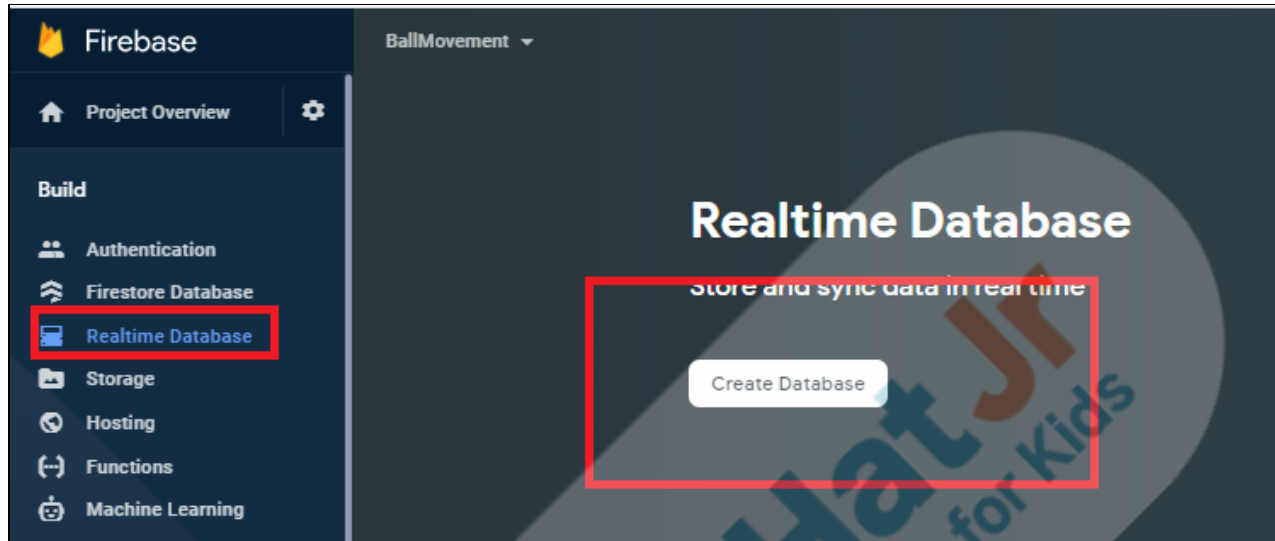
[Previous](#) [Create project](#)

* It takes a while to create the project; Once done, click the **Continue** button.

Step 5: Click **Build** and then on **Realtime Database** on the left-hand navigation panel.

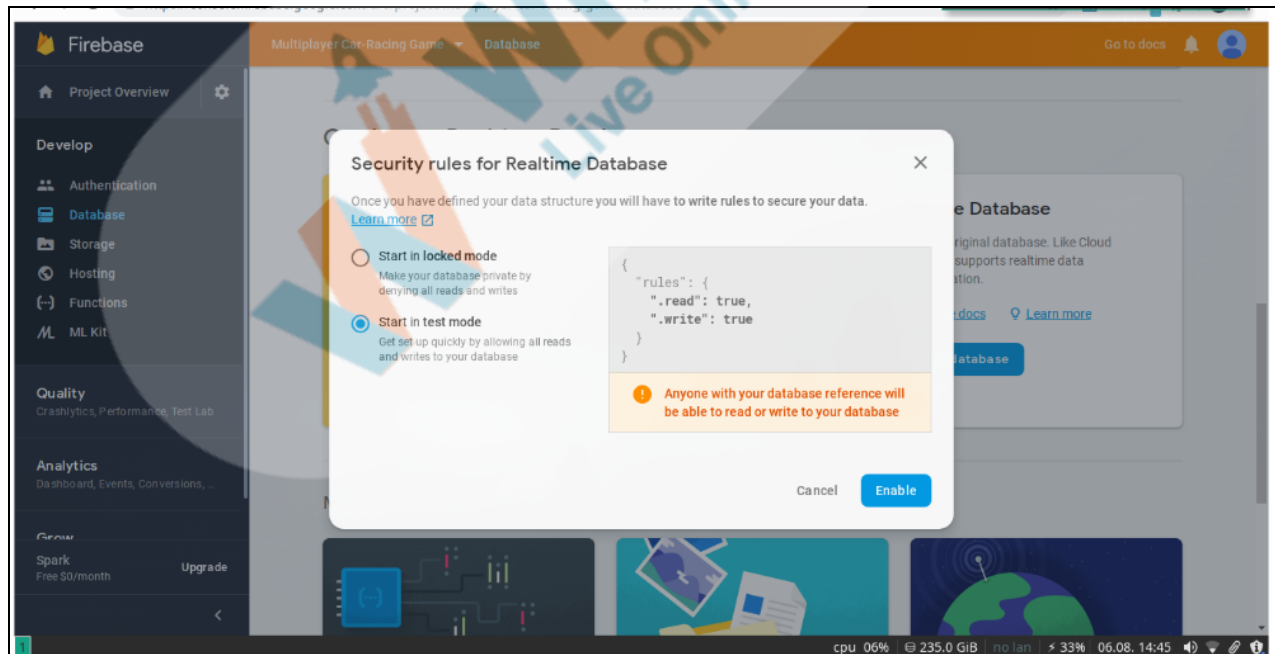


Step 6: Click 'Create Database'.

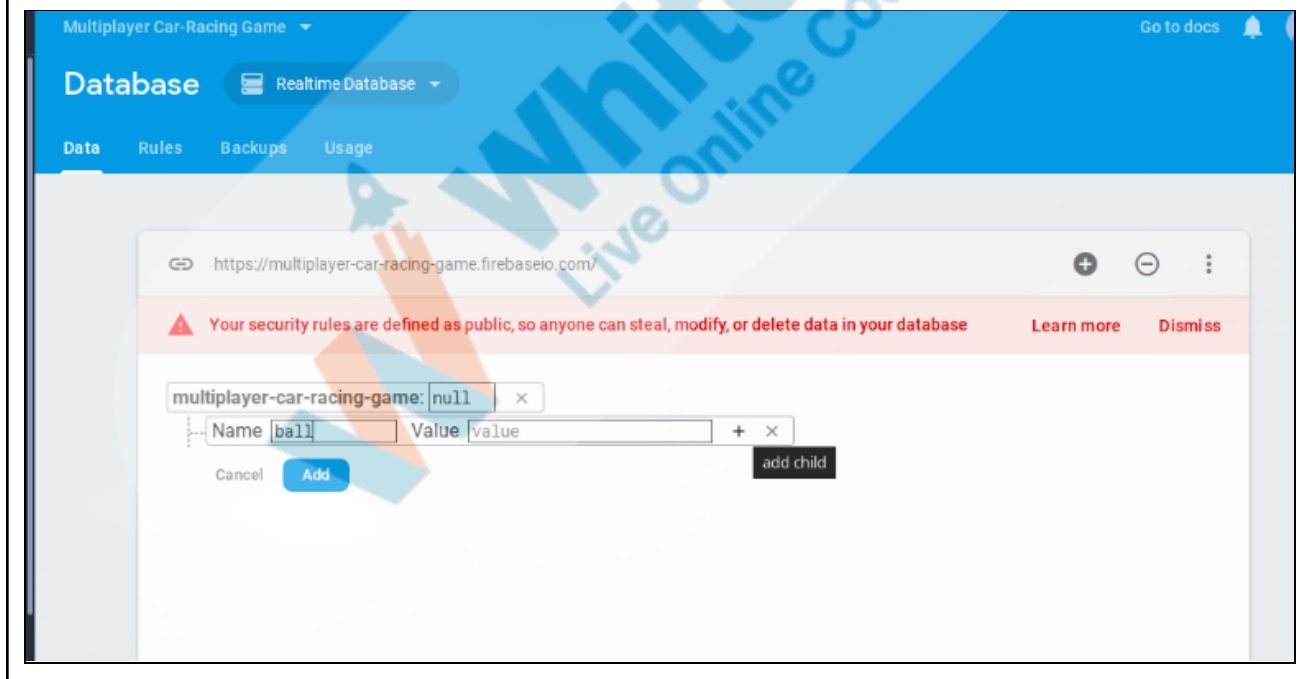
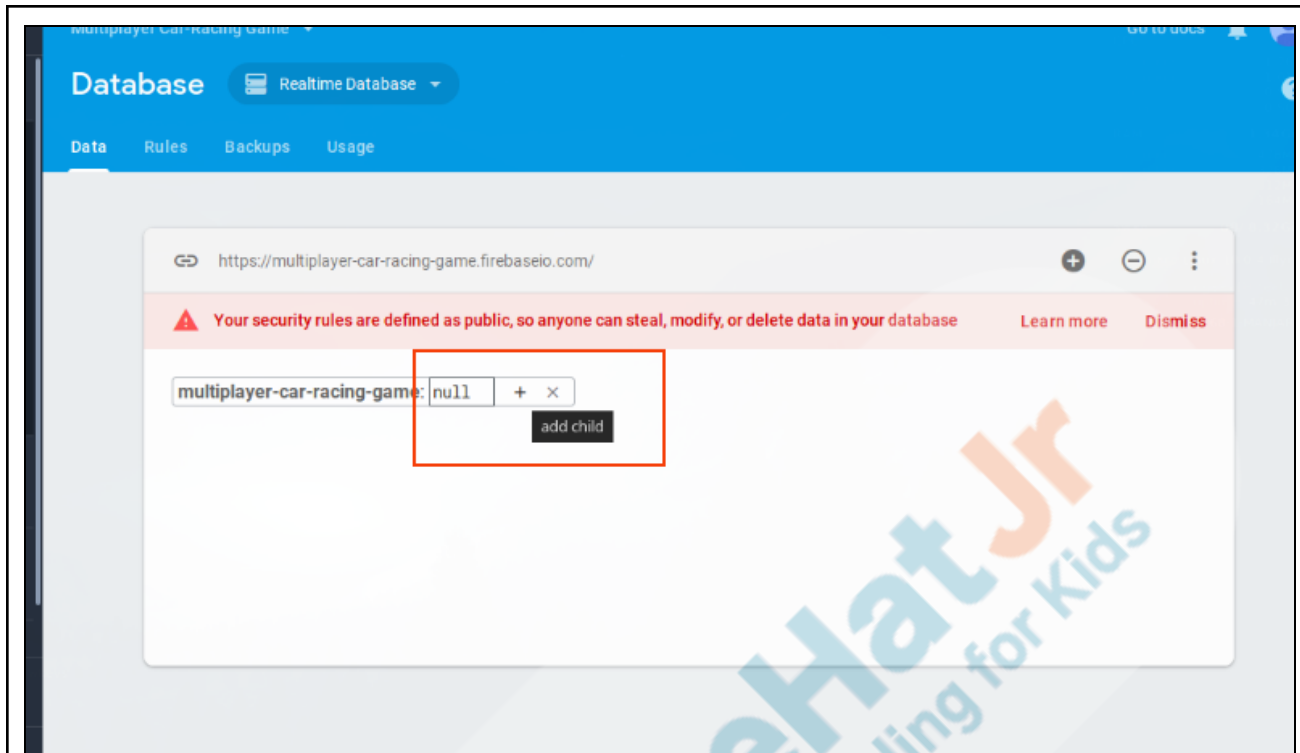


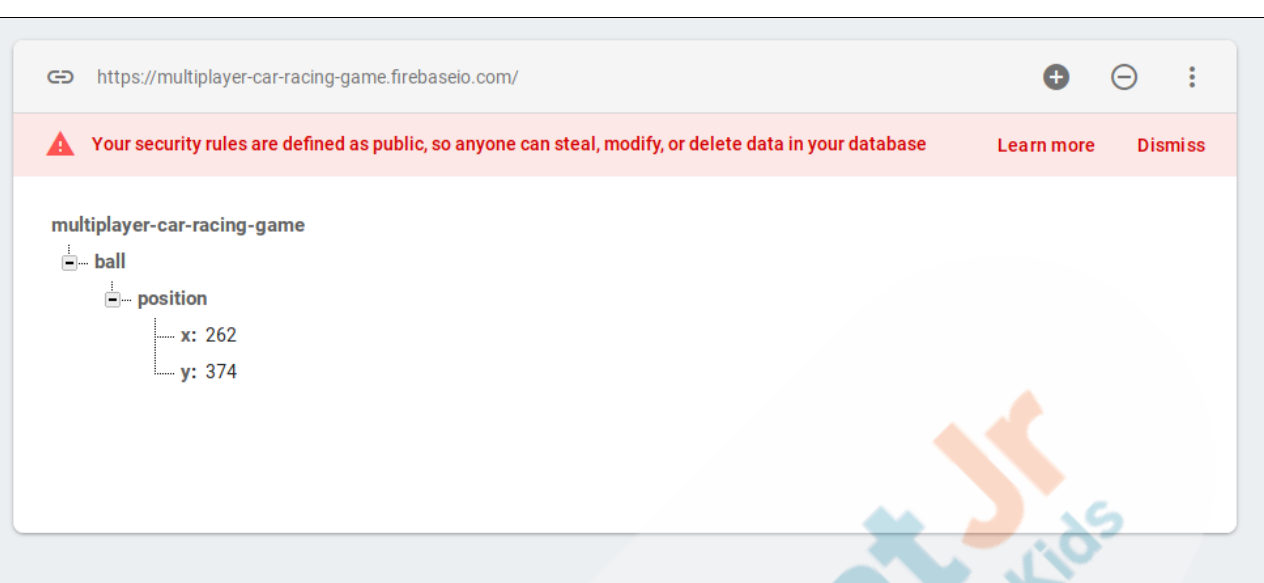
Step 7: Create the database in the test mode for now.

Note: 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 8: Add a child to create nodes, which can hold the ball's x and y positions.





Now that our database is ready, how do we access it from our code?

We need to connect the **Firestore Realtime Database** that we created to our application.

We will have to import a few Firebase libraries, which will allow us to read and write to our database.

We will also need some configuration settings, such as the API key, database URL, and so on, which can authenticate our application and allow it to write to the database.

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 to our HTML file.

The teacher shows how to connect the application to the Firestore Database and how to initialize the Firestore App with the Firestore config.

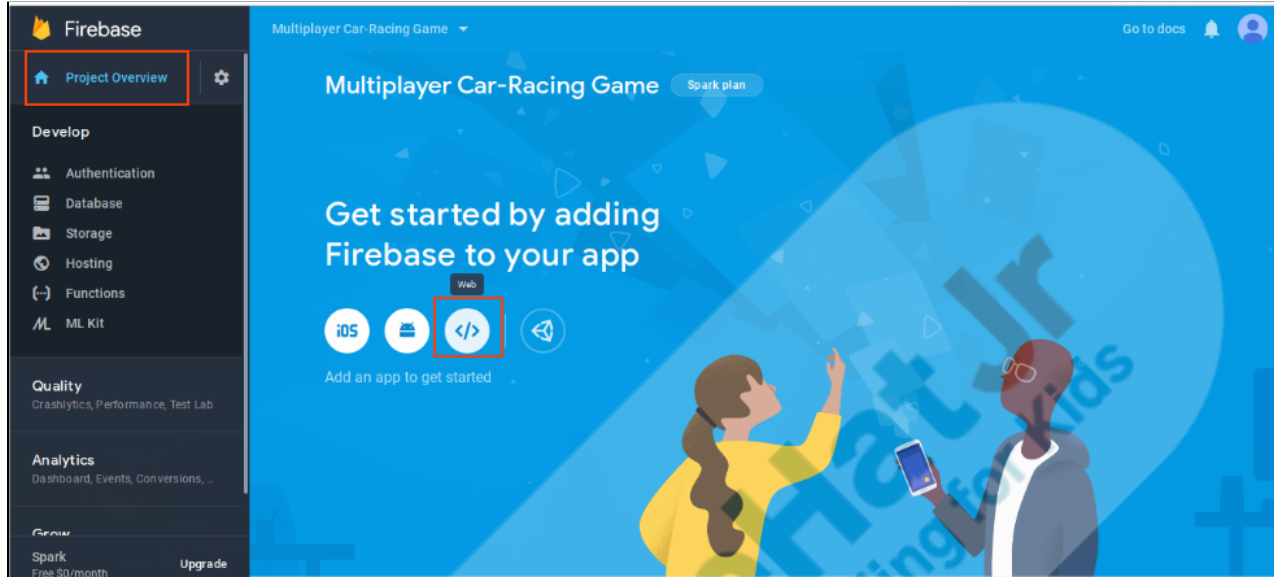
ESR: Varied

ESR:

The student observes and learns how to initialize the Firestore app.

Step 1: Click “Project Overview” and Project Settings.

Click on the icon of “add to the 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 the **SDK code** to the **index.html** file along with the **src** library for Firebase database.

```

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: "AIzaSyBYV9kw1jd-8z1RsYSLGIv2zBX4MhkNAo8",
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

```

The 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.

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

If there is an error in reading the values in the database, the **showError()** function is called.

The student observes, asks questions, and learns.


```

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");

```

In the **readPosition()** function, we can read the position of the value in the database.

The student learns how to read data from the remote real-time database.

What do we need to read from the database?

ESR: Position of a ball

We will read them and assign the x and y values of the ball position in the database to the ball sprite.

```

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

```

We will also write a **showError()** function which will be called in case there is an error in fetching the data from the database.

The **showError()** function is an optional function that can be avoided too.

```
function showError() {
    console.log ("Data not recived from the database");
}
```

The teacher runs the code

The output shows the position of the ball is moved to 100, 100 in both browsers. However, when we move the ball in one browser, it is still not reflecting in another browser.

How can we resolve that?

Yes.

.set() is used to set the value in the database.

Essentially the arrow presses are changing the position values of the ball only in the database. In our application, we are reading those values and displaying the ball at that position.

Now, why don't we set up the database at your system, and create a function to write the position there, so that you can experience it by yourself?

ESR: We need some function to write values in the database



ESR: Yes

Teacher Stops Screen Share

STUDENT-LED ACTIVITY - 20 mins

- **Ask the Student to press the ESC key to come back to the panel**
- **Guide the Student to Start Screen Share**
- **The teacher is getting into Fullscreen mode**

ACTIVITY

<ul style="list-style-type: none"> • Write and read from the remote database. • Debug and fix the errors. 	
<div>  </div> <p>Teacher starts slideshow :Slides 19 to 20</p>	
<p>Run and show the presentation in a slideshow from slides 19 to 21 to set the student activity context.</p> <ul style="list-style-type: none"> • Setting up Firebase database • Read and Write ball position in the database. 	
<div>  </div> <p>Teacher ends slideshow</p>	
Teacher Action	Student Action
<p><i>Guide the student to download the boilerplate code for the application.</i></p>	<p><i>The student downloads the code for Student Activity and opens it in the VS Code editor.</i></p>
<p><i>Guide the student to create a new real-time database in Firebase.</i></p>	<p><i>The student creates a new real-time database in Google Firebase and creates a new entry to store the ball's x and y positions.</i></p>
<p><i>Guide the student to connect the application to the database</i></p>	<p><i>The student connects the database to the application.</i></p>
<p>Before writing any further code, why don't you run the code in two browsers?</p>	<p><i>The student clicks on Go Live, and copies the same link in another browser.</i></p>

What do you see in both browsers?

ESR:

The initial position of the ball is 250, 250 and then it jumps to 100, 100 in both browsers. But when we move the ball in one browser, it does not show in another and also the value of the database is not changing.

Correct, that is because we are not writing back the changed position in the database.

Similar to reading values from the database, while writing values back to the database we need to provide location.

Do you remember, which function we use to referring to a particular location in the database?

Probe the student to refer to .ref() function given in boilerplate code.

ESR: .ref()

Yes, we will first provide location using **.ref()** and then we will use **.set()** to update the value of x and y.

Let us rename the **changePosition()** everywhere with **writePosition()**. Now, whenever a key is pressed we want to write new the position into the database.

In **writePosition()** instead of changing the ball.x +1 and ball.y +1, we will change it to position.x+1 and assign it to the 'x' field of the database. we will do the same for the 'y' field of the database.

The student writes the code for writePosition().

```

17
18 function draw(){
19   background("white");
20
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 function writePosition(x,y){
38   database.ref('ball/position').set({
39     'x': position.x + x ,
40     'y': position.y + y
41   })
42 }

```

Awesome, Lets us run the code and see the output.

The ball in both browsers seems to move in synchronized positions when the arrow keys are pressed.

There are a few bugs though, did you notice them?

If the arrow key is pressed immediately when the app starts, the app shows an error.

Also, when the app is started, the initial position is 250, 250 before it synchronizes to the database values of the ball's position.

Can you think of why these bugs are present and how we can fix them?

The student clicks on Go live to see the output and also checks the database after moving the ball.

ESR: Varied

Allow the student spends some time thinking about fixing the bug.

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.

Till then, the position value is undefined and the ball sprite is displayed at the default value of 250, 250 that we had used to create it.

When we press the arrow key immediately at 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

WRAP-UP SESSION - 5 Mins



Teacher starts slideshow from slides 21 to 32

Teacher Action	Student Action
<p>Run the presentation from slides 21 to 32</p> <p>Following are the objectives of the WARM-UP session:</p> <ul style="list-style-type: none"> • Appreciate the student. • Revise the current class activities. • Discuss the quizzes. 	<p>Discuss the current class activities with the student and the Student can ask and clear his/her doubts related to the activities.</p>
Quiz Time - Click the In-Class Quiz	
Question	Answer
<p>If a game object moves independently from each other in two different browsers, they are:</p> <ol style="list-style-type: none"> synchronous semi-synchronous asynchronous semi-asynchronous 	C
<p>_____ servers are computers which are remotely connected through the internet and maintain your data which you can use in your applications.</p> <ol style="list-style-type: none"> Console Database Databank Codebank 	B
<p>Which of the following databases did we use for our code?</p> <ol style="list-style-type: none"> Firebase Cloud Firestore Back4app Parse Firebase Realtime Database 	D

End the quiz panel	
<u>FEEDBACK</u> <ul style="list-style-type: none"> • Appreciate the student for their efforts in the class. • Ask the student to make notes for the reflection journal along with the code that they wrote in today's class. 	
<p>You get Hats off for your excellent work!</p> <p>In the next class, we will start working on a multiplayer car racing game using the Firebase Database.</p>	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="background-color: #0072bc; color: white; padding: 5px; margin-bottom: 5px; display: flex; align-items: center; gap: 10px;"> Creatively Solved Activities  +10 </div> <div style="background-color: #0072bc; color: white; padding: 5px; margin-bottom: 5px; display: flex; align-items: center; gap: 10px;"> Great Question  +10 </div> <div style="background-color: #0072bc; color: white; padding: 5px; display: flex; align-items: center; gap: 10px;"> Strong Concentration  +10 </div> </div>
<p>* This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.*</p> <p>Project Overview</p> <p>AIR BALLOON RIDE</p> <p>Goal of the Project: In Class 35, 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 have to apply what you have learned in the class and set up a real-time database for your game. Also, add a background image for the game and add an air balloon and add keyPress events.</p>	<p><i>Students engage with the teacher for the project.</i></p>

Story:

Kanchan went to an event and saw people enjoying a hot air balloon ride at the event. But due to lack of timing, she missed the hot air balloon ride. After coming back home, Kanchan plans to create her own virtual hot air balloon in which she can virtually travel with her cousins. However, Kanchan knows that she is not very good at coding.

Can you help her in creating this virtual hot air balloon ride game?

I am very excited to see your project solution and I know you will do really well.

Bye Bye!

Teacher ends slideshow



Teacher Clicks

✕ End Class

ADDITIONAL ACTIVITIES

Additional Activities

Encourage the student to write the reflection notes in their reflection journal using Markdown.

Use these as guiding questions:

- What happened today?
 - Describe what happened.
 - The 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?

The student uses the Markdown editor to write their reflections in a reflection journal.

Activity	Activity Name	Links
Teacher Activity 1	Boilerplate Code	https://github.com/whitehatjr/asynchronousBallMovement
Student Activity 1	Boilerplate Code	https://github.com/pro-whitehatjr/C35RV_SynchronousBallMoment_StudentActivity
Teacher Activity 2	Teacher Reference	https://github.com/whitehatjr/synchronousBallMovement
Teacher Reference	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO_C35_V3_lit_Withcues.html
Teacher Reference	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/79c09efa-3efa-4d68-a334-471acbc3cb3.pdf
Project Solution	Air Balloon Ride	https://github.com/pro-whitehatjr/b9954e8d6c517e38285e465c5f76de9a