CODE NINJAS®

Bronze Belt Ninja Guide

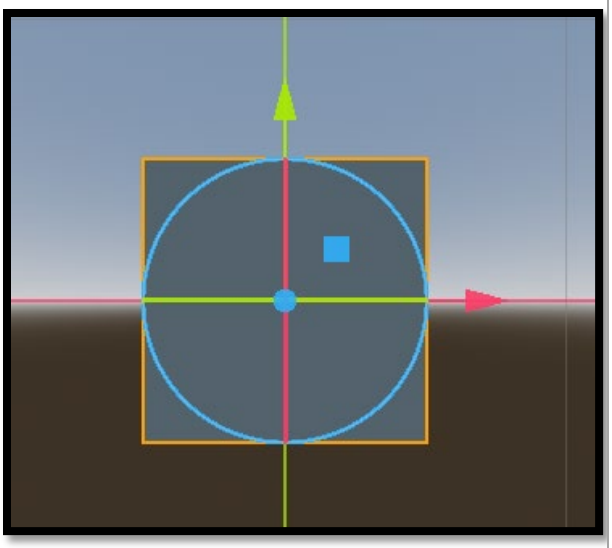
Activity 01: Dropping Bombs

2D and 3D Worlds

Games are typically developed in either a **2-Dimensional (2D)** or **3-Dimensional (3D) environment**. Each dimension offers different perspectives and design possibilities!

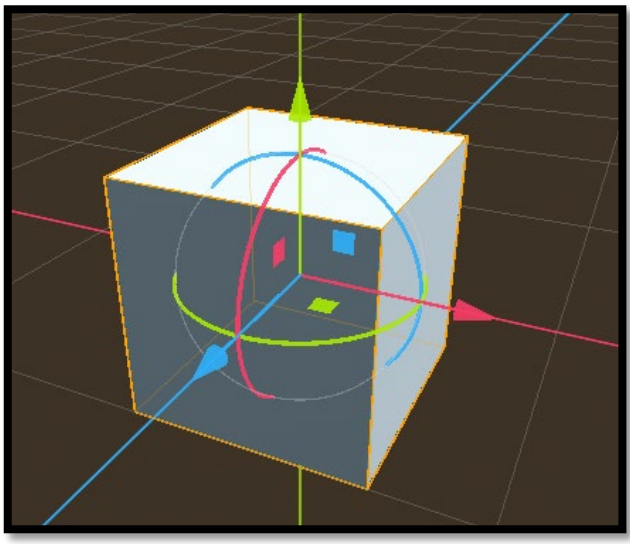
2D Environments

- Appears flat on screen
- Uses **x** axis to control **horizontal** movement
- Uses **y** axis to control **vertical** movement



3D Environments

- Appears to have depth and volume
- Uses **x** axis to control horizontal movement
- Uses **y** axis to control vertical movement
- Uses **z** axis to control forward/backward movement



Godot provides flexibility to create games in either dimension. You can start a Godot project in 3D and change the perspective of the camera to craft a 2D game using 3D elements. Dropping Bombs will be built with this method!

Sensei Stops

Pause for a **Sensei Stop!**

Throughout these projects, there will be **Sensei Stops.** A **Sensei Stop** is a checkpoint with a **Code Sensei** to make sure everything in your game is correct before moving on. Think of it like the yellow "Check Code" buttons in IMPACT. This will help you to debug projects and give you a chance to show off your amazing hard work!

At each **Sensei Stop**, make sure to:

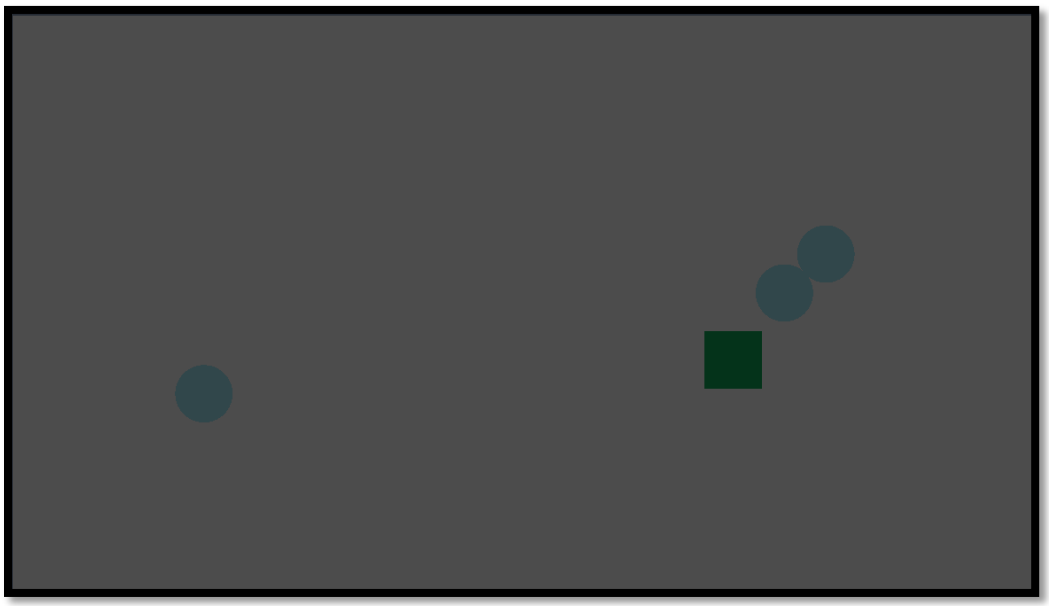
- Pause and call over a Code Sensei.
- While waiting for a Code Sensei, review previous steps and concepts.
- Show off the game and explain what you understand to the Code Sensei.
- Get the clear from a Code Sensei before continuing onto the next steps.

At this **Sensei Stop**, make sure to check in with a Code Sensei to tell them what you know about Godot, nodes, and trees so far!

Activity 01: Dropping Bombs

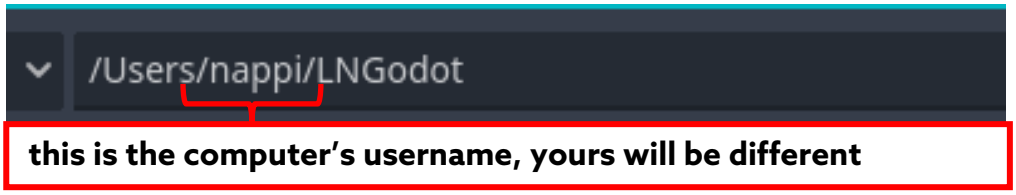
In this activity, you will control a player using direction buttons, or WASD keys, to avoid bombs dropping from the sky. This simple project will help you practice the basics of Godot and learn about the difference between 2D and 3D environments.

By the end of this activity, you will have explored how to create and manipulate nodes, cameras, animatable bodies, rigid bodies, and collisions.

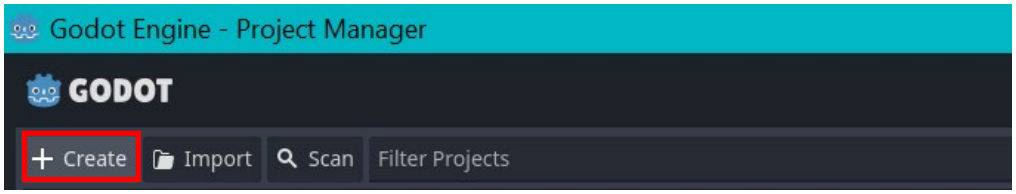


1 All projects in Godot will be stored in a path like:
/Users/[MyComputerUsername]/[MyInitials]Godot

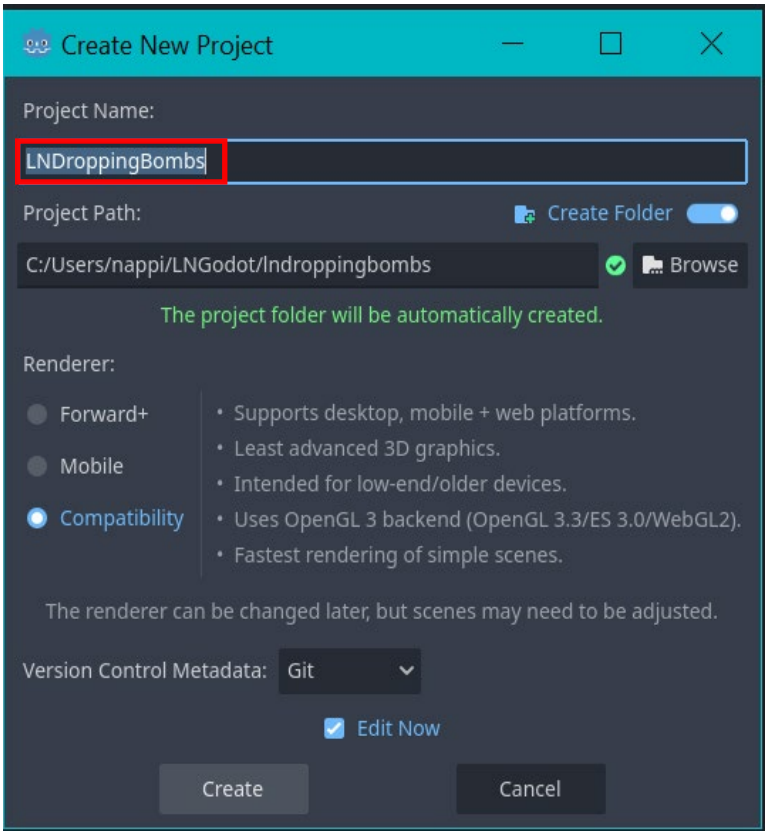
Don't worry if the path looks slightly different from the image shown! All computers have their own username.



2 After opening Godot, in the top left corner select **+ Create**.

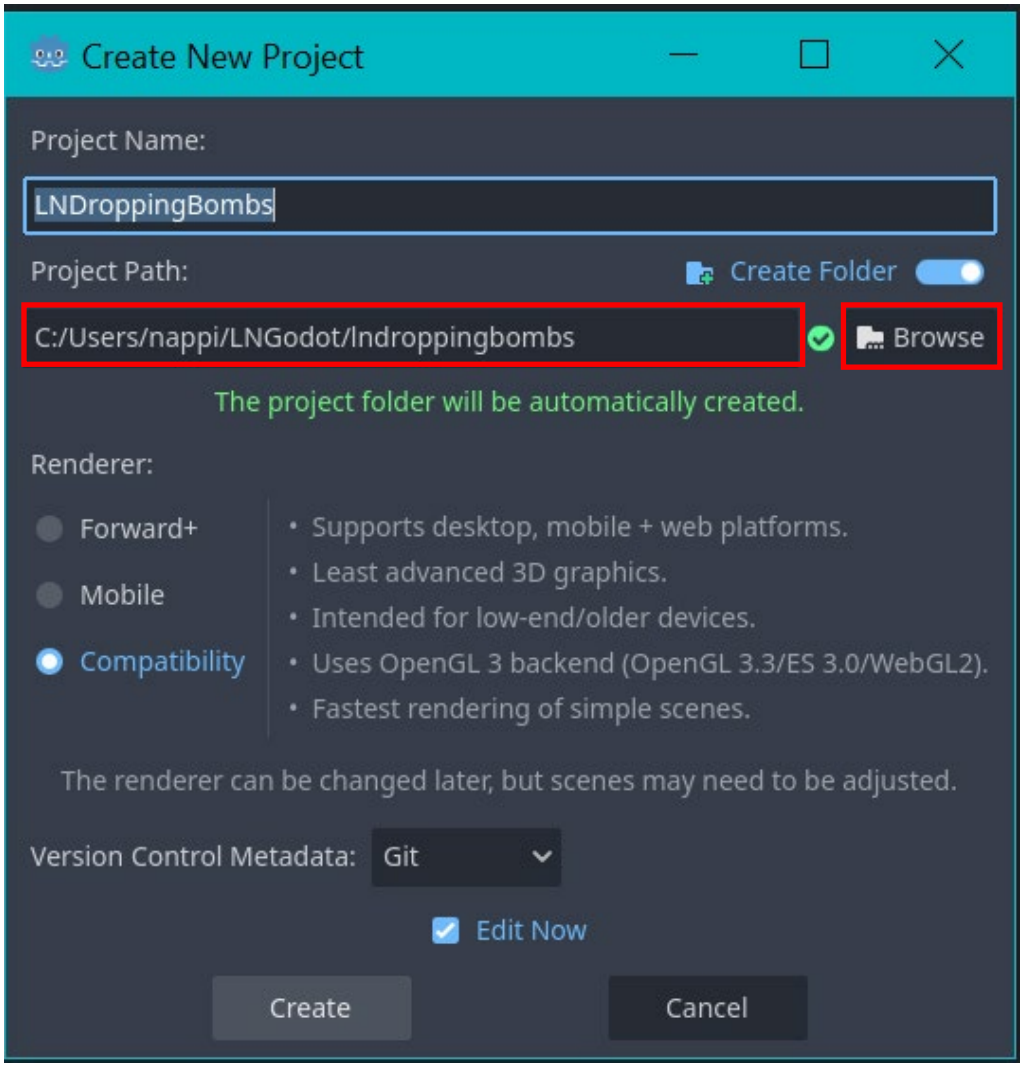


A **Create New Project** window will pop up. Name the project **[MyInitials]DroppingBombs.**



3 Make sure the **Project Path** is the same as specified from a Code Sensei in **Activity 00.** Refer back to **Step 1** for what the **Project Path** might look like.

Select **Browse** next to the **Project Path.** This will open a new window. At the top, type in the correct **Project Path.**



Once you have the correct path, click **Select Current Folder** at the bottom. This will bring you back to the **Create New Project** window.

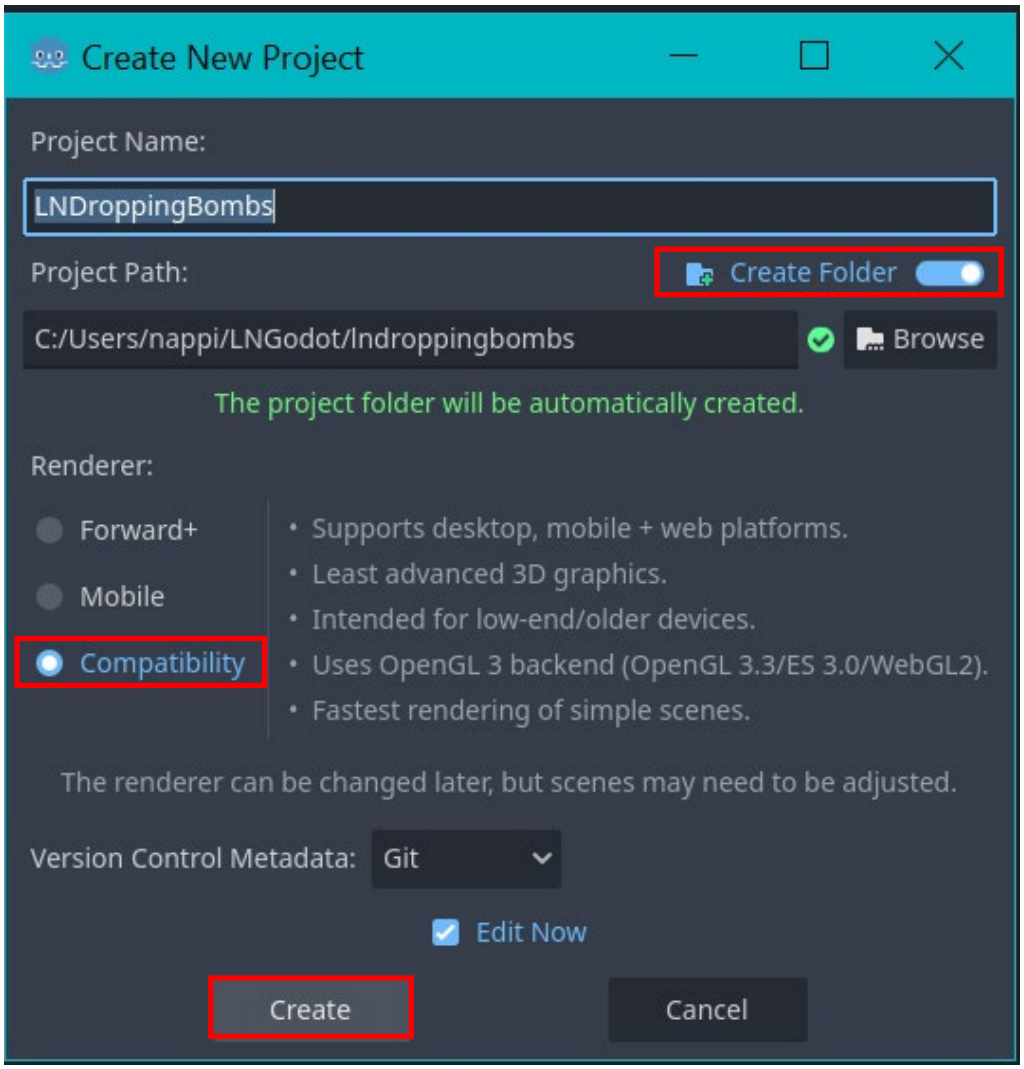
Pause for **Sensei Stop #1!**

Check in with a Code Sensei before moving on. Make sure the **Project Path** is correct.

4 The **Create New Project** window should be on the screen right now.

Make sure **Create Folder** is toggled **on** to keep things organized. Godot can only create the project in an **empty** subfolder.

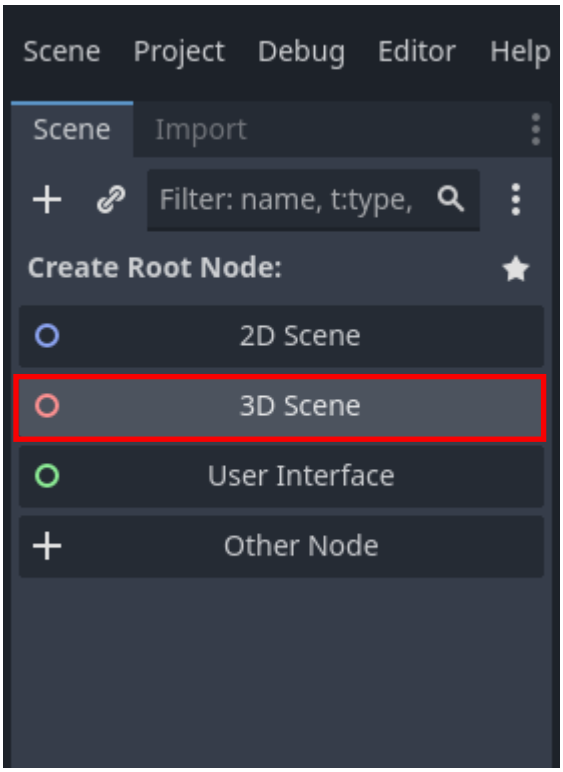
Select **Compatibility** as the renderer, then click **Create.**



5 First, the **Main root node** and **main scene** need to be created.

Recall that the **Main root** is like the root of a tree. This acts as the parent to all other nodes in the project. A tree of nodes can be saved to a file called a **scene**. Godot requires a **main scene** to run a project.

In the top left corner, find the **Scene** menu. This is where all nodes in the project are stored and where new nodes are created. Select **3D Scene** to create a **Node3D** as the **Main root** to a new **scene.**

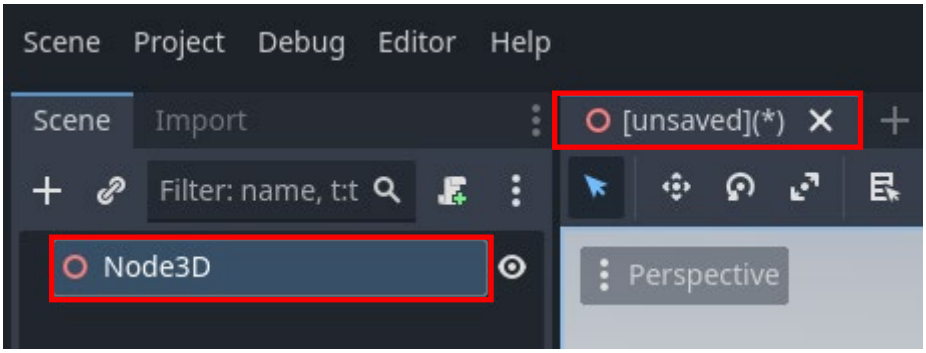


Pro Tip:

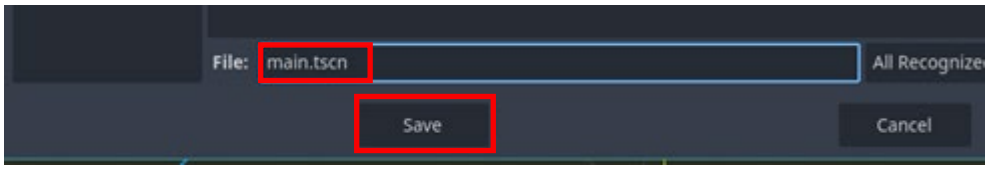
The "Scene menu" is where it says "Scene" and has a blue line over it. The other "Scene", above it, is a dropdown menu that handles advanced Scene options.

6 Once the **Node3D** has been added, look around the editor. What has changed?

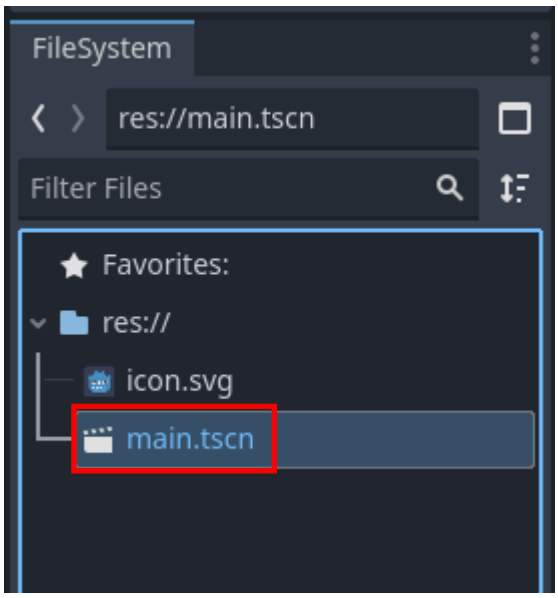
Notice in the top middle of the editor that the new **scene** is **unsaved.**



7 Press **CTRL** and **S** at the same time on the keyboard to save the scene. In the **Save Scene As...** window, name the scene **main** and select **Save**.

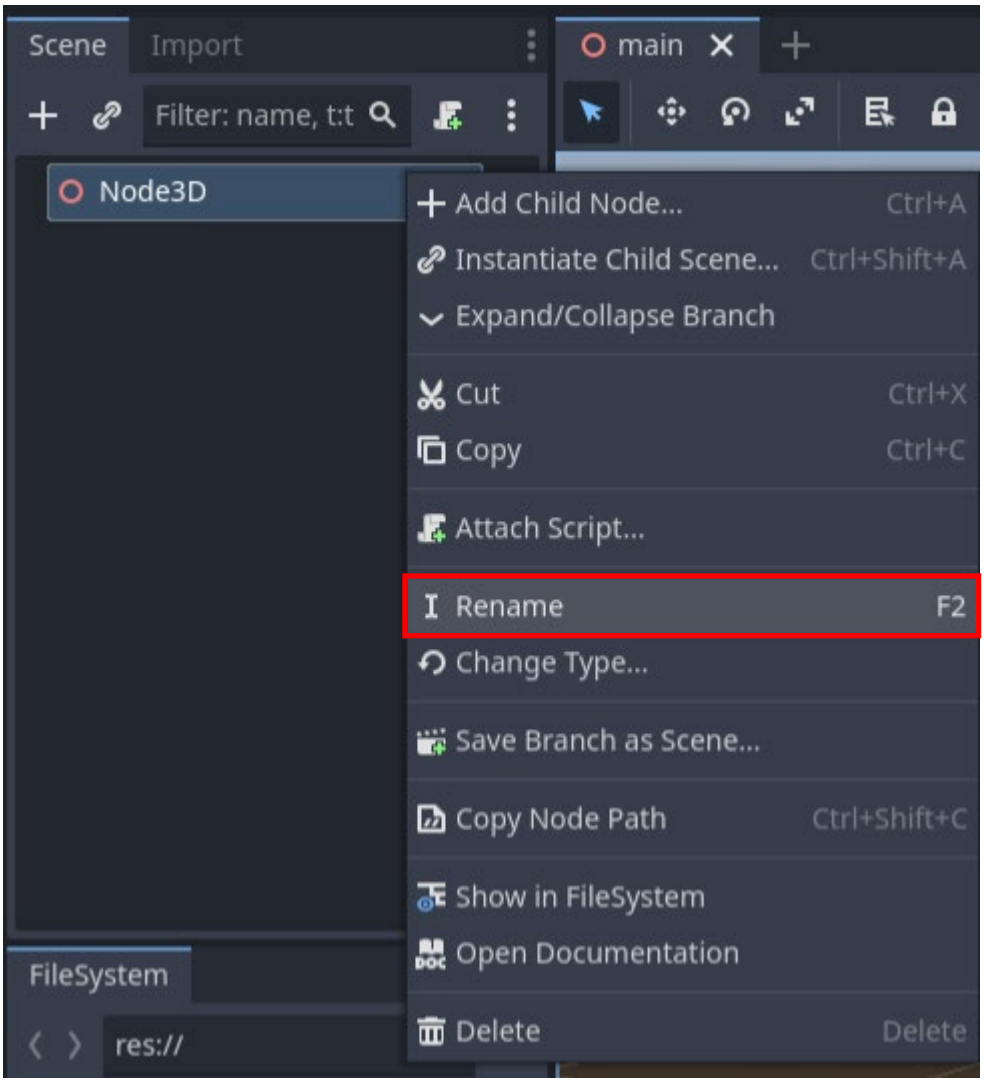


Find **FileSystem** in the bottom left corner of the editor. This is where all the scenes and scripts of the project will be stored. Notice that the **main scene** is now present.



8 Lots of nodes will be added to the project. To stay organized, rename the **Main root node**.

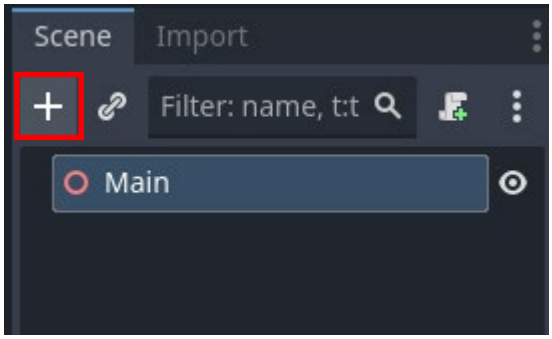
In Scene, **right-click** on **Node3D** and select **Rename**. Replace **Node3D** with **Main**.



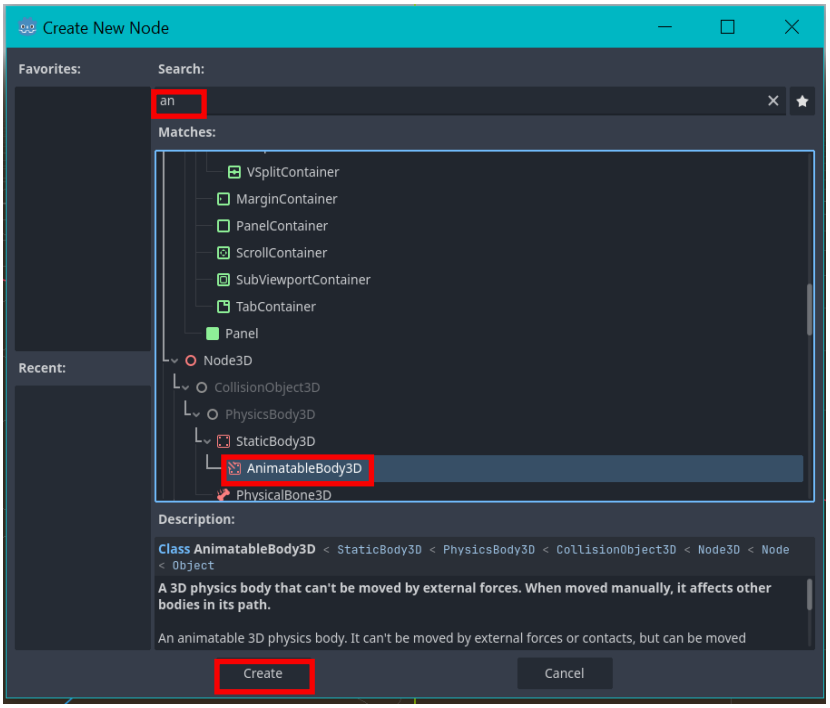
9 Notice the (*) symbol next to the **main scene.** This means the scene is currently **unsaved**. Make sure to press **CTRL + S** on the keyboard frequently to save the game to prevent losing progress.

10 Add a simple square player to the game.

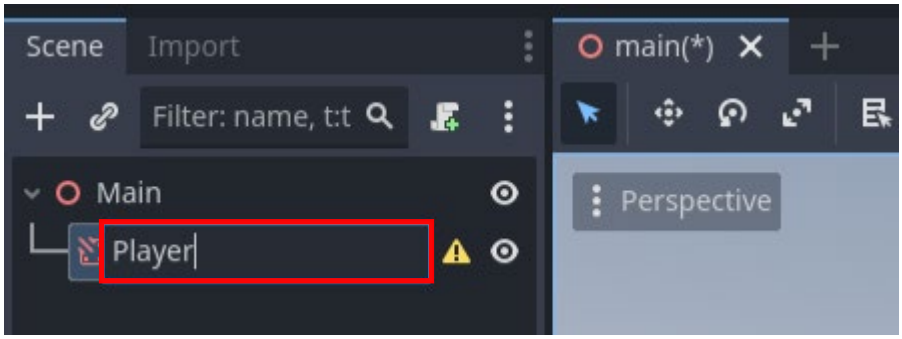
In **Scene**, select the **+** button to add a new child node. The player should only move when controlled; it should not move if any other objects collide with it. In cases like these, an **AnimatableBody3D** node is best.



In the search bar, begin typing **AnimatableBody3D** until the node shows up, then select it. Click **Create** to add the node to the game.

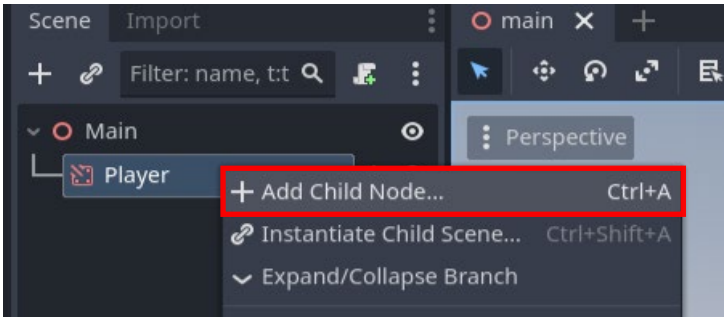


11 **Double-click** on **AnimatableBody3D** and type **Player** to rename it. This is another way to rename a node.

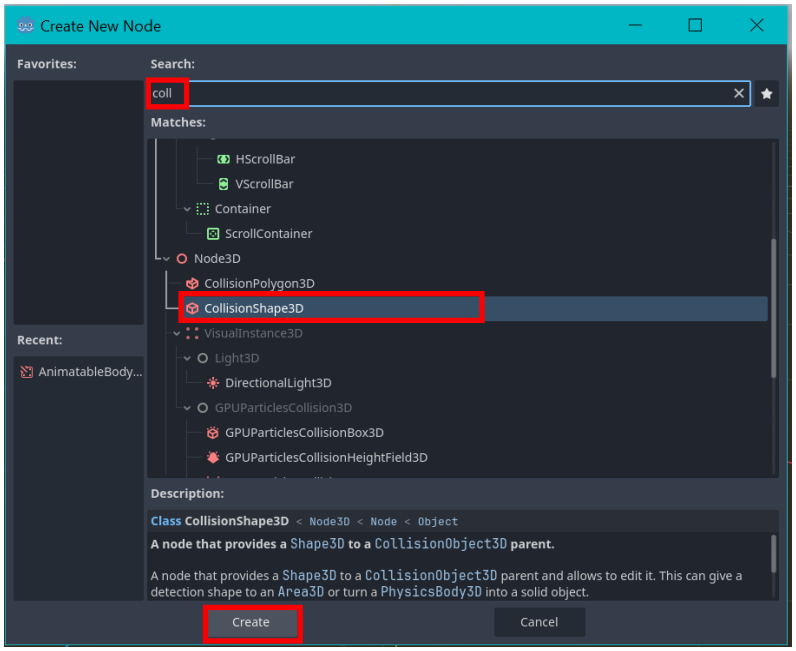


12 Notice the hazard symbol next to the **AnimatableBody3D** node under the **Main** parent node. This is a **Node Configuration Warning**, which signals that the node is not set up correctly, because it has no shape. This means it cannot collide with or interact with other objects.

Add a child **CollisionShape3D** node to the Player to fix this. **Right-click** on **Player** and select **Add Child Node**.



Search for **CollisionShape3D** and once found, select it and click **Create.**

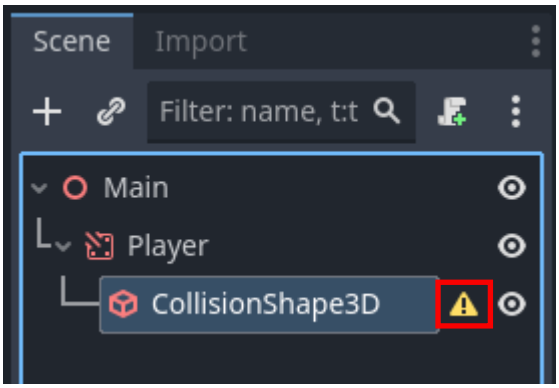


Pro Tip:

Notice Godot has a “Recently Used Nodes” section located in the bottom left when creating a new node. How might this be helpful?

13 In **Scene**, make sure the hierarchy of nodes matches the image, and that **CollisionShape3D** is a child node to **Player.**

Notice the hazard symbol is now on **CollisionShape3D.** This is because the shape for the collision still needs to be added.



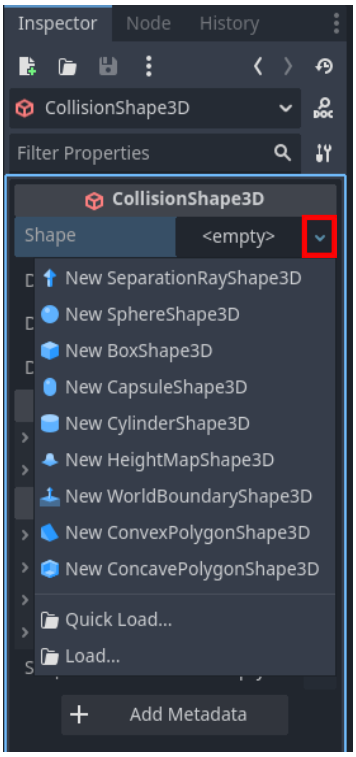
Pro Tip:

Nodes can be dragged to other nodes to reparent or rearrange them.

14 To add a collision shape, find the **Inspector** window on the right side of the screen. Locate **Shape <empty>,** then select the **drop-down arrow** to add a shape.

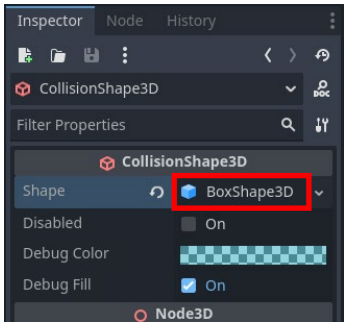
Try out different shapes to see how they look.

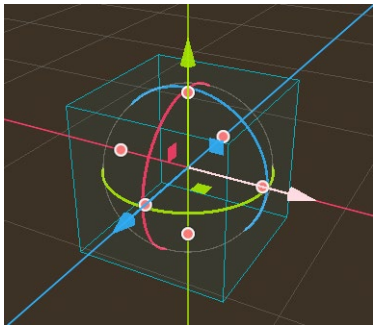
For this project, the player is going to be a **cube.** Select a 3D shape that best resembles this.



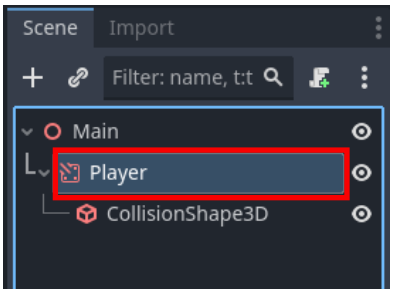
15 Select **New BoxShape3D**. In the game window in the center of the screen, there should now be a **blue cube**. This is the collision's shape, not the **Player's** shape.

Now, an object won't be able to go through this space, it will collide with the Player instead.

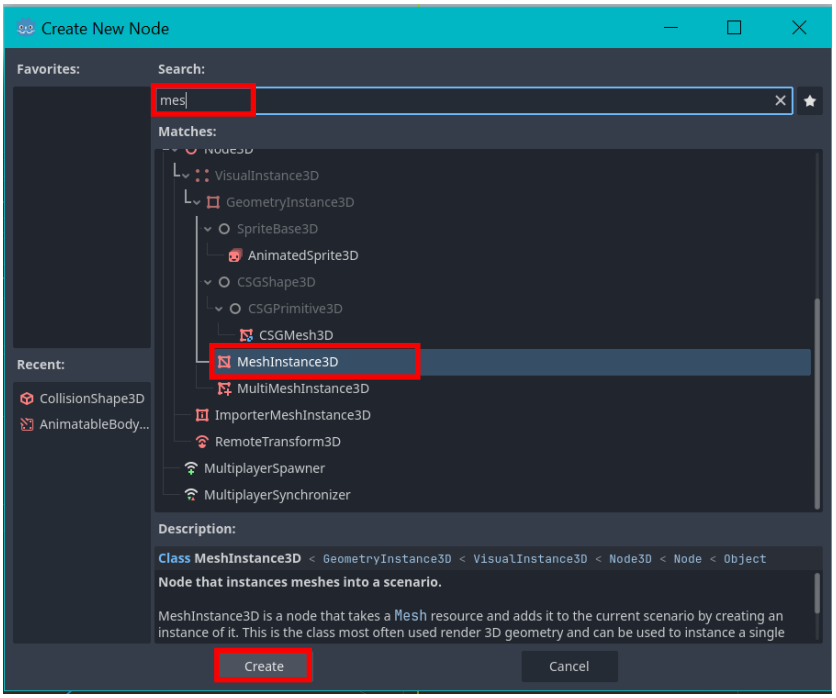




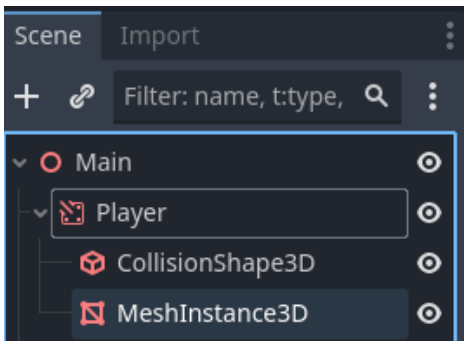
16 Give the Player a shape. In **Scene**, select **Player**. Press **CTRL** and **A** at the same time to add a new node.



Search for **MeshInstance3D**. Select the node, then click **create**.



Make sure the **MeshInstance3D** node is a child node of the **Player** by checking the hierarchy shown in the image below.

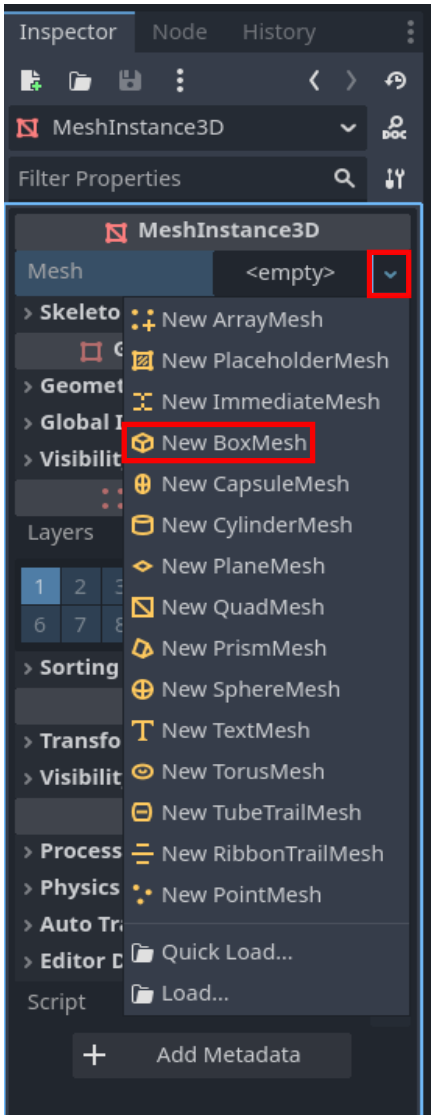


17

Select the shape of the mesh for the **MeshInstance3D** node.

In **Inspector,** locate **Mesh <empty>** then select the **drop-down arrow** to add a shape.

Select **different** meshes, then notice how they change appearance. After, select a mesh that **matches** the cube shape of the collision.



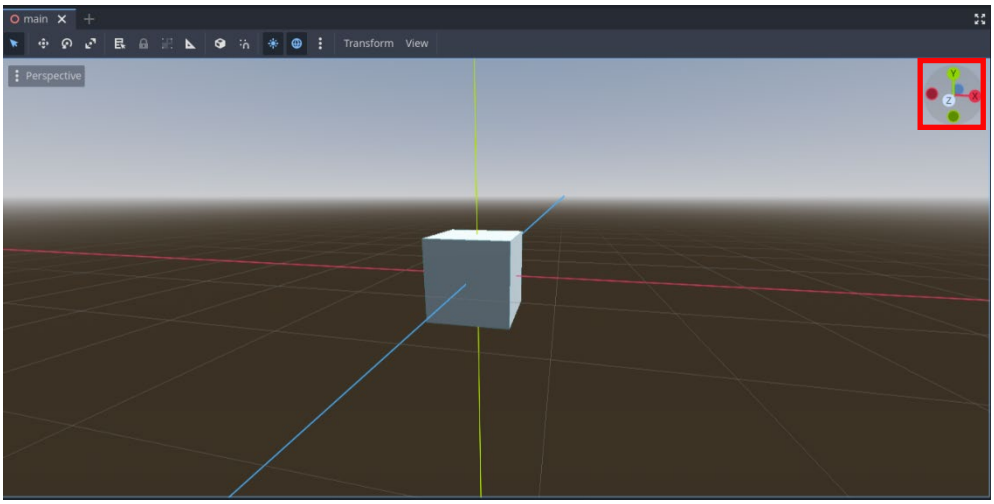
Pause for **Sensei Stop #2!**

Check in with a Code Sensei before moving on.
Make sure the Player's nodes are set up correctly.

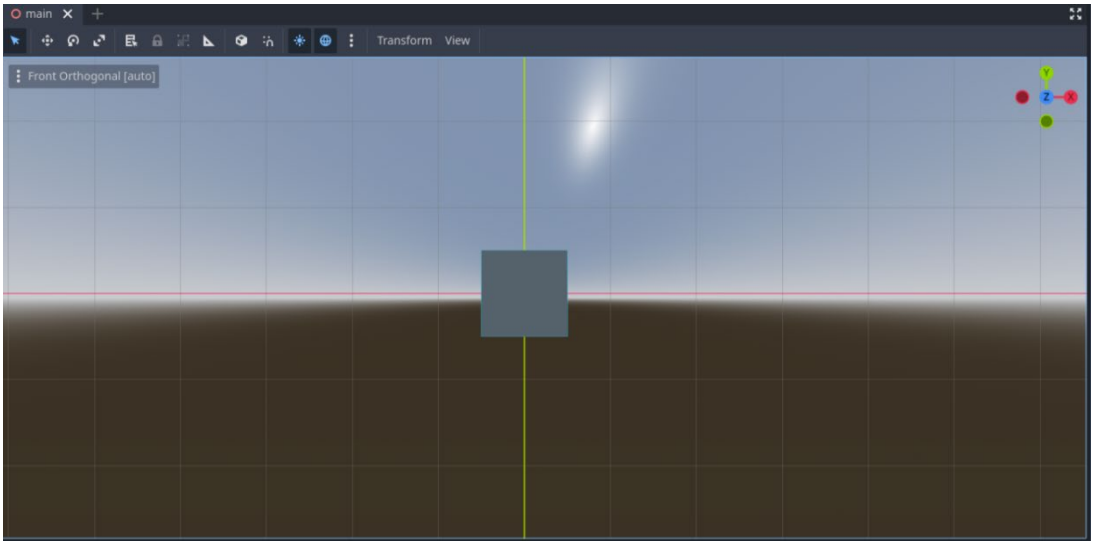
Reminder: Press **CTRL + S** to save your work!

18 Even though this is a 3D project, the game will only use movement in **2 Dimensions** (up, down, left, right). Update the view so that it is strictly **2D**.

In the top right corner of the game window, click on the blue **Z** to change the view to **front orthogonal**. Now the 3-Dimensional scene can be viewed from the front, which makes it appear 2-Dimensional.



Notice the Player appears to change from a cube to a square.



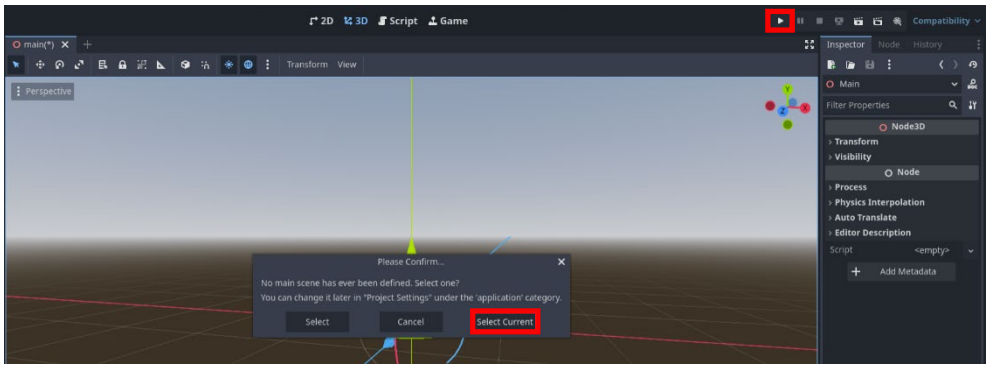
Another way to do this is to select the **Perspective** tab in the top left corner of the game window and click on **Front View**.

Pro Tip:

After reopening the game, the perspective may need to be changed back to **Front View.**

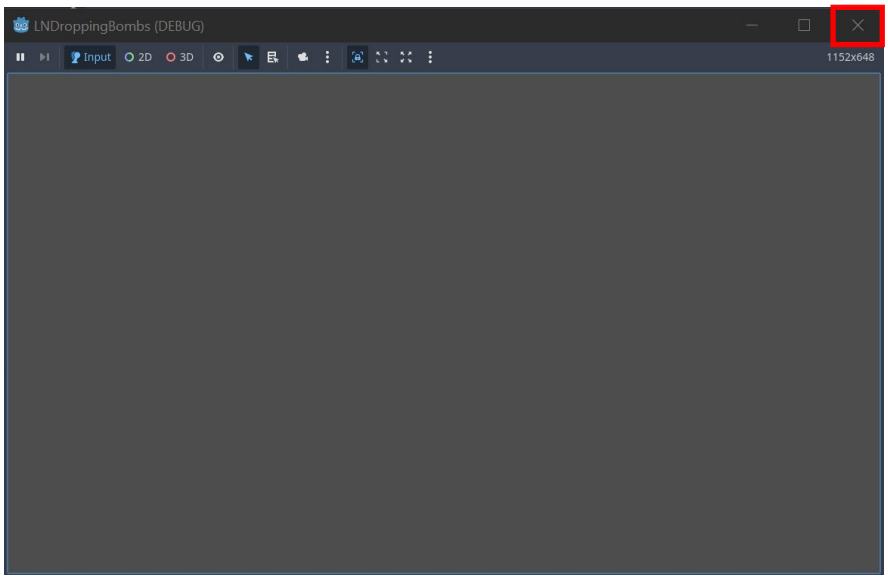
19 In the top right corner, click the **play** button to playtest the project.

A pop-up will appear to select the scene where the playtest will take place. Click **Select Current**.



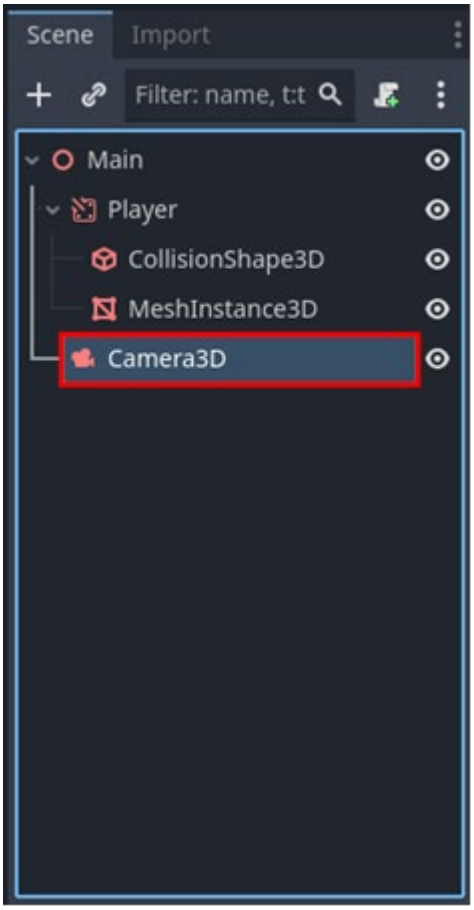
A playtest window will then appear. Notice that it is just a grey screen right now; that is because a camera hasn't been added to the scene yet!

Click on the **X** button in the top right corner of the playtest window to return to the editor.

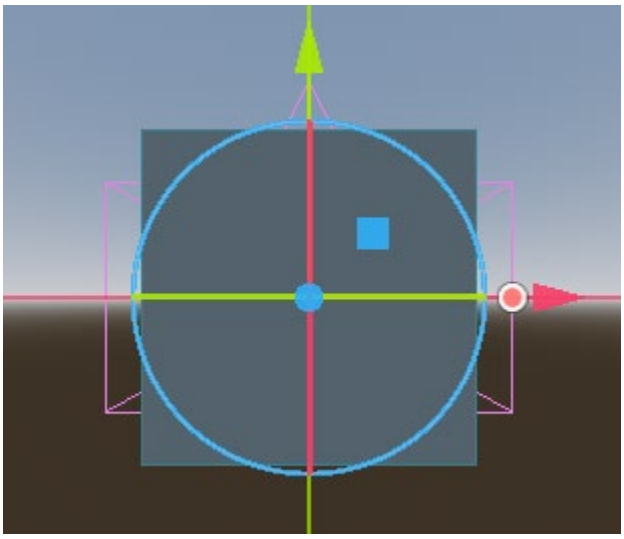


20 In **Scene**, add a **Camera3D** child node to the **Main root**.

Make sure the hierarchy of nodes match the image and that **Camera3D** node is a **child** node to **Main**.

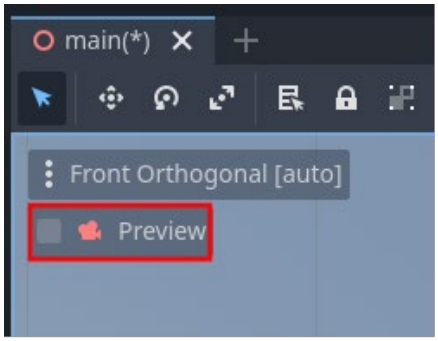


21 In the game window, notice that **pink lines** have been added. This is the outline of what the camera is capturing.

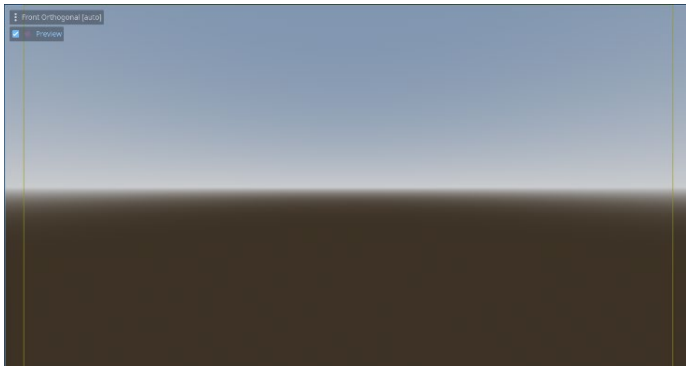


To get a glimpse at what the camera sees, make sure the **Camera3D** node is selected. This will add a **Preview** option to the game window.

In the top left corner of the game window, click **Preview**.



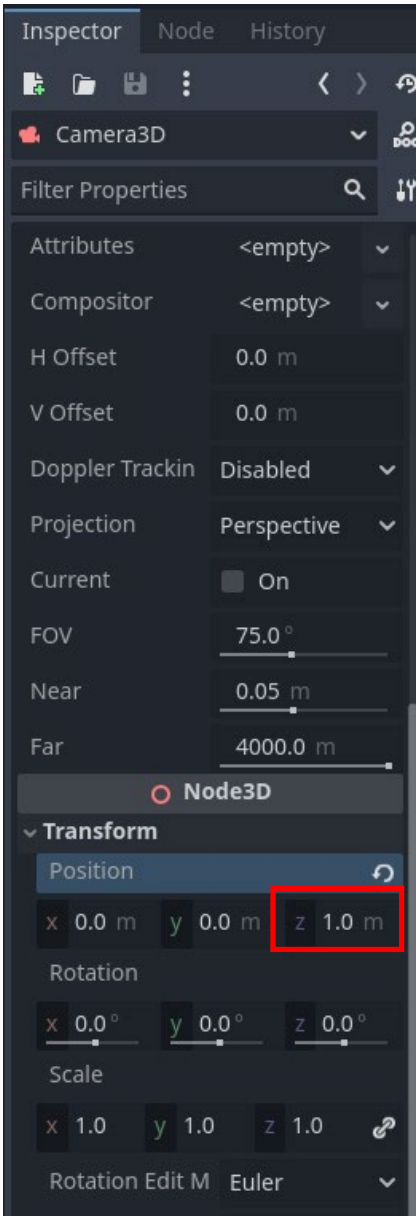
Still nothing can be seen yet. Without changing anything, what might be done to fix this?



22 With the **Camera3D** still selected, check the **Inspector** and scroll down to click on **Transform**. This will show position, rotation, and scale for the camera.

Under **Position**, what value might be changed so that the camera is *behind* the Player?

Change the **z** position to **1.0**.



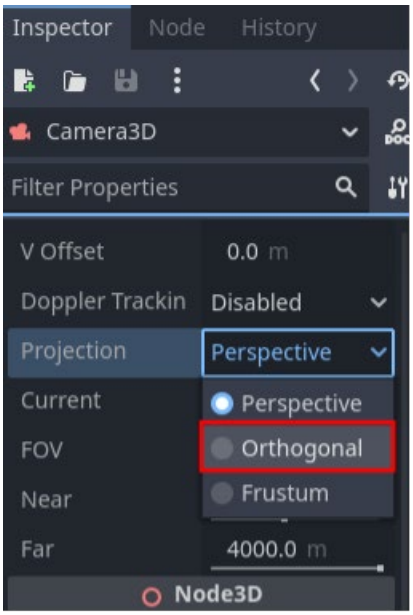
By increasing the **z** position, the camera is now placed *behind* the Player. Now, it can capture the Player's image like a camera in real life!

Don't worry – in the next steps, the camera will be adjusted so that the Player does not take up the entire screen.

23

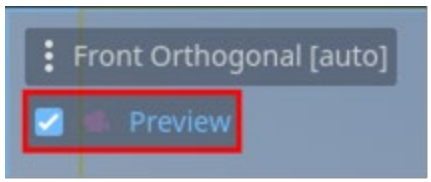
Change the **Projection** and **size** of the camera to view the Player.

Scroll back up to the top of the **Inspector** then change the **Projection** of the camera to **Orthogonal**. Since the view of the game is **orthogonal**, the projection of the camera must match this.



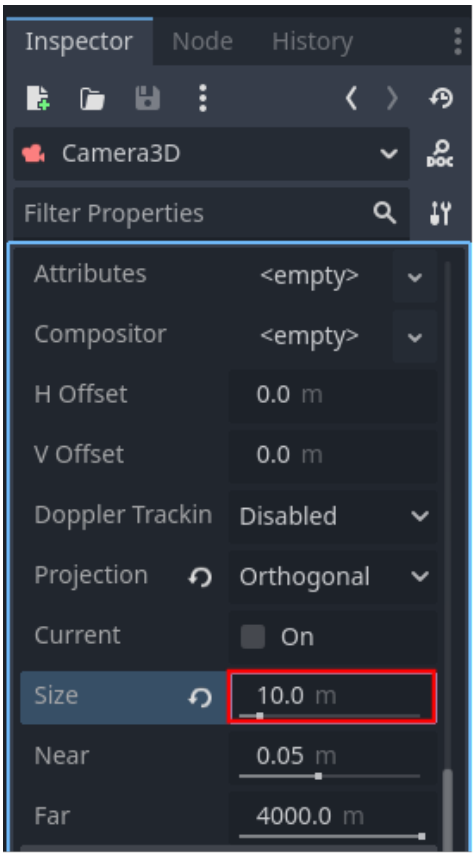
24

In **Scene**, make sure the **Camera3D** node is selected, then exit the camera's preview by deselecting **Preview** in the game window.

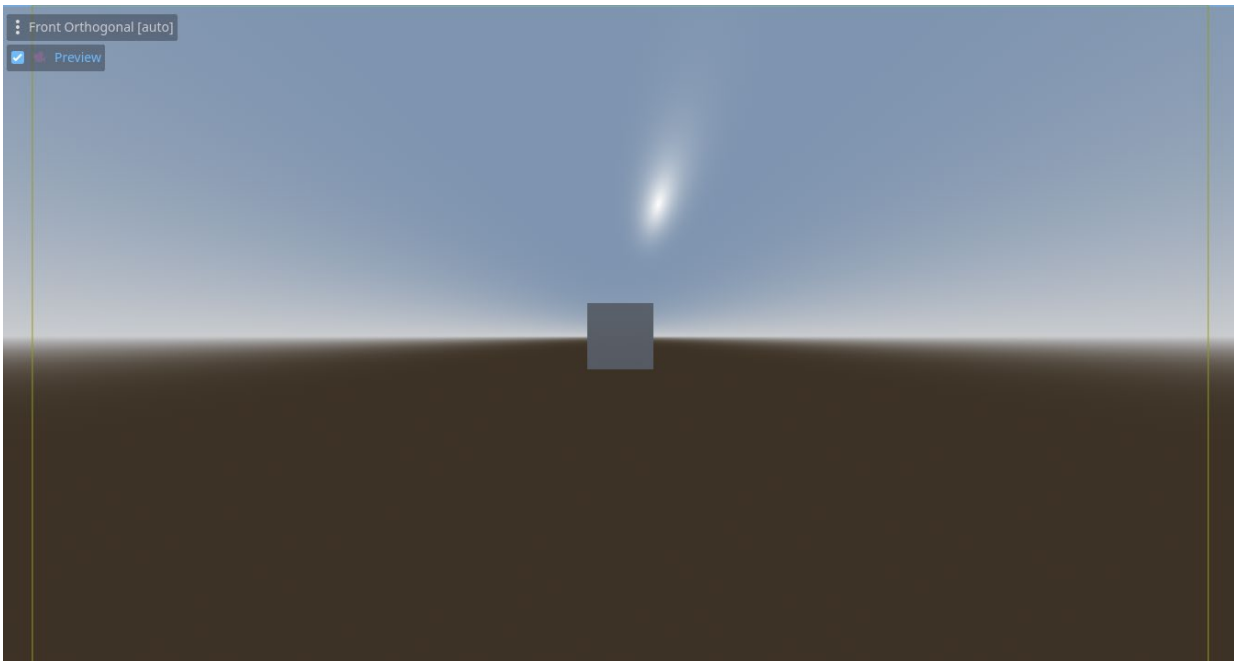


In the **Inspector**, tinker with the **size** of the camera. Notice how the camera's pink outline changes as the camera's size is updated.

Set the size to **10.**



Preview the camera again to see what has changed. The camera preview should be the same as shown.



Pro Tip:

The game window may need to be zoomed out to see the camera's outlines. Remember, this can be done by scrolling down with the mouse.

Pause for **Sensei Stop #3!**

Before continuing, check with a Code Sensei to make sure the **Camera3D** node is set up correctly.

Once reviewed, make sure uncheck **Preview** in the top-left of the viewport.

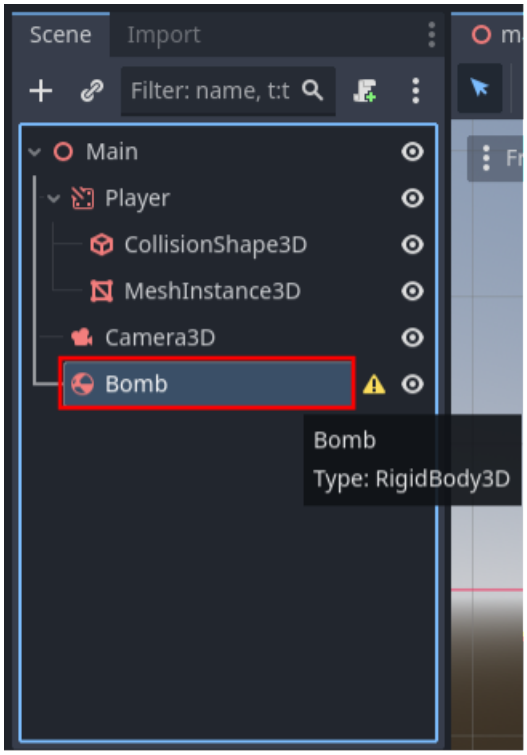
Reminder: Press **CTRL + S** to save your work!

25 Now it's time to add some bombs. The bombs should appear as if they are falling from the sky due to gravity.

To add an object that is affected by gravity, let's use a **RigidBody3D** node. The movement of this node cannot be controlled directly by the user; instead, it is controlled by forces.

Add **RigidBody3D** as a child node to the **Main root** and rename it to **Bomb**.

Notice the hazard symbol next to the Bomb. Add the nodes that can fix this hazard symbol and give them a shape that best resembles a bomb.

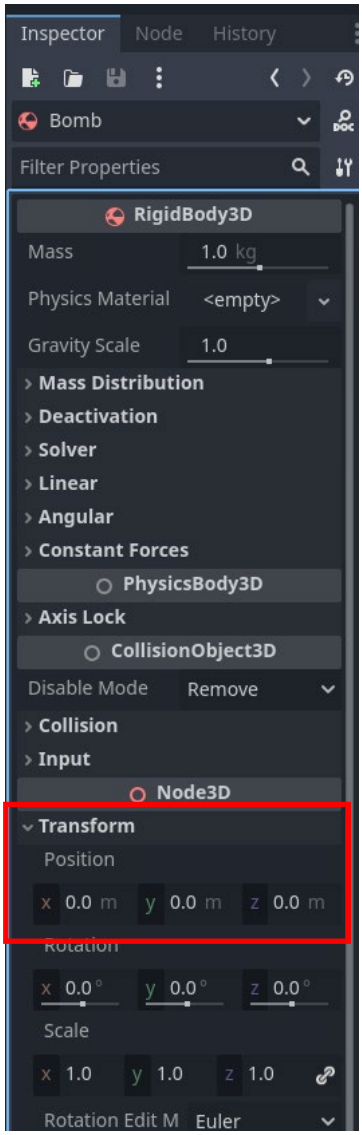


Make sure to add both the **CollisionShape3D** and **MeshInstance3D** nodes as child nodes to the **Bomb.** These nodes can be found in the recently used nodes section when creating a new node. Then, set their **Shape** and **Mesh** to **SphereShape** and **SphereMesh**, respectively.

New Concept: RigidBody3D

A RigidBody3D is used as the base for the Bomb in this project so force and gravity can be applied.

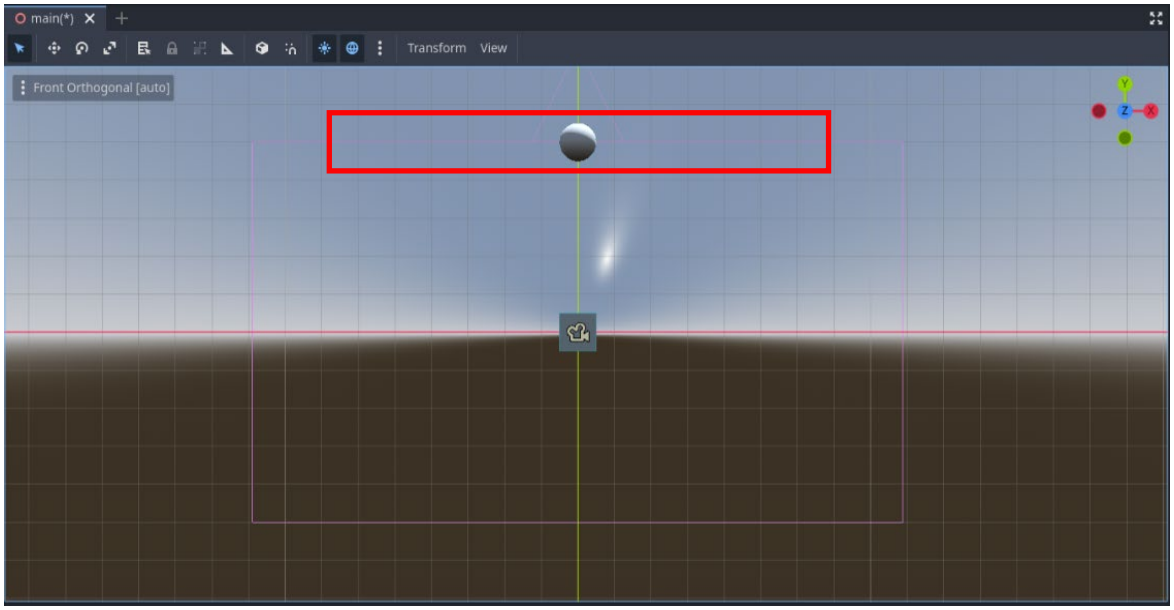
26 Reposition the bomb so that it appears above the player when the game starts. In **Scene**, click on the **Bomb** node, then in the **Inspector**, select **Transform.**



Which **position** value might be changed so that the bomb appears above the Player? Try out different values to place the bomb in line with the camera's pink outline.

Note: In IMPACT, larger **y** values result in a *lower* position on the screen while in Godot, increasing the **y** value will place the object in a higher position on the screen.

Test out the game to see the Bomb fall from the sky, then continue tinkering with its position.



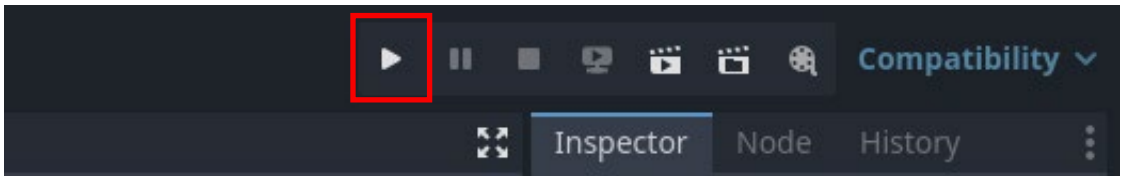
Pro Tip:

Make sure to exit the camera's preview mode. In **Scene**, select **Camera3D** and uncheck **Preview** in the game window.

27 Click the **play** button in the top right corner to playtest the project. What happens?

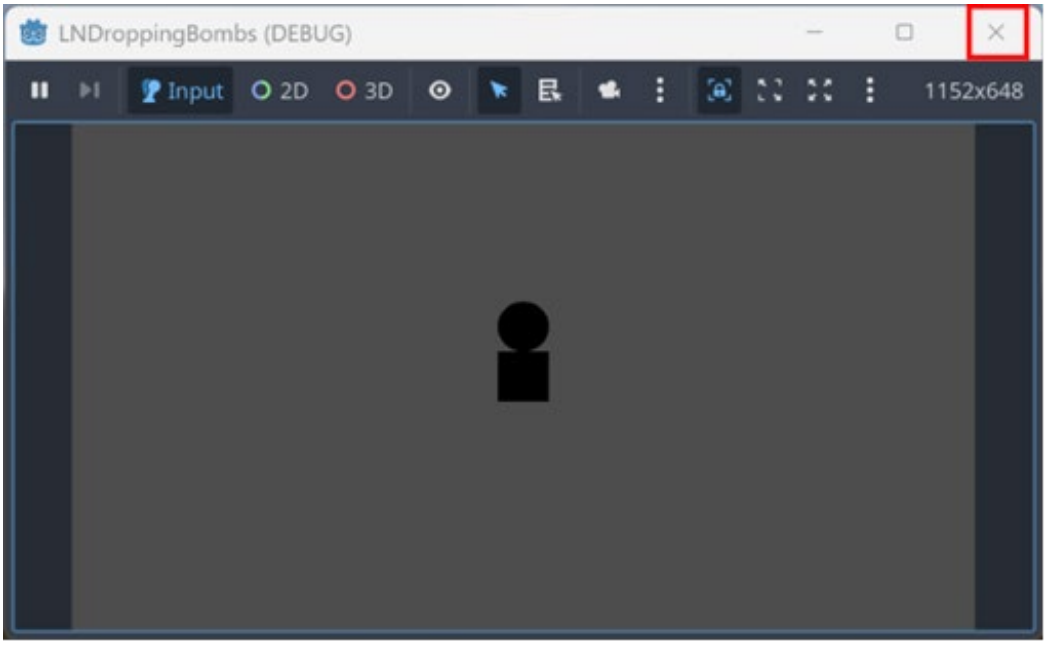
If the Bomb's position is directly above the Player, the Bomb will land on top of the Player and stay there until the Player moves.

If the Bomb's **x** position was changed in **Step 26**, then the Bomb is no longer directly above the Player and the Bomb will fall of the screen.



28 Click the **X** button to stop playtesting the project.

Always exit a playtest before continuing to add to the project.



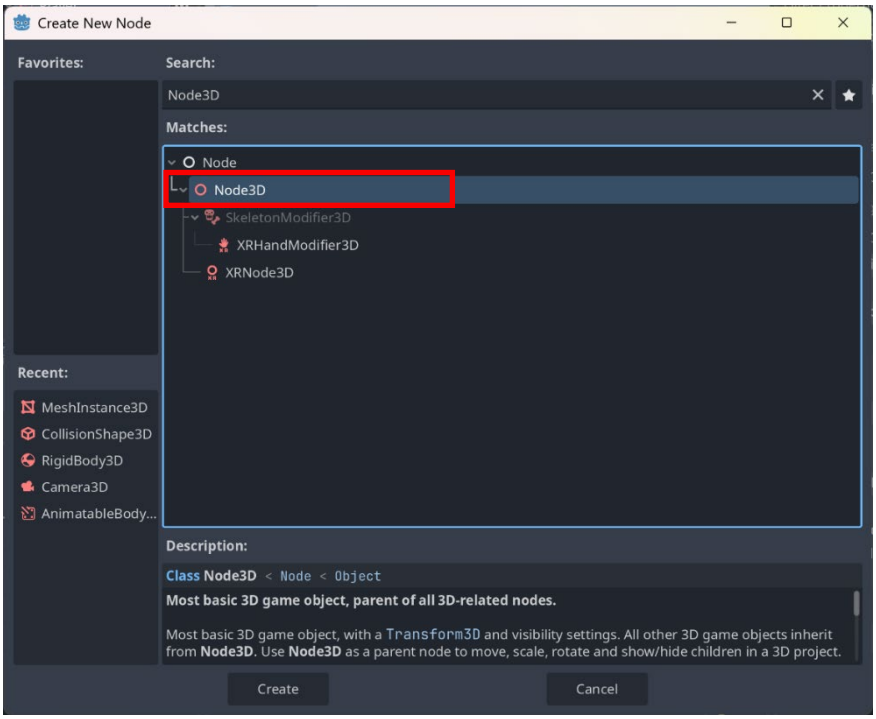
Pause for **Sensei Stop #4!**

Great work so far! After trying to add the Bomb yourself, check in with a Code Sensei to make sure that it is set up correctly.

Reminder: Press **CTRL + S** to save your work!

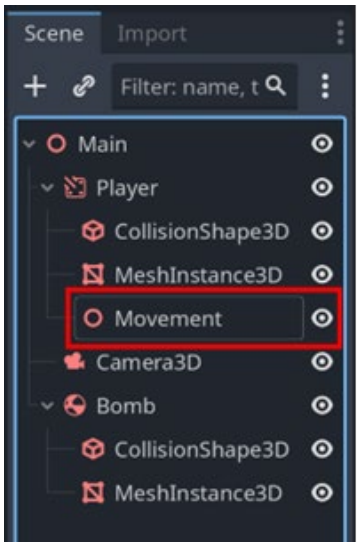
29 Let's add some movement to the game!

Add a child node to the **Player**. Type **Node3D** into the search bar, select the node and click **Create**.



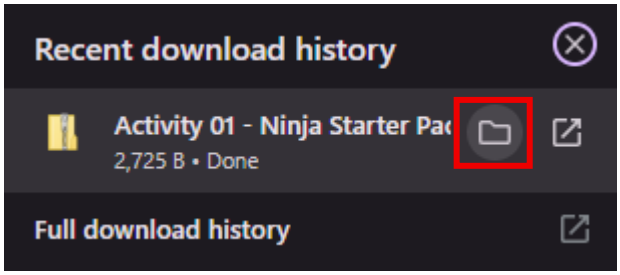
30 Change the name of the new **Node3D** to **Movement.**

Double check that the hierarchy of the nodes matches this image. **Movement** should have been added as a child to the parent node, **Player**.



31 To make the Player move, a **script** needs to be attached. A script is just a file full of code. Although this project has pre-made scripts, there will be opportunities in future projects to create your own scripts.

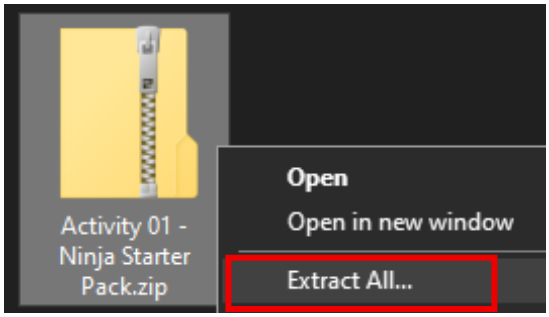
After downloading **BB Activity 01 – Ninja Starter Pack.zip,** a downloads window should pop up at the top right of the screen. Click on the **folder icon** to open the location of this download.



If the **Downloads** window disappears, click **CTRL + J** on the keyboard to bring it back. Then click the folder icon to open its location.

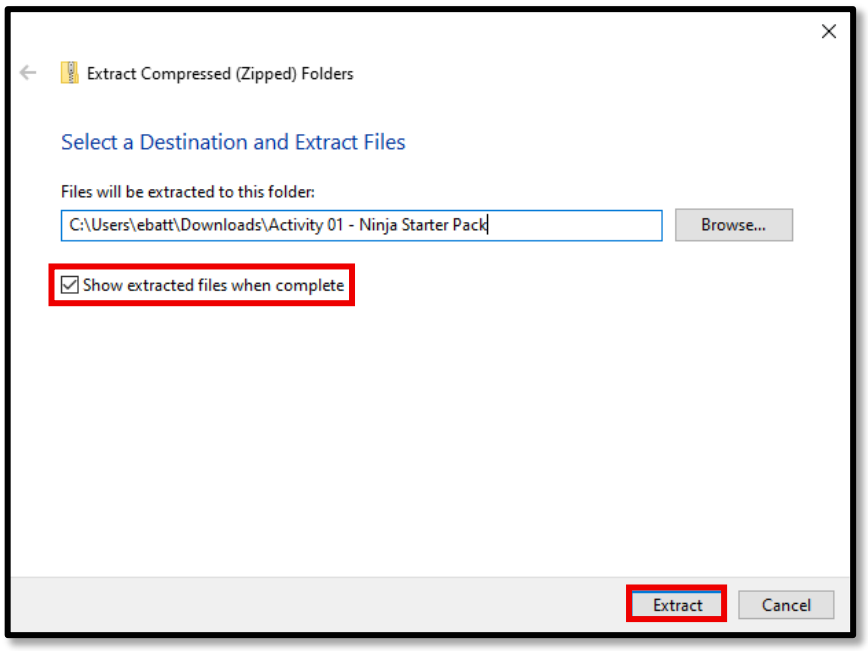
32 Notice the folder is zipped up. Files can't be opened when they are zipped up like this. To open a zipped folder, it must first be unzipped or extracted.

Right-click on the **zip** file and select **Extract or Extract All**.



33 This will create a new window to select a place to put the extracted files.

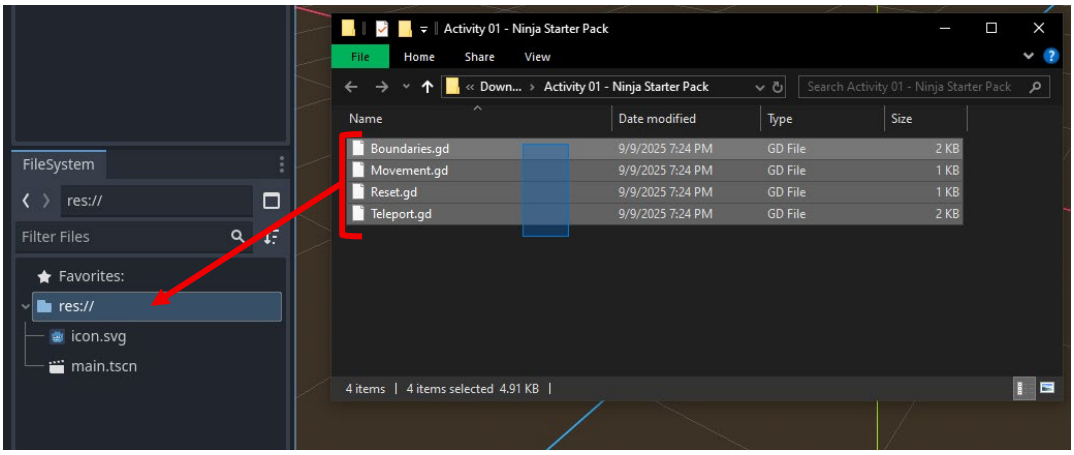
Check the **Show extracted files when complete** then click **Extract.** This will create another **File Explorer** window exactly where the extracted files are, **do not close that window!**



Reminder:

The path will look slightly different from the image shown because all computers have their own username!

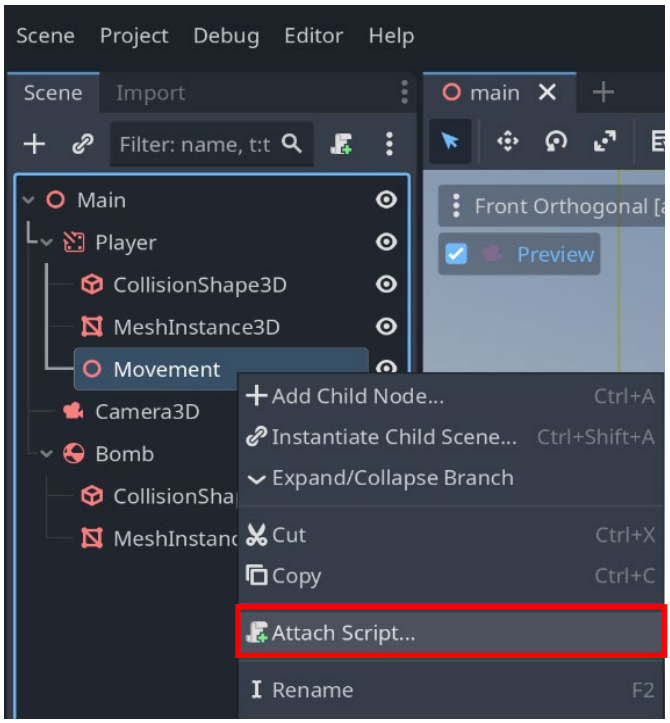
34 Have both Godot and the **File Explorer** with the extracted files open. Select **all** the files and drag them into the **res**:// in **FileSystem** as seen in the image.



Reminder:

File Explorers can be reopened in the browser by pressing **CTRL + J** on the keyboard and clicking the **folder icon**.

35 Time to attach the movement script. **Right click** on the **Movement** node and select **Attach Script.**

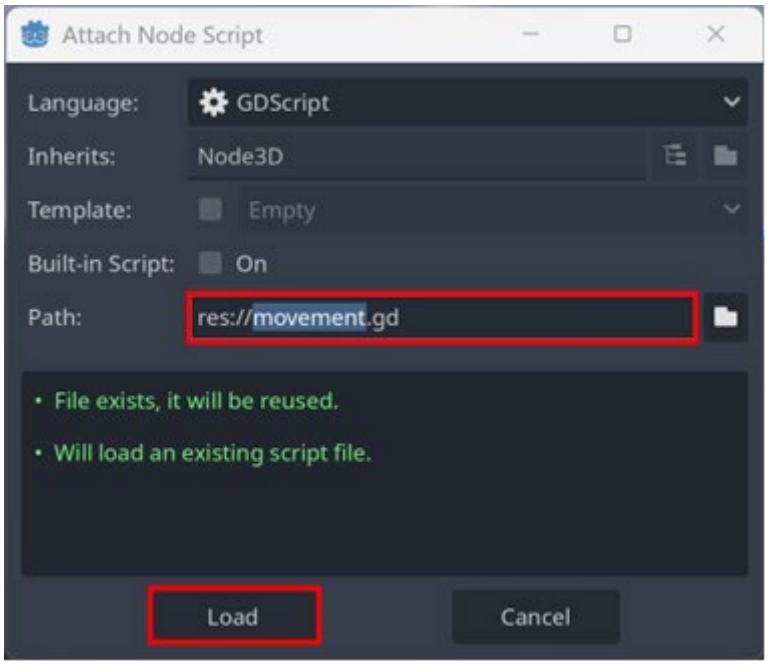


36 Godot automatically matches the name of the script to the node it's being attached to.

The path should show **Movement.gd**. If the path does not show this, click the **folder** icon and select the **Movement.gd** script.

Important Note: Godot should say "*File exists, it will be reused*". If not, refer to **Step 33** about importing files and ask a Code Sensei for help.

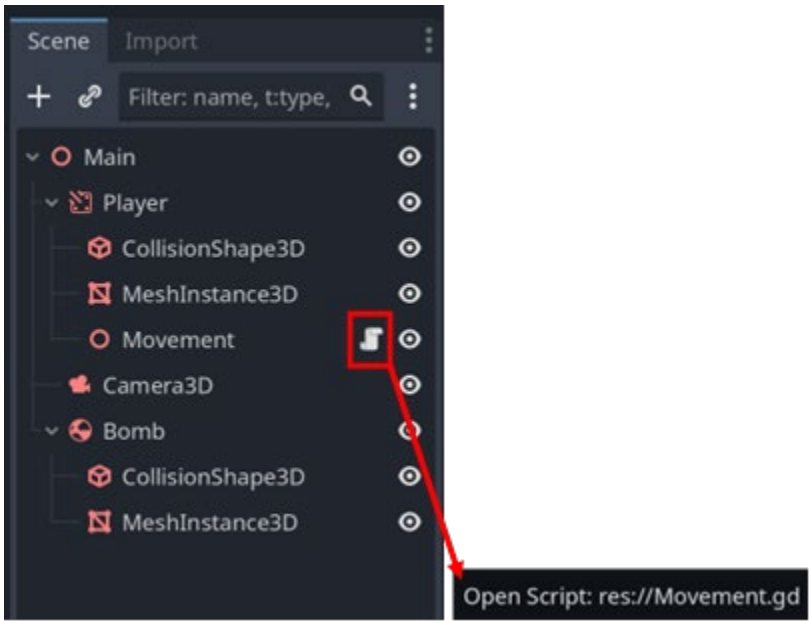
Click **Load** to attach the movement script.



Once the script is attached, the view will change to **Script** and the **Movement** script will open in the editor. **Do not make changes to the script.** Navigate back to the game window by selecting **3D** at the top of the screen.

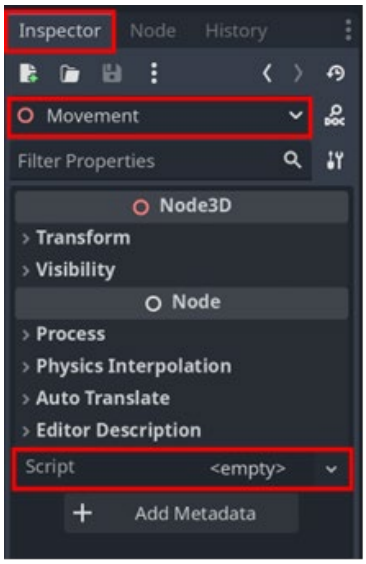
37 A new icon can be seen beside the **Movement** node. This means the node has a script attached.

Hover over the icon with the cursor to see which script is attached.

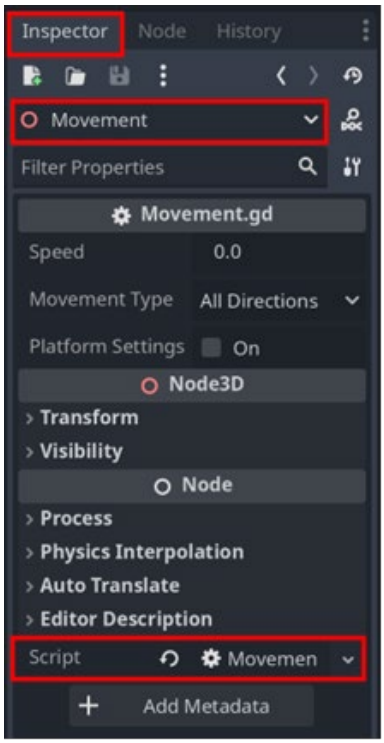


38 The Movement node's **Inspector** also shows which script has been attached. Select the **Movement** node and check the **Inspector** on the right side of the screen.

If the Script property shows <**Empty**>, a script has **not** been attached.



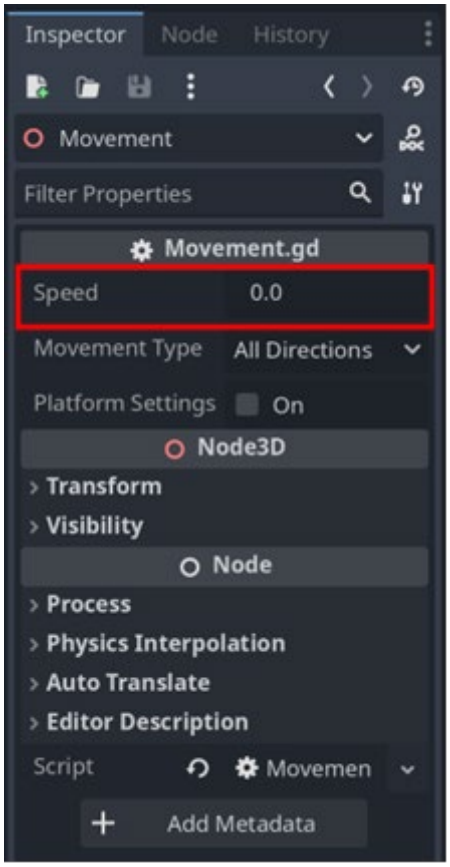
If the Script property shows **Movement**, the Movement script has been successfully attached to the Movement node.

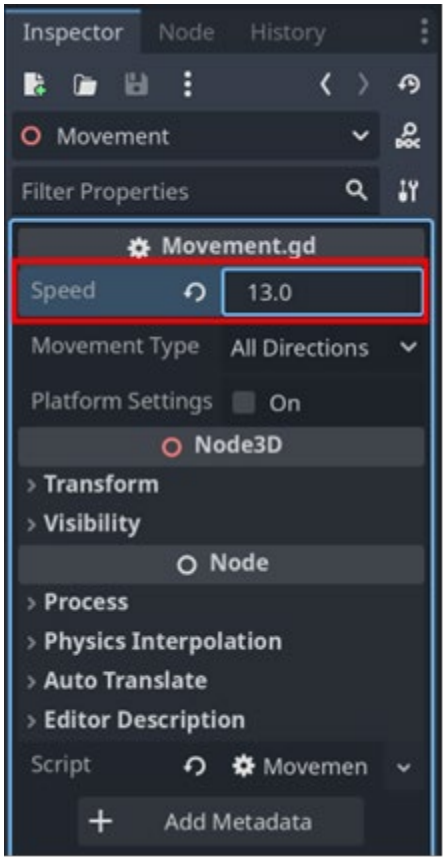


39 Playtest the project and try to move the Player with **direction buttons** or **WASD keys** on the keyboard. What happens?

40 Oh no! The Player is not moving. Double check the movement script in the **Inspector**.

Notice the **Speed variable** is set to 0. Change the value of **Speed** and playtest the project using direction buttons or WASD keys.





Pause for **Sensei Stop #5!**

Before continuing, check with a Code Sensei and make sure the **Movement** script was added correctly.

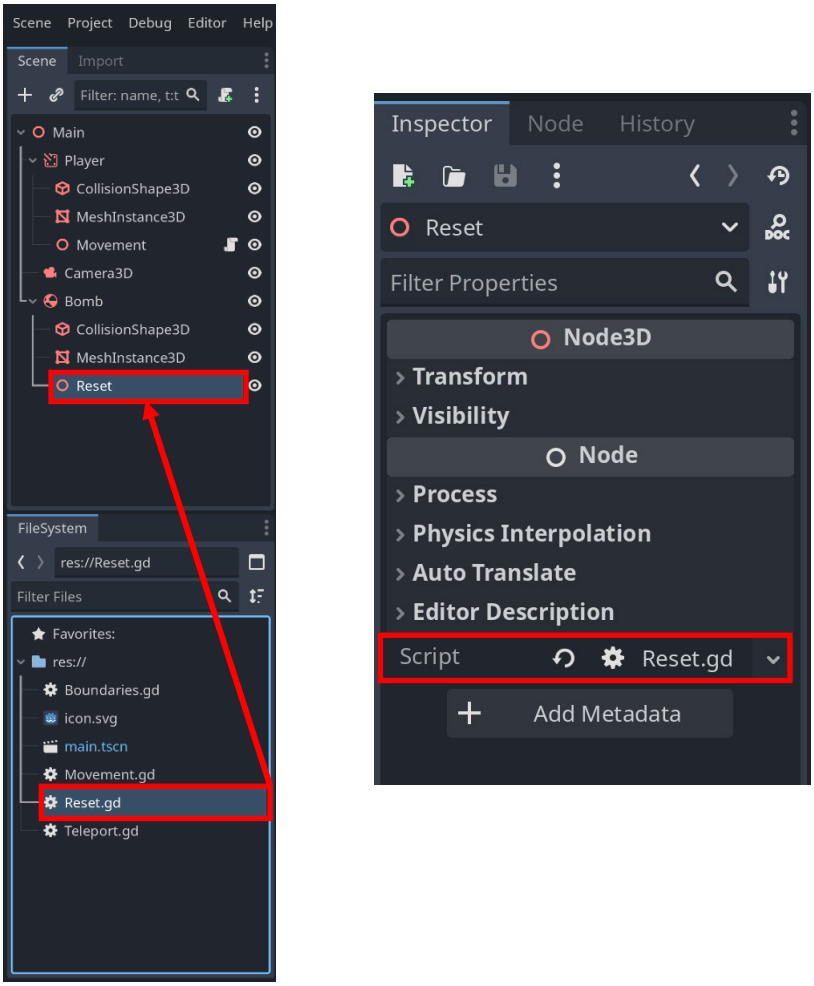
Reminder: Press **CTRL + S** to save your work!

41 When the **Bomb** and **Player** collide, the game does not reset. This can be fixed by adding in another script.

Add a new **Node3D** to the parent node **Bomb** and rename it Reset. Refer back to **Steps 29-30** if needed.

Attach the script by **dragging Reset.gd** to **Reset.** This is another way scripts can be attached to nodes.

Check that the correct script was attached in the Reset node's inspector.



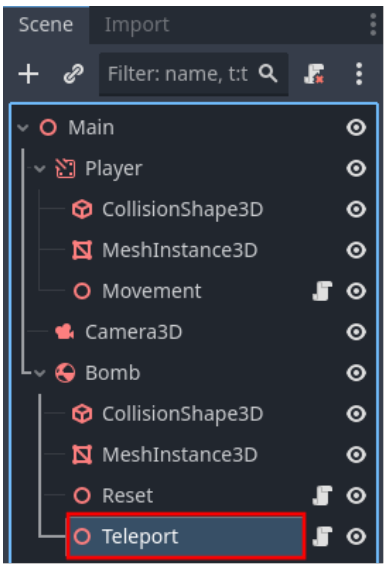
42 Playtest the project again and only move the player *downwards* to see what happens when the **Bomb** hits the **Player**. Then, try to dodge the Bomb. What happens?

43 When the Bomb does not collide with the Player, the Bomb continues to fall off the screen forever. This does not make for a very exciting game!

Let's fix this by adding a new script called **Teleport**.

This script checks if the Bomb exits the view of the camera, then resets Bomb to a new random position at the top of the screen. When the Bomb falls off the screen, it is teleported back to the top of the screen to fall again.

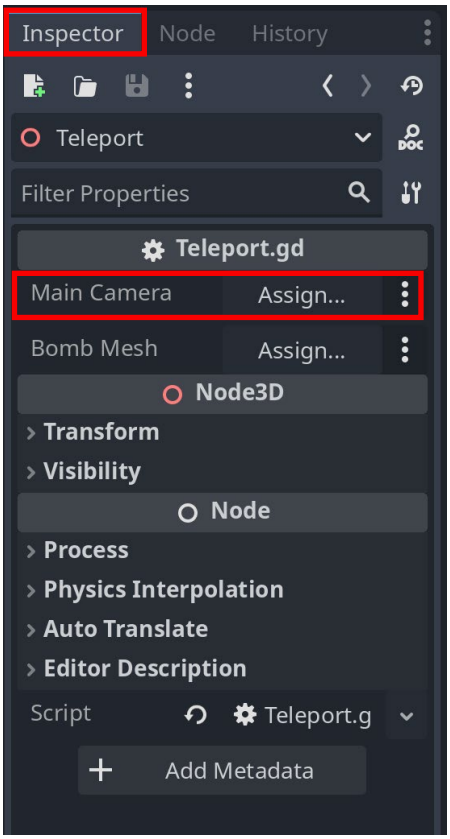
Add a new **Node3D** as a child node to the parent node Bomb, rename it **Teleport**, and attach the **Teleport** script.

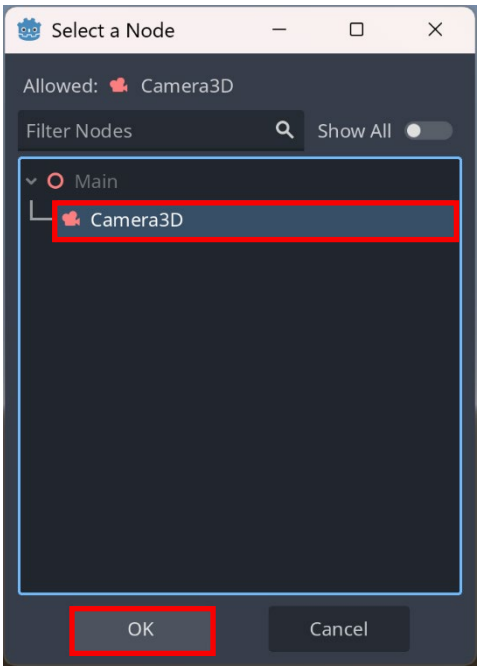


44 Open the **Teleport** node's **Inspector**.

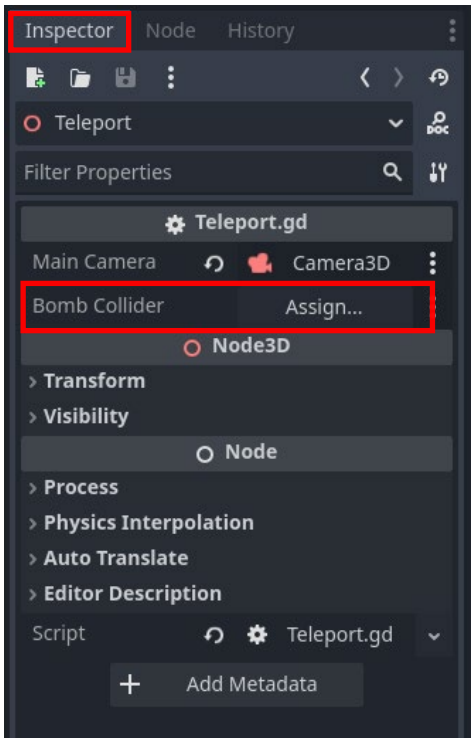
These are two values that need to be assigned: **Main Camera** and **Bomb Collider**.

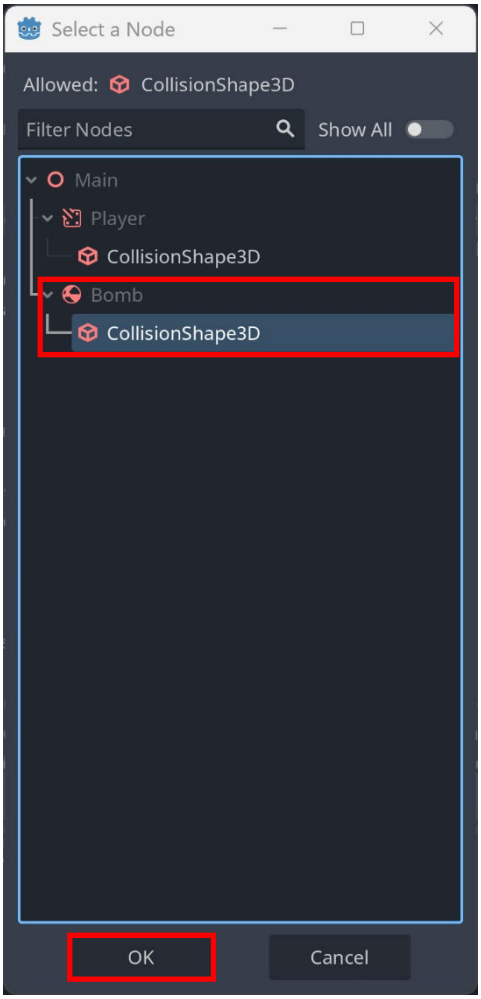
Click on **Assign** beside **Main Camera**, then select the **Camera3D** and click **OK**.





Click on Assign beside Bomb Collider, then select CollisionShape3D under Bomb and click OK.





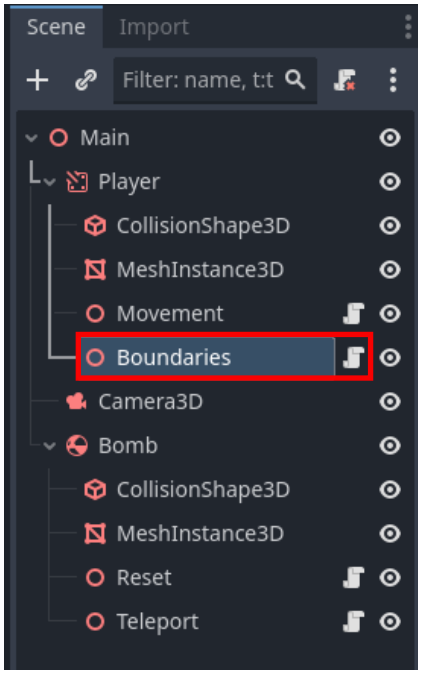
45 Playtest the project.

What happens to the **Bomb** when it exists the camera view?

Can the **Player** exit the camera view?

46 The Player can leave the screen to avoid the Bomb. Prevent this from happening by adding a new script.

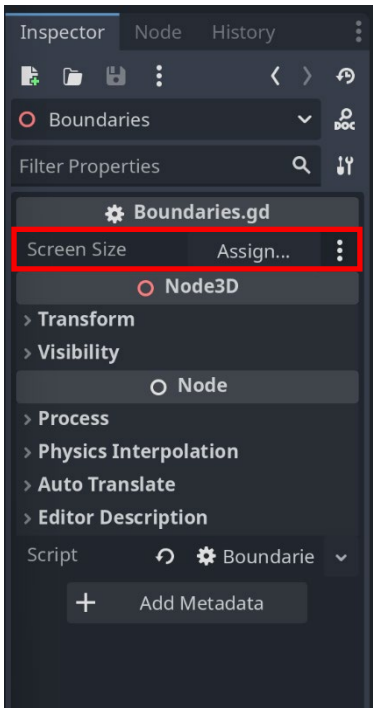
Add a new **Node3D** as a child node to the parent node **Player,** rename it to **Boundaries** and attach the Boundaries script.

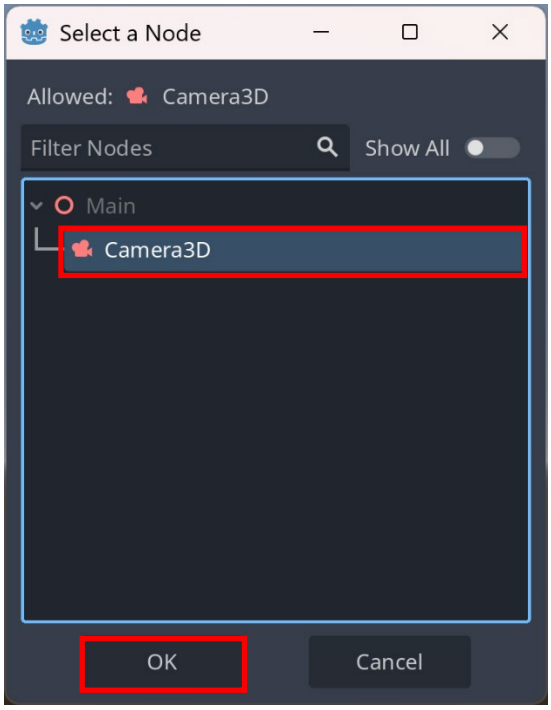


47 The Screen Size needs to be assigned to keep the Player inside the camera view.

Open the **Boundaries** node **Inspector**.

Click **Assign** beside Screen Size. Select **Camera** and click **OK.**





48 Playtest the project. Can the player leave the camera view?

Pause for **Sensei Stop #6!**

Before continuing, check with a Code Sensei to make sure the **reset, teleport, and boundaries** scripts are working correctly.

Reminder: Press **CTRL + S** to save your work!

49

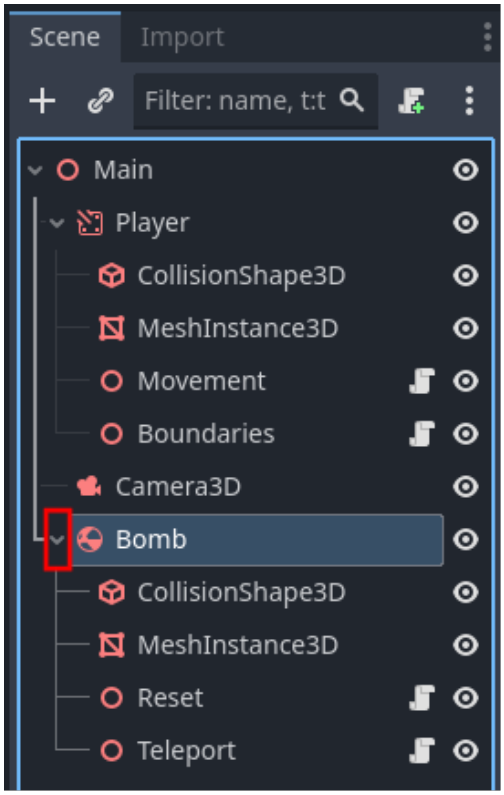
The game isn't very exciting yet. Let's duplicate a node to add more Bombs to the project!

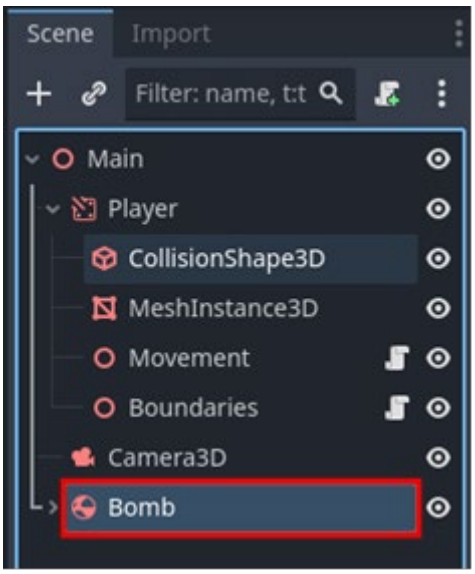
Before doing this, click the **V** arrow on the left side of the **Bomb** to collapse the Bomb node.

This does not remove any of the child nodes from Bomb, it only hides them.

The **>** arrow beside the Bomb can be clicked to extend the Bomb hierarchy and view the child nodes inside.

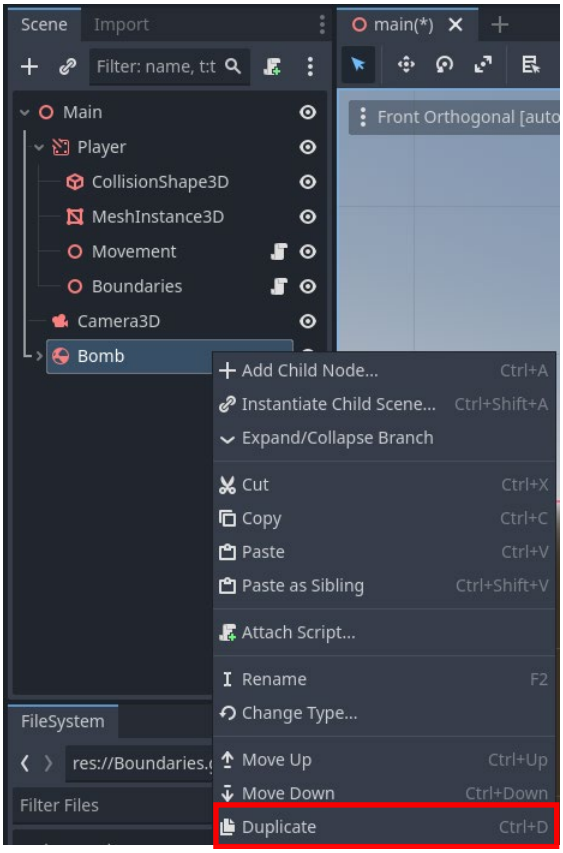
This can be helpful when managing nodes in the Scene menu.

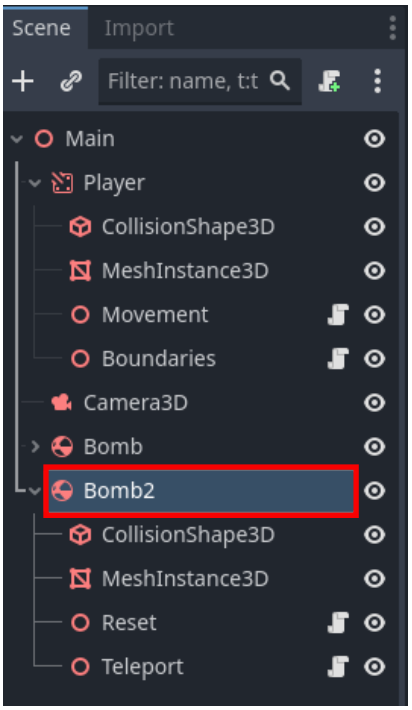




50 **Right click** on **Bomb** and select **Duplicate** to create a copy.

A new node, **Bomb2**, will appear underneath Bomb. This new node is an exact duplicate of the **Bomb** node. **Bomb2** node contains the same child node and scripts.

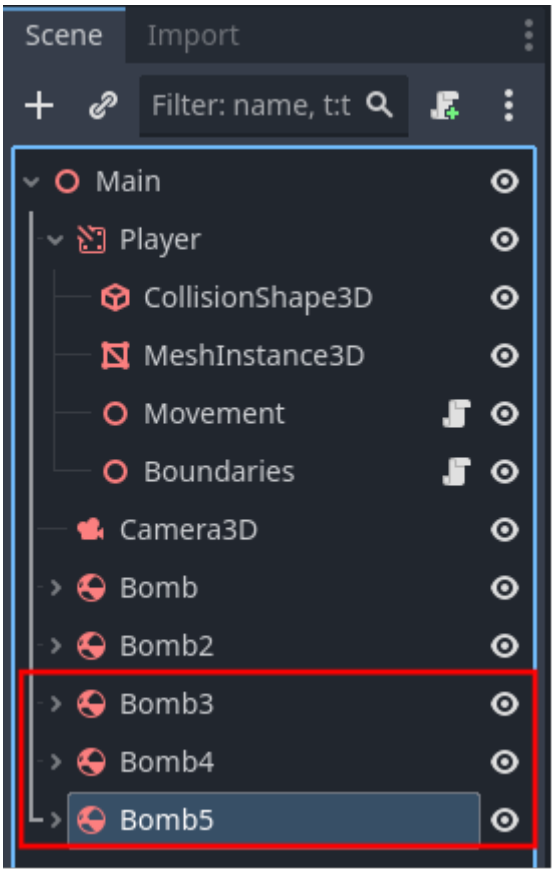




51 Duplicate a few more **Bombs** to the project by pressing **CTRL + D** on the keyboard.

After adding **at least 3** more Bombs, playtest the game and try to avoid the dropping bombs!

Note: If changes are made to **Bomb**, those changes also need to be made to **Bomb2** and any other duplicates of **Bomb**. A simpler way to do this will be covered in future projects.

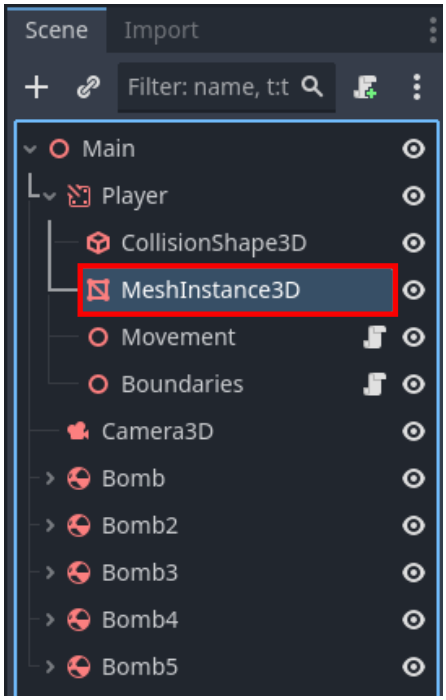


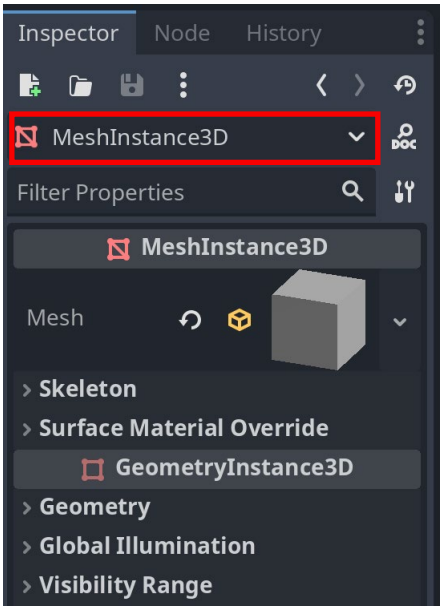
Reminder:

Click the **arrow** next to parent nodes to hide their child nodes.

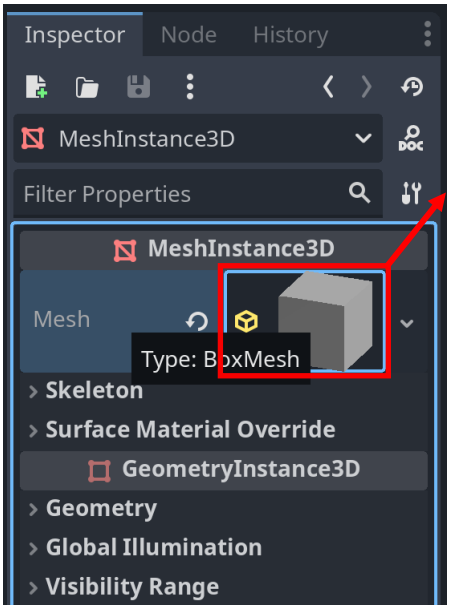
52 Let's add some color to the project!

Open the **Inspector** for the **Player's MeshInstance3D.**

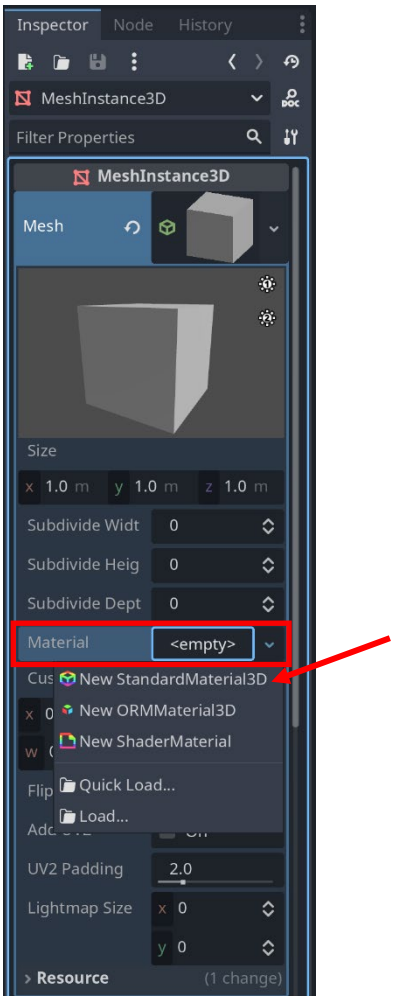




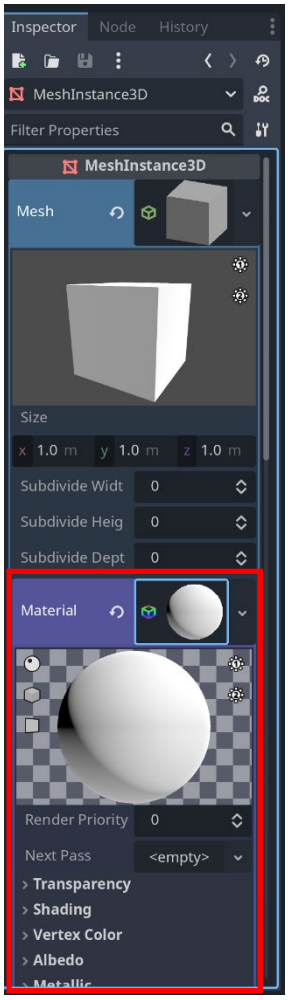
53 Click on the **cube** beside Mesh to open the **Mesh Menu**.



54 Find **Material**, click **<empty>**, and select **New StandardMaterial3D.**

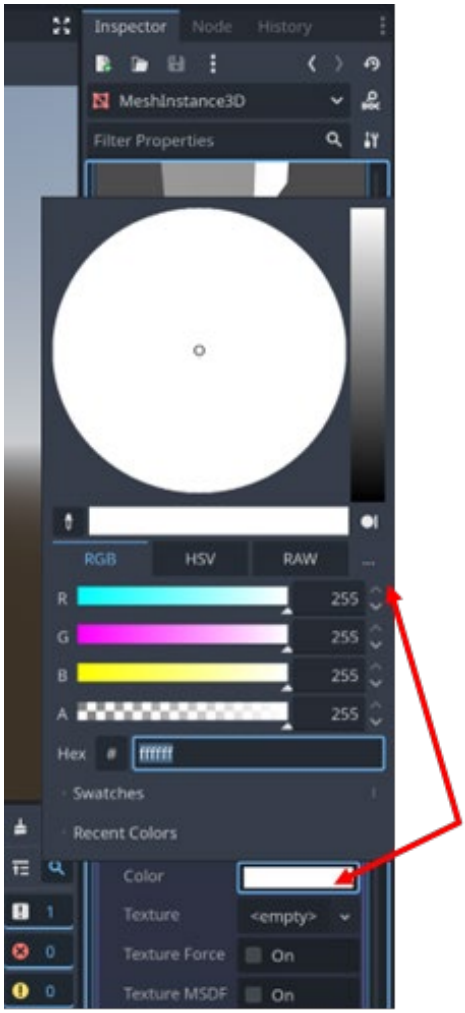


55 The **Material Menu** should open in the inspector once the **New StandardMaterial3D** is added.



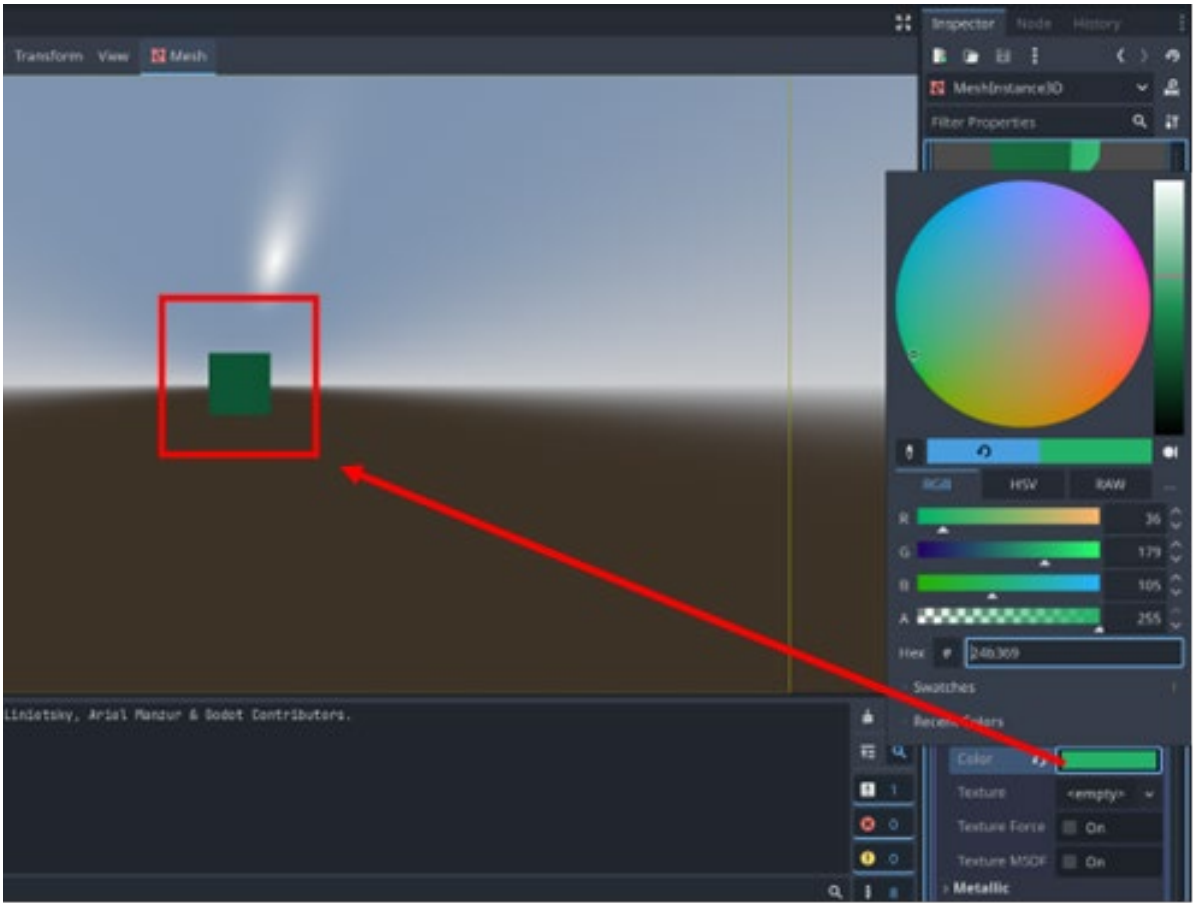
56 Find and open the **Albedo** drop-down menu. Albedo is the base color for the material.

57 Next to **Color**, click to open the Color Picker. This is where the color of the meshes can be changed.



58 Drag the **R G B** arrows and circle value to pick a color for the **Player.** Use the sliders to tinker with different values to see how the colors change.

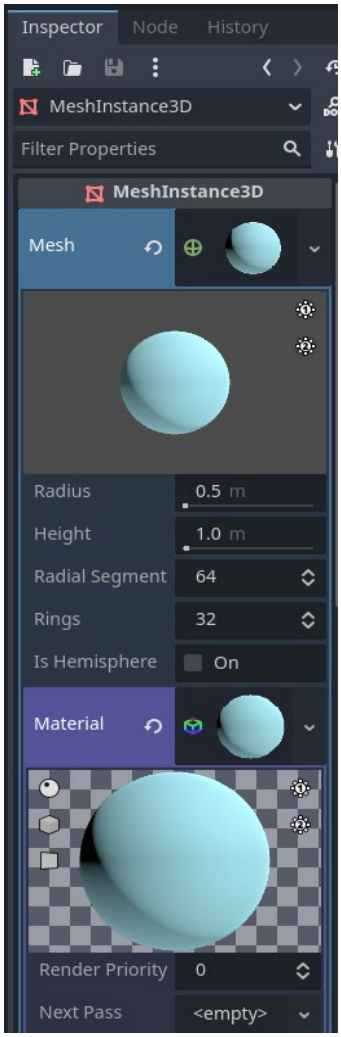
59 The scene will provide a more accurate idea of how the color will look in the game.



60 Change the color of the bombs by selecting the **mesh** of one of them. For this project, it does not matter which bomb mesh is selected.

Refer back to **Steps 52-59** for additional guidance.

Note: Adding a material and changing the color will **affect all Bombs**. This is because the same mesh is on all Bombs in the project. This will be covered in later projects.

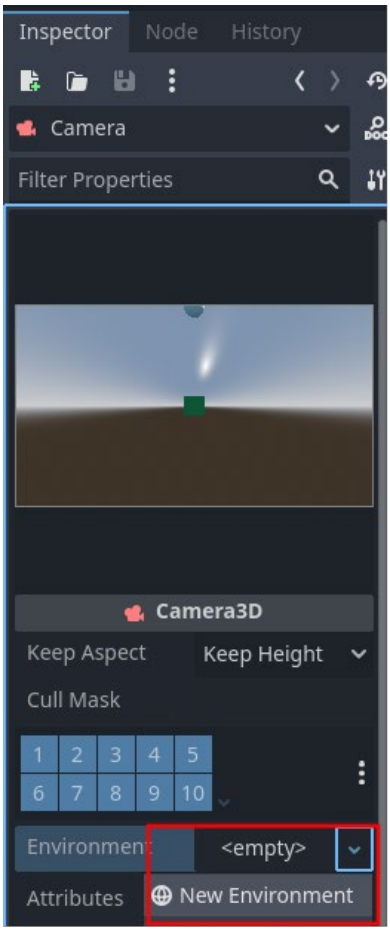


61 Try playtesting the project.

Oh no! The colors can't be seen yet! This is because the **StandardMaterial3D** used for the **meshes** needs **light** to reflect the color, otherwise it will appear completely black. To add light to the project, an **Environment** needs to be added to the camera. More about this will be explained in a later project.

Open the **Inspector** for **Camera3D.** Select the **drop-down arrow** and click on **New Environment**.

Playtest the project again to see the new colors that were added!



Pause for **Sensei Stop #7!**

Before continuing, check with a Code Sensei to make sure the **custom colors** are added correctly.

Reminder: Press **CTRL + S** to save your work!

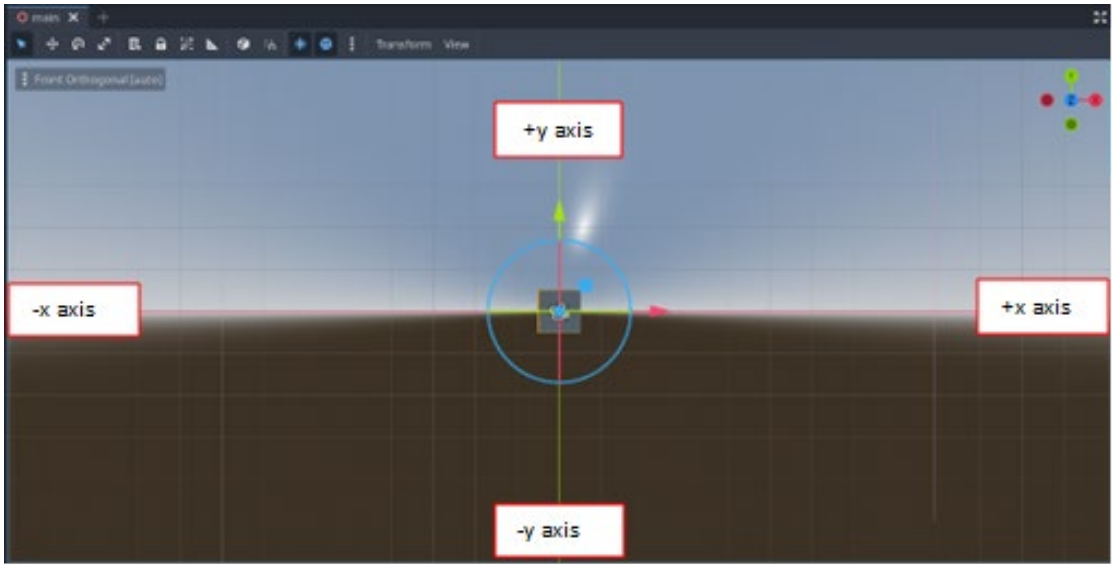
62

Adjust the Player's starting position.

In Inspector, adjust the Player's **x** and **y** position. Make sure **y** is *between* **0** and **-4**, and that **x** is *between* **-8** and **8** to keep the Player inside the Camera View.

Think about how the x and y axis are different in Godot than IMPACT.

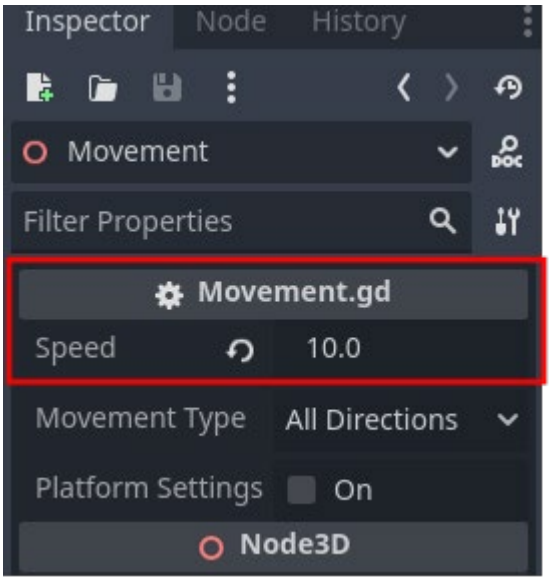
Note: The **z** value must remain 0.



63 Adjust the Player's speed to make the game more or less difficult. This can be any value as long as the game isn't impossible to beat.

Does a smaller speed make it easier or harder to avoid the Bombs? What about a higher speed?

Refer back to **Step 40** for instructions on how to adjust the Player speed if needed.



64 Playtest the final version of your project!

Does everything work as expected? What else might need to be updated?

Pause for **Sensei Stop #8!**

Congrats on creating your first project in Godot! Way to go!

- What is the biggest difference when working in Godot compared to other game editors like IMPACT or Unity?
- What did you enjoy most when creating this project?
- What was something you found difficult? Why?

Reminder: Press **CTRL + S** to save your work and submit!