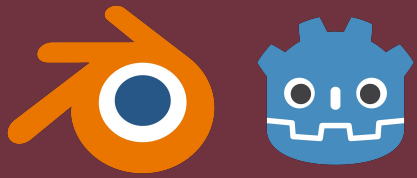
Sign in here!



DevLUp FSU GBM #2

Intro to Godot

January 30, 2025

Welcome to DevLUp at FSU!

Meeting Schedule

Date	GBM #	GBM Title	Secondary Event	Presenter
9 Jan		(No Meeting)	(oops, no involvement fair!)	
16 Jan	1	Introductions and Design Activity		All
23 Jan		(No Meeting)	SNOW!!!	
30 Jan	2	Intro to Godot (Orbital Odyssey Minigame)		Dion
6 Feb	3	Intro to Game Design	ASLC Showcase	Jake
13 Feb	4	Art Fundamentals (for artists and non-artists)		Parker
20 Feb	5	Accessibility in Games		Ares
27 Feb	6	Godot Tidbits		Dion
6 Mar	8	Intro to Unity		Jake
13 Mar		(No Meeting)	Spring Break	
20 Mar	7	Intro to Stencyl (Point & Click)		Whalen
27 Mar	9	1 Hour Game Jam (or Design Sprint)		Whalen
3 Apr	10	Game Jam Fundamentals	Game Jam?	Dion
10 Apr	11			
17 Apr	12		Innovators Showcase	
24 Apr	13			
1 May		(No Meeting)	Finals	

ASLC Indie Games Fest!

- We’re tabling for DevLUp and showing off some of our member’s games
- **Sunday, Feb. 9th, 2-6PM**
- Submit your game to us (here or in the announcement)

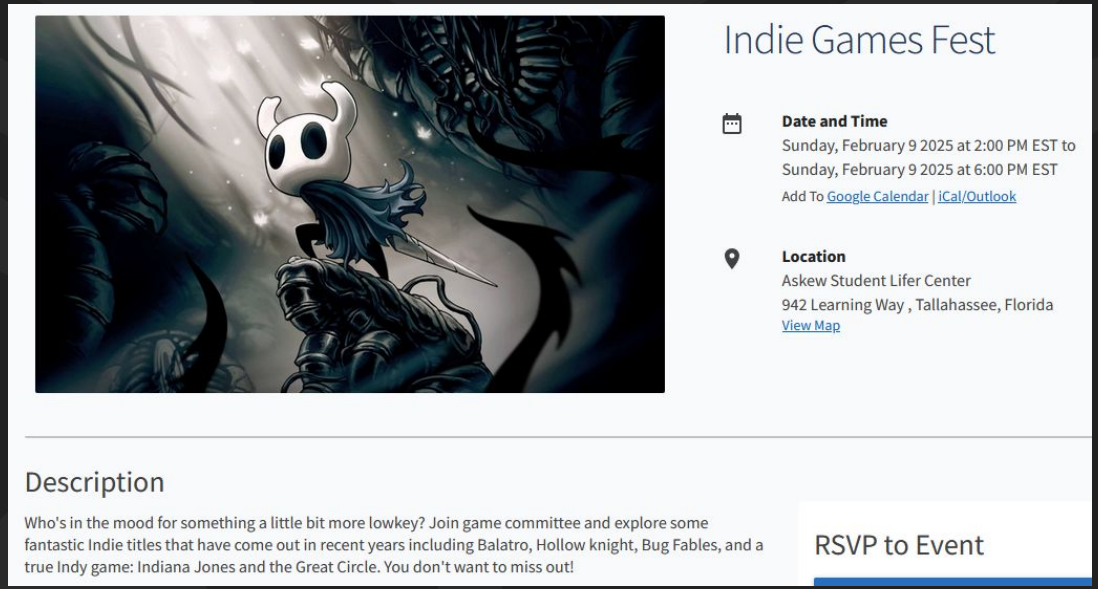
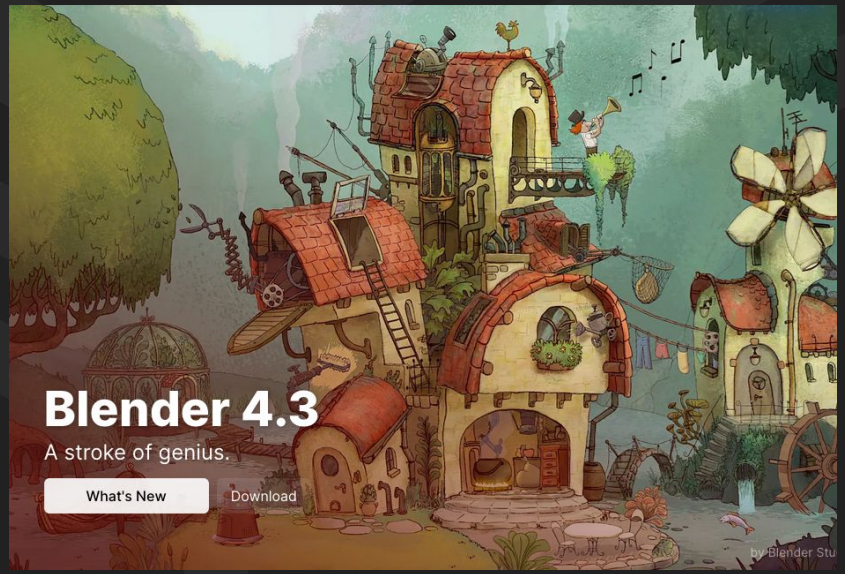


Figure χ: They haven’t made one of those cool promotional graphics yet :(

Download Blender

https://www.blender.org/

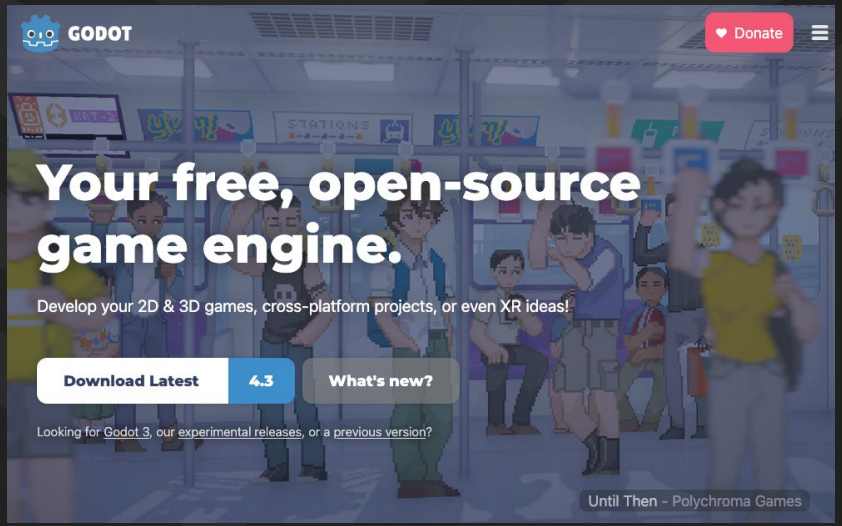
Latest version (v4.3)



Download Godot

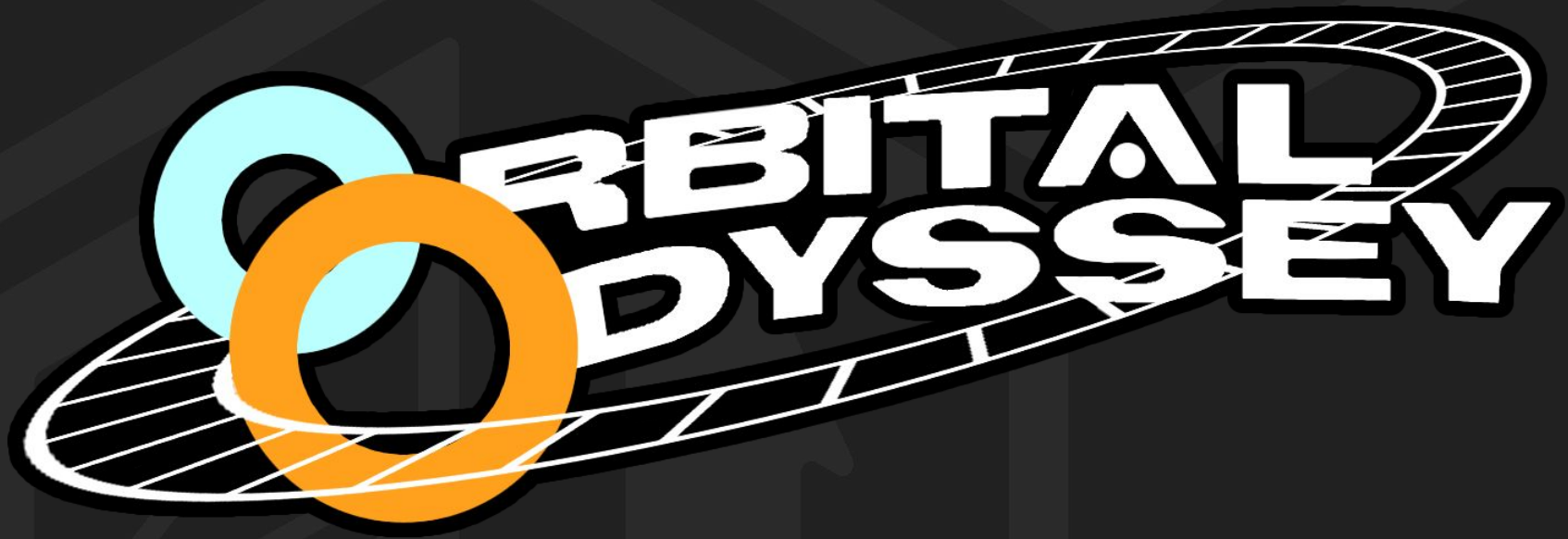
https://godotengine.org/

Latest version (v4.3)



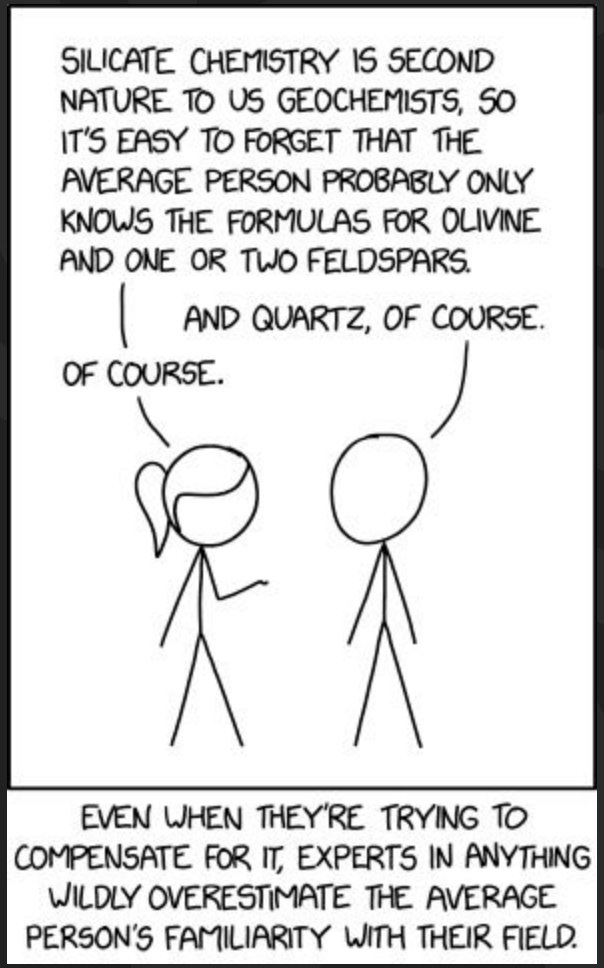
Download Orbital Odyssey Starter Files

https://github.com/devlup-fsu/party-game/archive/831e1a11a1533a874551dd58469e2fea1349b7e1.zip



#👀showoff

Recommended Further Learning


SILICATE CHEMISTRY IS SECOND NATURE TO US GEOCHEMISTS, SO IT'S EASY TO FORGET THAT THE AVERAGE PERSON PROBABLY ONLY KNOWS THE FORMULAS FOR OLIVINE AND ONE OR TWO FELDSPARS.
AND QUARTZ, OF COURSE.
OF COURSE.
EVEN WHEN THEY'RE TRYING TO COMPENSATE FOR IT, EXPERTS IN ANYTHING WILDLY OVERESTIMATE THE AVERAGE PERSON'S FAMILIARITY WITH THEIR FIELD.

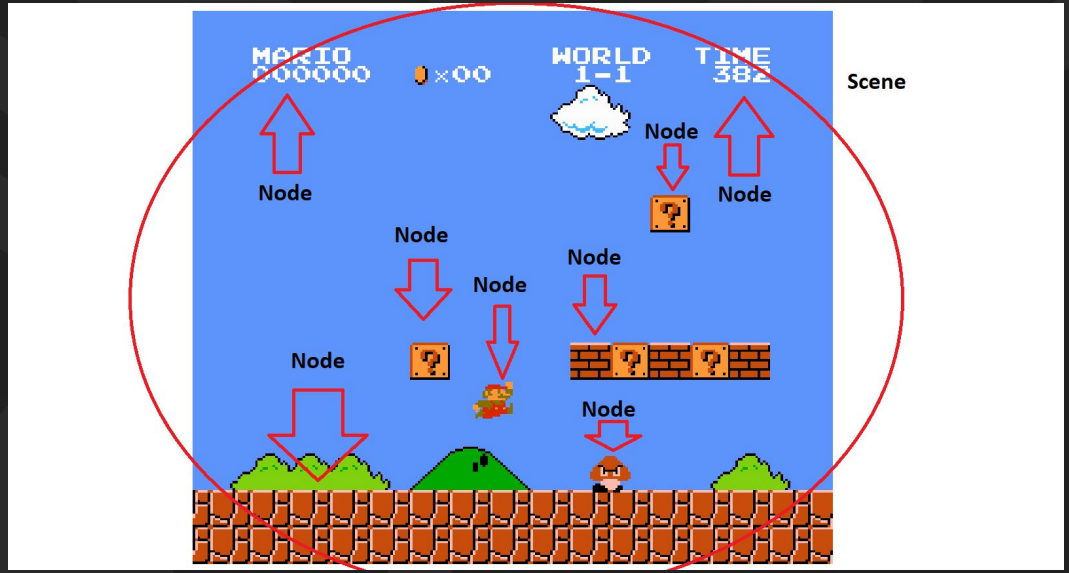
xkcd 2501

Scenes and Nodes

- The building blocks of a game

- The nodes are the individual parts of the game from Mario to the score to the enemies on the stage

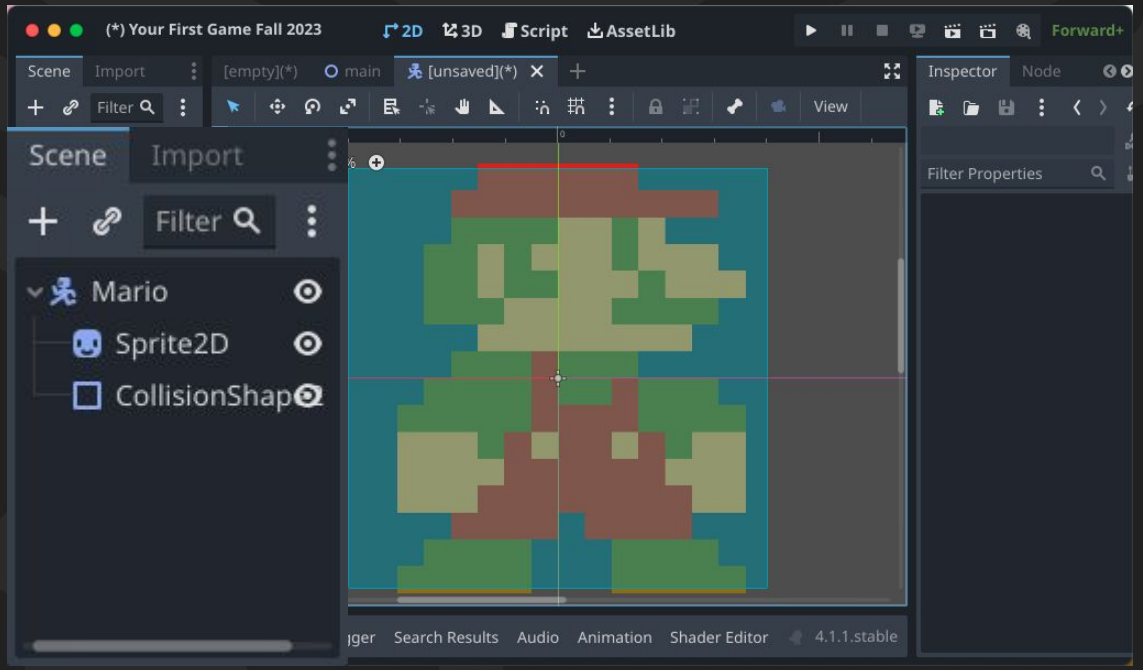
- Can think of the scene as the level of a game or a collection of nodes



Nodes *are* Scenes

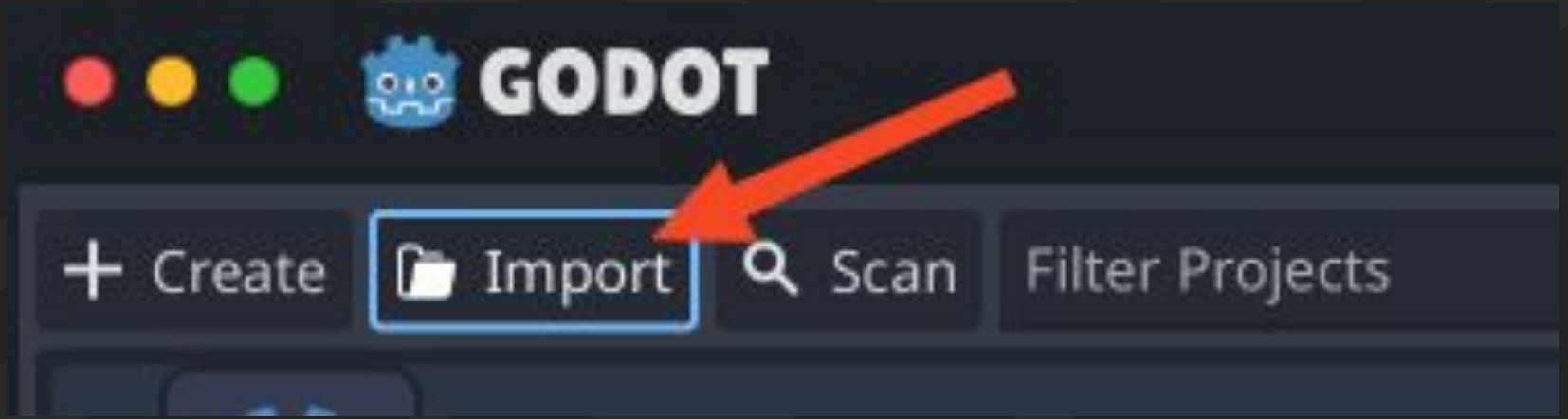
-Nodes can also have children too

-Children nodes can contain functionality like the collision of the parent node or the texture

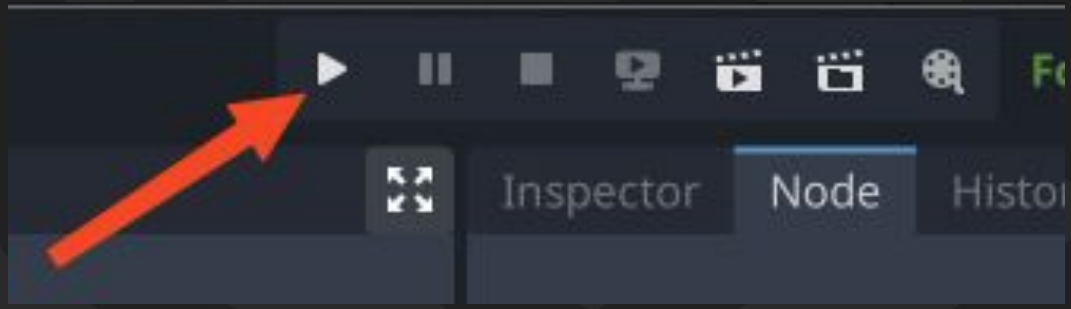


Setup the Project

Import Orbital Odyssey



Run the Game


Inspector
Node

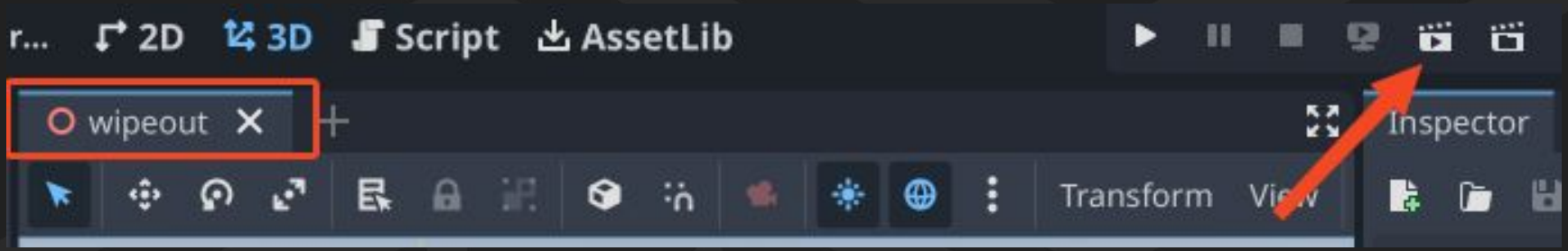

Party Game (DEBUG)
Player 1
Player 2
Player 3
Player 4
A
B
X
Y

Making a Minigame

What We’re Making Today

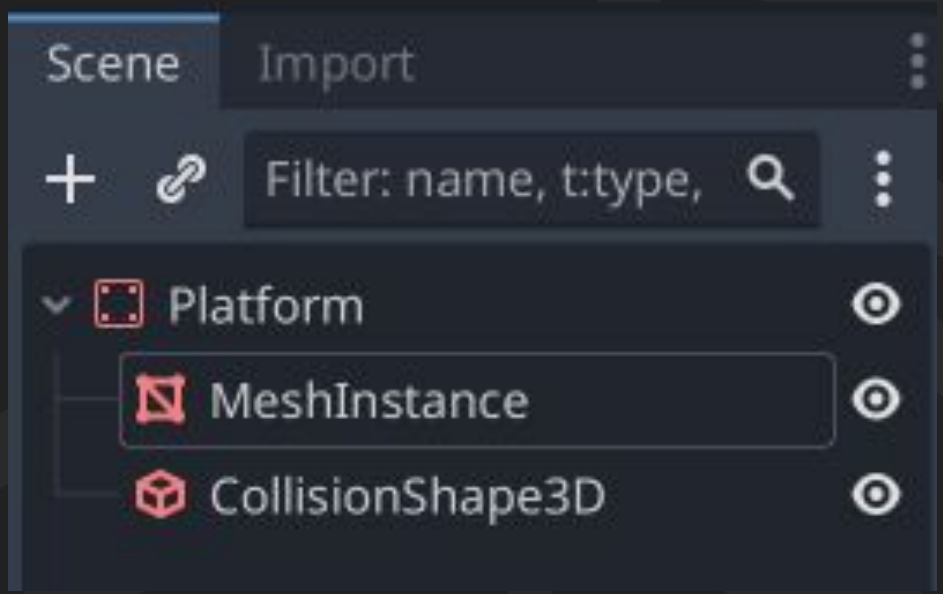
Each Minigame Has Its Own Folder

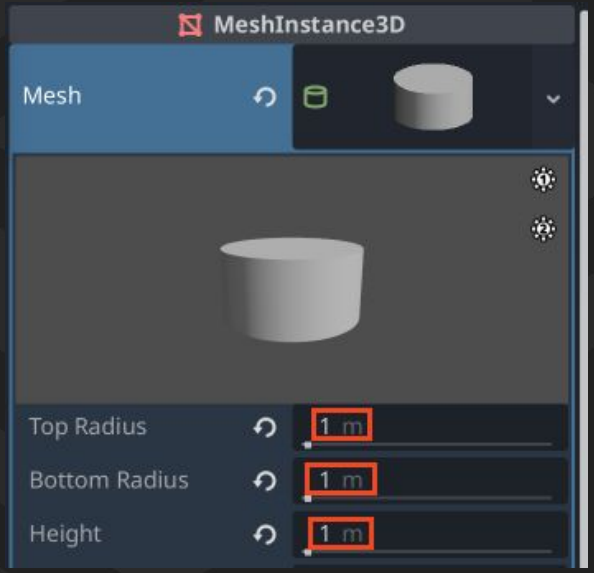
Run Your Minigame Directly

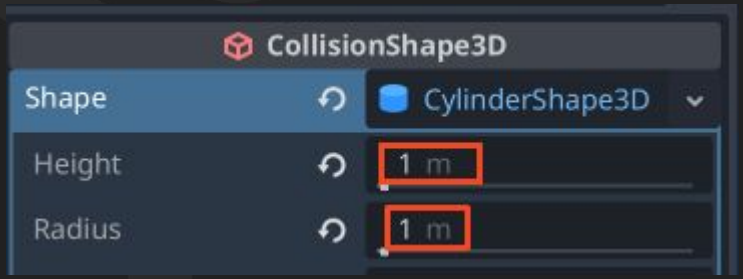


Creating the Platforms

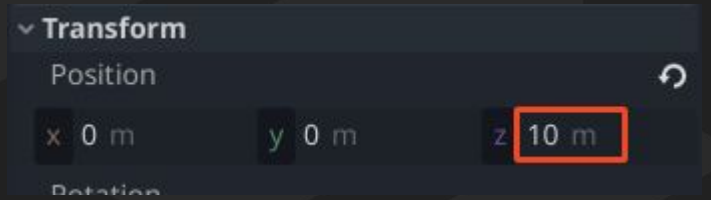
Create the Platform Scene

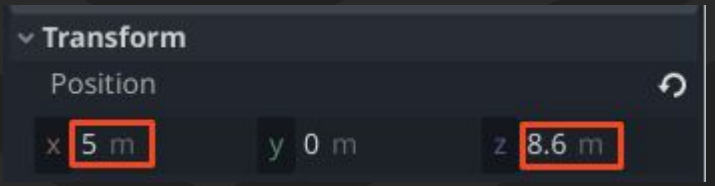


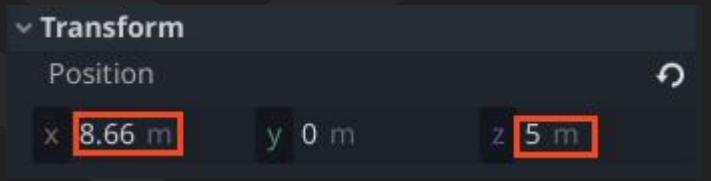


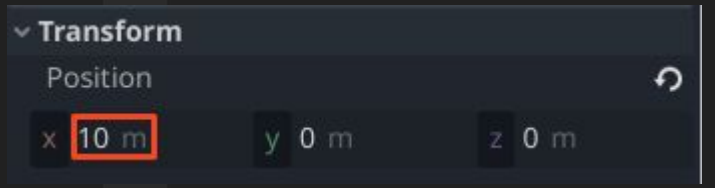


Place the Platforms

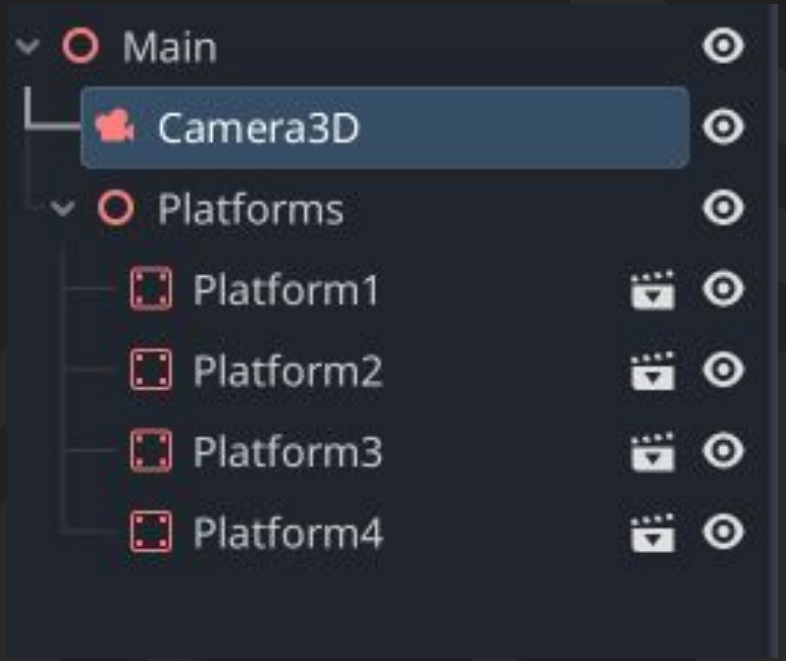




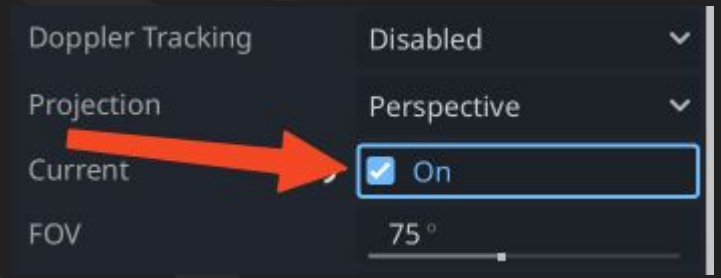




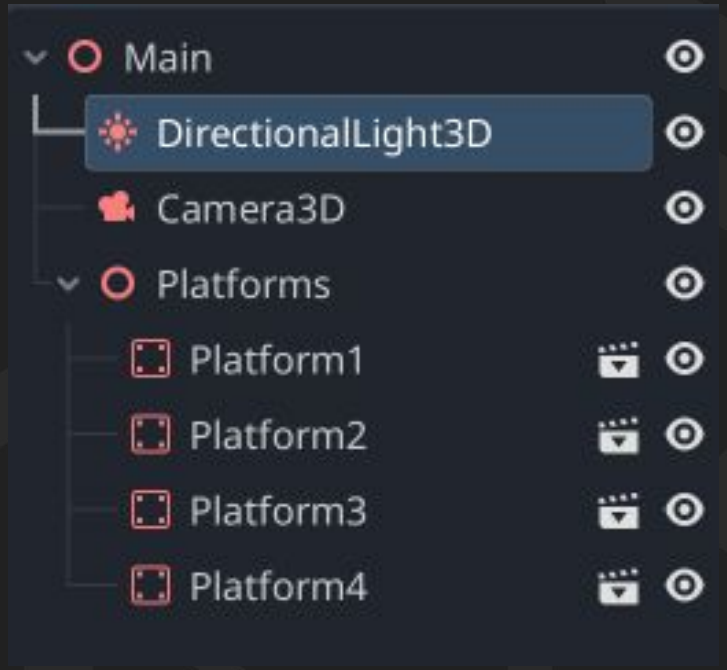
Add a Camera


Main
Camera3D
Platforms
Platform1
Platform2
Platform3
Platform4

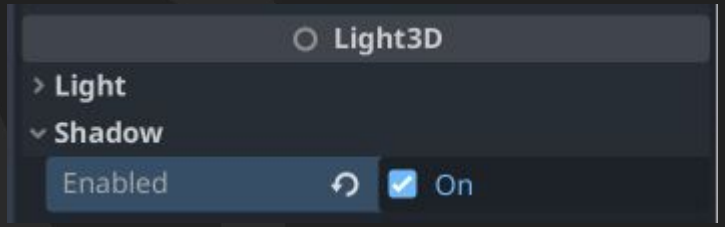

Transform
Position
x 9 m
y 4.5 m
z 9 m
Rotation
x -30 °
y 45 °
z 0 °
Scale


Doppler Tracking
Disabled
Projection
Perspective
Current
On
FOV
75 °

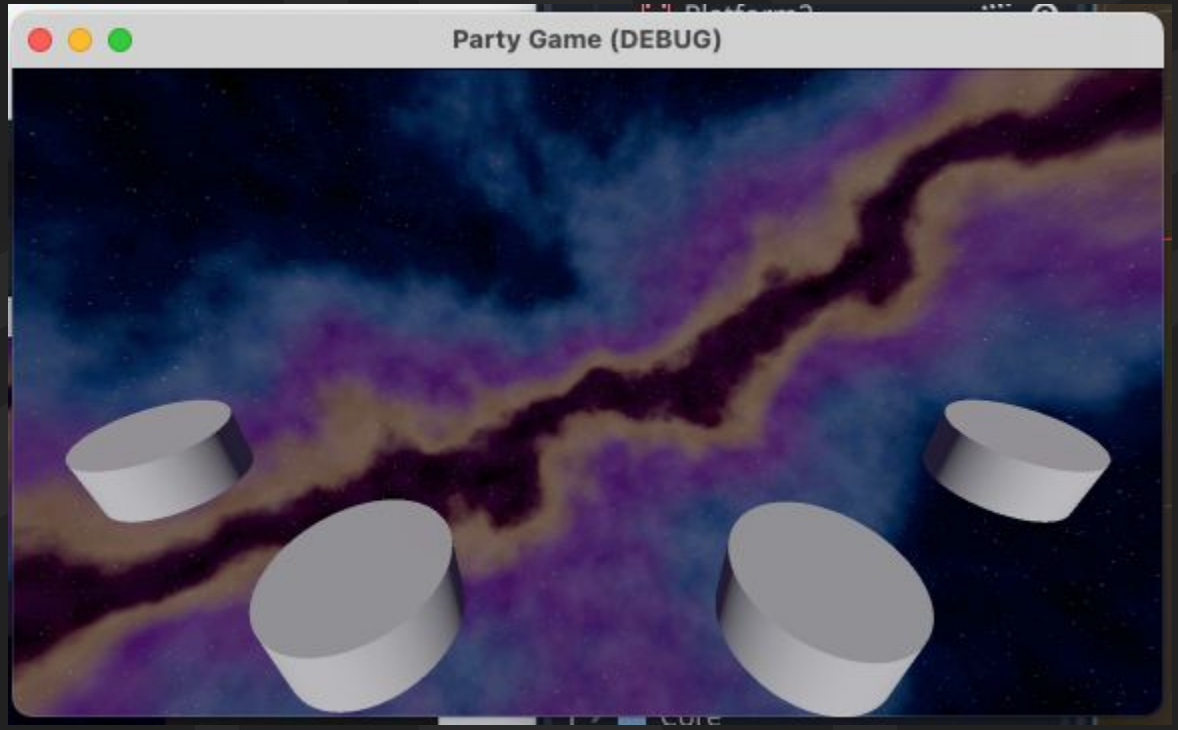
Add Light


Main
DirectionalLight3D
Camera3D
Platforms
Platform1
Platform2
Platform3
Platform4


Transform
Position
x 0 m
y 0 m
z 0 m
Rotation
x -30 °
y 45 °
z 0 °

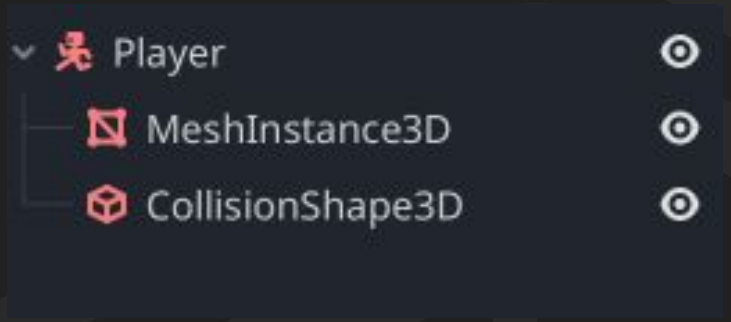

Light3D
Light
Shadow
Enabled
On

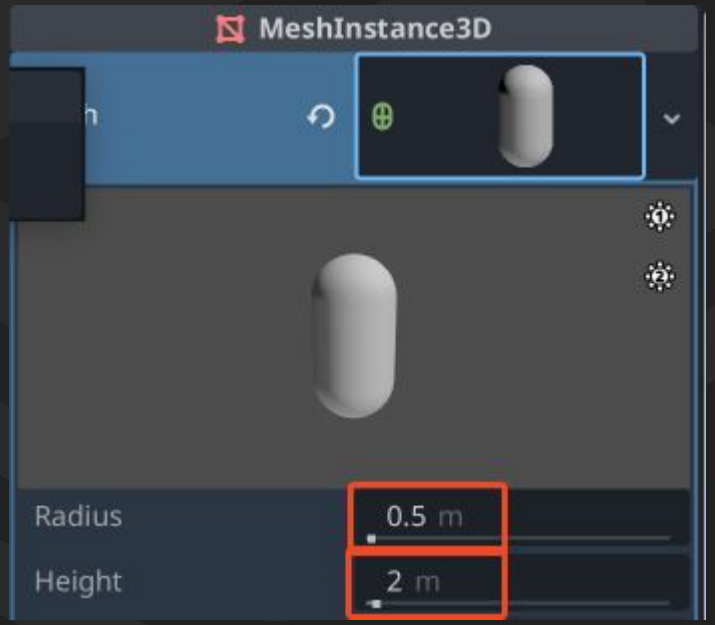
Run the Game

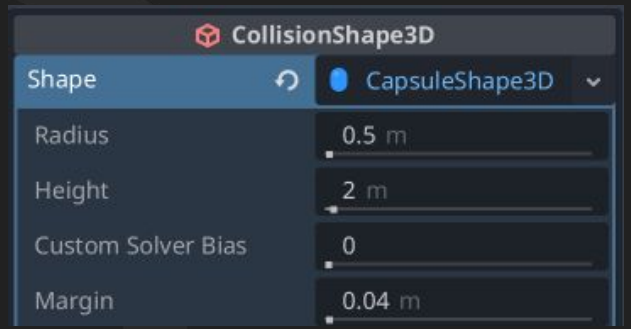


The Player

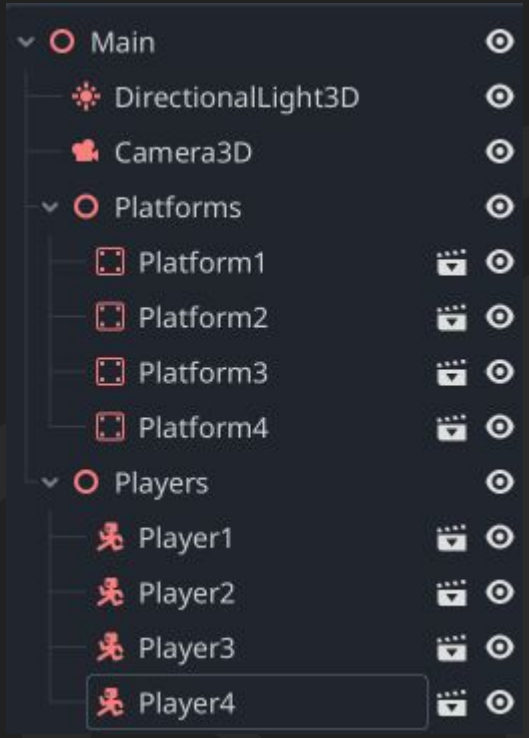
Create the Player Scene

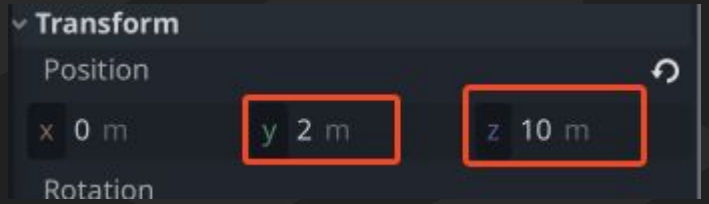


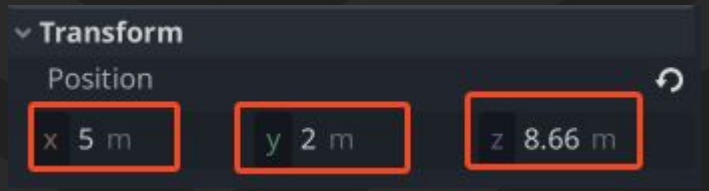


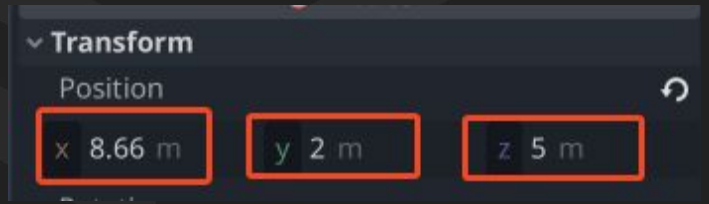


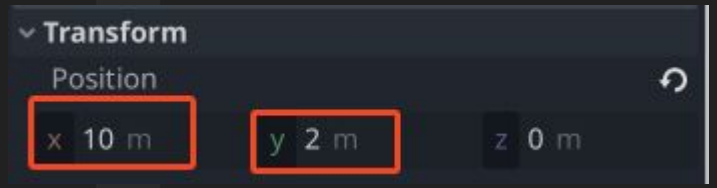
Place the Players









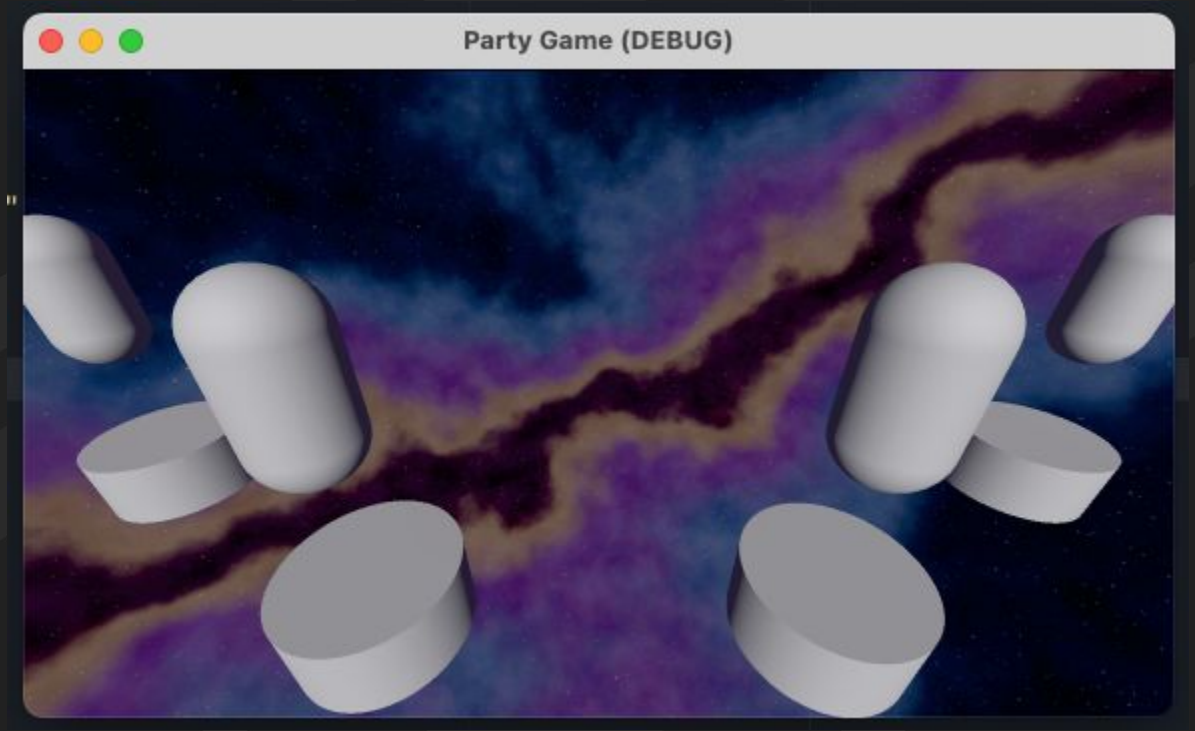


Add Jumping

Code Snippet #1 - player.gd

```gdscript
extends CharacterBody3D


const JUMP_VEL = 4
const GRAVITY_ACC = -4


func _physics_process(delta: float) -> void:
    # Add the gravity.
    if not is_on_floor():
        velocity.y += GRAVITY_ACC * delta

    # Handle jump.
    if Input.is_action_just_pressed("core_player_jump") and is_on_floor():
        velocity.y = JUMP_VEL

    move_and_slide()

```

Run the Game

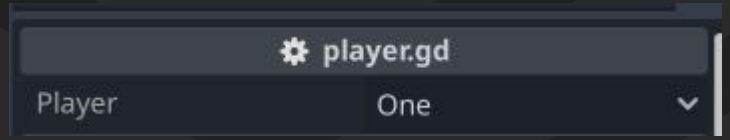


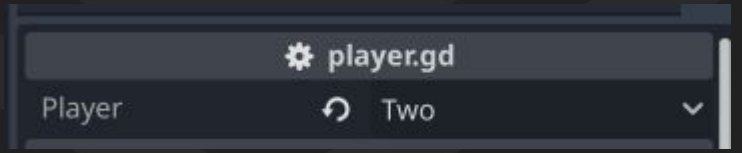
Support Multiple Players

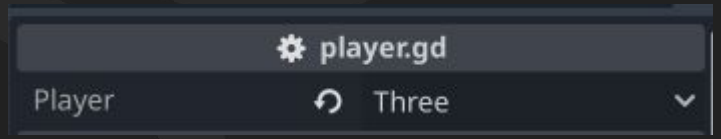
Code Snippet #2 - player.gd

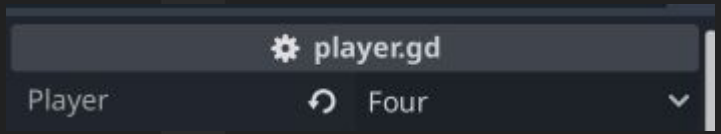
```gdscript
extends CharacterBody3D


const JUMP_VEL = 4
const GRAVITY_ACC = -4

@export var player: Controls.Player


func _physics_process(delta: float) -> void:
	# Add the gravity.
	if not is_on_floor():
		velocity.y += GRAVITY_ACC * delta

	# Handle jump.
	if Controls.is_action_just_pressed(player, "core_player_jump") and is_on_floor():
		velocity.y = JUMP_VEL

	move_and_slide()

```

Set Each Player

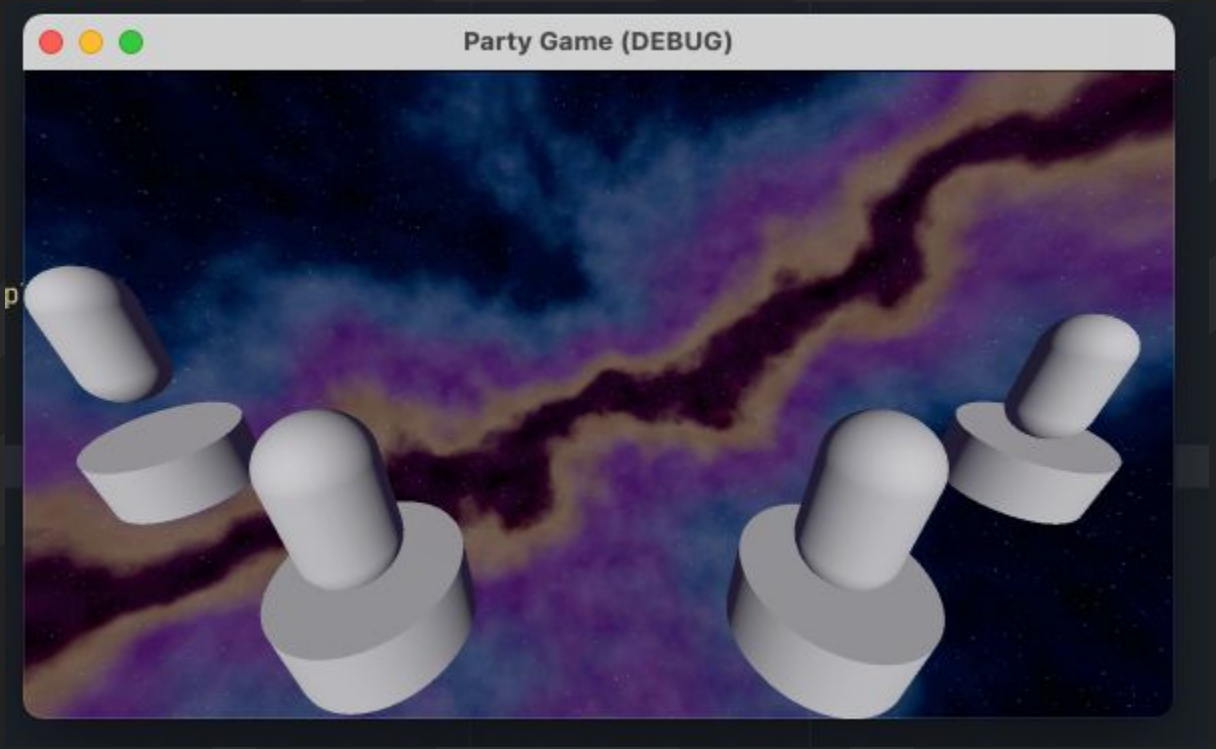






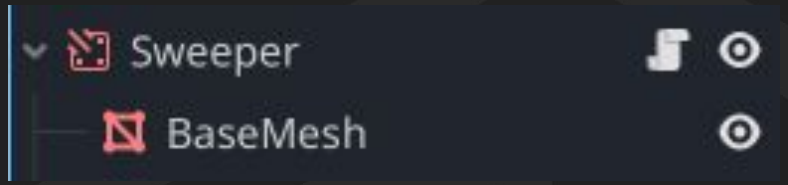


Run the Game

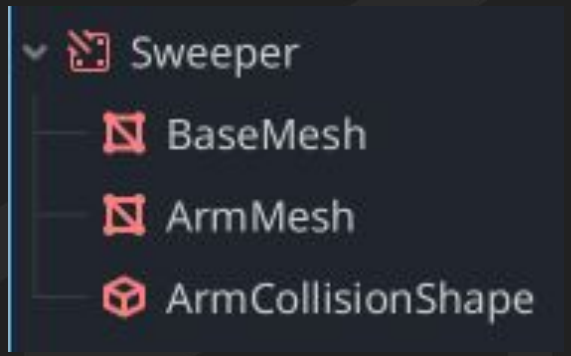


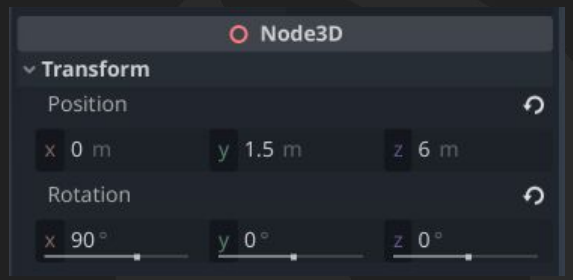
The Sweeper

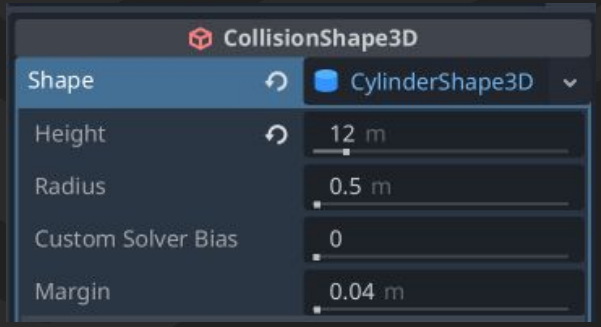
Create the Sweeper Scene



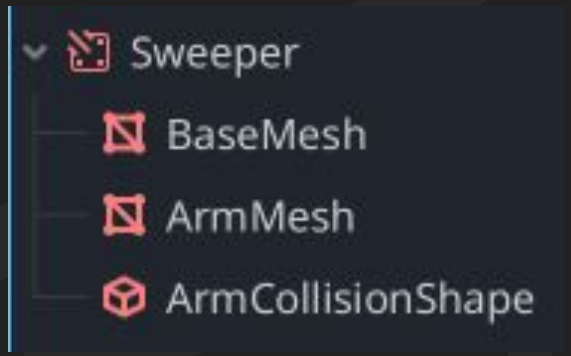
The Sweeper Arm

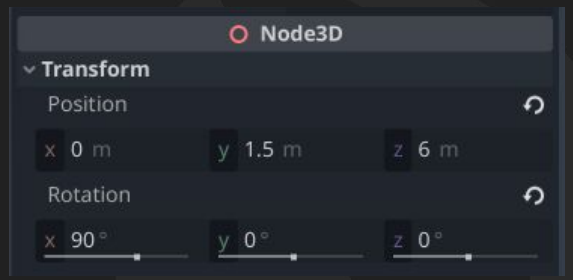


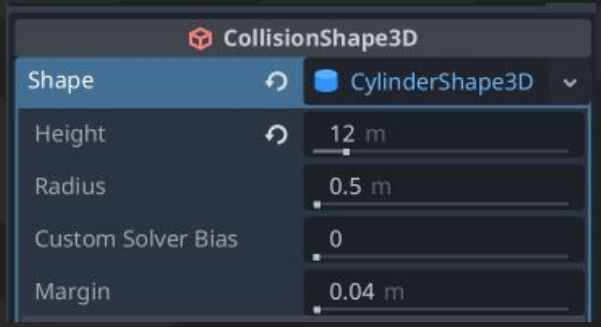




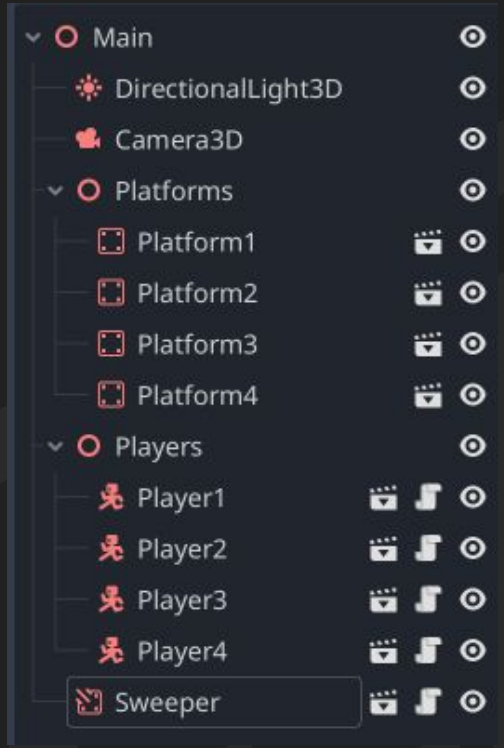
The Sweeper Arm

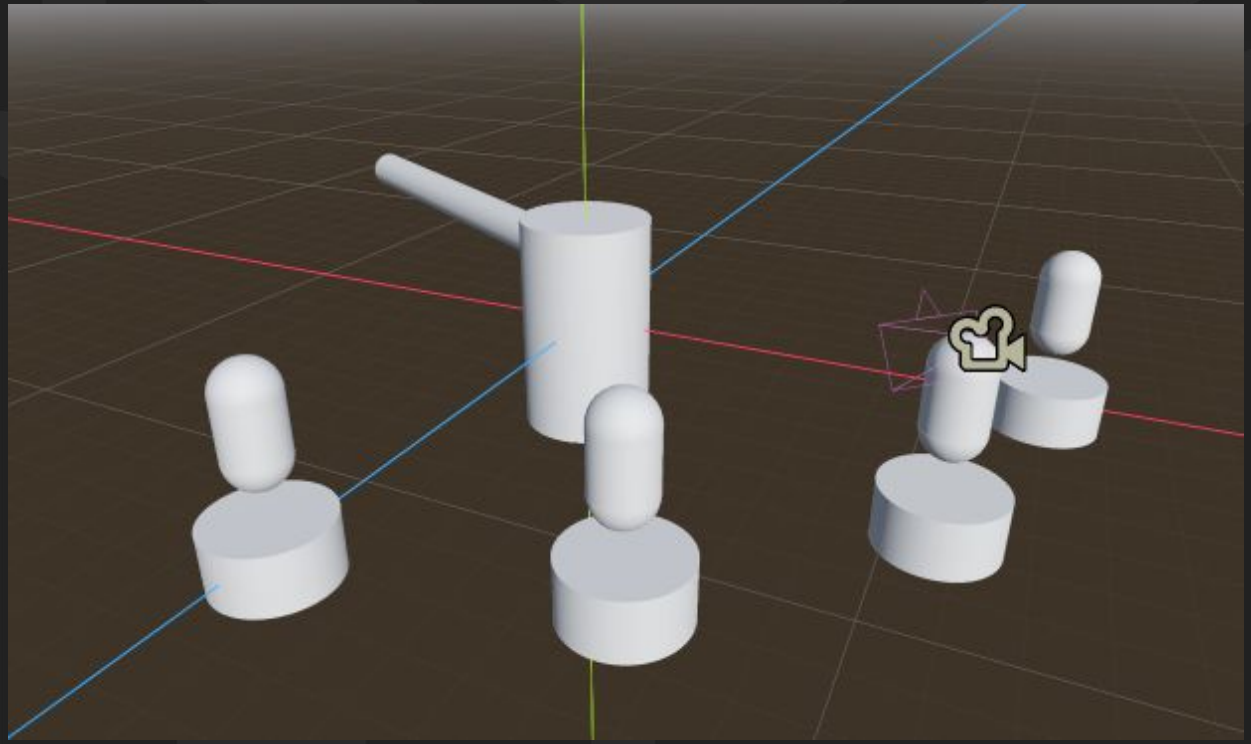






Add the Sweeper the Wipeout Scene



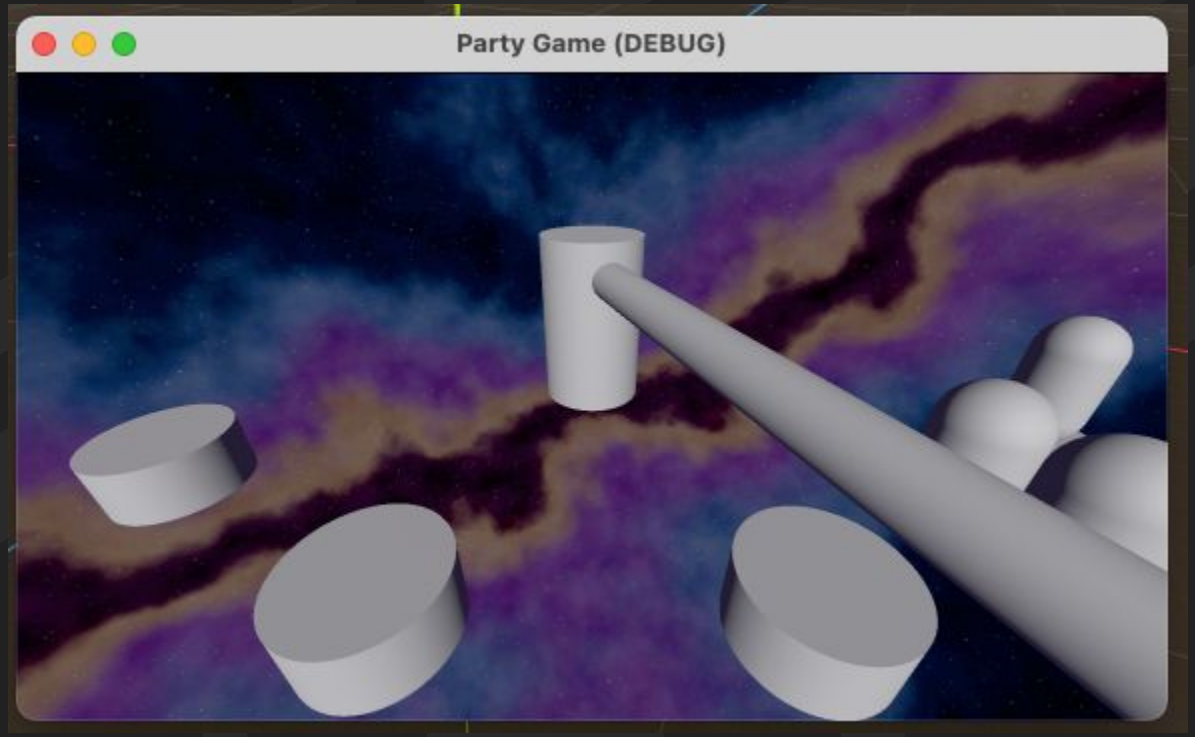


Sweeper Rotation

Code Snippet #3 - sweeper.gd

```
1  class_name WipeoutSweeper
2  extends AnimatableBody3D
3
4  var rotational_vel_deg = 45
5
6
7  func _physics_process(delta: float) -> void:
8      global_rotation_degrees.y += rotational_vel_deg * delta
9
```

Run the Game



Ending the Game

Detect Collisions

Code Snippet #4 - player.gd

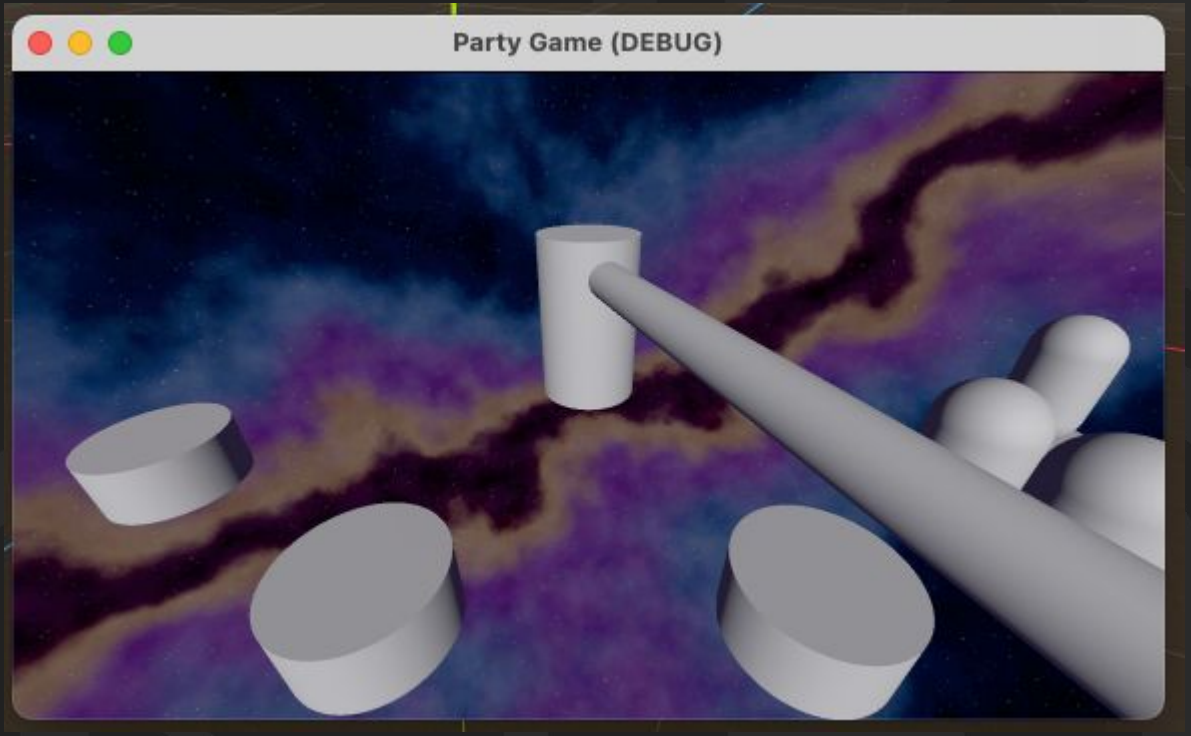
```gdscript
class_name WipeoutPlayer
extends CharacterBody3D

const JUMP_VEL = 4
const GRAVITY_ACC = -4

signal hit_by_sweeper(player: WipeoutPlayer)

@export var player: Controls.Player


func _physics_process(delta: float) -> void:
    # Add the gravity.
    if not is_on_floor():
        velocity.y += GRAVITY_ACC * delta

    # Handle jump.
    if Controls.is_action_just_pressed(player, "core_player_jump") and is_on_floor():
        velocity.y = JUMP_VEL

    move_and_slide()

    for collision_idx in range(get_slide_collision_count()):
        var collision = get_slide_collision(collision_idx)
        if collision.get_collider() is WipeoutSweeper:
            hit_by_sweeper.emit(self)
            $CollisionShape3D.disabled = true

```

Track Hits

Code Snippet #5 - wipeout.gd

```gdscript
extends Node3D

var players: Array[WipeoutPlayer] = []
var player_hit_order: Array[Controls.Player] = []

func _ready() -> void:
    for player in $Players.get_children():
        if player is WipeoutPlayer:
            players.push_back(player)
            player.hit_by_sweeper.connect(_on_player_hit_by_sweeper)


func _process(_delta: float) -> void:
    if players.is_empty():
        SceneManager.exit_minigame(
            player_hit_order.find(Controls.Player.ONE),
            player_hit_order.find(Controls.Player.TWO),
            player_hit_order.find(Controls.Player.THREE),
            player_hit_order.find(Controls.Player.FOUR)
        )


func _on_player_hit_by_sweeper(player: WipeoutPlayer) -> void:
    players.erase(player)
    player_hit_order.push_front(player.player)
```

Run the Game




```
1st: Player 4
2nd: Player 3
3rd: Player 2
4th: Player 1
```

If we have time…

- Better collision detection
- Sweeper rotates faster over time
- More sweepers
- Sweeper moves up and down
- UI?

Sign in here!

DevLUp FSU GBM #2

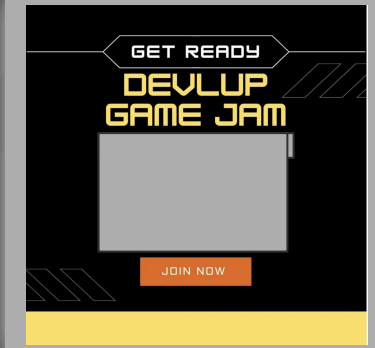
Intro to Godot

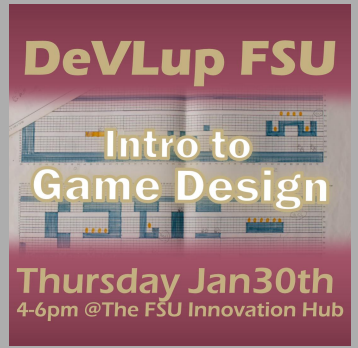
January 30, 2025

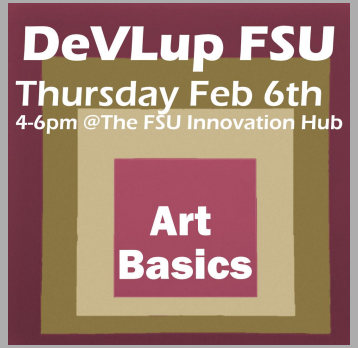
Join our Discord and visit our website!

Join our Discord server using the QR code to stay in touch with the club!

Meet other DevLUp FSU members, “showoff” your work, share resources, take part in game jams, and stay up to date on all our future events!







Discord

fsu.devlup.org