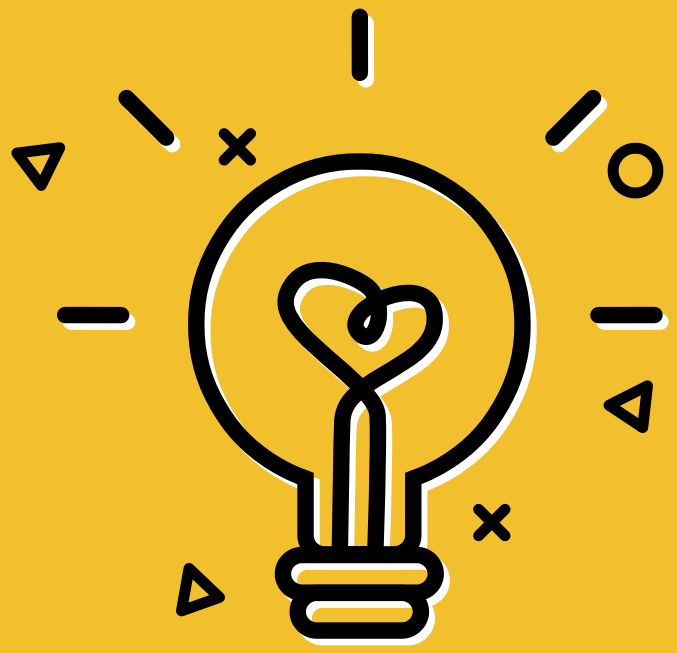


My Game: Weightless

Maxim and Alicia

Hi, my name is Maxim. I am enthusiastic about programming, and have coded a website before this project.

My Project

I made a game about a strongman that could lift weights. My goal was to learn more about coding and create a game, without following a step-by-step tutorial for everything

How did I make the game?

01 Level and Tilemap

Create a level, with a tilemap. Upload your desired tilemap image, and tell Godot what the tile size is. Add a physics layer to your tilemap, and dray the hitbox for the tiles. Also add and occlusion layer to your tilemap, and draw the shapes for that too

02 Player

Add a player, then use the built-in script to add basic movement functionality. Replace the inputs with your own custon inputs. Don’t forget to add a camera, sprite and hitbox as well.

03 Weight and Button

Add a rigid body to act as the weight. Add a sprite and hitbox, and change physics to layer 2. On the tilemap, add the physics layer 2. For the button, use a static object, and do the same thing.

04 Interactions

In the player script add a signal that checks if you’re touching a weight. Connect this to the level. Write a signal that connects back to the player, telling it what weight it is touching.

Weightless – My game

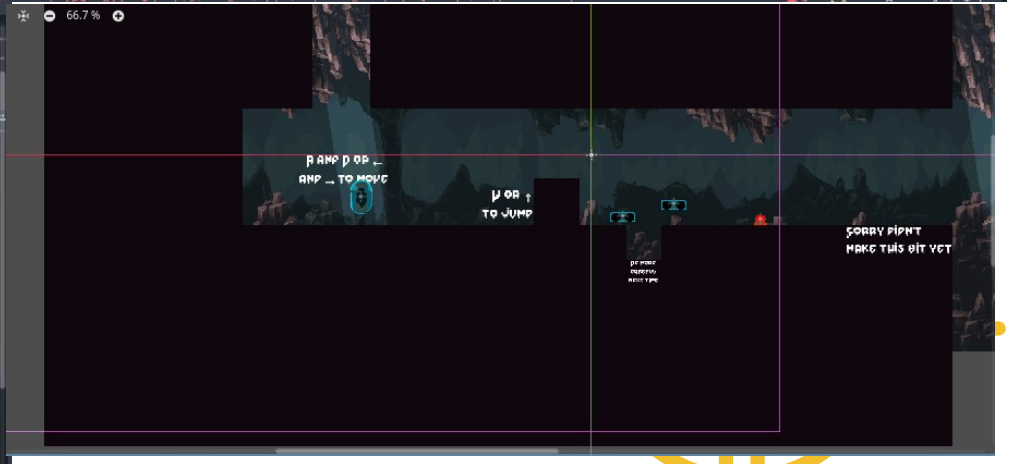
My Code

Screenshots

Here are some screenshots of my code:

```gdscript
if not is_on_floor():
    velocity.y += gravity * delta

# Handle Jump.
if Input.is_action_pressed("jump") and is_on_floor():
    velocity.y = JUMP_VELOCITY

var direction = Input.get_axis("left", "right")
if direction:
    velocity.x = direction * SPEED
else:
    velocity.x = move_toward(velocity.x, 0, SPEED)

if Input.is_action_just_pressed("intereact"):
    emit_signal("pick_check")

move_and_slide()


func _on_level_player_pick_return(ok_status, weight):
    print(ok_status, ' ', weight)
    if weight != null and not weight amount >= 2:
```

```gdscript
signal player_pick_return(ok_status : bool, weight)


# Called when the node enters the scene tree for the first time.
func _ready():
    pass


# Called every frame. 'delta' is the elapsed time since the previous frame.
func _process(_delta):
    #print(get_children())
    weights = []
    for object in get_children():
        #print(object)
        if "Weight" in object.name:
            #print("yup, weight")
            weights.append(object)

func _on_player_pick_check():
    var found = false
    #print(weights)
    for weight in weights:
```



Weightless – My game

My Code

Explanation

The code is mostly made using the godot built-in code, but I also added some code to use my own inputs and pick up and drop weights. The code for the weights checks if the player is touching a weight. If the player is touching a weight, it picks it up. If the player is not touching a weight, but holding one, the player drops the weight. If the player does not have a weight and is not touching one, the player does nothing

```
func _on_player_pick_check():
    var found = false
    #print(weights)
    for weight in weights:
        #print("doing a 4")
        if abs($Player.position.x - weight.position.x) <= 16:
            emit_signal("player_pick_return", true, weight)
            found = true
    if not found:
        emit_signal("player_pick_return", false, null)
```

```
func _on_level_player_pick_return(ok_status, weight):
    print(ok_status)
    if ok_status and not weight_amount >= 2:
        level.remove_child(weight)
        weight_amount += 1
    elif weight_amount == 1:
        weight_instance.position = position
        weight_amount -= 1
        level.add_child(weight_instance)
```

Problems and Solutions

1. Weight Intereactions

I couldn’t get the player to be able to detect if it is touching a weight, but I figured out that I could connect a signal from the player to the level. I made all the logic for the weights in the level

2. Lighting Problems

I had trouble working out the directional light, but I used an online tutorial that helped mr understand how to use the Godot lighting system

3. Version Control

Learning how to use git was hard, but my dad helped me so that I could:

- Commit the files
- Save my work properly
- Access commit history

Reflections

Here are some of my reflections:

I didn’t choose my project quickly enough so I didn’t have time to work on the game. I took about a month to choose it when I should have taken about a week or so. This had the consequences of delaying my project, so I didn’t have time to work on it. If I chose my project quickly, I would be able to add more features and focus on making it look nice.

Also, when I didn’t have a plan, I couldn’t make my project efficently because I didn’t know what I had to do next, so I couldn’t make my project expecting that feature. But when I wrote a plan, I could work on the project much faster. When I didn’t have a plan, I wasn’t sure what I was doing.

Next time, I will use the Godot game engine again, as it comes with a lot of built-in functionality, so I don’t have to write much code. It makes making the game easier, because of it’s beginner-freindly interface and easy-to-use documentation.

Yellow Idea Template for LibreOffice Impress.
Credit by : @ealita.id, 2020.

THANK YOU