Download Starting Project

1. Open bit.ly/2I46RDy in a browser to download “crash-course-godot-3.zip”
2. Unzip the file.
3. In your file browser, open the folder within the zip, “crash-course-godot-3”.

Crash Course Godot 3

Making 3D games with open-source software

Goal

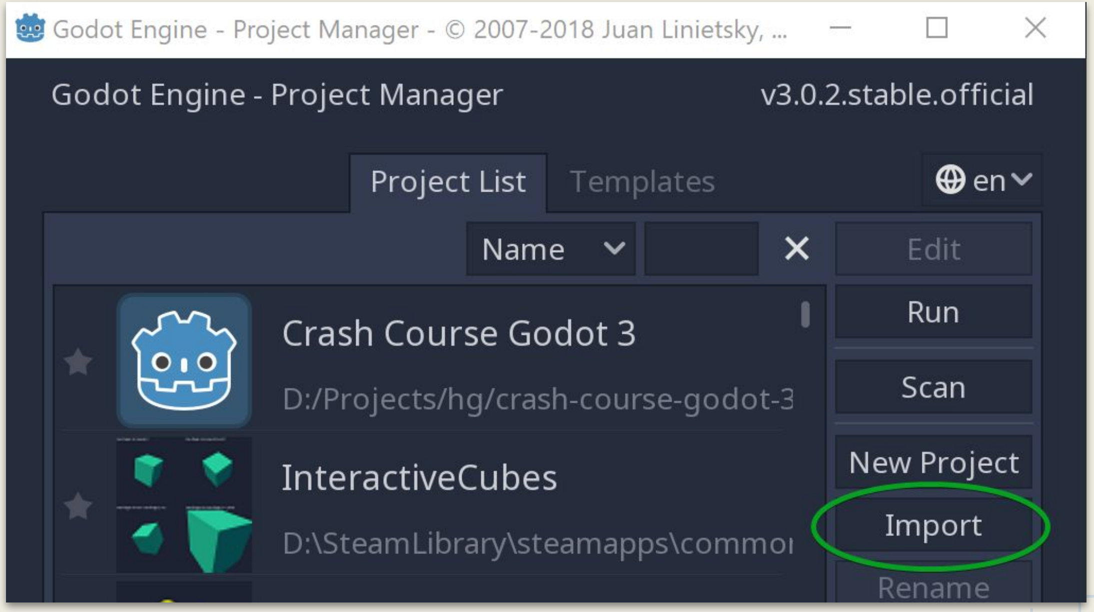
- Get comfortable with Godot game editor
- Create an interactive 3D environment
- Learn lots of 3D development terms
- A brief introduction in visual scripting

Supplementary materials

- docs.godotengine.org/en/3.0/getting_started/step_by_step
    a. Official site providing tutorials on how to use the game engine.
    b. Doubles as an in-depth manual!
- youtu.be/-D-IcbsdT04
    a. Recommended video tutorial

Importing Tutorial Project

Step 1: Click “Import.”



Godot Engine - Project Manager - © 2007-2018 Juan Linietsky, ...

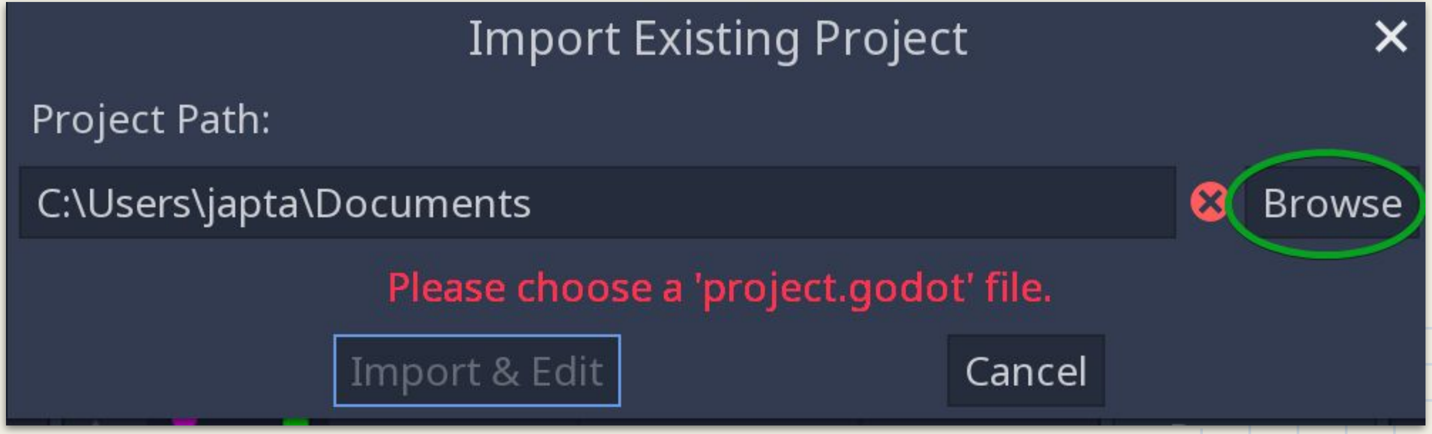
Godot Engine - Project Manager v3.0.2.stable.official

Crash Course Godot 3
D:/Projects/hg/crash-course-godot-3

InteractiveCubes
D:\SteamLibrary\steamapps\commoı

Importing Tutorial Project

Step 2: Click on “Browse.”



Import Existing Project

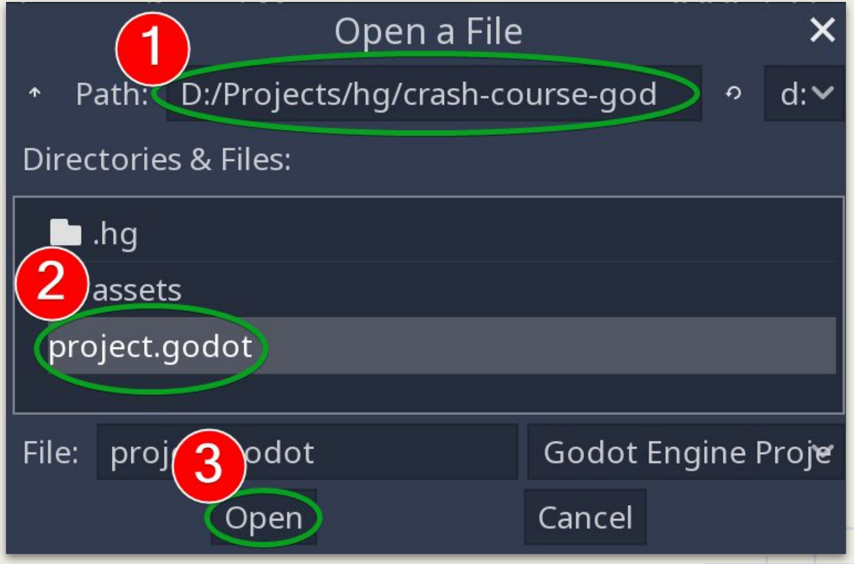
Project Path:

C:\Users\japta\Documents

Please choose a 'project.godot' file.

Importing Tutorial Project

Step 3: Navigate to the “crash-course-godot-3,” and find the file, “project.godot”. Click “open.”



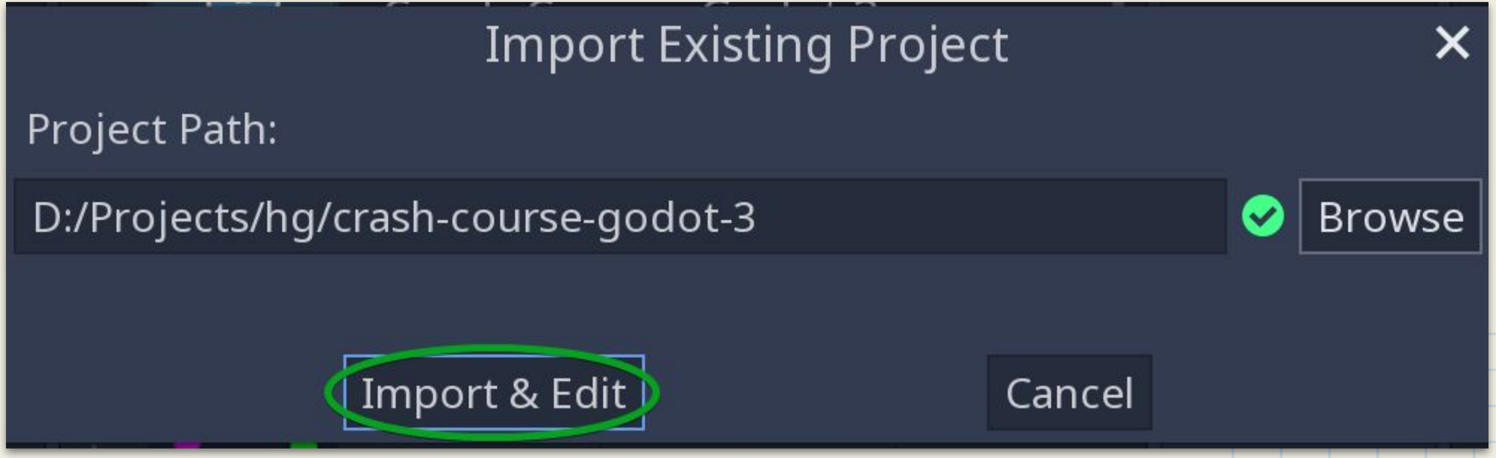
Path: D:/Projects/hg/crash-course-god

File: proj odot

Godot Engine Proje

Importing Tutorial Project

Step 4: Finally, click on “Import & Edit.”



Import Existing Project

Project Path: D:/Projects/hg/crash-course-godot-3

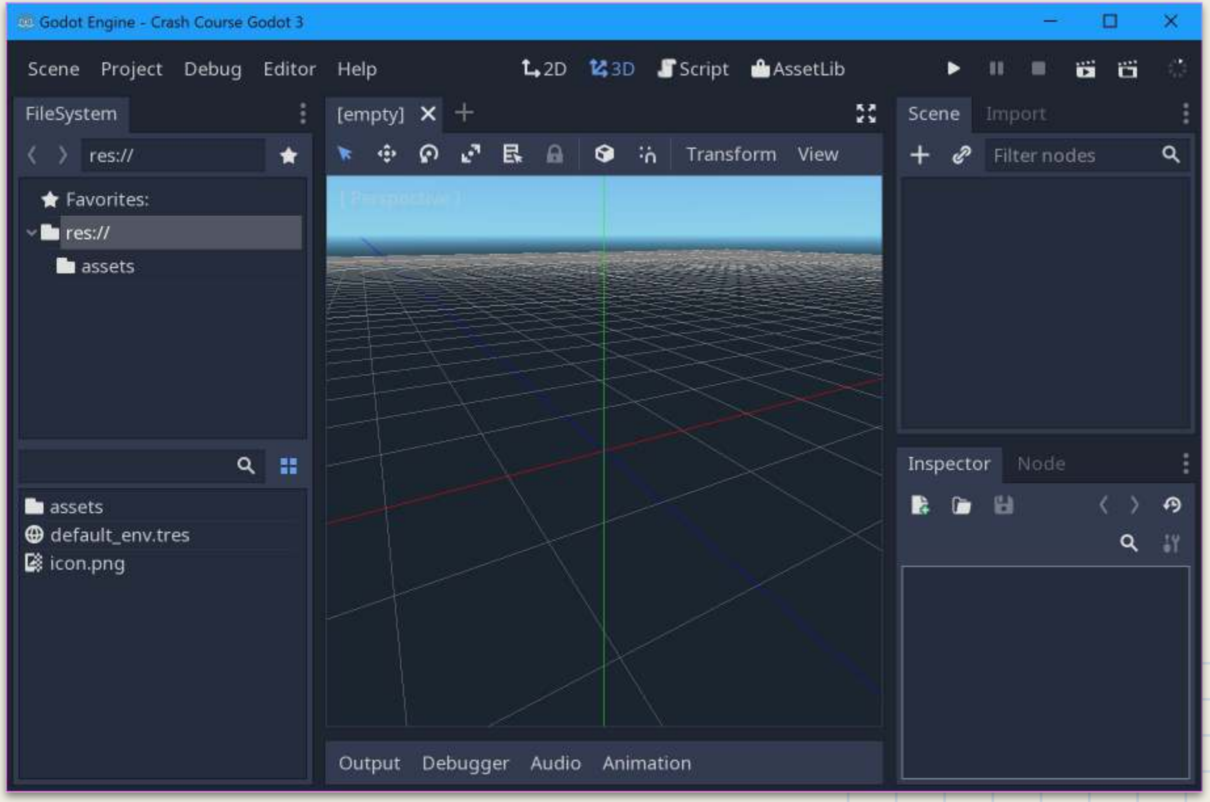
Asset License

Original files obtained from: http://opengameart.org/content/machu-picchu

- **level.obj** is a modified version of MPFull.blend from ctdabomb, released under CC-by-sa 3.0
- **grass.png** is from samuncle, released under CC-by-sa 3.0
- **rock.jpg** is from Marianne Gagnon , release under CC-by-sa 3.0

Link to CC-by-sa 3.0 license: https://creativecommons.org/licenses/by-sa/3.0/legalcode

About Godot 3



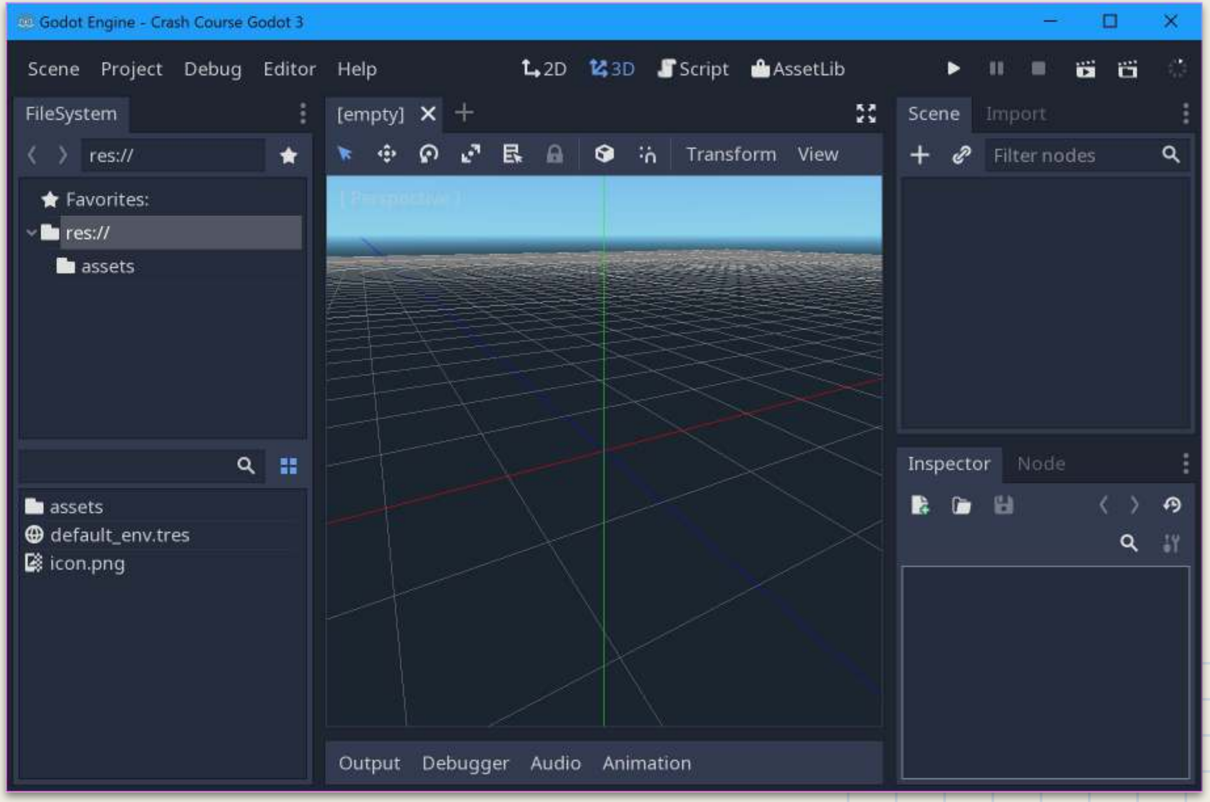
What is Godot 3?

- A What-You-See-Is-What-You-Get (WYSIWYG) 3D & 2D Game Engine
- Many built-in features
  - Physics, Sound, Scripting, Gamepad support, Plugins, and more!
- Builds to many platforms
  - PC, Mac, Linux, HTML5 + WebGL, iOS (iPhone + iPad), Android, Windows 10

Licenses and Fees

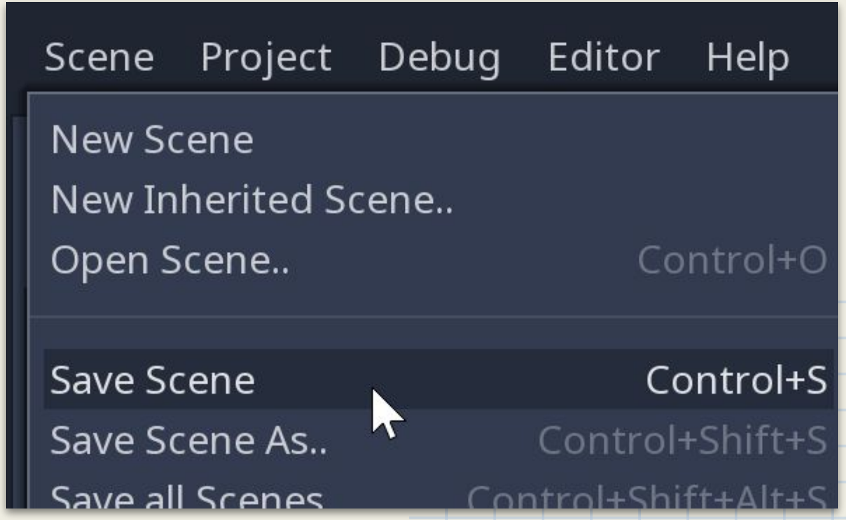
- Free forever!
  - Under the open-source MIT license
    - You can change the engine code if you want to!
  - Build to Windows, Mac, Linux, HTML5 + WebGL, iOS, Android, and Windows 10
  - Has VR and AR support
  - C#, Python-ish GDScript, and visual scripting support
    - Extensions available for actual Python, D, etc.
  - Totally OK to sell your game! There's no royalty fees.

Making a game



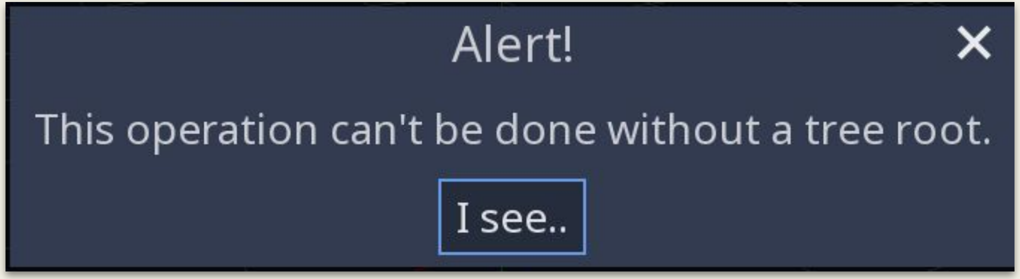
Setting Up the Project

- Godot has a base assumption all scenes must meet.
- Try selecting “Scene -> Save Scene”



Setting Up the Project

- An error indicating that the scene must have exactly one tree root node.
  - What does that mean?
- Close pop-up by clicking, “I see…”

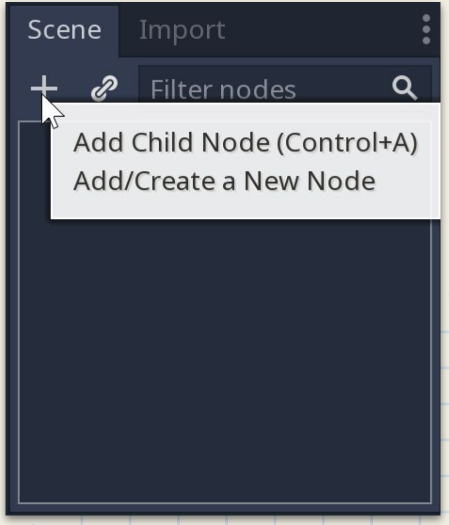


Alert!

This operation can't be done without a tree root.

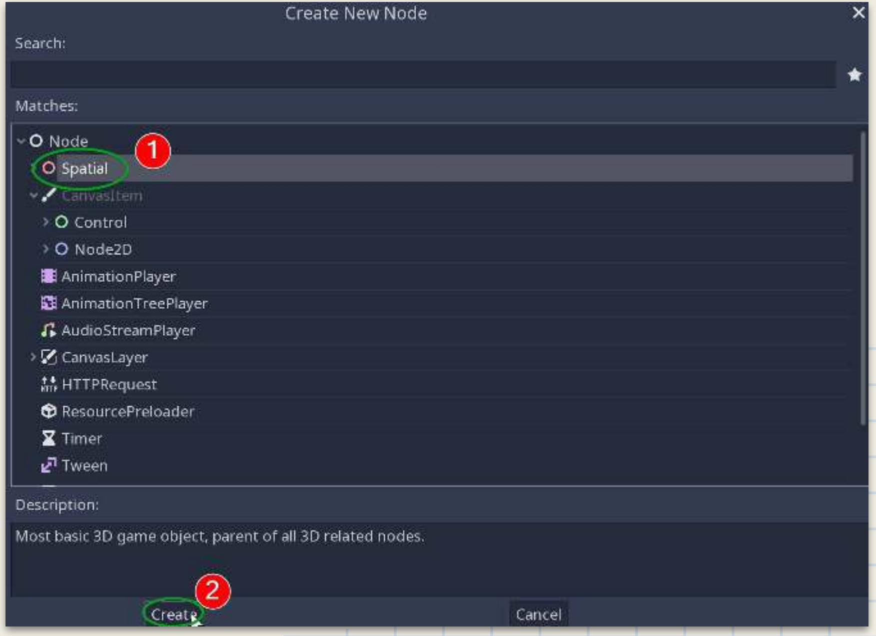
Setting Up the Project

- On the Scene Tree dock in the upper-right hand corner, click on the plus sign.



Setting Up the Project

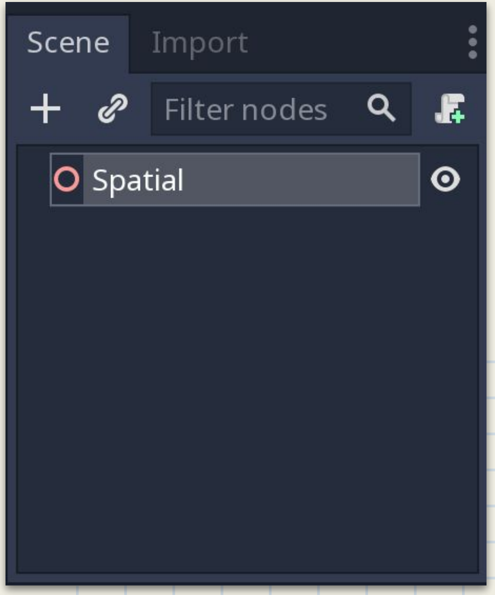
- Select “Spatial” on the pop-up dialog, and click, “Create.”
  - This creates a new root node!
  - Spatial is a node with… spatial information.



Most basic 3D game object, parent of all 3D related nodes.

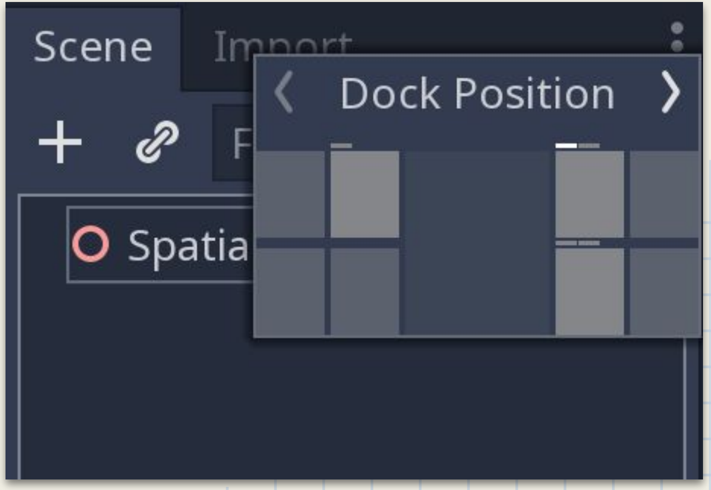
Scene Tree Dock

- Displays the content of a scene in a tree hierarchy.
- Every scene must have *exactly* one root node.
- You can change the order of the objects by dragging them up and down.
- Dragging objects into another turns that object into a child (I'll go over this later).



As an Aside

- Clicking the 3-dot button on the upper-right hand corner of every dock let's one change it's position.

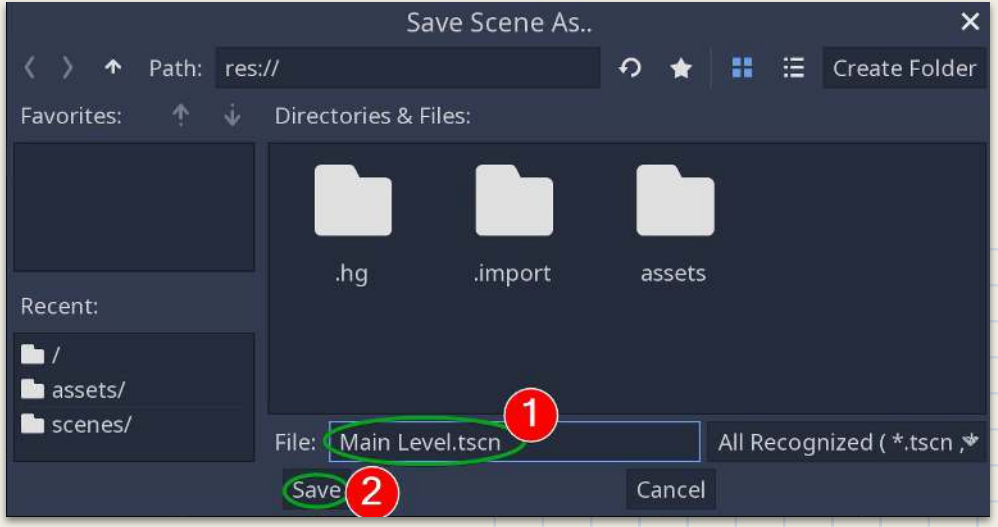


Quick lexicon review

- Node
  - Godot’s term for any individual object.
  - Each node specializes in a certain functionality.
  - Every entry in the Hierarchy pane is a node.
- Scene
  - Godot’s term for files storing a collection of nodes.
  - Store references to assets in the project.
  - Has a *.tscn file extension.
- Assets
  - Industry-wide term for any files used in the game.
  - For Godot, that’s anything inside the project folder.

Setting Up the Project

- Select “Scene -> Save Scene”
- Enter a file name. This lesson uses “Main Level.tscn”
    - Name must end in .tscn
- Click “Save.”



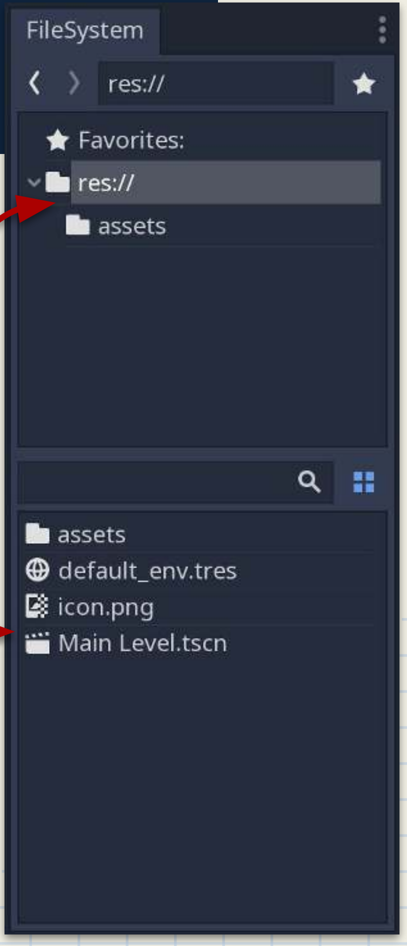
Save Scene As..

Path: res://

File: Main Level.tscn

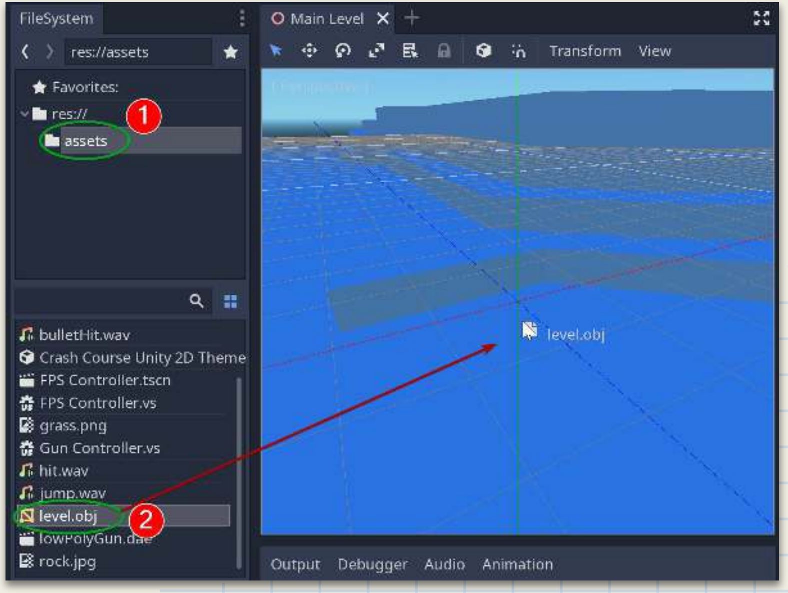
File System Dock

- Displays the contents of the project.
  - Automatically syncs with the folder if there's any changes
- Top-half displays the folders.
  - "res://" is the project folder itself.
- Bottom-half displays the content of the selected folder.
  - Has a search bar to make it easier to find assets



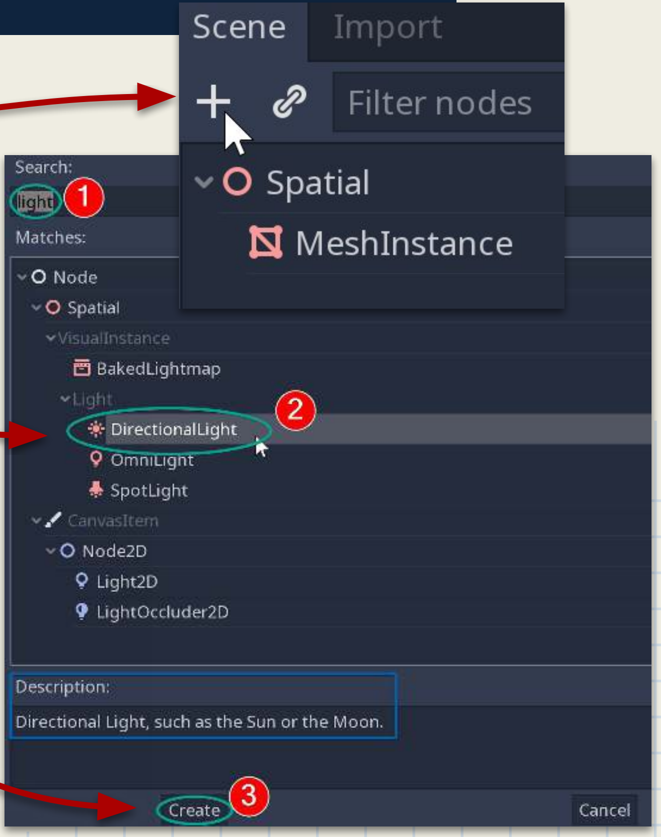
Add a Model to The Scene

- Click on the folder, “assets” on the File System dock.
- Drag-and-drop **level.obj** into the center of the screen.



Add a Light to The Scene

- On Scene Tree dock, click the plus button.
- Search for “light” and select “DirectionalLight.”
- Click “Create.”



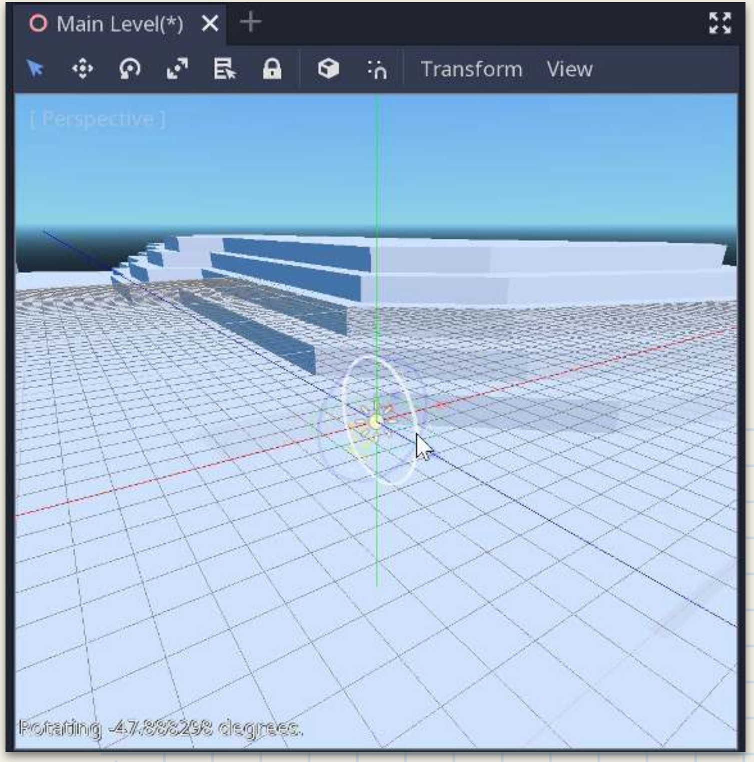
Search: light

Description:

Directional Light, such as the Sun or the Moon.

Add a Light to The Scene

- Rotate the directional light in the Scene pane by clicking-and-dragging the transparent rings.
- Press Ctrl+S/Cmd+S to save the scene



Rotating -47.888298 degrees.

Quick lexicon review

- Models
  - Industry-wide term for 3D sculpture files.
  - Can contain animations.
  - Usually made in a specialized program, e.g. Maya, Blender, etc.

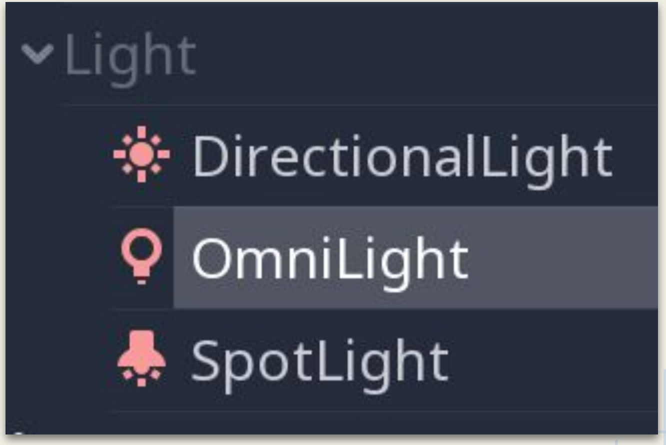
Importing 3D models

Godot can natively import:

- COLLADA (*.dae)
  - Supports UV-mapping, animations.
- Wavefront (*.obj)
  - Supports UV-mapping.
- glTF 2.0 (*.gltf)
  - Supports UV-mapping, animations, and materials.

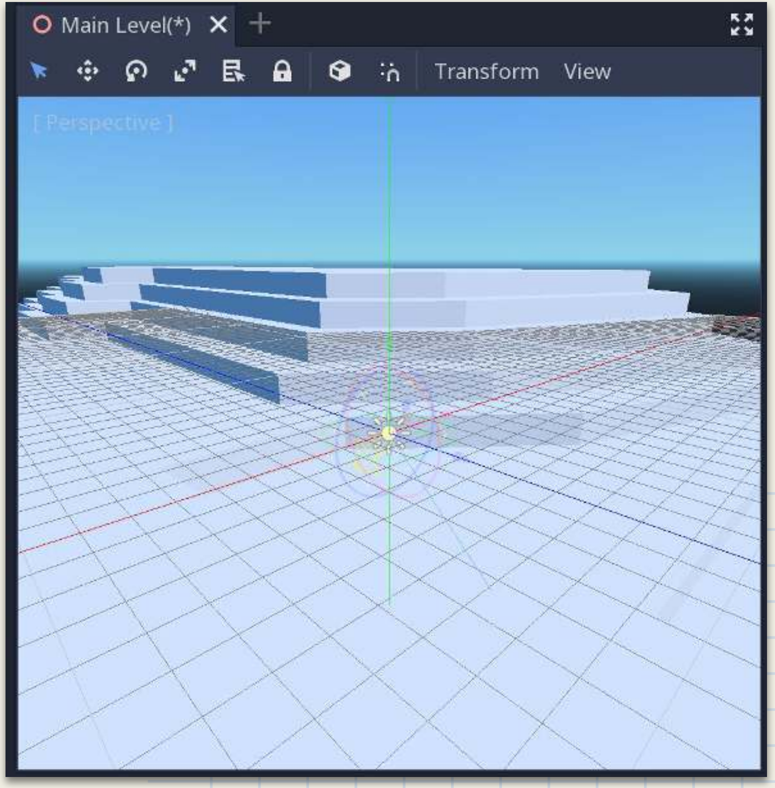
About Lighting

- Directional Light
  - A sunlight emitted in one direction
- Omni Light
  - A glow emitting from a single point
- Spotlight
  - A cone-shaped light used to simulate flashlights and spotlights



Scene Pane

- A 3D, tabbed view of a scene where objects can be positioned, rotated, and scaled.
- Nodes selected in the Hierarchy-pane are also selected in the Scene-pane, and vice versa.

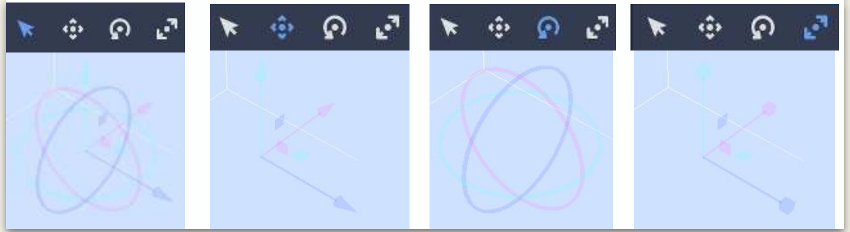


Navigating the scene pane

- 3-button mouse:
  - Left-click to select objects.
  - Hold right mouse button to rotate camera around camera position.
  - Scroll wheel to zoom in and out.
  - Click and hold on the scroll wheel to rotate camera around the point the camera is focusing on.
  - Hold shift, then click and hold on the scroll wheel to pan.

Manipulating game objects

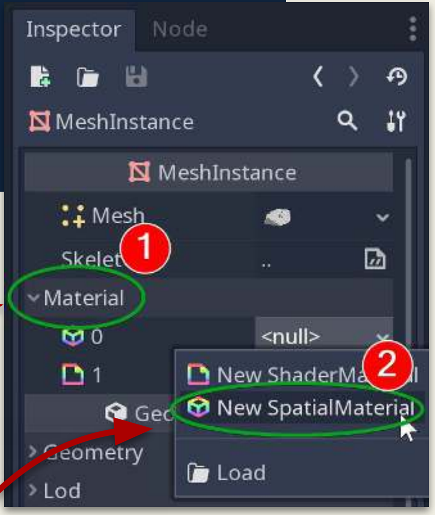
- Object controls, from left to right:
  - All (Q)
  - Translate (W)
  - Rotate (E)
  - Scale (R)

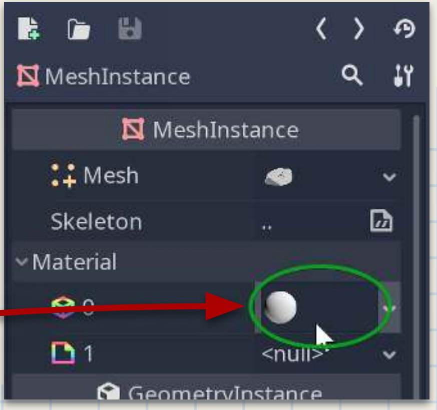


- Controls to toggle object’s reference point:
- Play Game controls, from left to right:
  - Play game from starting scene (more on that later)
  - Pause game
  - Stop game
  - Play current scene
  - Play a selected scene (opens a pop-up)

Creating Materials

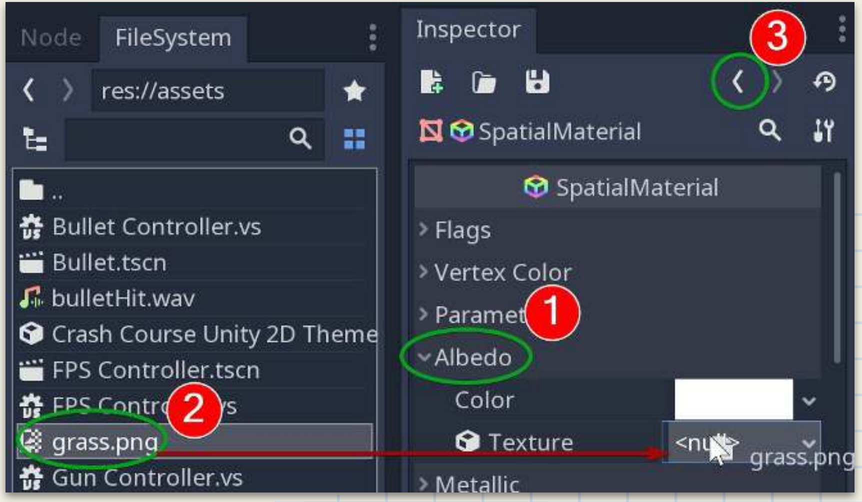
- Select “MeshInstance” in the Scene Tree dock.
- In the Inspector dock, at the lower right-hand corner, expand “Material.”
- Under material “0,” click the “v” button and select “New SpatialMaterial.”
- Click on the new material.





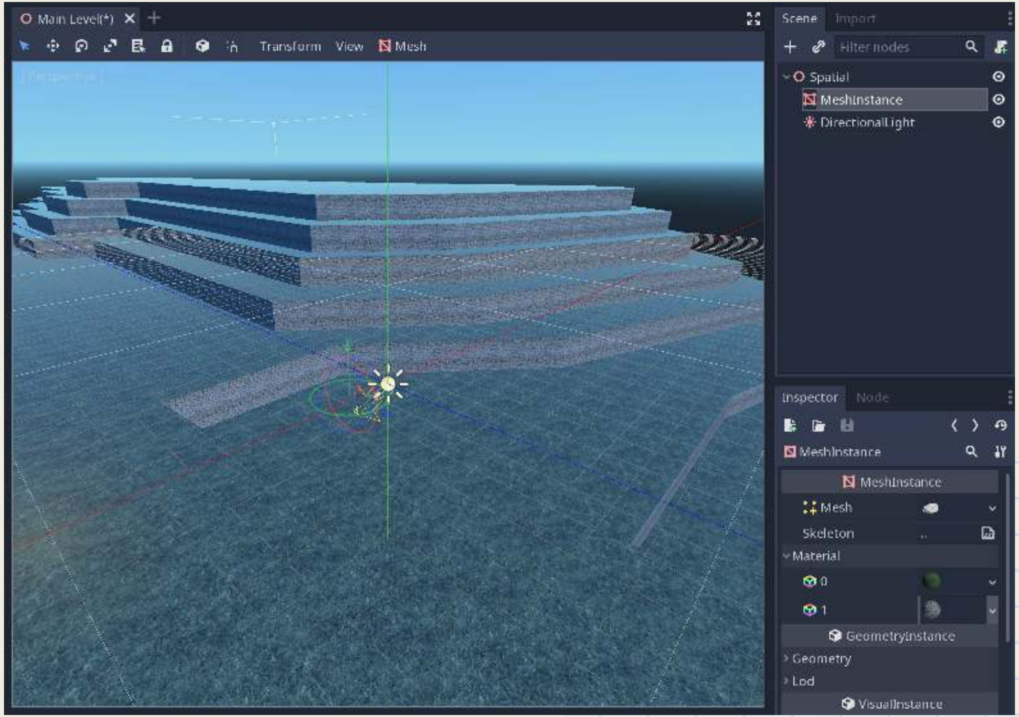
Adding Texture to Model

- Expand the “Albedo” group.
- From the File System dock, under the “assets” folder, drag-and-drop the “grass.png” to the “Texture” field.
- Click the back button.
  - On upper right-hand corner of the dock.



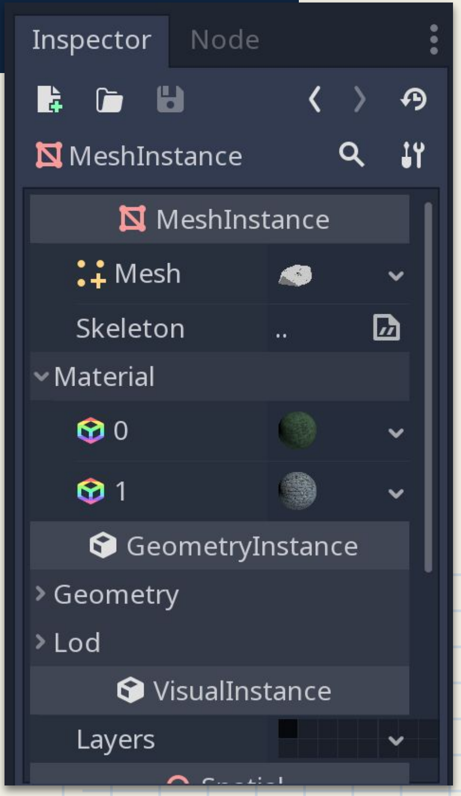
Adding Texture to Model

- Repeat the steps for Material "1," using file "rock.jpg".



Inspector Dock

- Displays the properties and details on a selected node in the Scene Tree dock.
- One can edit the properties of an object here.
- Some properties has yet another inspector; use the 2 arrows on the upper right-hand corner to move forward or back inspectors.



Quick lexicon review

- Material
    - Industry-wide term for what the quality surface of a model is supposed to look like (e.g. metal, plastic, non-shiny stuff)
    - In Godot, materials can be either embedded property or files (*.tres).
        - Changing a material file's properties will update all game objects with the same material
    - Models retain information on how materials are mapped to its surface
- Textures
    - Industry-wide term for images that represents how a model is supposed to be painted
    - Models usually contains coordinates (called UV) that indicate how a texture is supposed to be mapped on the model

Supported Images

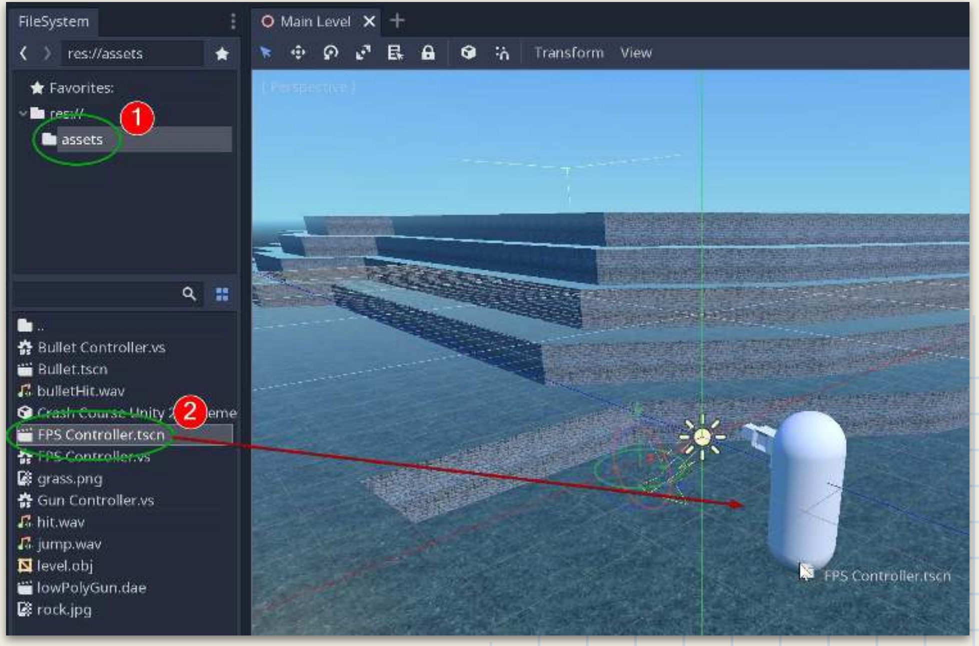
In Godot,

- PNGs (*.png)
- JPEGs (*.jpg, *.jpeg)
- and more!

Adding Camera

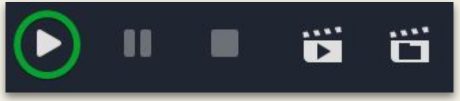
Godot can embed existing scenes into another scene.

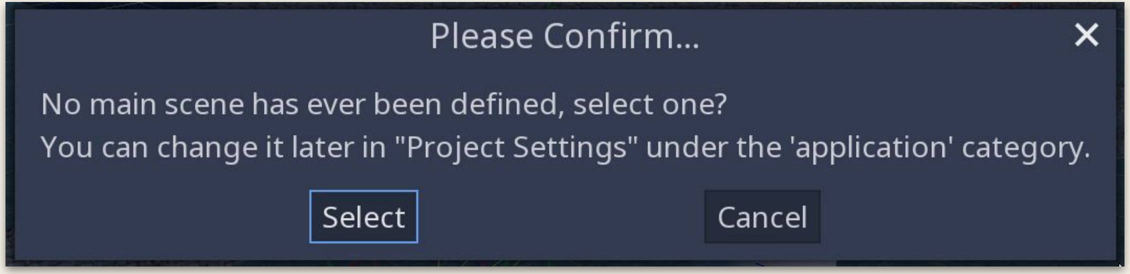
- From the File System dock, under the “assets” folder, drag “FPS Controller.tscn” into the Scene editor.



Playing the Game

- Press the play button.
- Observe that we get another pop-up.
- What does it mean?





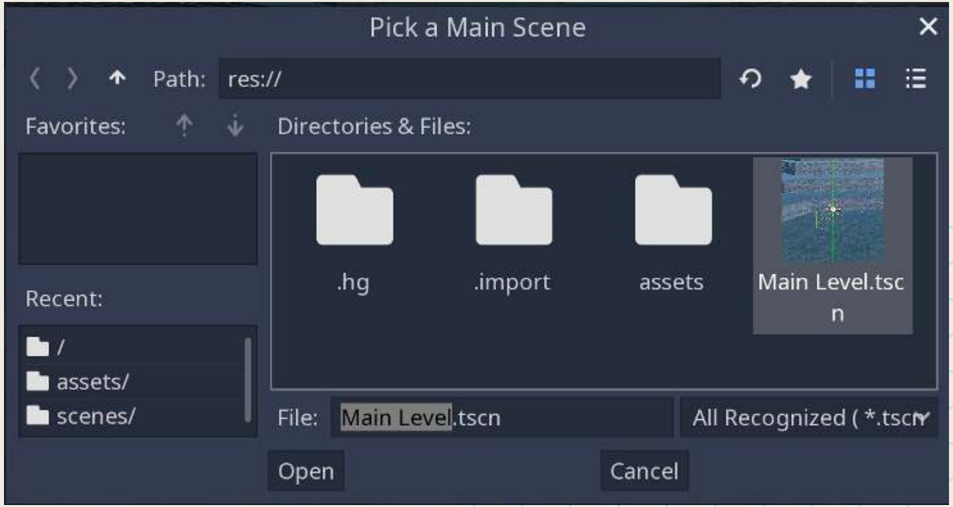
Please Confirm...

No main scene has ever been defined, select one?
You can change it later in "Project Settings" under the 'application' category.

Playing the Game

Godot requires that the project settings defines which scene to start the game on. Fortunately, there's a convenient shortcut to set this setting.

- Click on "Select."
- On the file browser, the current scene should be selected. Just click, "Open."
- This sets the setting!



Pick a Main Scene

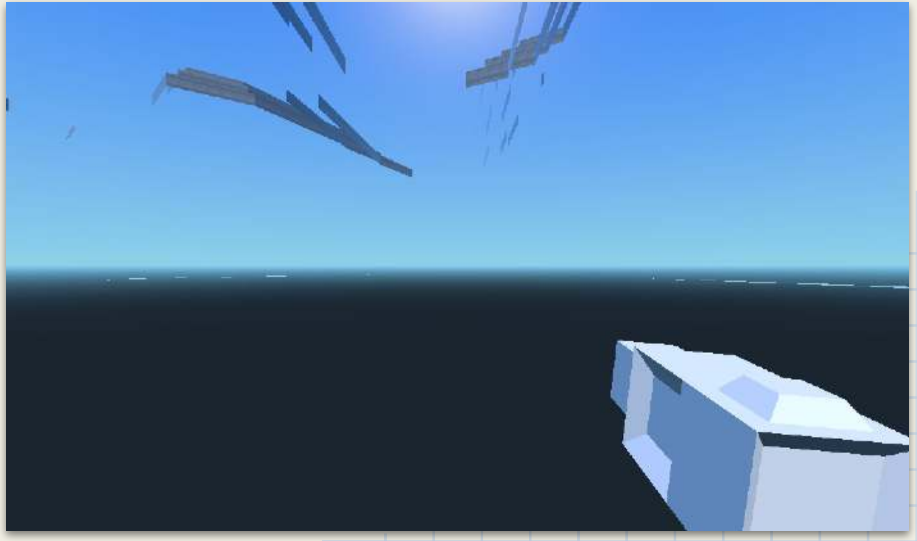
Path: res://

File: Main Level.tscn

Playing the Game

The game should now play!

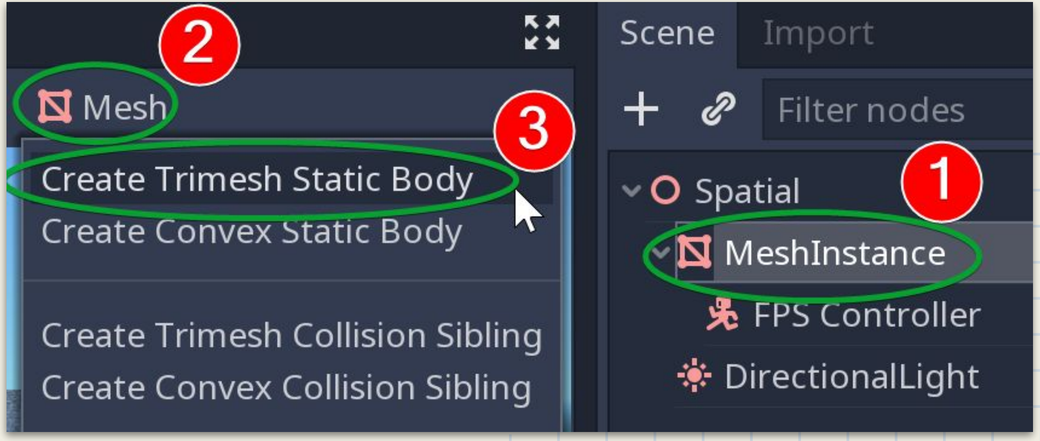
- ...except the player is falling through the floor...
- What's missing?



Adding a Collider

The ground needs a collider with the same shape as the model. Fortunately, Godot has a shortcut.

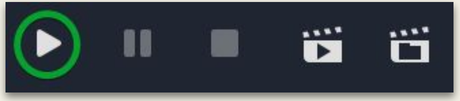
- Select “MeshInstance” in the Scene Tree dock.
- In the Scene editor, click the “Mesh” button.
- Select “Create Trimesh Static Body.”



2
Mesh
3
Create Trimesh Static Body
Create Convex Static Body
Create Trimesh Collision Sibling
Create Convex Collision Sibling
Scene
Import
Filter nodes
1
Spatial
MeshInstance
FPS Controller
DirectionalLight

Playing the Game

- Press the play button.



- Use the mouse to look around, arrow keys (or WASD) to move, and space to jump.
- Left-click to shoot.

Quick lexicon review

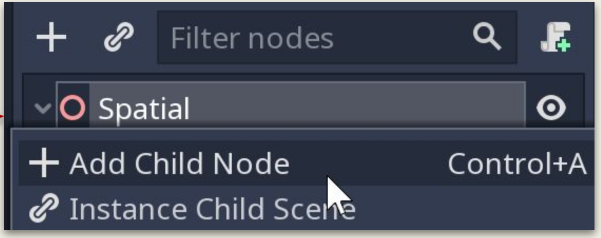
- Colliders
  - Industry-wide term for shapes representing the boundaries of an object
  - Used by the physics engine to determine where objects collide

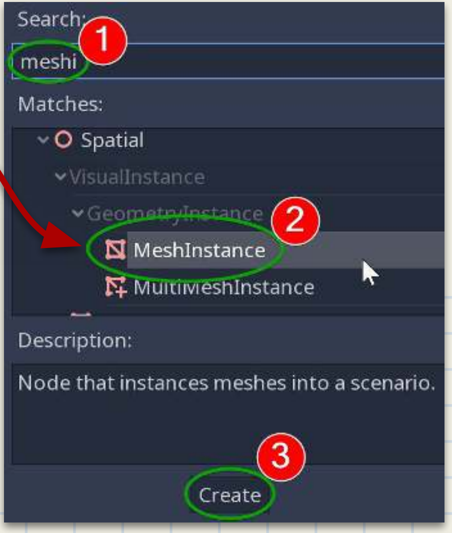
Types of Collider Nodes

- Static Body
  - A node for handling non-interactive objects.
  - No physics, e.g. gravity, is applied to this node.
  - Can be animated.
  - Needs at least one collider node as children for it to do anything.
- Collision Shapes
  - A collider node for a predefined shapes.
  - Can be generated from model's data.
  - Requires a parent Body node (like Static Body).
- Collision Polygon
  - A collider node where the polygon can be defined by the user.
  - Requires a parent Body node (like Static Body).

Adding Premade Graphics

- Under the Scene Tree dock, right-click the root node, “Spatial.”
- In the pop-up, select MeshInstance, and click, “create.”
- Observe that a new node, MeshInstance2 was created, but nothing appears in Scene Editor.
  - Note: for brevity, these node creation steps will be shortened to, “create X”




Search:

meshi

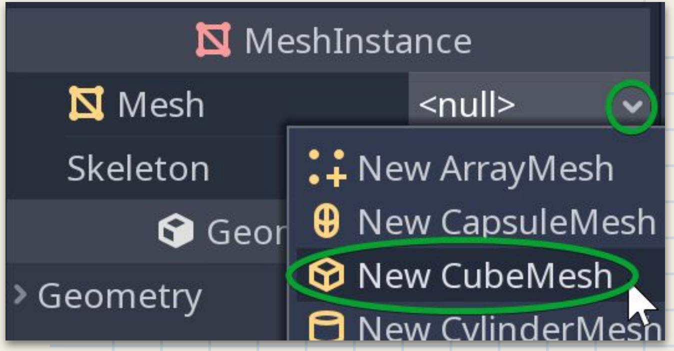
Matches:

Description:

Node that instances meshes into a scenario.

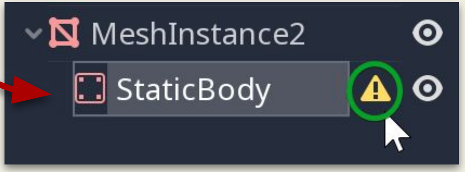
Adding Premade Graphics

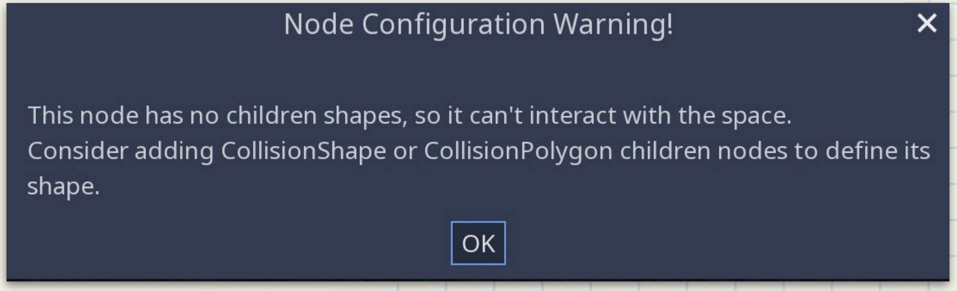
- In the Scene Tree dock, left-click MeshInstance2.
- In the Properties dock, change the mesh to "New CubeMesh."
- A floating cube will appear in the Scene editor, but we know from previous example this cube has no colliders.



Adding Premade Collider

- Right-click “MeshInstance2,” and create a StaticBody node.
- Clicking on the warning symbol brings up a pop-up explaining the StaticBody expects children.
- Click “OK” to close the dialog.
- Right-click StaticBody and create a CollisionShape.

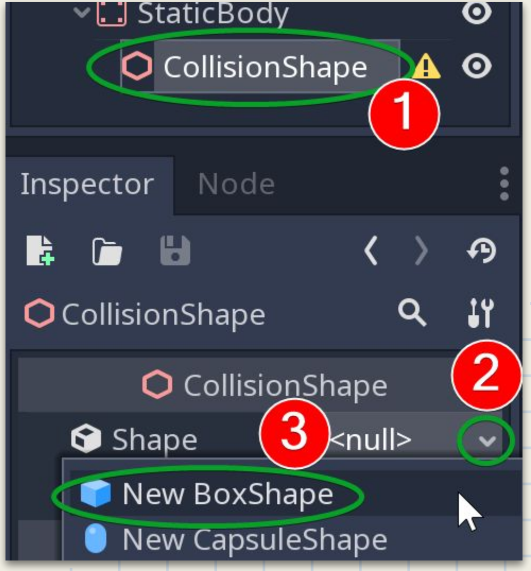



Node Configuration Warning!

This node has no children shapes, so it can't interact with the space. Consider adding CollisionShape or CollisionPolygon children nodes to define its shape.

Adding Premade Collider

- CollisionShape node will have a warning indicating no shape is defined.
- Select the CollisionShape to update the inspector.
- In the Inspector dock, click on the arrow next to the Shape field.
- Select "New BoxShape."
- Play the game, and shoot the cube!

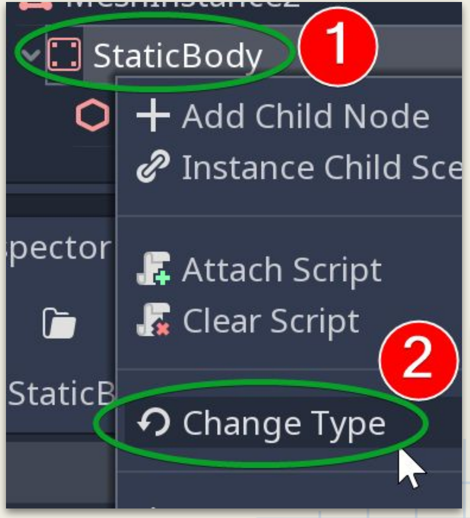


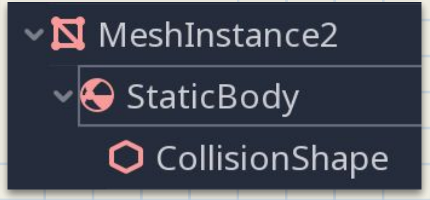
Adding Premade Collider

- Why go through such a lengthy method to put together a collider?
  - A shorter method to create tri-mesh collider exists, after all.
- Tri-mesh collider works well for complex shape that rarely moves, but it’s bad for performance.
- Premade collision shapes, like cubes and spheres are much faster!
- Let’s make the cube react to the bullets (and gravity)!

Adding Interactive Stuff

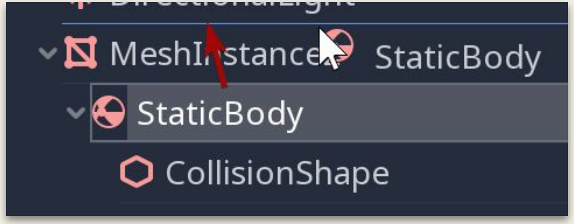
- Right-click on StaticBody, and click, “Change Type.”
- Change the type to Rigidbody.
  - Notice the icon changes, but name doesn’t.
  - Icon represents type.
  - Name is customizable.
- Play the game, and shoot the cube.
- Try shooting at the ground below the cube.



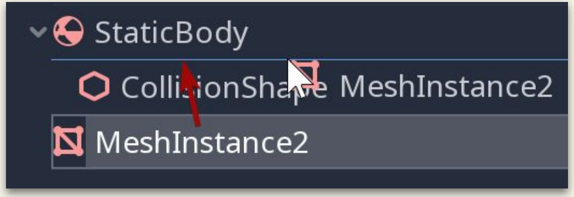


Adding Interactive Stuff

- Drag-and-drop the StaticBody above the MeshInstance2.



- Next, drag MeshInstance2 under StaticBody. Play the game!

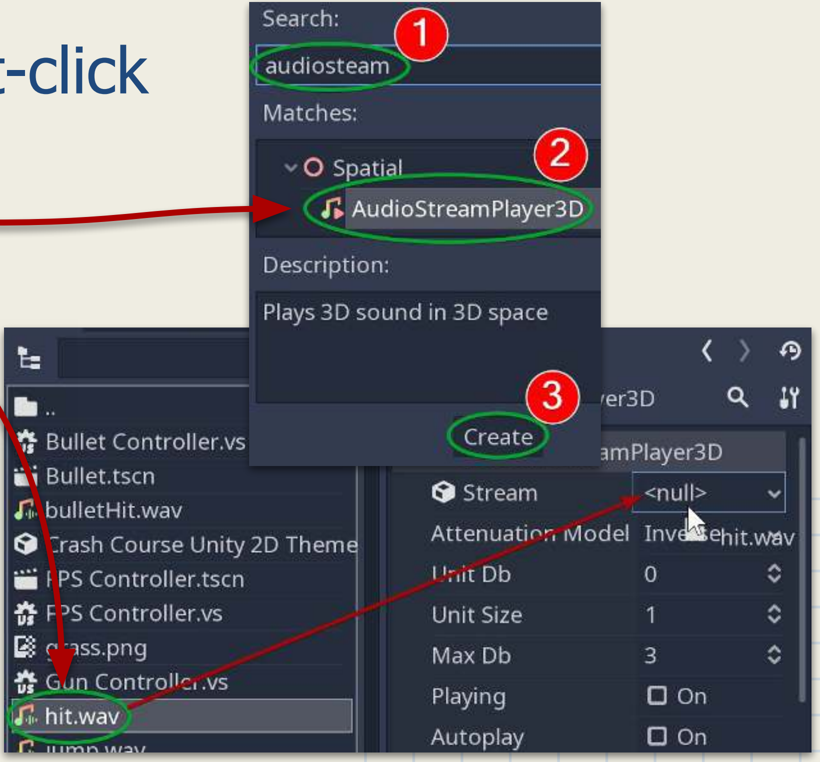


Quick lexicon review

- Rigid Body
  - Industry-wide term for a moving and/or interactive physics objects
  - Contains information such as mass, drag, and center-of-gravity
  - Turns a group of colliders (including those in the children) into a single, interactable shape
- Child
  - Industry-wide term for an object whose position, scale, and rotation follows that of another object: the parent
  - In Godot, they appear as nested entries in the Hierarchy tree view
  - Trivia: this “following parent” calculation process is known as forward kinematics, a term in 3D animations

Adding a Sound

- In the Scene Tree dock, right-click StaticBody, and create AudioStreamPlayer3D node.
- Drag file “hit.wav” to the the property, “Stream.”
- Now we need the sound to play each time the cube hits Something. How?



Search:
audiosteam

Matches:
Spatial
AudioStreamPlayer3D

Description:
Plays 3D sound in 3D space

Create

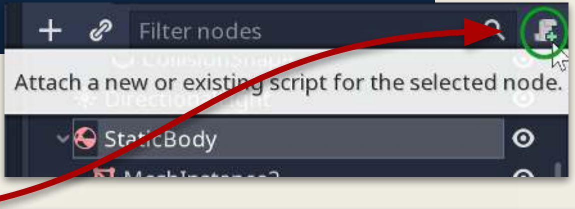
..
Bullet Controller.vs
Bullet.tscn
bulletHit.wav
Crash Course Unity 2D Theme
FPS Controller.tscn
FPS Controller.vs
grass.png
Gun Controller.vs
hit.wav
jump.wav

er3D
amPlayer3D

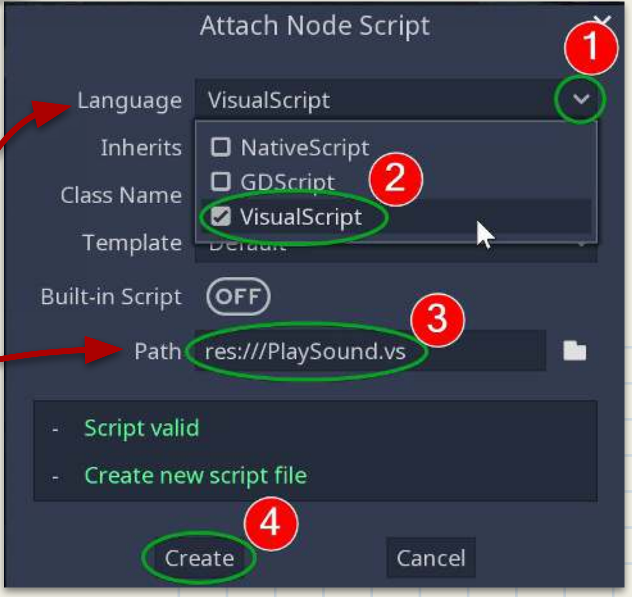
Stream	<null>
Attenuation Model	Inverse hit.wav
Unit Db	0
Unit Size	1
Max Db	3
Playing	□ On
Autoplay	□ On

Adding a Script

- Select StaticBody, then click the scroll with the plus sign on the upper right-hand corner of the dock.
- In the pop-up, select the Language, "VisualScript," and change the path to "res:///PlaySound.vs"
- Click, Create.



Attach a new or existing script for the selected node.



Language: VisualScript
- ☐ NativeScript
- ☐ GDScript
- ☑ VisualScript

Template: Default

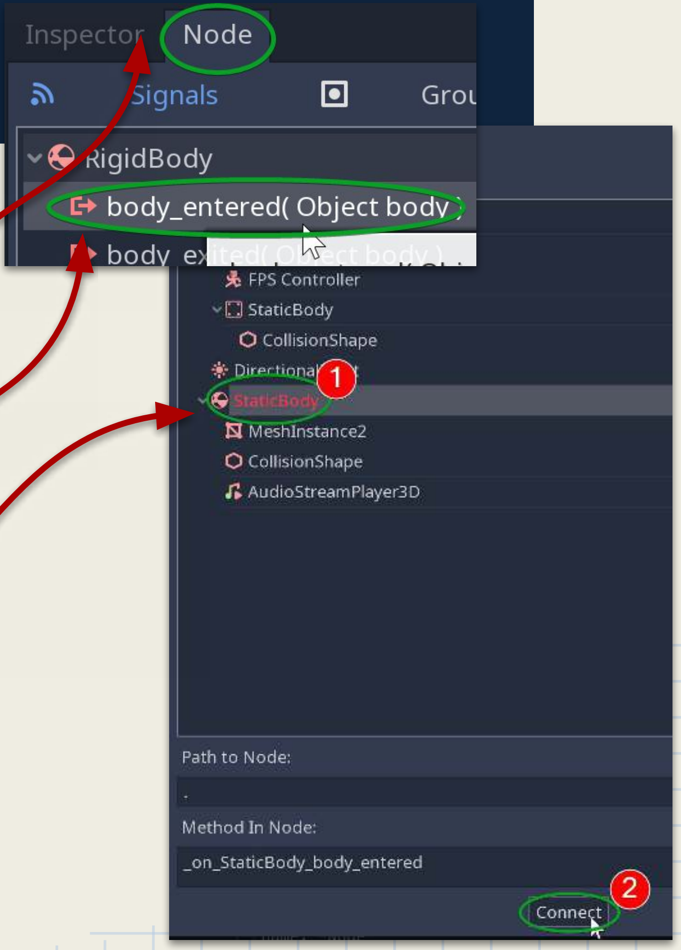
Built-in Script: OFF

Path: res:///PlaySound.vs

- Script valid
- Create new script file

Code the Script

- While StaticBody is still selected, click on the "Node" tab next to the Inspector.
- Double-click on "body_entered(Object body)"
- In the pop-up, select "StaticBody," and click, "Connect."

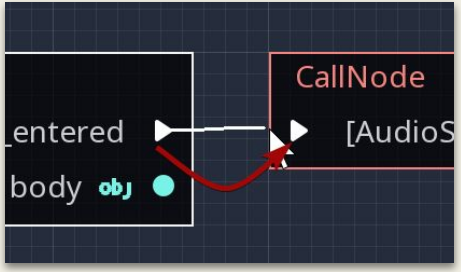


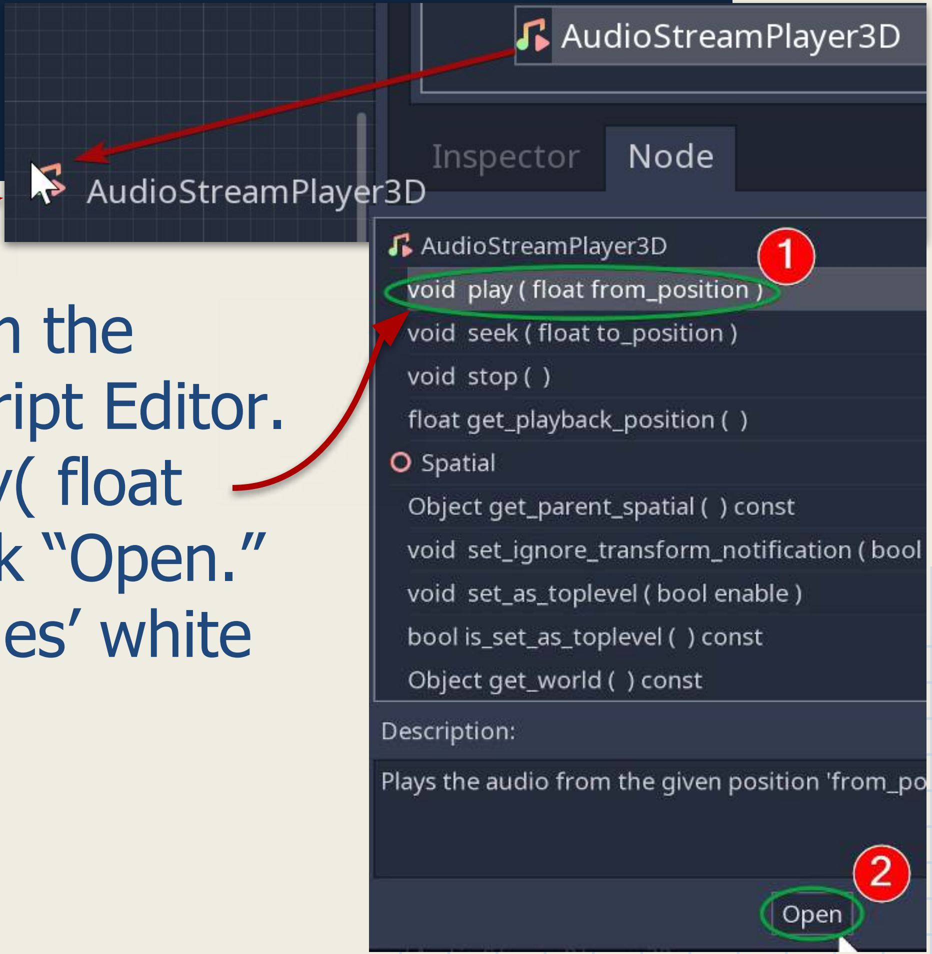
Path to Node: .

Method In Node: _on_StaticBody_body_entered

Code the Script

- Drag-and-drop the AudioStreamPlayer3D from the SceneTree node to the Script Editor.
- In the pop-up, select “play(float from_position)”, then click “Open.”
- Then connect the two nodes’ white arrows:





AudioStreamPlayer3D

void play (float from_position)

void seek (float to_position)

void stop ( )

float get_playback_position ( )

Spatial

Object get_parent_spatial ( ) const

void set_ignore_transform_notification (bool

void set_as_toplevel (bool enable)

bool is_set_as_toplevel ( ) const

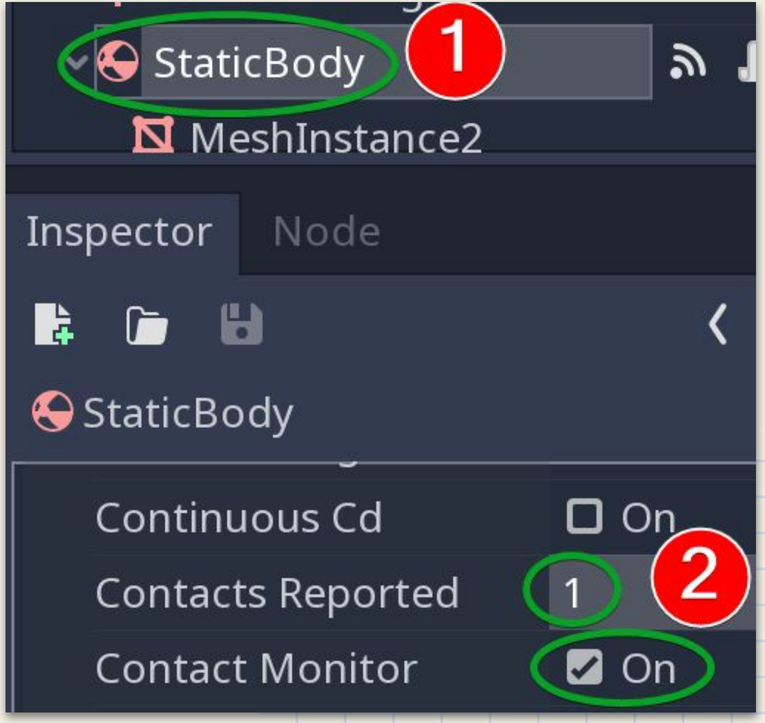
Object get_world ( ) const

Description:

Plays the audio from the given position 'from_po

Finishing the Sound Effect

- Play the game.
  - Notice there's no sound effect. Something is missing!
- In the Scene dock, select StaticBody.
- Click the Inspector tab.
- Check the "Contact Monitor" property.
- Change "Contacts Reported" to 1. Play the game.



Continuous Cd □ On

Contacts Reported 1

Contact Monitor ☑ On

Script Summary

- Visual Scripting takes a graph-based approach.
- First, a function based on an event or signal triggers.
- Then the functions follows the white line to the next node.
- This node process the command; it then decides which white triangle to follow through next (if any).
- If there is a white line from that triangle, it follows that to the next node.
- Repeat until it reaches an end.

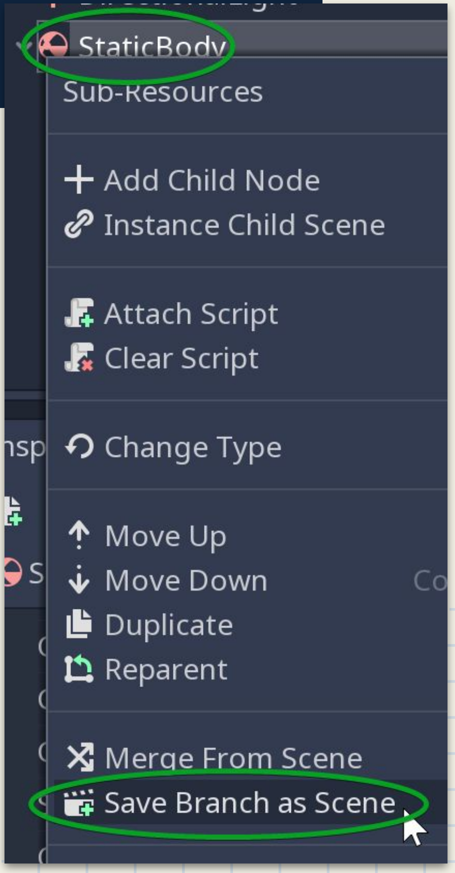
Importing Sound

Godot can natively import:

- WAV (*.wav)
  - Best for short sound effects
- OGG (*.ogg)
  - Best for music, especially PC and consoles

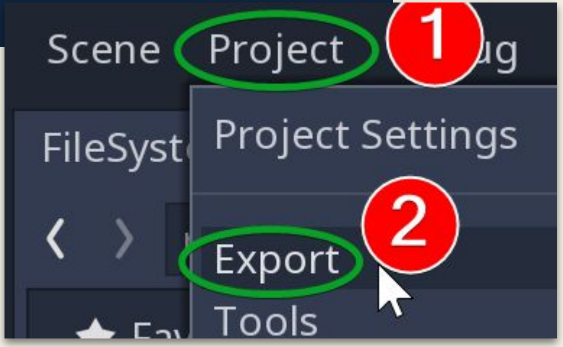
Duplicating the Cube

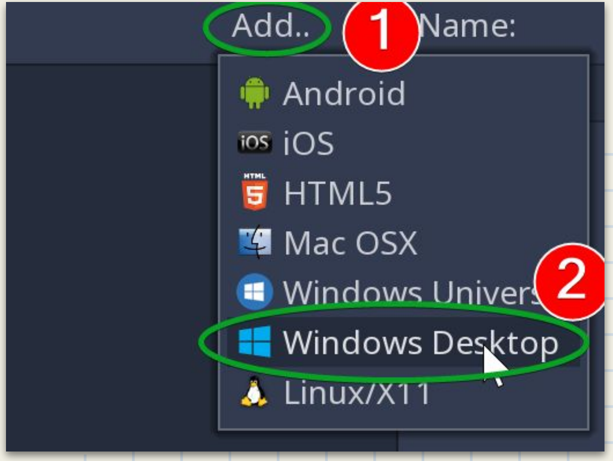
- Right-click on StaticBody, and select "Save Branch as Scene."
- Save the scene as "Cube.tscn"
- Select Spatial (the root node) in Scene Tree dock.
- Drag-and-drop Cube.tscn in the Scene editor everywhere!
  a. Remember, scenes can be added to another like a normal node!



Building an Executable

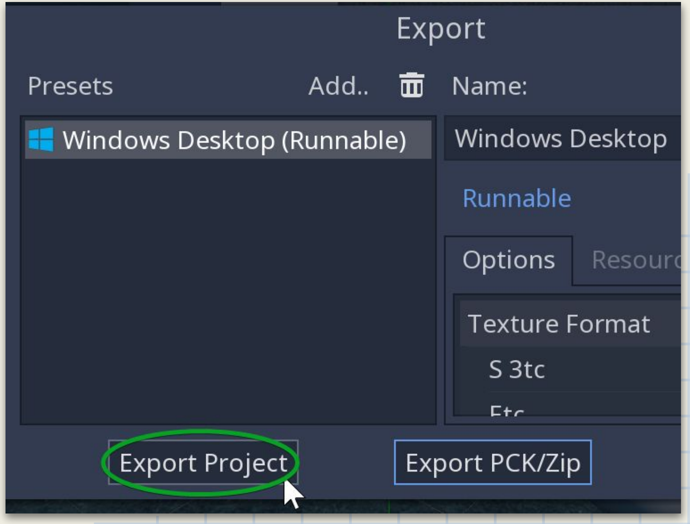
- Save the scene with Ctrl+S/Cmd+S.
- In the file menu, select "Project -> Export"
- In the pop-up dialog, click “Add…” and select your platform.





Building an Executable

- Click “Export Project.”
- Select a folder that *isn’t* in your project.
- Open the folder, and play the build!

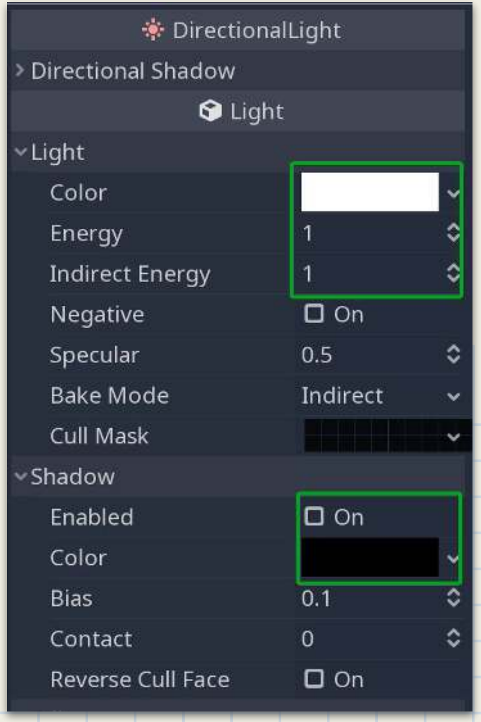


Congratulations!

Any questions?

Polishing Lighting

- Select “DirectionalLight,” and take a look at the properties.
- Try changing any of the properties highlighted on the right.
    - Highly recommend checking the “Shadow -> Enabled” checkbox!



DirectionalLight

› Directional Shadow

Light

⌄ Light

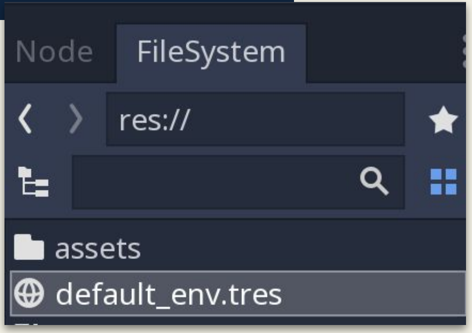
Color	⬜
Energy	1
Indirect Energy	1
Negative	□ On
Specular	0.5
Bake Mode	Indirect
Cull Mask	

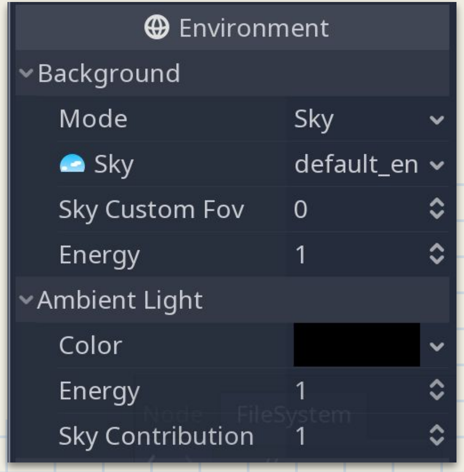
⌄ Shadow

Enabled	□ On
Color	⬛
Bias	0.1
Contact	0
Reverse Cull Face	□ On

Polishing Post-Processing

- Double-click the file, “default_env.tres,” then check the Inspector.
- See what happens if you change any of the “Background” properties.
- See what happens if you change any of the “Ambient Light” properties.





Environment

Background

Mode: Sky

Sky: default_en

Sky Custom Fov: 0

Energy: 1

Ambient Light

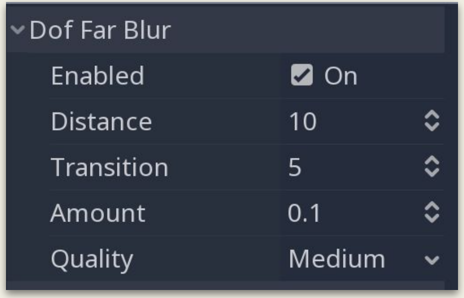
Color: ■

Energy: 1

Sky Contribution: 1

Polishing Post-Processing

- See what happens if you change any of the “Dof Far Blur” properties.
    - Note: DOF = Depth-of-Field.
- See what happens if you change any of the “Glow” properties.



Dof Far Blur

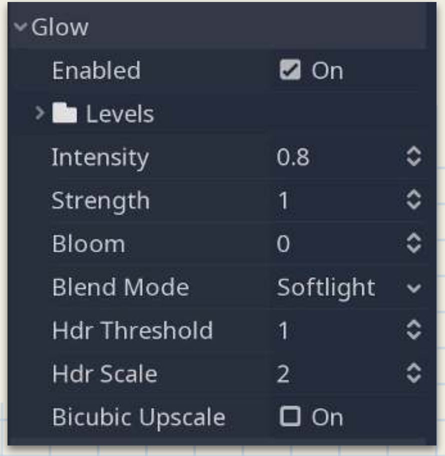
Enabled ☑ On

Distance 10

Transition 5

Amount 0.1

Quality Medium



Glow

Enabled ☑ On

Levels

Intensity 0.8

Strength 1

Bloom 0

Blend Mode Softlight

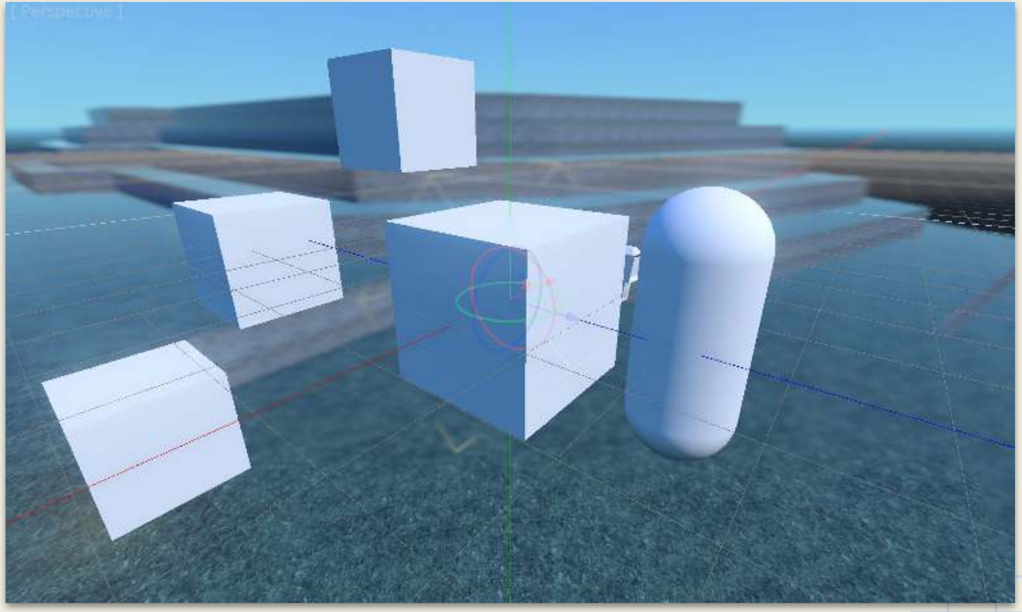
Hdr Threshold 1

Hdr Scale 2

Bicubic Upscale ☐ On

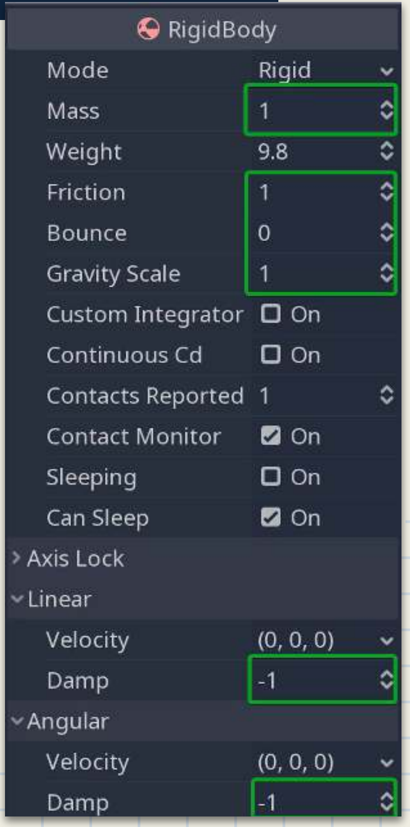
Polishing Post-Processing

- Did anything change?



Polishing Physics

- Select “StaticBody,” and take a look at the properties.
- Try changing any of the properties highlighted on the right.
  - Remember to play the game on each change, then shoot the cube to see what happens!

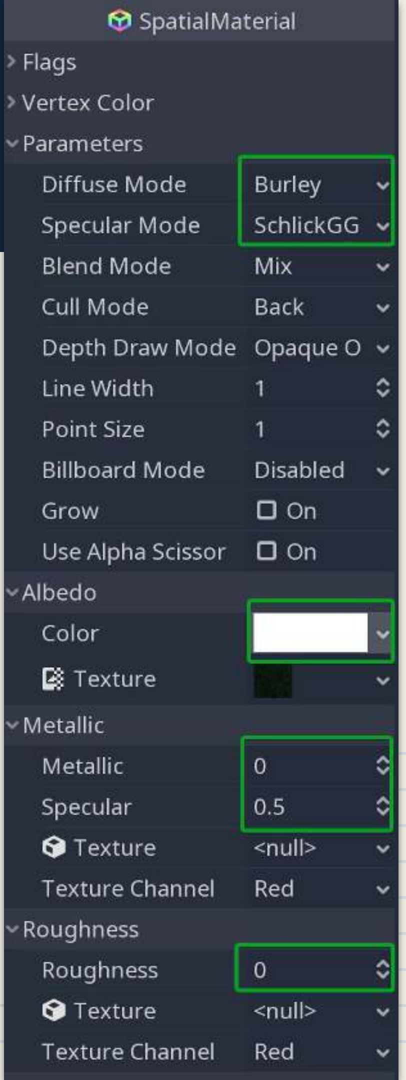


RigidBody

Mode	Rigid
Mass	1
Weight	9.8
Friction	1
Bounce	0
Gravity Scale	1
Custom Integrator	□ On
Continuous Cd	□ On
Contacts Reported	1
Contact Monitor	☑ On
Sleeping	□ On
Can Sleep	☑ On
Axis Lock	
Linear	
Velocity	(0, 0, 0)
Damp	-1
Angular	
Velocity	(0, 0, 0)
Damp	-1

Polishing Materials

- Select “MeshInstance,” then click on one of the materials under the Properties dock.
- Try changing any of the properties highlighted on the right.
  - Highly recommend adjusting the “Metallic” and “Roughness” properties.



SpatialMaterial

› Flags

› Vertex Color

⌄ Parameters

Diffuse Mode	Burley
Specular Mode	SchlickGG
Blend Mode	Mix
Cull Mode	Back
Depth Draw Mode	Opaque O
Line Width	1
Point Size	1
Billboard Mode	Disabled
Grow	☐ On
Use Alpha Scissor	☐ On

⌄ Albedo

Color	⬜
Texture	

⌄ Metallic

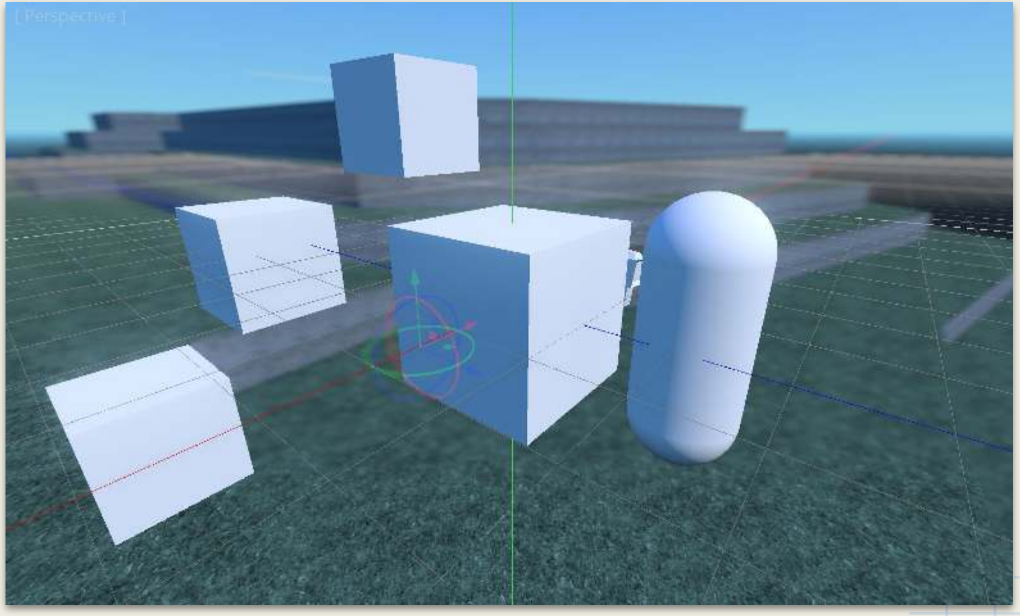
Metallic	0
Specular	0.5
Texture	<null>
Texture Channel	Red

⌄ Roughness

Roughness	0
Texture	<null>
Texture Channel	Red

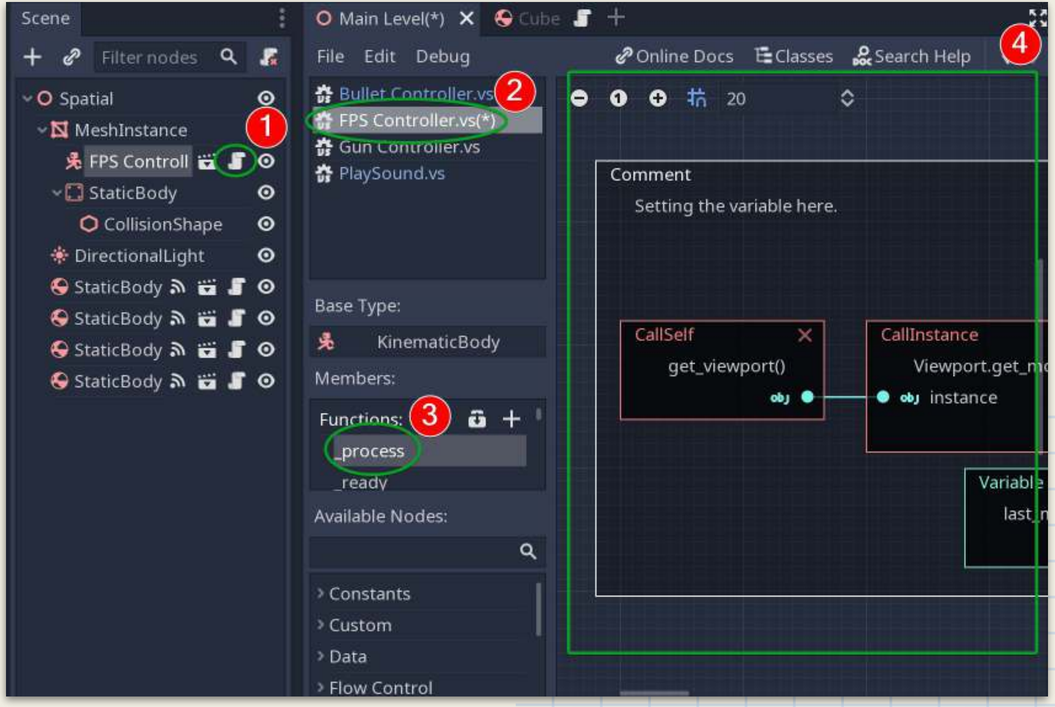
Polishing Materials

- Did anything change?



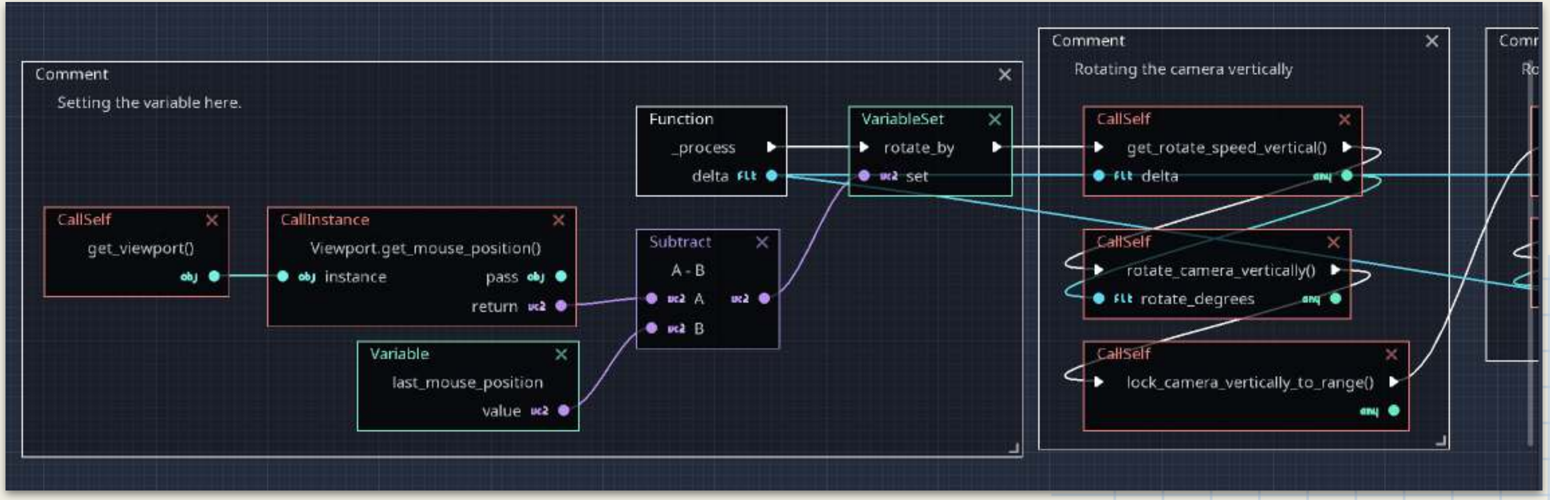
Reading Scripts

- Click on the scroll icon next to “FPS Controller” in Scene Tree dock.
- In the script editor, click “FPS Controller.vs”
- Finally, click the “_process” function.



Reading Scripts

- The function “_process(float delta)” runs every frame (i.e. all the time).
- Can you decipher what’s going on with the script?



Comment

Setting the variable here.

Comment

Rotating the camera vertically

Polishing Wrap-Up!

- What kind of polish did you make?