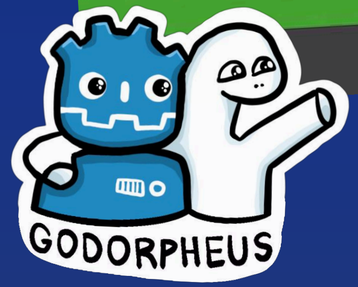
HACK CLUB

JUMP START!

# GODOT PLATFORMER GAME

EASY, BEGINNER FRIENDLY

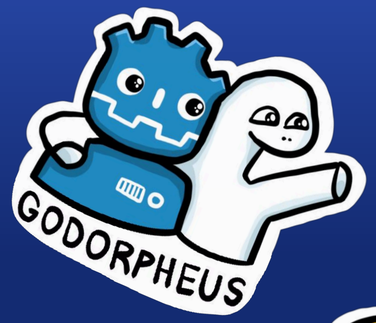
with Estella!



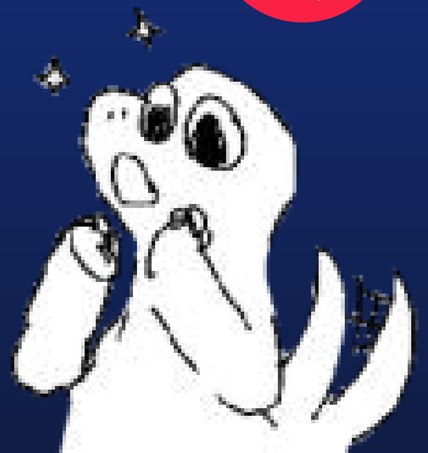
# OVERVIEW

You'll build a 2D game with a player that can run and jump around. You will use Godot, a game engine.

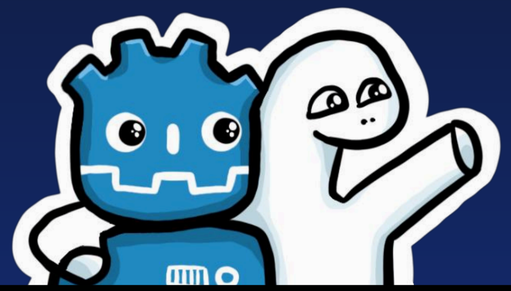
# STEPS




1. Setup Godot
2. Make a Github repo
3. Build a basic platformer
4. Change your game
5. Publish on Itch.io
6. Submit your game



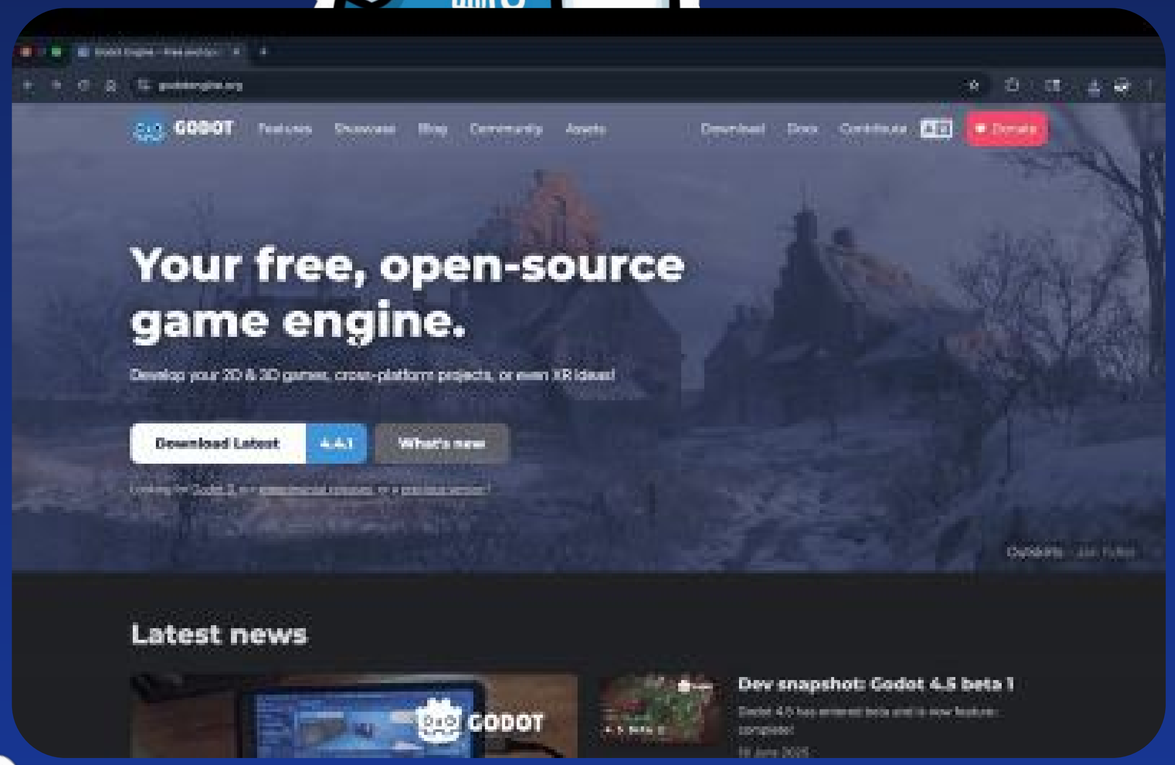
# SETUP GODOT



1. Godot is a free game-making tool. Go to godotengine.org to install it.

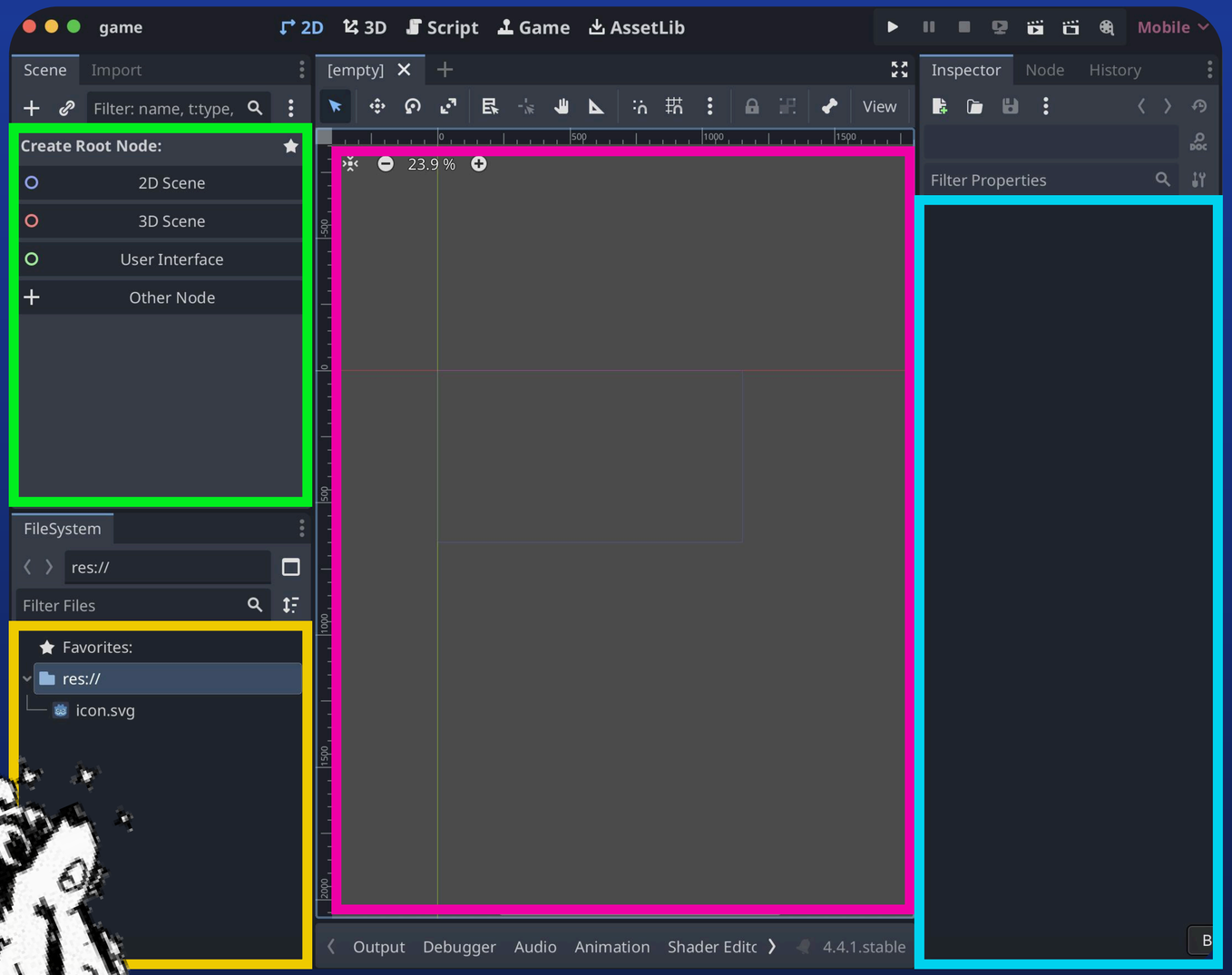
2. If you cannot install it, you can also use the web editor editor.godotengine.org.

3. Open Godot, click Create New Project, name your game, select Mobile render, and click Create.



# GODOT LAYOUT

# IMPORTANT AREAS




## Main viewport

See what your game looks like. Purple rectangle is the gameview.

## Scene and nodes

Nodes and scenes are game parts. They will be listed here.

## FileSystem

Find all your files and folders. Images, sounds, code, and more.

## Inspector

With a node selected, edit its properties in here.

# MAKE GITHUB REPO

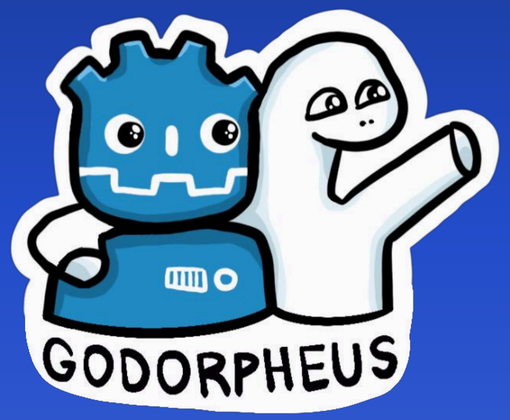
1. Go to github.com and make an account.
2. Create a new repository. Name it, make it public, choose no README, no license, and no .gitignore.
3. Open terminal and cd into your Godot game folder.
4. Run git init.
5. Run git branch -M main.
6. Copy your repo's web URL from Github, then run git remote add origin <your-link>.
7. Run git add . .
8. Run git commit -m “init”.
9. Run git push -u origin main.

# IMPORTANT COMMANDS

## GODOT

Save your project often using cmd/ctrl+s.

Play your game with cmd/ctrl+b.

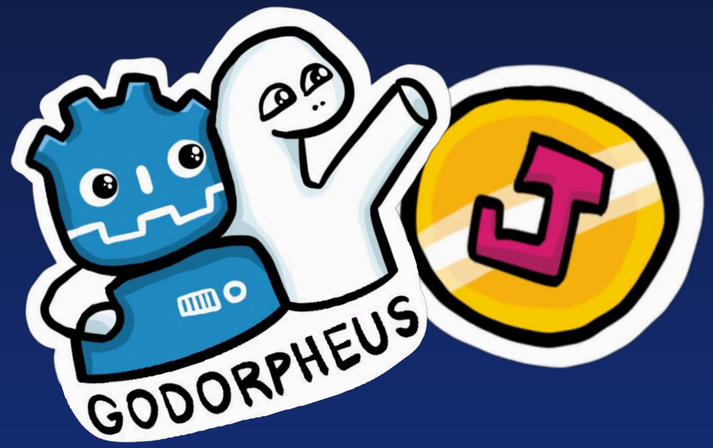



## GITHUB

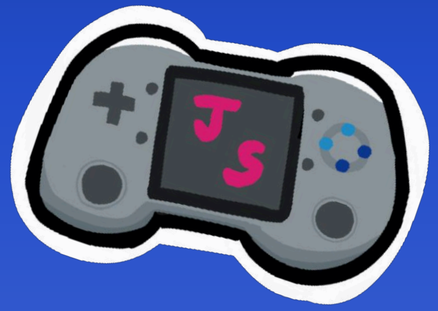
To update your repository:

```
git add .
git commit -m "things-added"
git push
```

"Things-added" is your commit message. It should be descriptive. Update often.



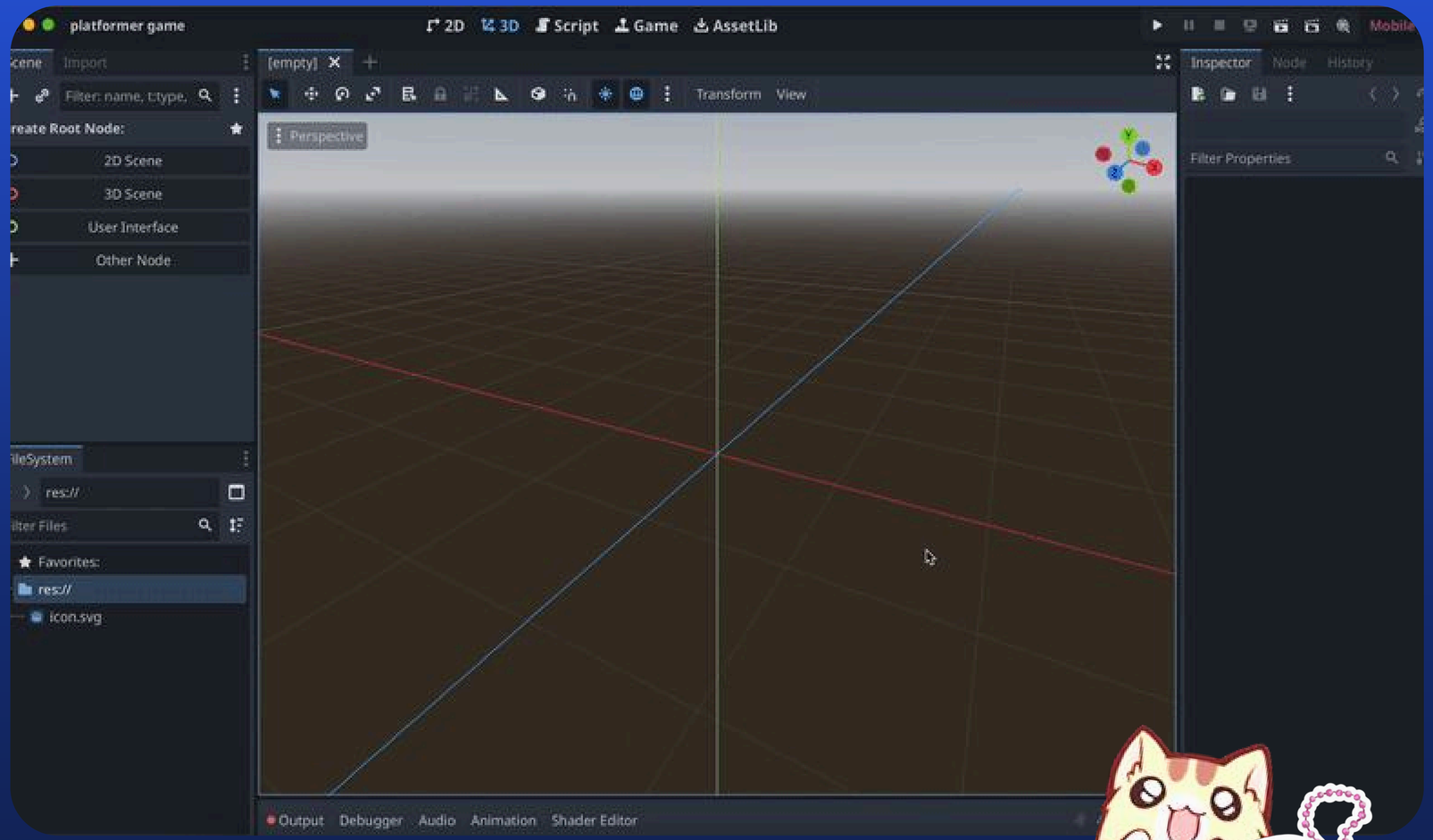
# TIME TO START BUILDING YOUR GAME!!



# CREATE SCENE

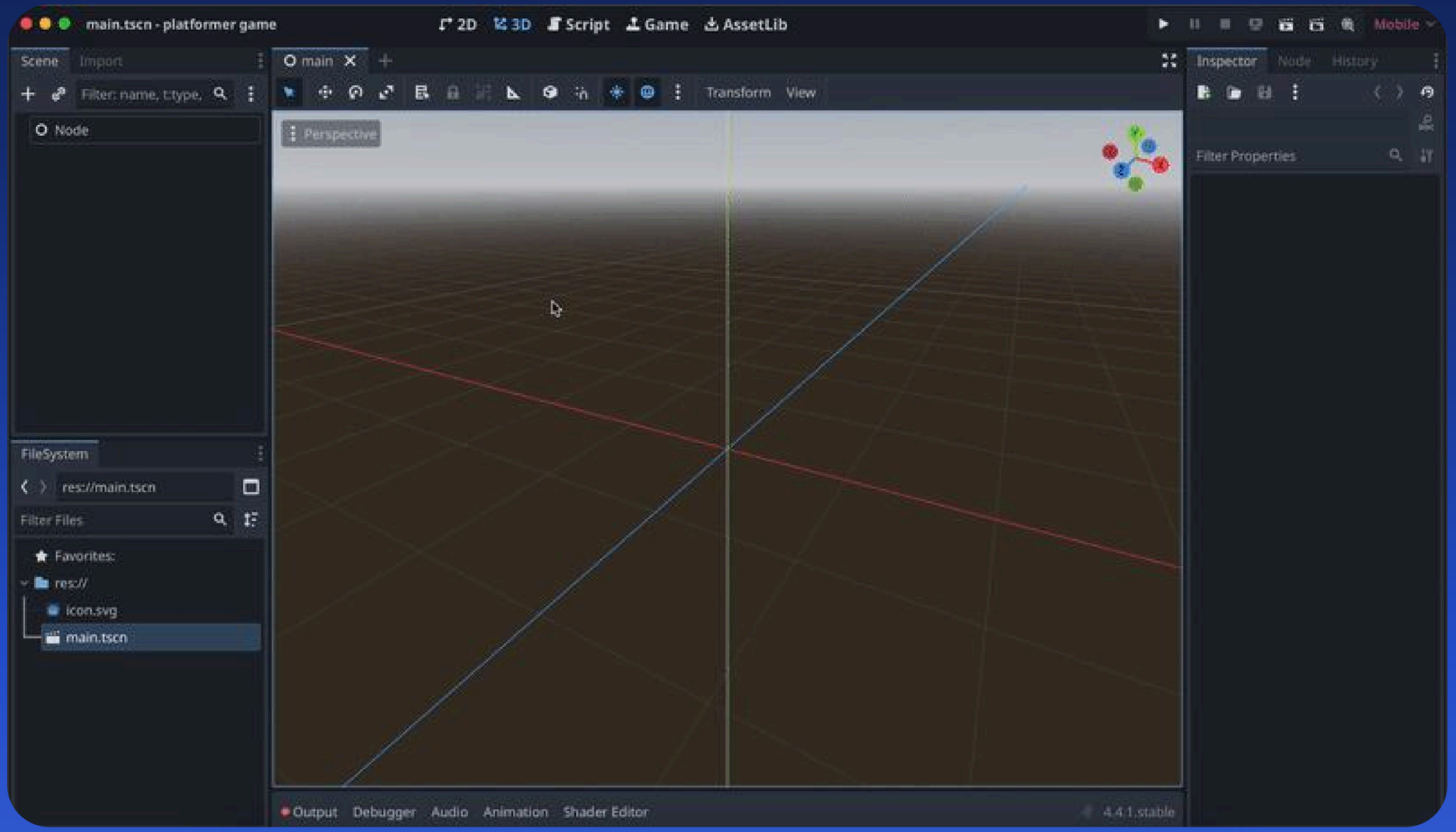
In the **Scene** section, choose **Other Node** and select **Node** (first one).

**Save** your game, and call it **main.tscn**. You will see it show up in the **FileSystem**.



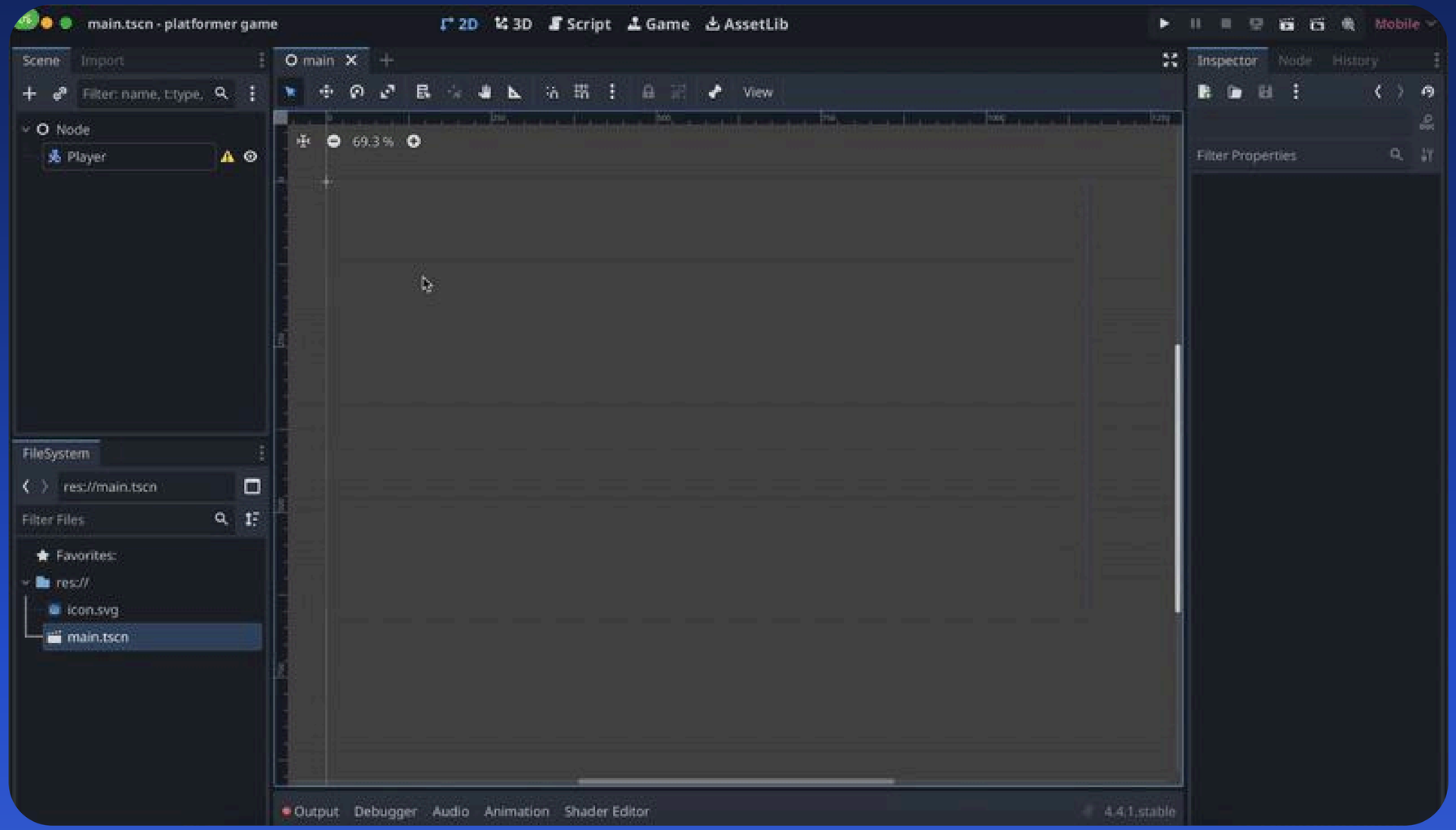
# CREATE PLAYER (1)

Under your Node, create a new node with cmd/ctrl+a or the plus. Select Character Body2D. Retitle it Player.



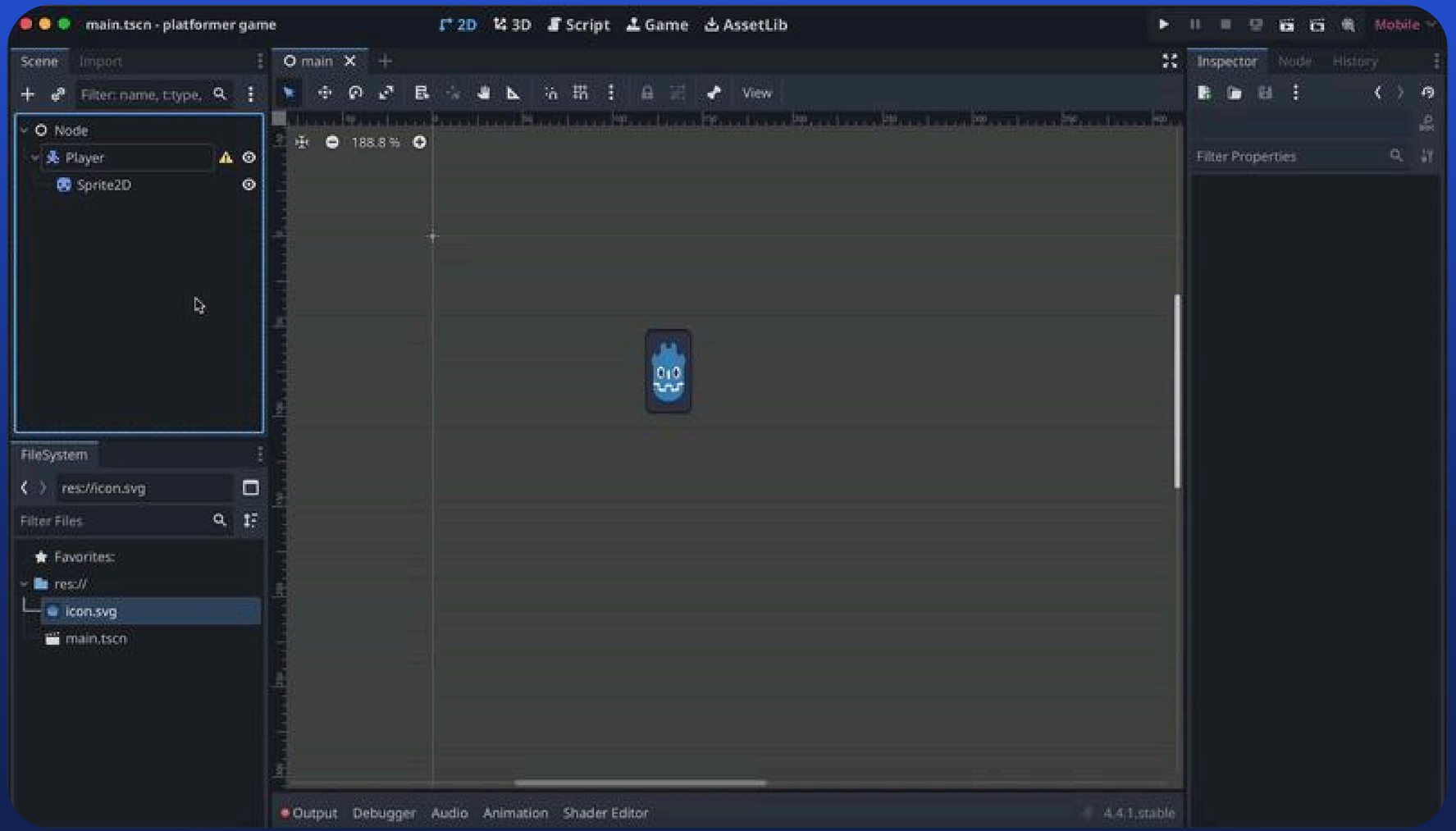
# CREATE PLAYER (2)

Under Player, create a Sprite2D node. Drag icon.svg from FileSystem into the empty box of Texture in Inspector. Resize in viewport.



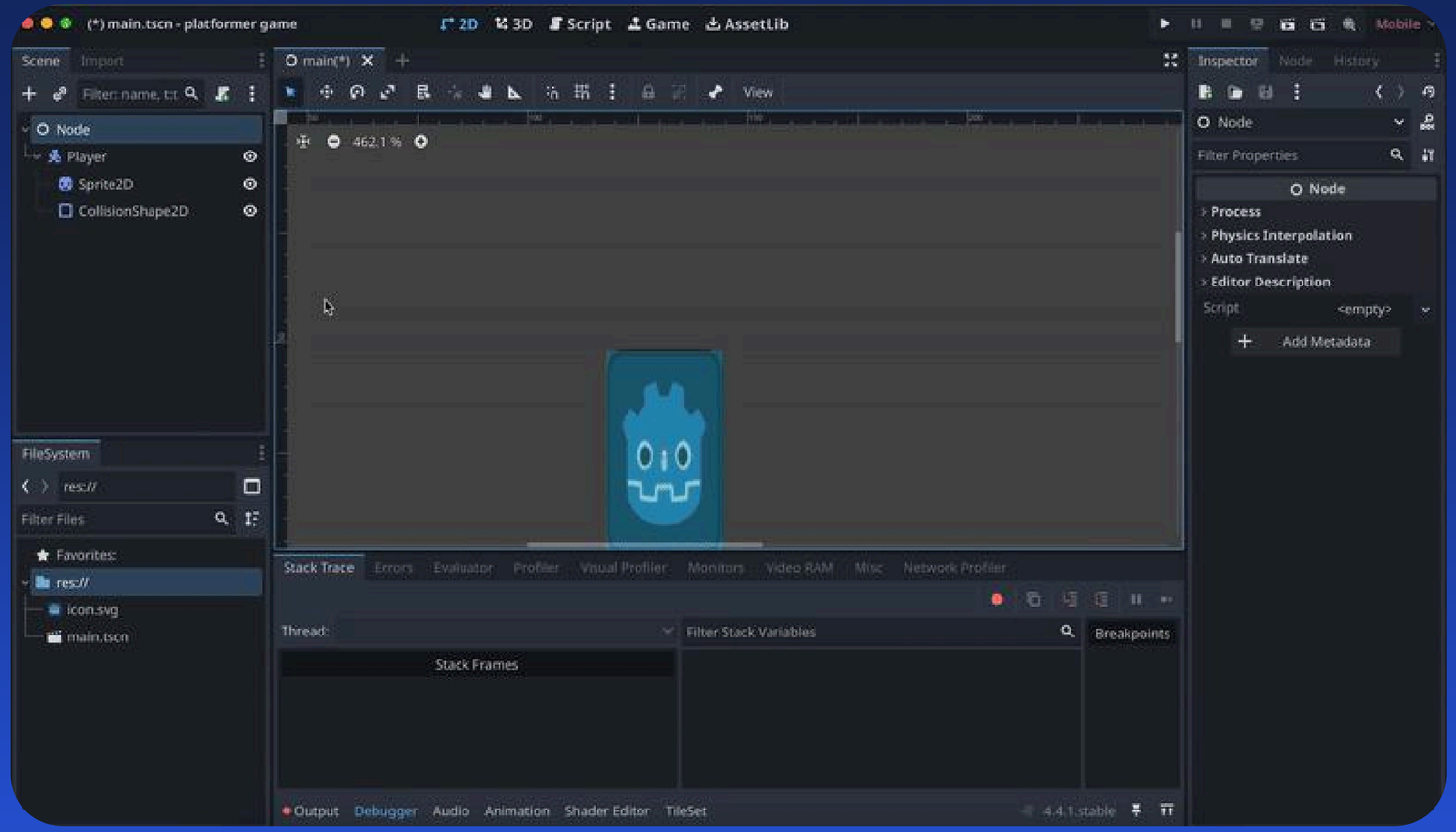
# ADD COLLSION BOX

Under Player, create a CollisionShape2D node. In Inspector, select Rectangle for Shape. Resize the blue box to cover the player. This is for physics.



# CREATE A TILESET

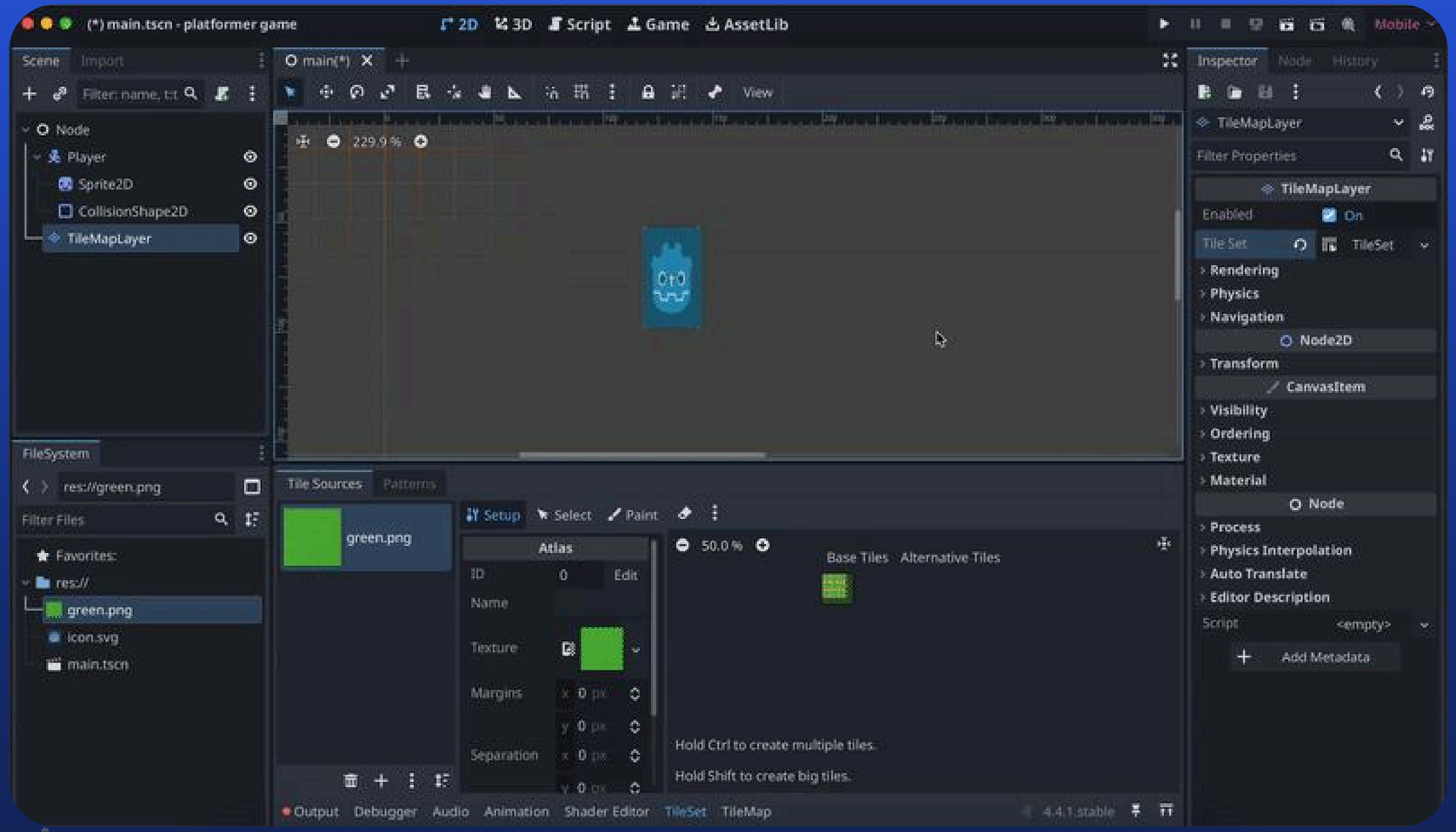
Under Player, create a TileMapLayer node. In Inspector, select New Tileset for Tile Set. In the bottom bar, go to TileSet. Import an image to FileSystem, then drag it into TileSet area. Select Yes to modifications.




Tilemaps are used to easily make grid-based maps.

# ADD COLLISION FOR TILESET

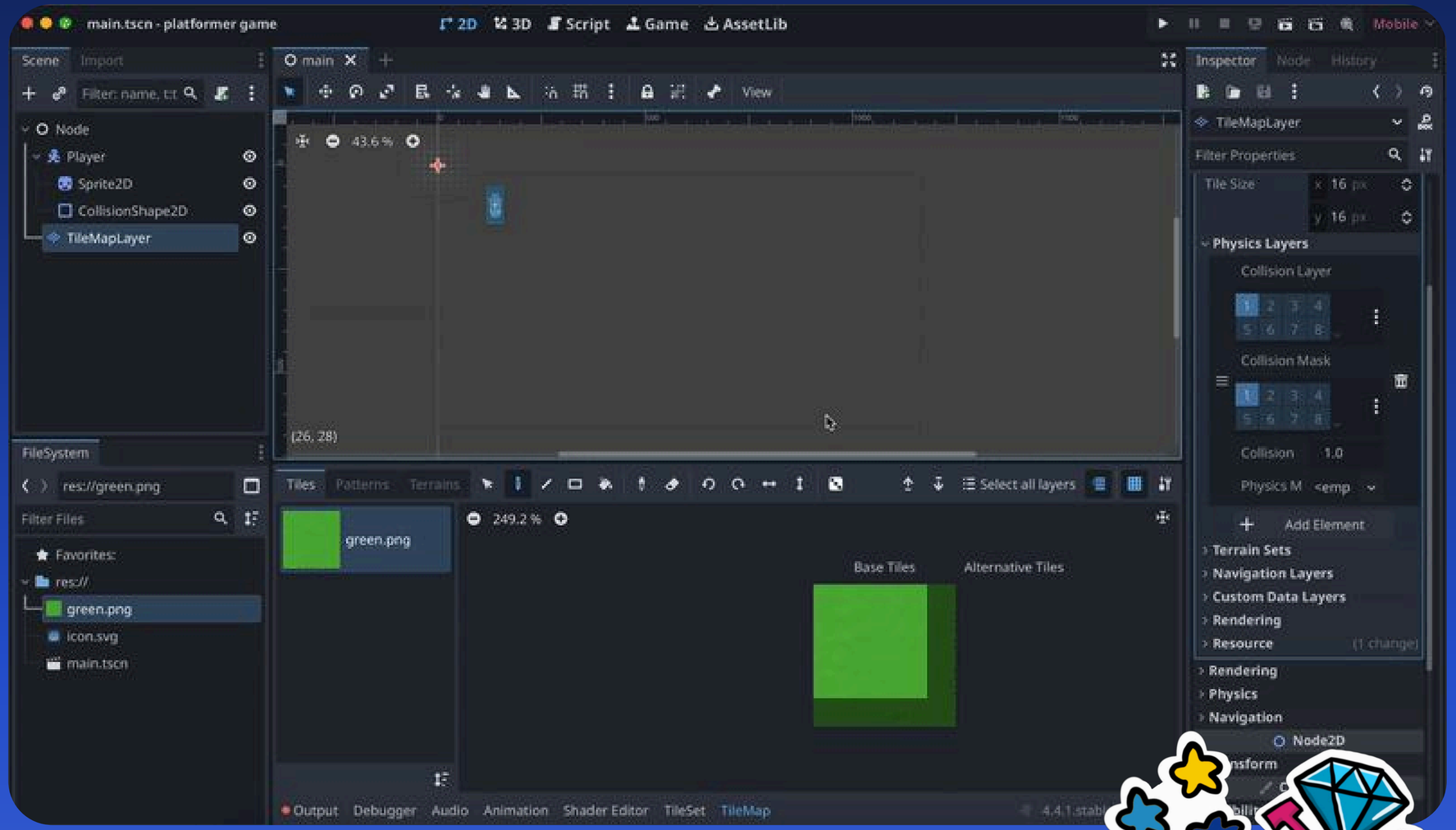
Open TileSet in Inspector. In Physics Layers, click Add Element. In TileSet, change to Paint and select Physics Layer 0 as Paint Property. Select all tiles blue.



Just like the Player, each block of the Tileset needs a collision box too.

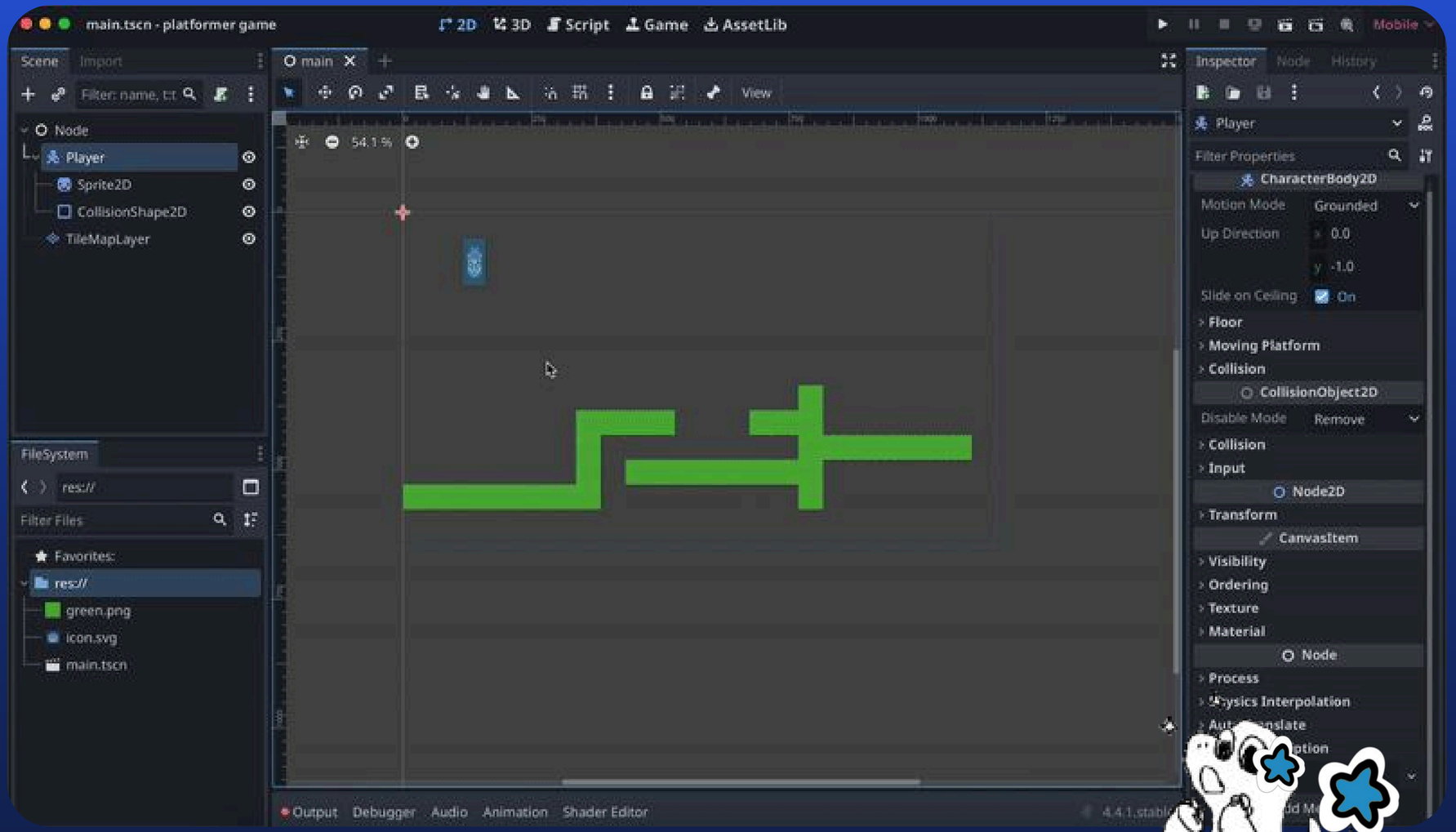
# DRAW A MAP

In the bottom area TileMap, use the Pencil tool and select one tile from the right. Draw a map. Aadjust the size of each tile in Inspector → Transform → Scale.



# CREATE PLAYER SCRIPT

Select Player node, and make a script by clicking the Paper with Green Plus. Choose GDScript, check Template and choose the Basic Movement one.



# UNDERSTAND TEMPLATE CODE

Run your game with ctrl/cmd+b or play button. It works, but the template code is the minumum and we will improve it, but first, read through to understand the code.

```
extends CharacterBody2D

const SPEED = 300.0
const JUMP_VELOCITY = -400.0

func _physics_process(delta: float) -> void:

    if not is_on_floor():
        velocity += get_gravity() * delta

    if Input.is_action_just_pressed("ui_accept") and is_on_floor():
        velocity.y = JUMP_VELOCITY

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

    move_and_slide()
```

this tells Godot that this script is connected to the **CharacterBody2D** node

**const** means the value doesn't change in the game **SPEED** is how fast the player goes, **JUMP_VELOCITY** is the jump height. in Godot, negative numbers go up and 0 is the top of the screen.

the **physics_process** function runs many many times every second and **delta** is the time since the last frame, it helps keep the animation smooth

if the player is **not on the floor**, then they are in the air and should fall because of gravity. **get_gravity** is the gravity strength, and **velocity** makes it so the player falls faster over time

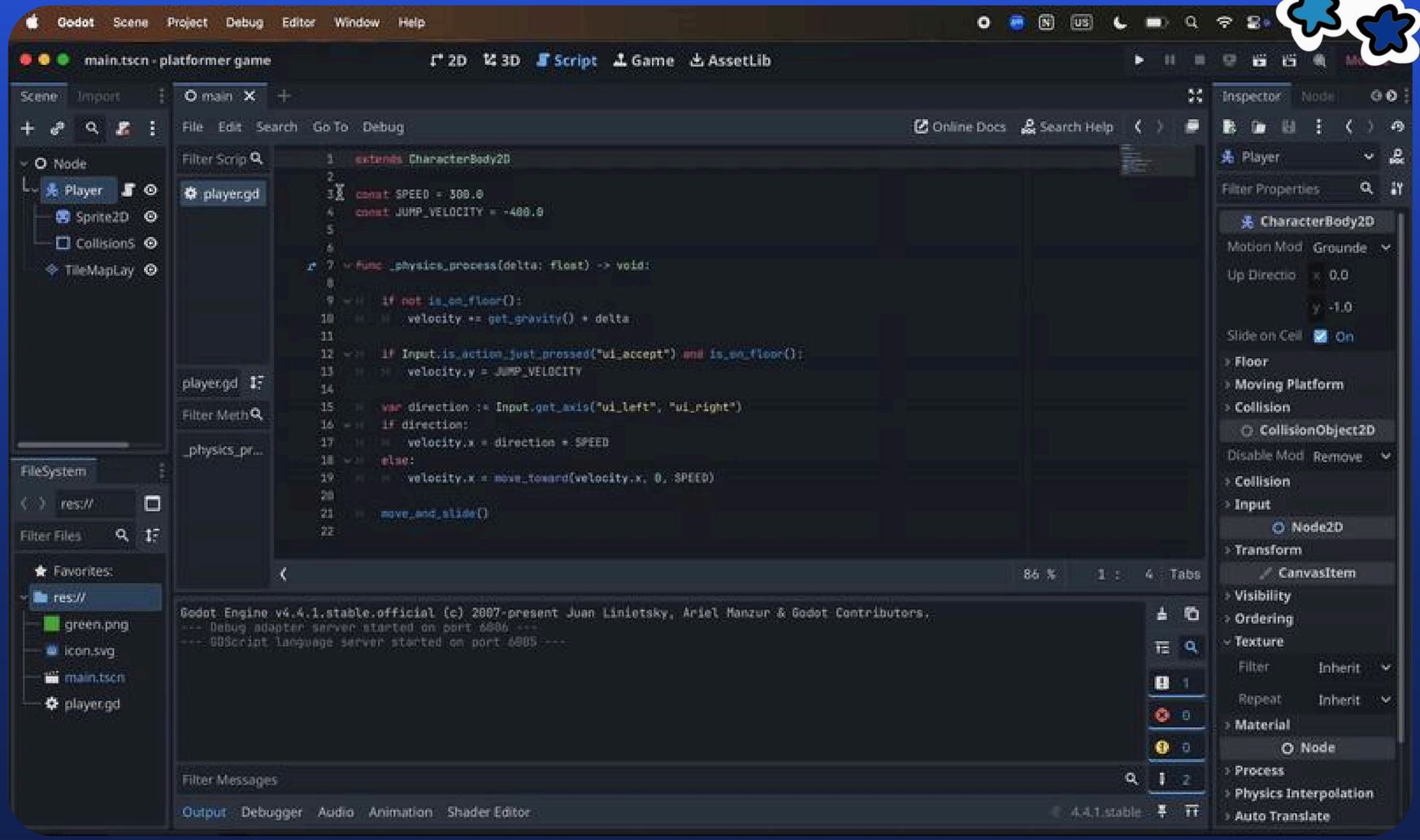
**ui_accept** is space or enter and if the player is **on the floor**, then sets **velocity** to go up, so the player can jump

**Input.get_axis** returns -1 when the **left** key is pressed and 1 when the **right** key is pressed. **velocity.x** makes the player move in the correct **direction** at **speed.** if they are not pressing right or left, then the **velocity** is slowly moved towards **0** to a stop.

this is the final important function which applies all of this to the player!!

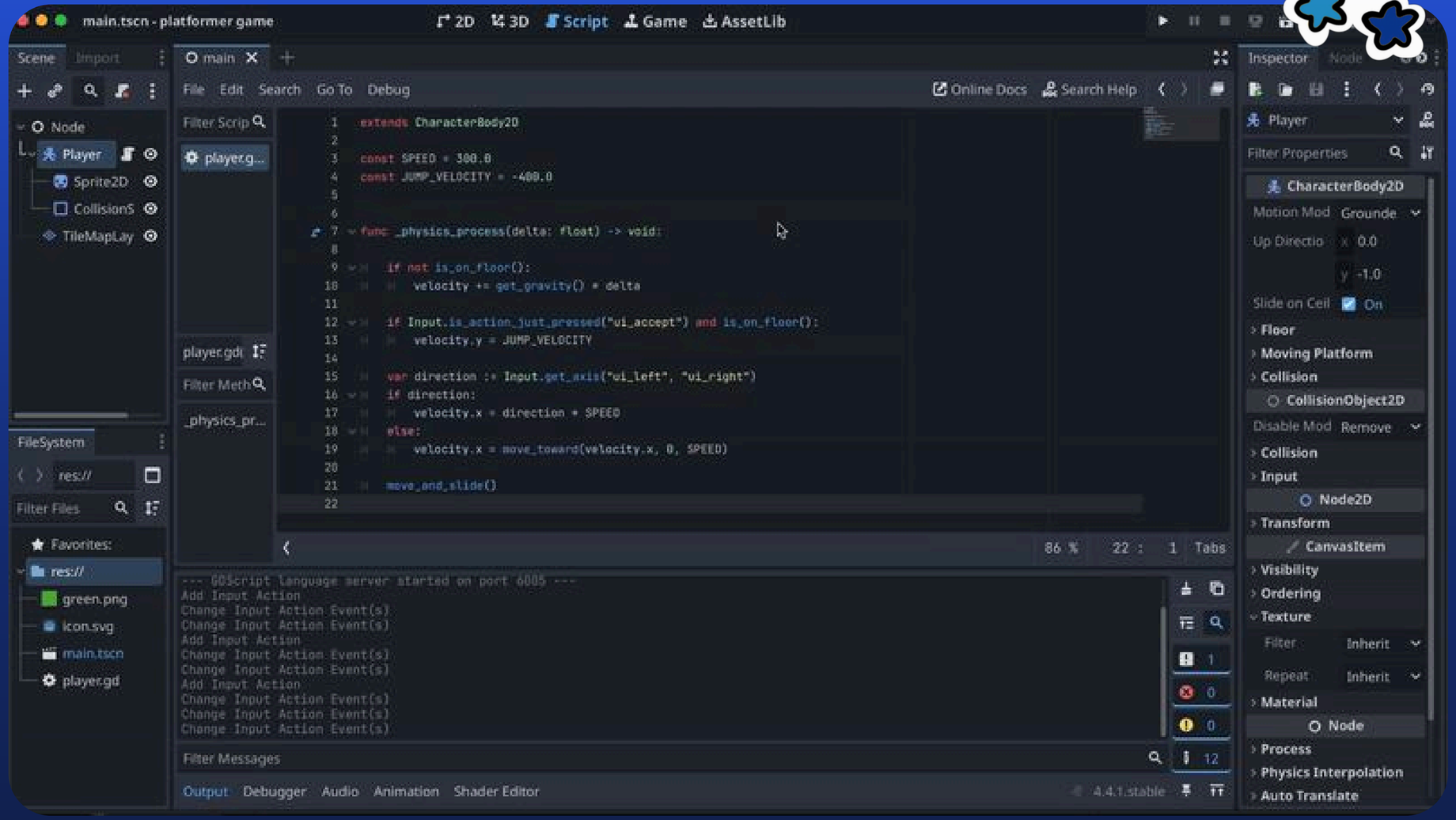
# MAKE INPUT MAP (1)

Go to Projects → Project Settings → Input Map. Create an action for jump, right, and left. Add keys to each one (WASD, arrows, space).



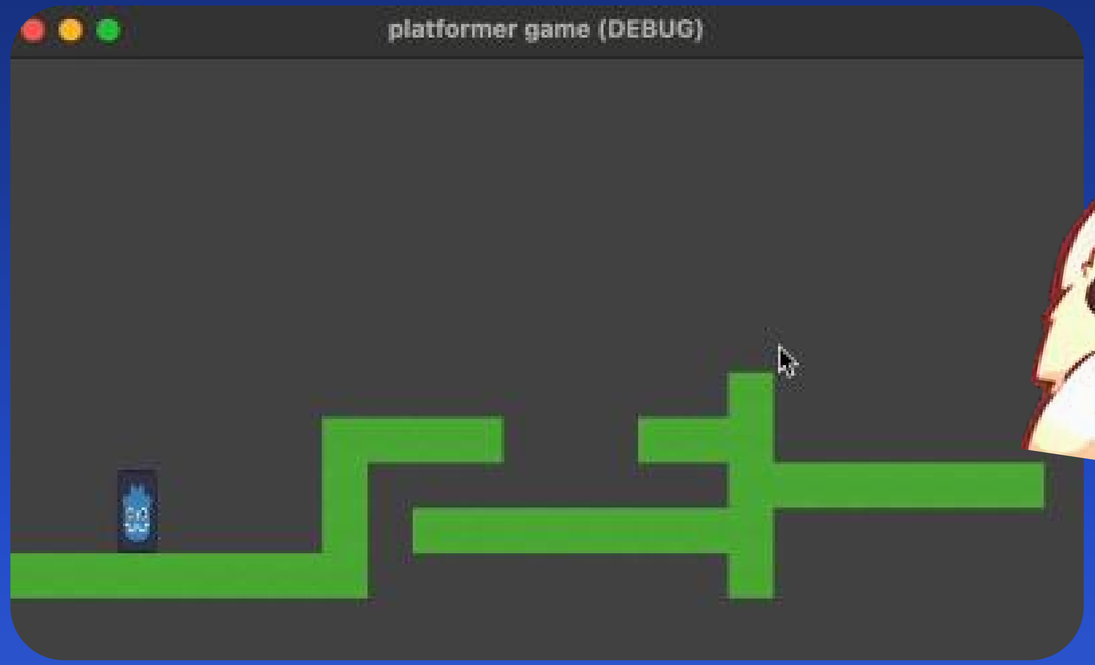
# MAKE INPUT MAP (2)

In player.gd script, replace “ui_accept”, “ui_right” “ui_left” with “jump”, “right” “left.

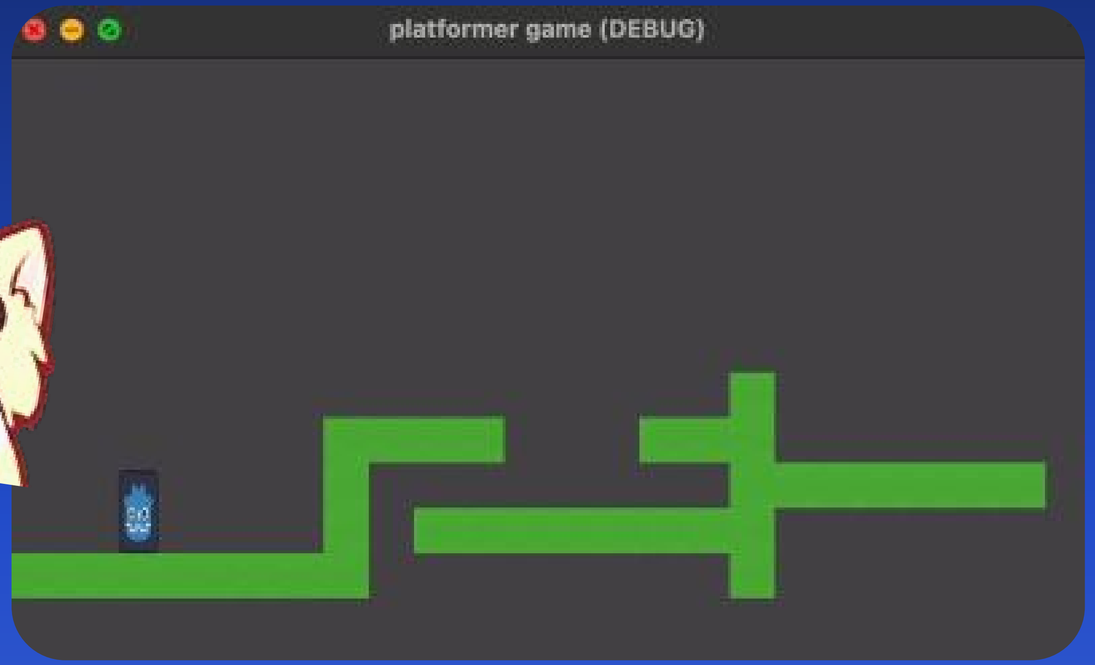


# TWEAK MOVEMENT IN CODE

Change SPEED to 900, and JUMP_VELOCITY to -900. Go to Project → Project Settings, search for Gravity and make it 2500. Use ctrl/cmd+b to run your game and test.



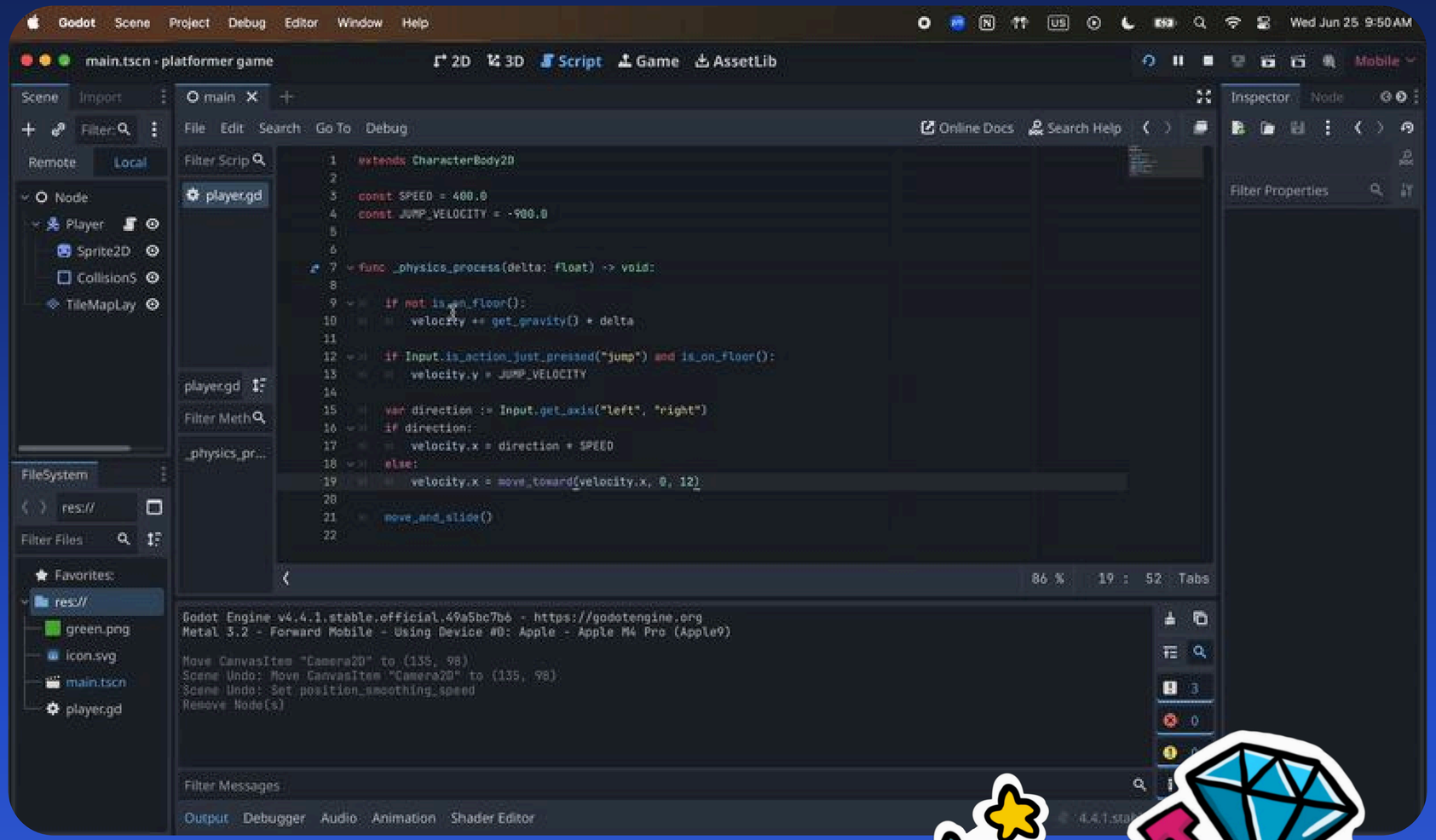

BEFORE CHANGES



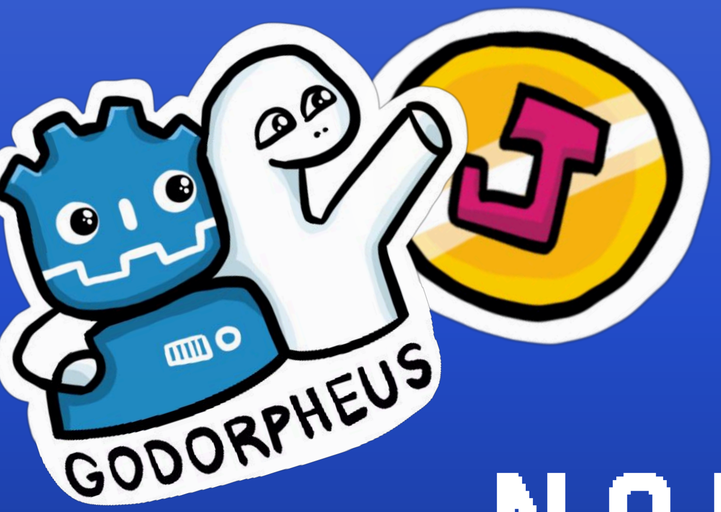

AFTER CHANGES

# ADD CAMERA

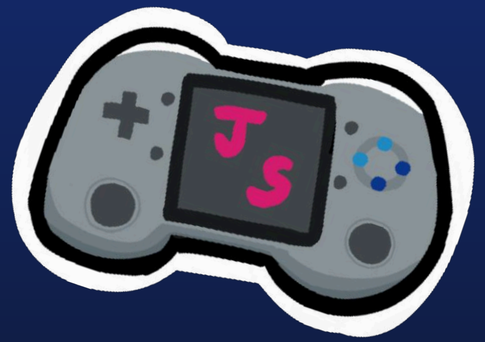
Under Player, add a Camera2D node. In viewport, drag the camera's center to the player. For a cool lag effect, enable Position Smoothing in Inspector and set to 6.



# CONGRATS! YOU HAVE A BASIC GAME NOW...



# NOW IT'S TIME TO CUSTOMIZE



# MAKE YOUR GAME UNIQUE

Time to customize your game! You have all the base pieces. Pick a fun theme for your game. Who is your player? Where are they? Where are they going?

EXAMPLE: a space game where you are an astronaut collecting spaceship parts and avoiding aliens

**FIRST, CHANGE**

your player's image
the background
the tilset

*You can draw these or find assets online.

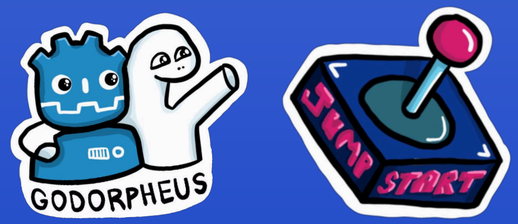
**THEN, YOU CAN ADD**

collectables
enemies
a score system
sound effects / music
anything else you want!

*There are tons of great tutorials online to help you!

Find links for good tutorials on jumpstart.hackclub.com!

# EXPORT YOUR GAME




1. In Godot, in the top bar go to Project → Export → Add → Web.

2. Then, Manage Export Templates → Download and Install.

3. Once it's done, go back to Export, select the Web preset and click Export Project.

4. Make a folder called Exports and save the index.html to it.

5. In File Explorer, compress all the contents in Exports folder into a .zip. Name it your-game-web.zip.

# PUBLISH ON ITCH.IO

1. Itch.io is a website to find games and publish your own games for others to play. Go to itch.io and make an account.

2. In Itch.io's dashboard, click Create New Project. Title it, add a description. For Kind of Project, choose HTML.

3. Upload the .zip file you made. Check the This file will be played in the browser box.

4. Change Viewport Dimensions to the ones in Godot, Project → Project Settings → Window.

# P2: PUBLISH ON ITCH.IO

5 Check the Fullscreen button and SharedArrayBuffer support box.

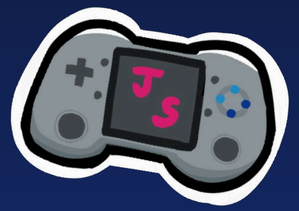
6 Add tags, a cover image, or anything else you want to. You can edit theme (colors and font) of your game page.

7 You need to Save the page before making it Public. It's is currently a Draft. Click Save & view page. Then go back, and select Public under Visibility and Acess.

8 Share your game page's URL for others to play!

*If you make updates in Godot, reupload a new .zip.

# SHIP YOUR PROJECT!



5 **Congratulations**!! You've build an AMAZING game in Godot.

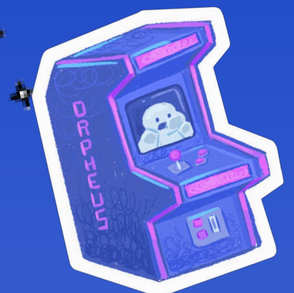
6 Go to **forms.hackclub.com/jumpstartdaydream** and submit!

**Submission Checklist! Make sure that...**

Your game is published on itch.io and the link works. Make sure it is public!

Your code is on GitHub and the repository is public. There should be .gd and .tscn files in it.

You've tested your game and it runs without any errors.



GODORPHEUS
J*S
ZOR
YOU'RE DONE!
CONGRATS!
JUMP START
DRPHEU
JS

# QUESTIONS? PROBLEMS?

Ask your Daydream organizer.

Email estella@hackclub.com.

Your project will be reviewed in about 1 weeks time. You will get a email. Be patient! :]

Once you get approval, CONGRATS! You've secured some super cool merch for yourself at Daydream.