GODOT

Game engine

MeetConf 2022

meet™

Agenda

How will our session look like?

- Lecture
  - What is Godot?
  - GDScript intro
  - Godot GUI
  - Importing/coding map & player
- Make your own games!
- Micro-Feedback time

meet™

What is Godot?

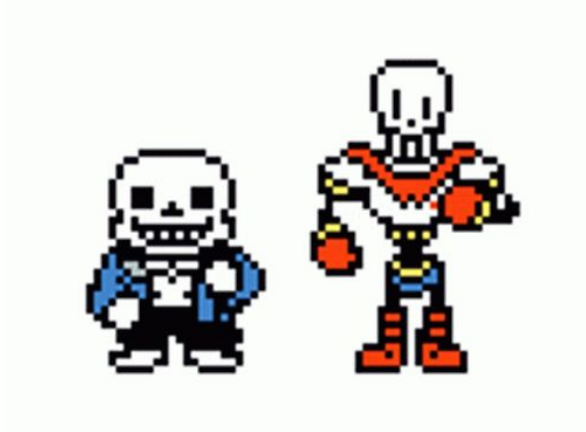
- Free & open-source game engine
- Developed by a community of volunteers

meet™

Why use Godot?

What makes it so ***Awesome***?

- Intuitive framework
- Feature packed (2D/3D, shaders, audio, light, anims)
- Cross-platform and exports
- Community support



meet™

What can we create with Godot?

Example Game

GDScript Basics

It's Python but *slightly* different

Variables

Python

```
x = 42                      #Int
y = 29.3                    #Float
name = “Lill”               #String
I_Love_Godot = True         #Boolean
```

GDscript

```
var x  = 42                 #Int
var y = 29.3                #Float
var name = “Lill”           #String
var I_Love_Godot = True     #Boolean
const GRAVITY  = 10         #Constant
```

Conditionals

Python

```
if condition:
    # Code to be executed if condition is true
else:
    # Code to be executed if condition is false
```

GDscript

```
if condition:
    # Code to be executed if condition is true
else:
    # Code to be executed if condition is false
```

Conditionals

Extra conditional

```
switch value:
    case first_case:
        # Code to be executed if value is first_case
    case second_case:
        # Code to be executed if value is second_case
    default:
        # Code to be executed if value is not
first_case or second_case
```

meet™

conditionals

What will this code do ?

It will print in the console `Value is apple`

Conditionals

```
# Declare a variable to hold the value we want to switch on
var value = "apple"

# Use a switch statement to execute different code based on the value of the
variable
switch value:
    case "apple":
        print("Value is apple")
        # Additional code to be executed if value is "apple"
    case "banana":
        print("Value is banana")
        # Additional code to be executed if value is "banana"
    default:
        print("Value is neither apple nor banana")
        # Additional code to be executed if value is neither "apple" nor
"banana"
```

Using switch

meet™

conditionals

What will this code do ?

```
# Declare a variable to hold the value we want to switch on
var value = "apple"

# Use a switch statement to execute different code based on the value of the variable
switch value:
    case "apple":
        print("Value is apple")
        # Additional code to be executed if value is "apple"
    case "banana":
        print("Value is banana")
        # Additional code to be executed if value is "banana"
    default:
        print("Value is neither apple nor banana")
        # Additional code to be executed if value is neither "apple" nor "banana"
```

Using switch

meet™

Loops

Python

```
for var in range(n):
    # Code to be executed

while condition:
    # Code to be executed
```

GDScript

```
for var in range(n):
    # Code to be executed

while condition:
    # Code to be executed
```

meet™

Functions

Python

```
def myfunc():
    # Code to be executed when function is ran
```

GDScript

```
func myfunc():
    # Code to be executed when function is ran
```

Functions

What will this code do?

```
func my_function(arg1, arg2):
    print(arg1)
    print(arg2)

my_function(“banana”, “nanana”)
```

meet™

Functions

It will print in the console:

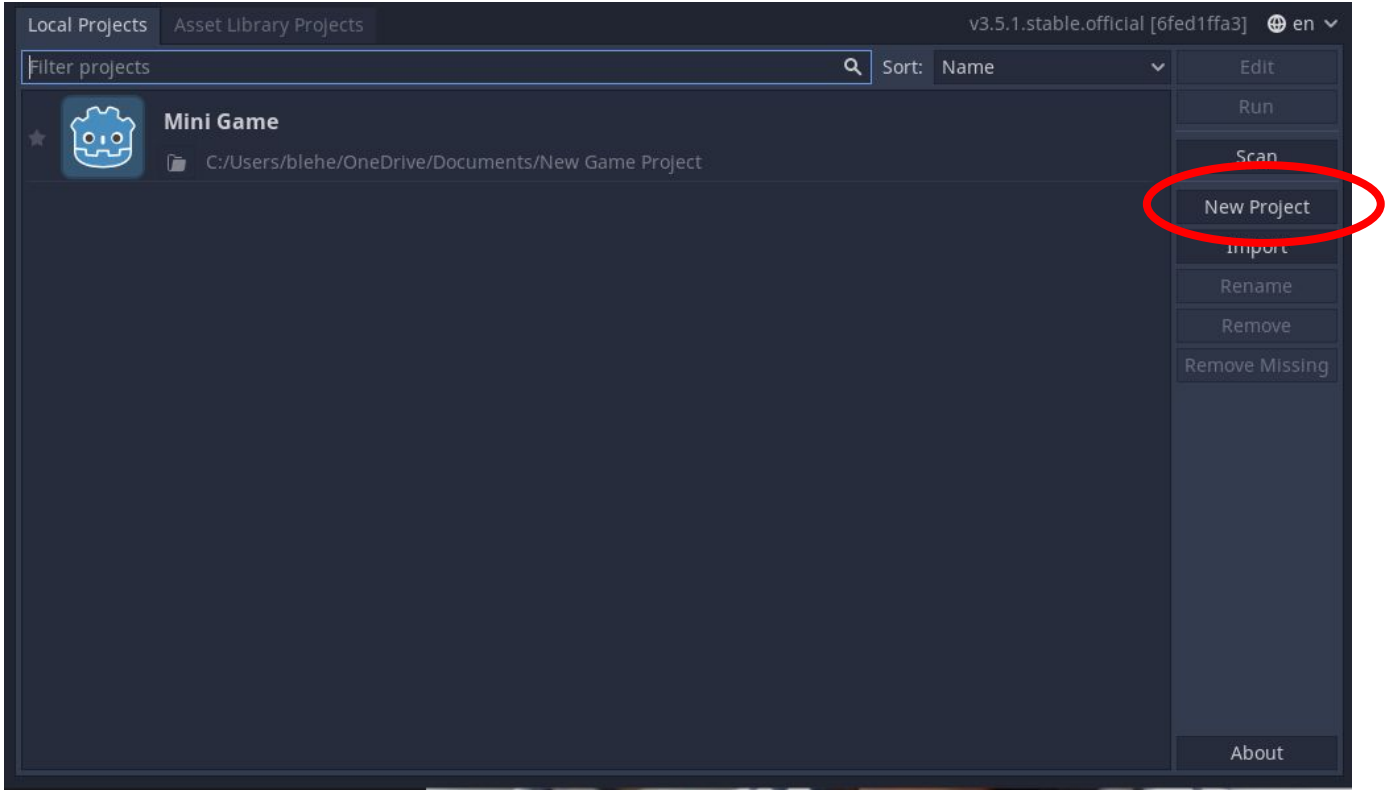
Banana

Nanana

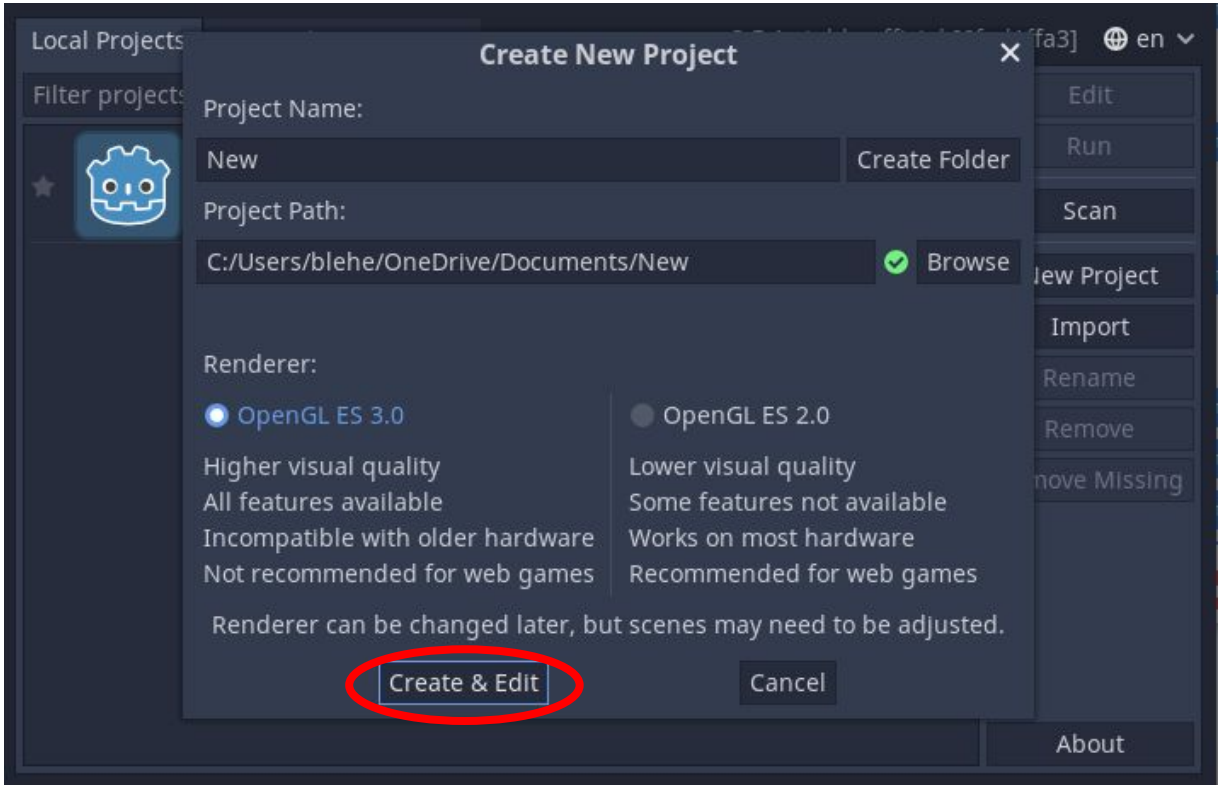
Godot GUI

Let's get familiar with the Godot Interface!

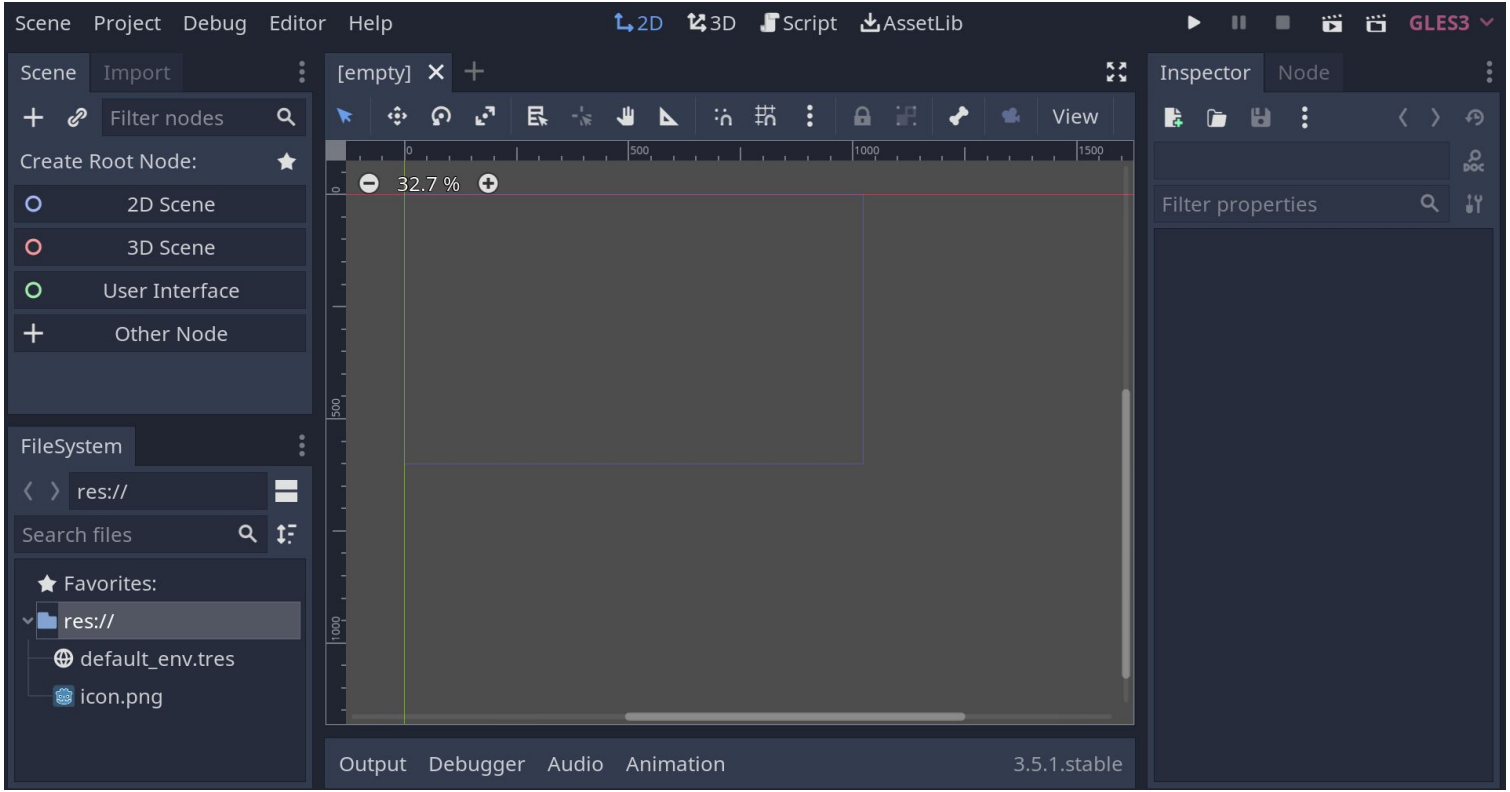
Creating your First Game



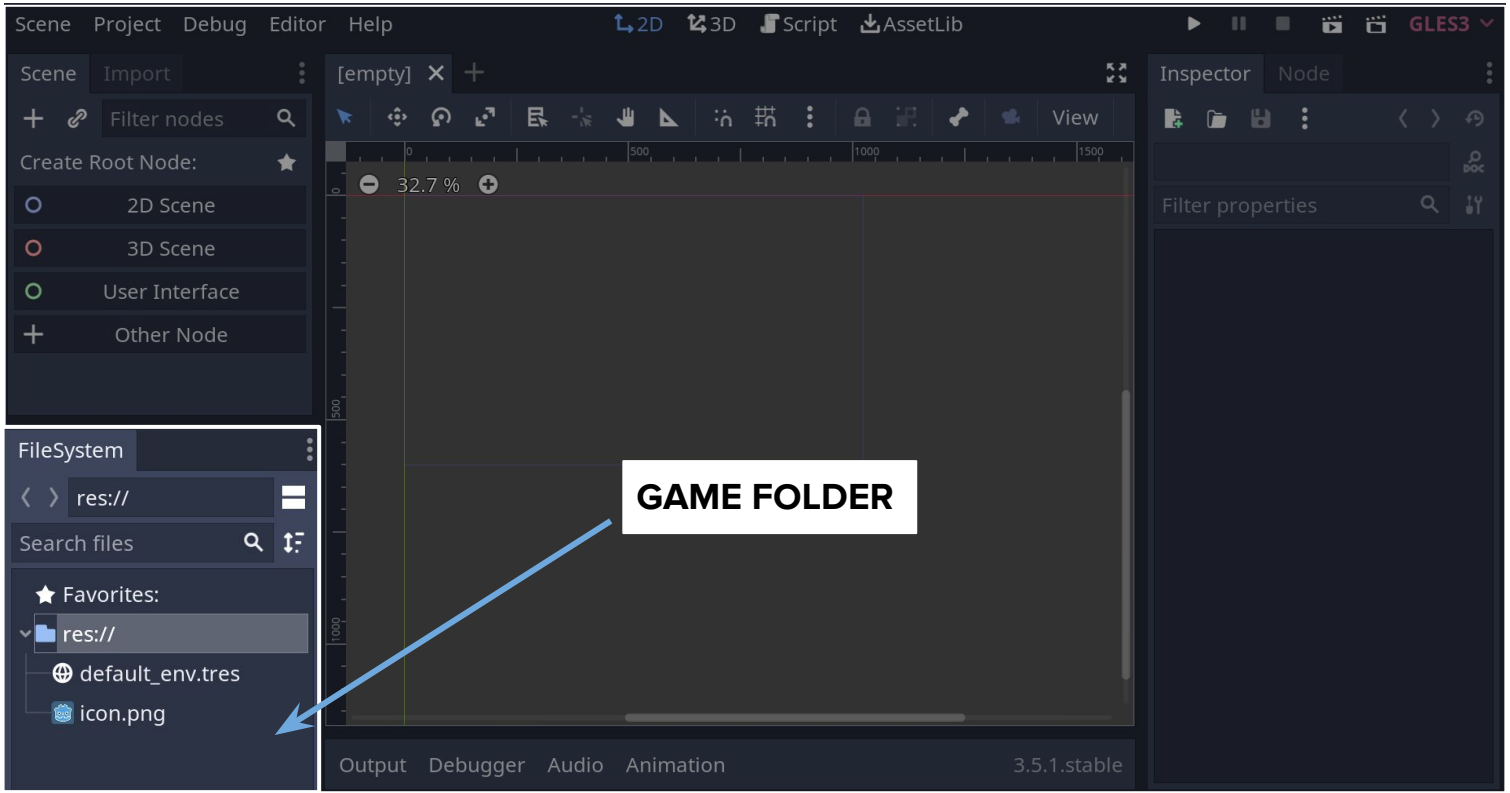
Creating your First Game



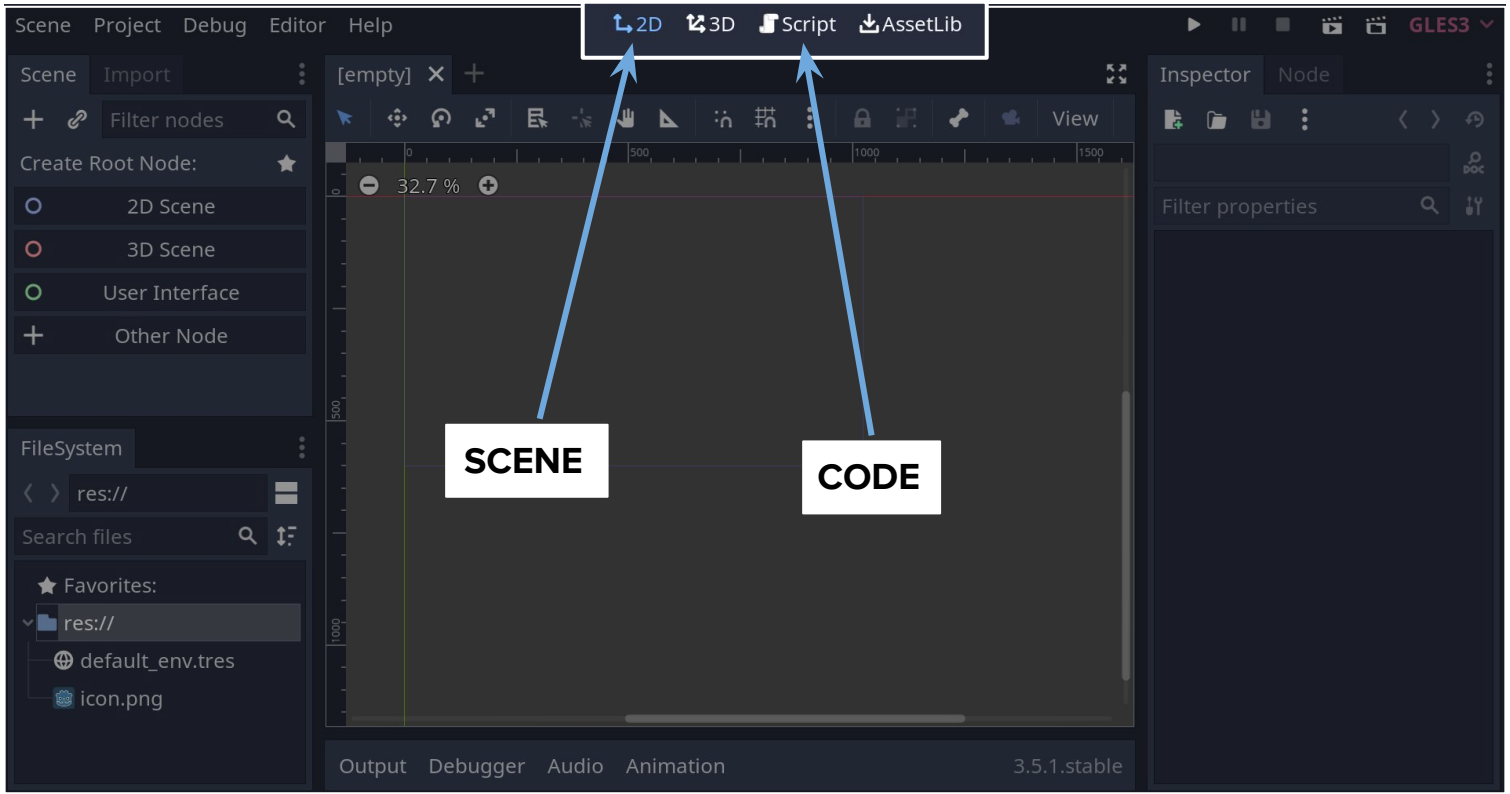
Main Godot GUI



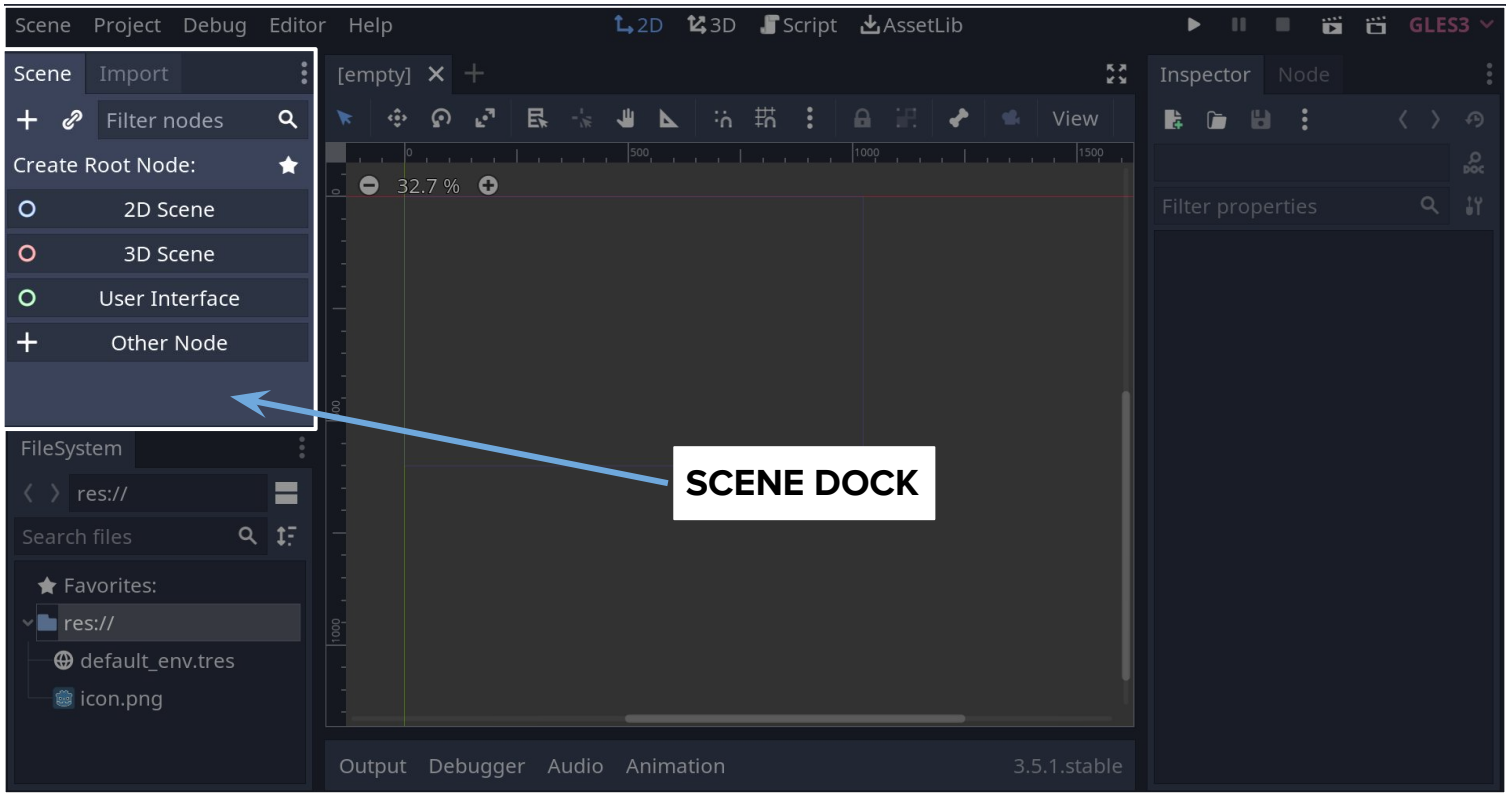
Files Dock


Scene
Project
Debug
Editor
Help
2D
3D
Script
AssetLib
GLES3
Scene
Import
Filter nodes
Create Root Node:
2D Scene
3D Scene
User Interface
Other Node
[empty]
View
32.7 %
Inspector
Node
Filter properties
FileSystem
res://
Search files
Favorites:
res://
default_env.tres
icon.png
GAME FOLDER
Output
Debugger
Audio
Animation
3.5.1.stable

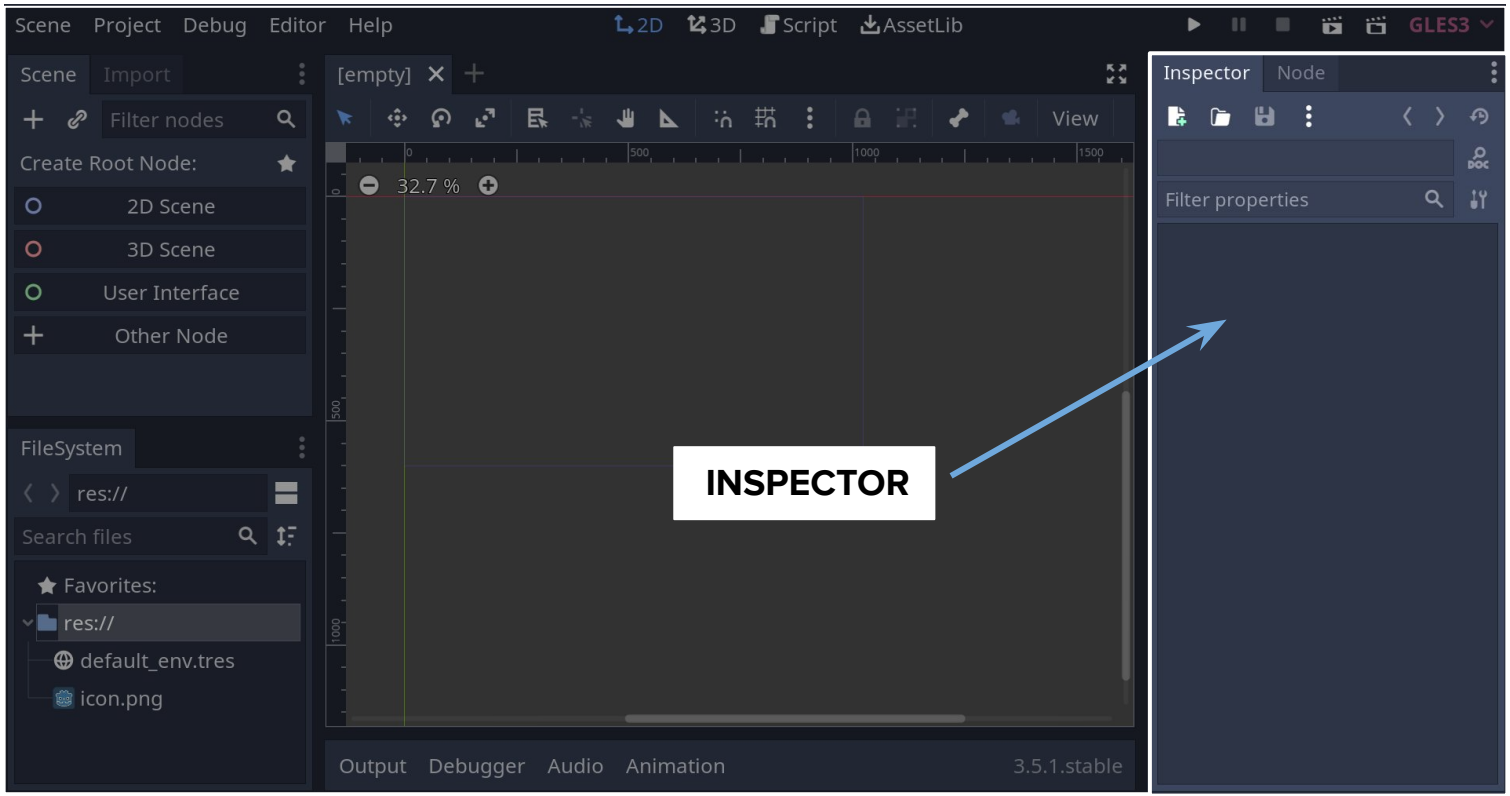
Main Godot GUI



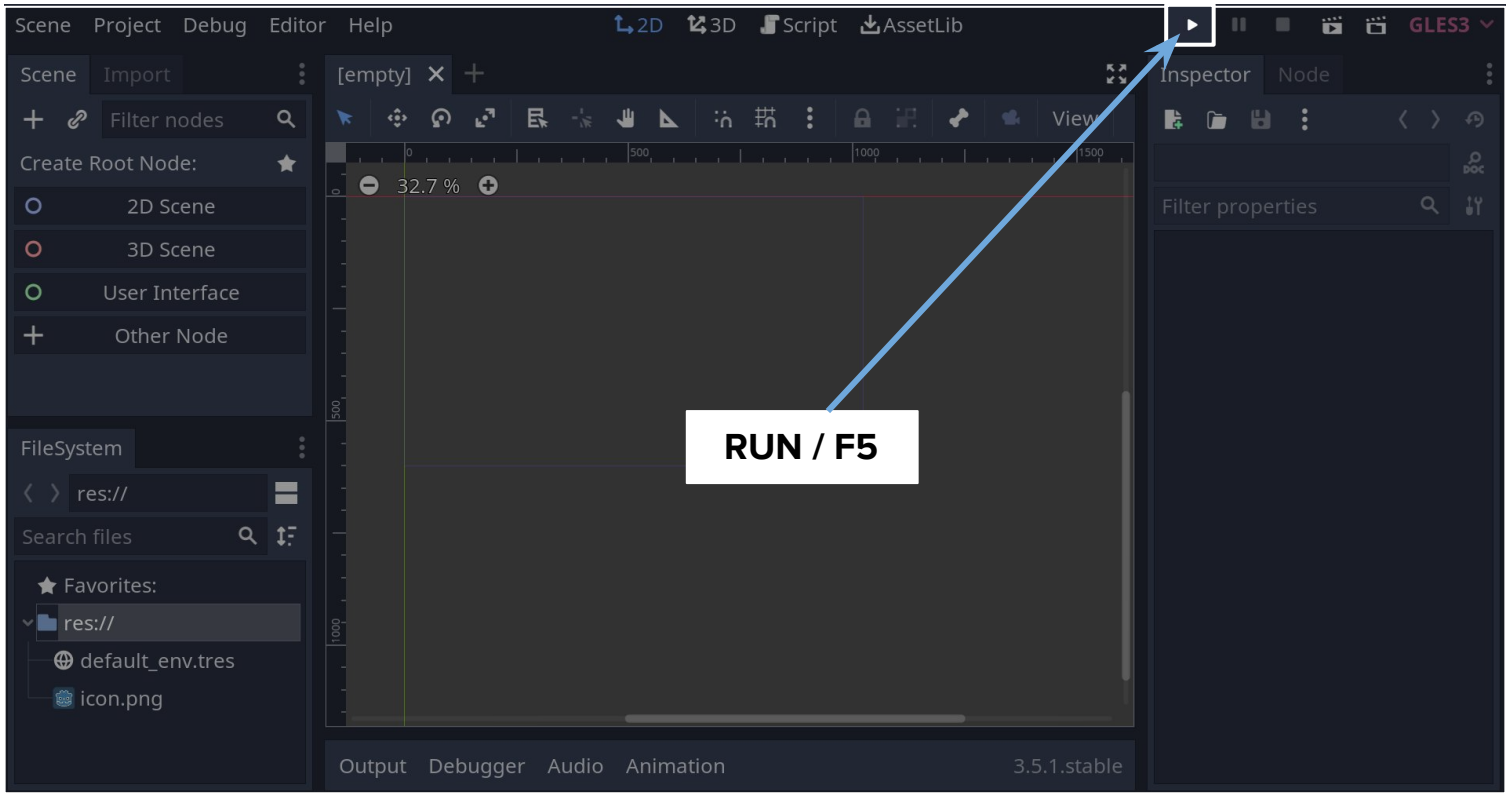
Scene Dock


Scene
Project
Debug
Editor
Help
2D
3D
Script
AssetLib
GLES3
Scene
Import
Filter nodes
Create Root Node:
2D Scene
3D Scene
User Interface
Other Node
[empty]
View
32.7 %
Inspector
Node
Filter properties
SCENE DOCK
FileSystem
res://
Search files
Favorites:
res://
default_env.tres
icon.png
Output
Debugger
Audio
Animation
3.5.1.stable

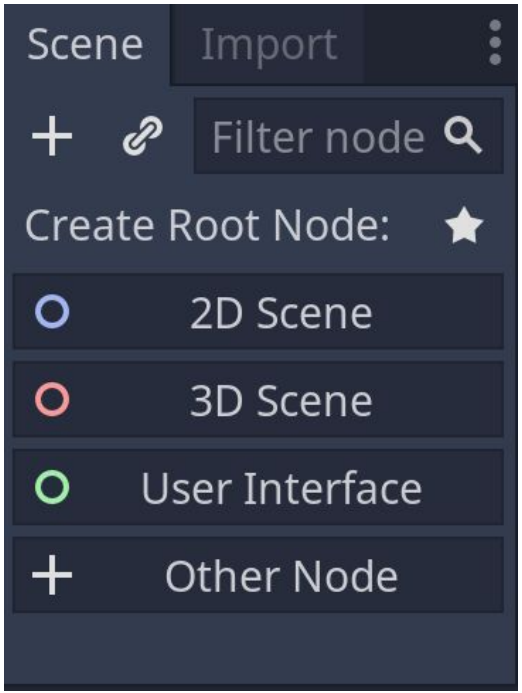
Inspector


Scene
Project
Debug
Editor
Help
2D
3D
Script
AssetLib
GLES3
Scene
Import
Filter nodes
Create Root Node:
2D Scene
3D Scene
User Interface
Other Node
[empty]
View
32.7 %
FileSystem
res://
Search files
Favorites:
res://
default_env.tres
icon.png
INSPECTOR
Inspector
Node
Filter properties
Output
Debugger
Audio
Animation
3.5.1.stable

Running your game



Scene Dock

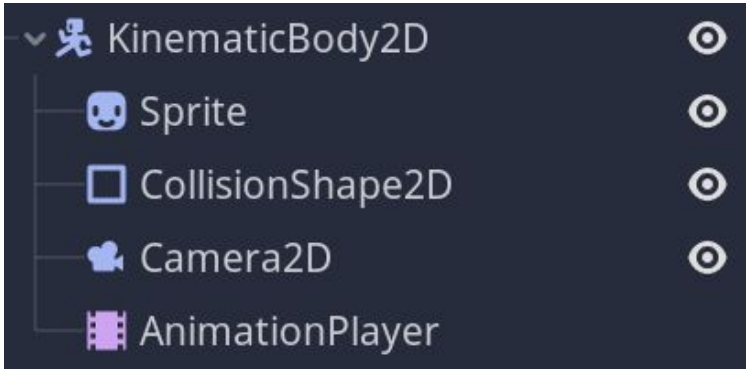


- Godot's most essential object is a Node.
- Nodes can be: sprites, collision shapes, scenes, kinematic bodies (the player), etc...

meet™

Tree structure

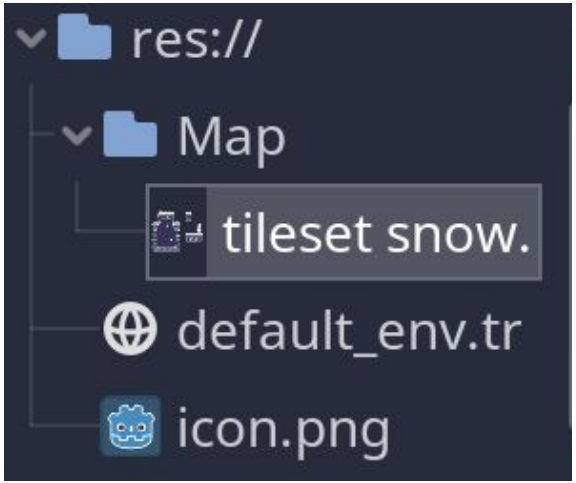
- The tree structure allows grouping nodes together: For example, for your player you can use this structure or a similar one-



Content of the game.

Importing your map

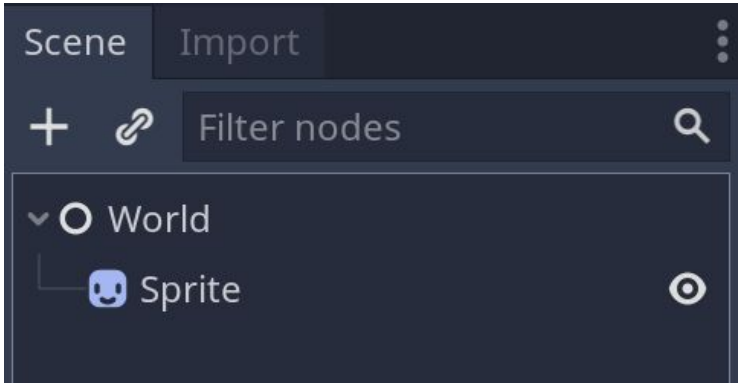
- Go the the linked folder
- Choose a map
- Create a map folder in your Files Dock
- Download and drag to your Map folder

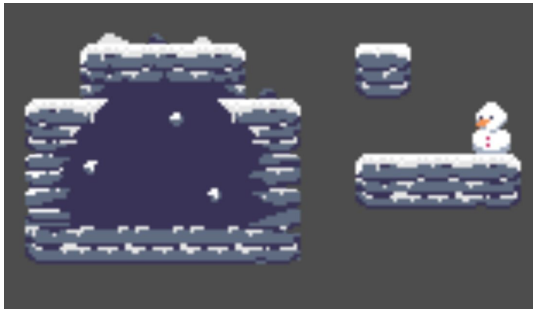


Folder

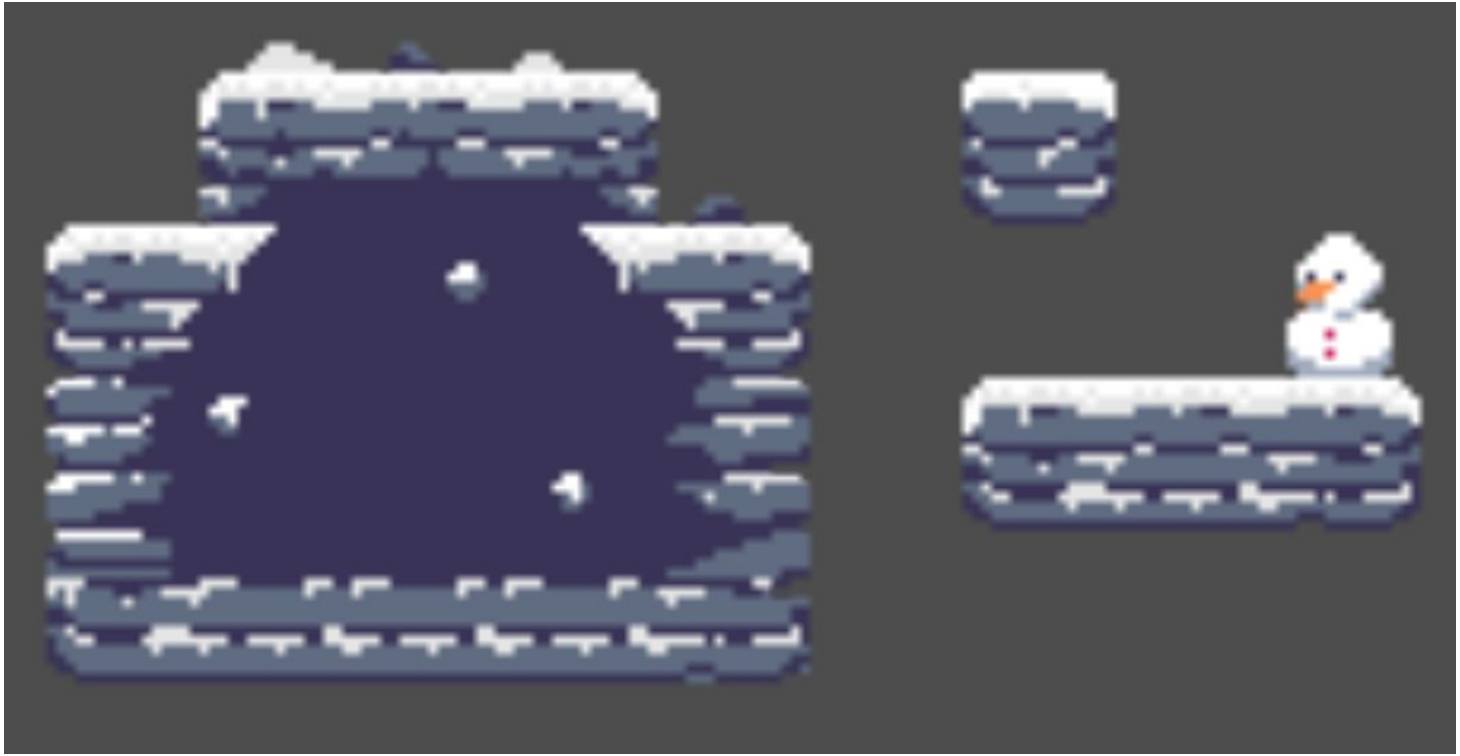
Building your map

- Go to the Scene Dock
- Create a new node called “World”
- Add a Sprite
- Drag your map under Texture in the Inspector Dock
- Make sure your map is visible





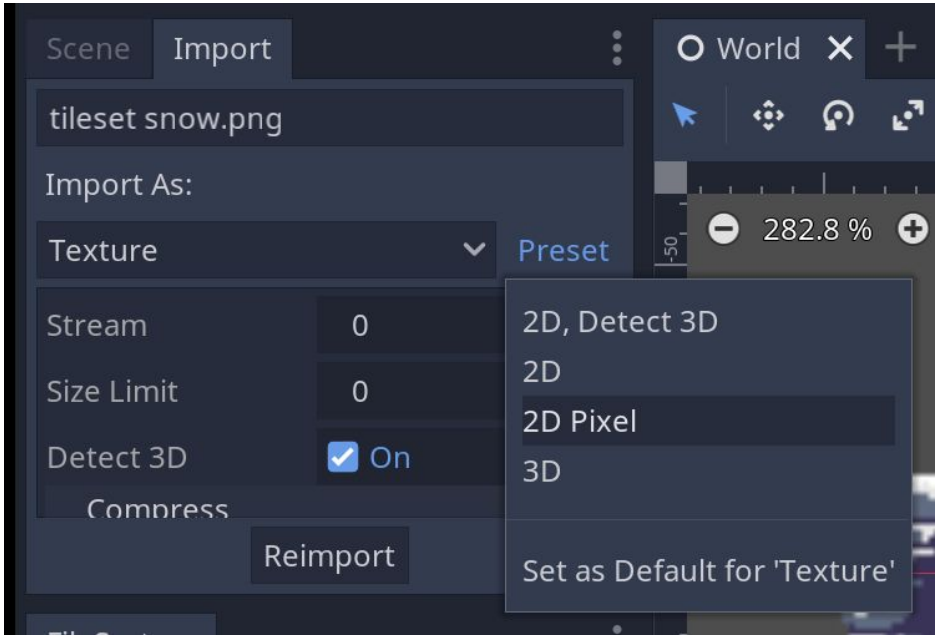
Blurry?



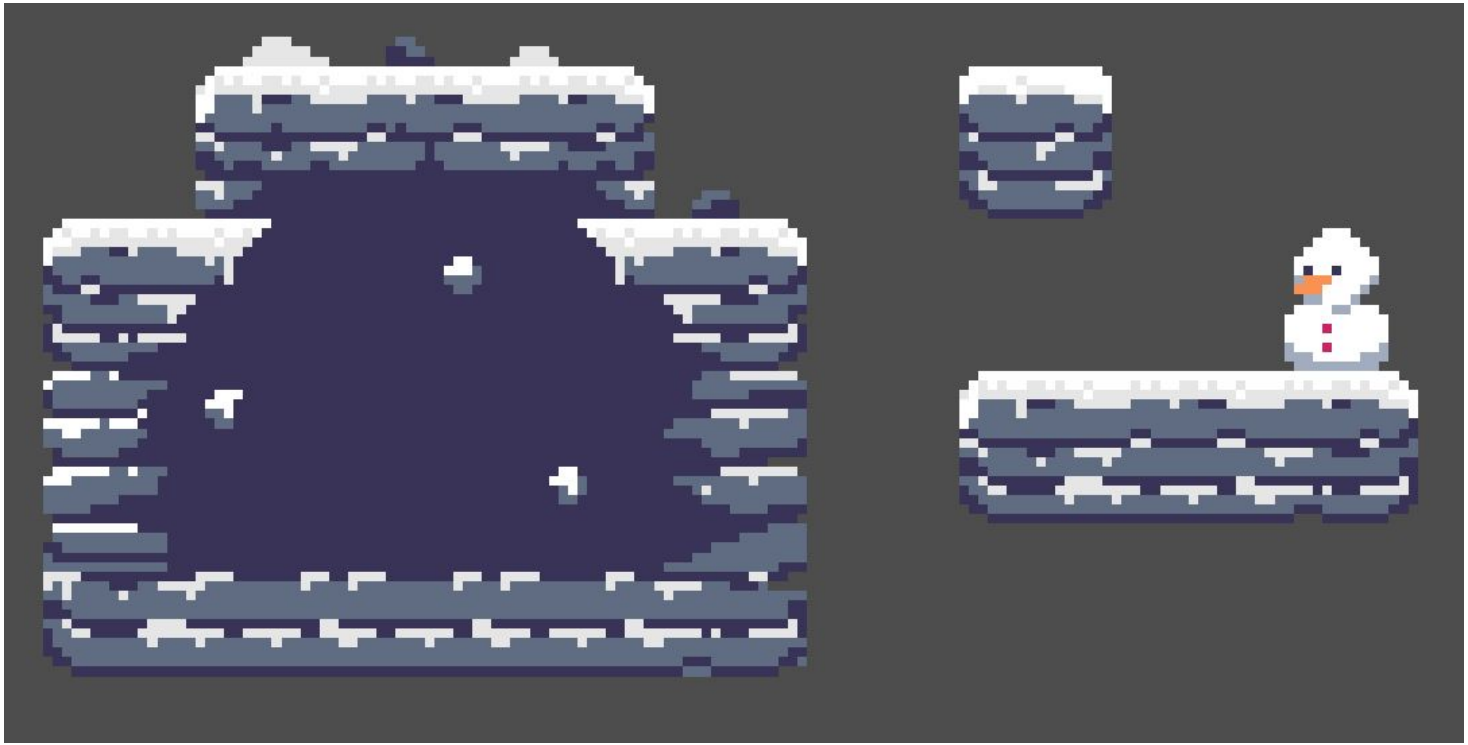
Building your map

To fix blurriness:

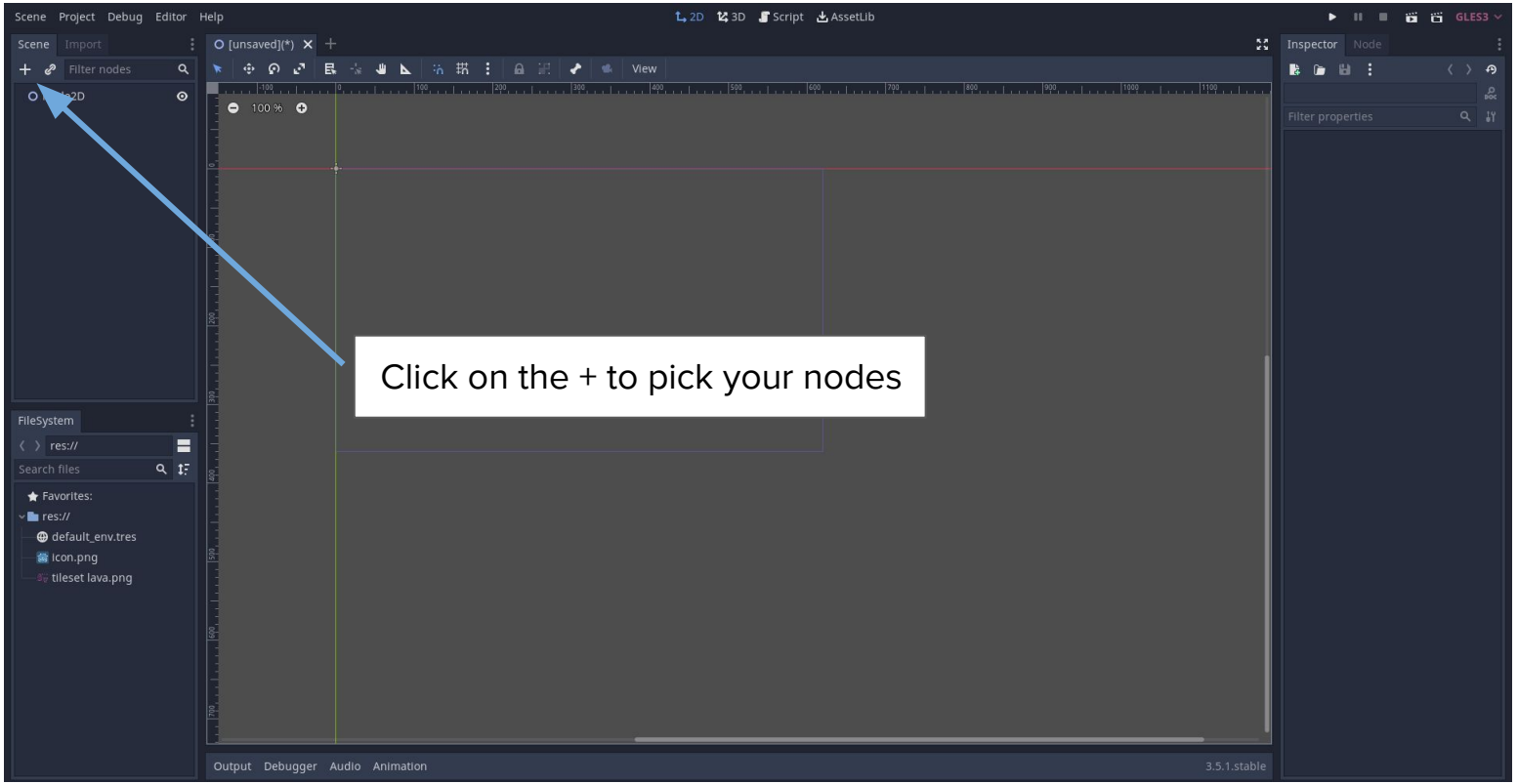
- Go to the Scene Dock
- Click Import
- Hover on Preset
- Select “2D Pixel”
- Set as Default for ‘Texture’
- Click Reimport



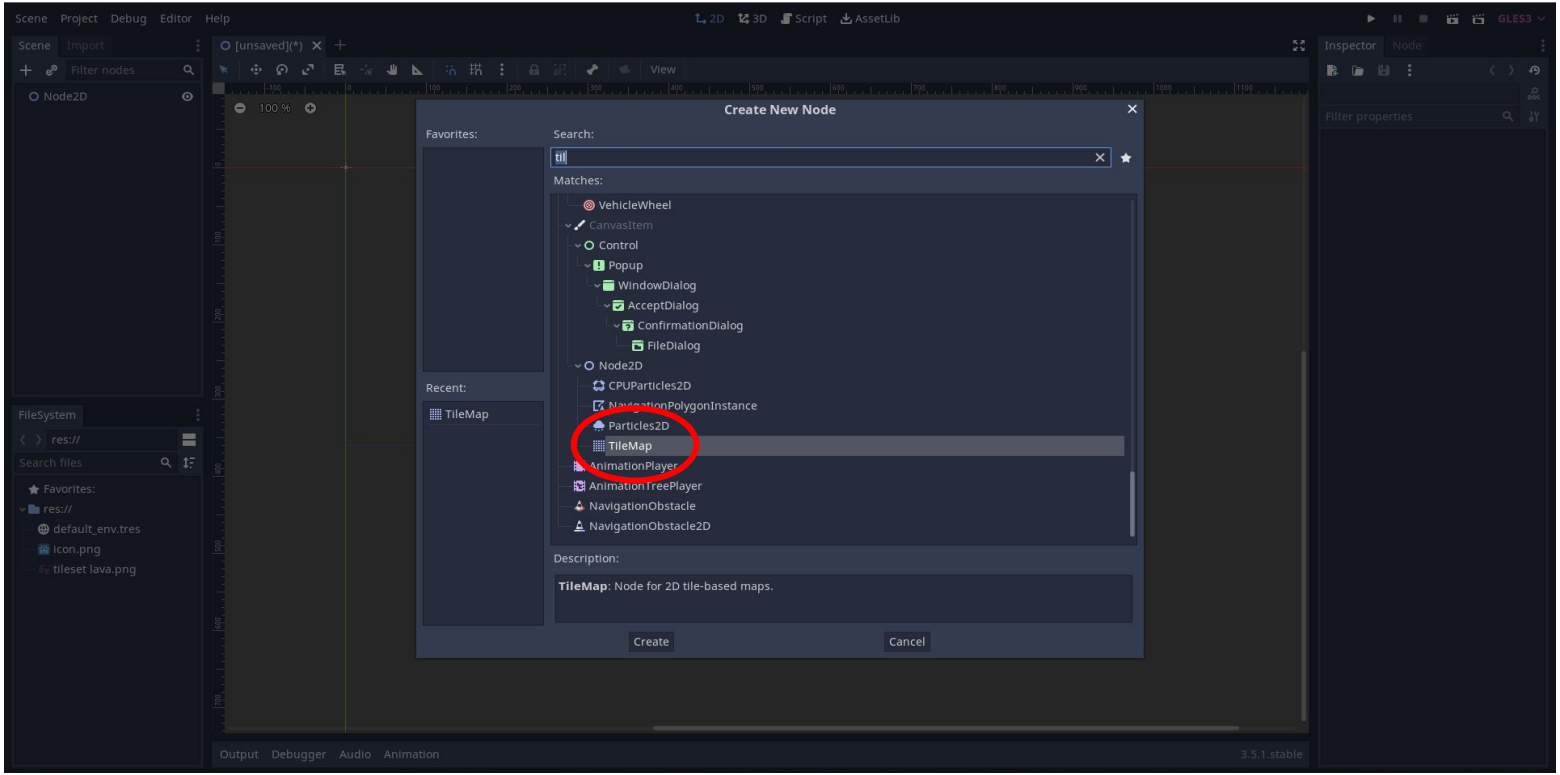
Much better!



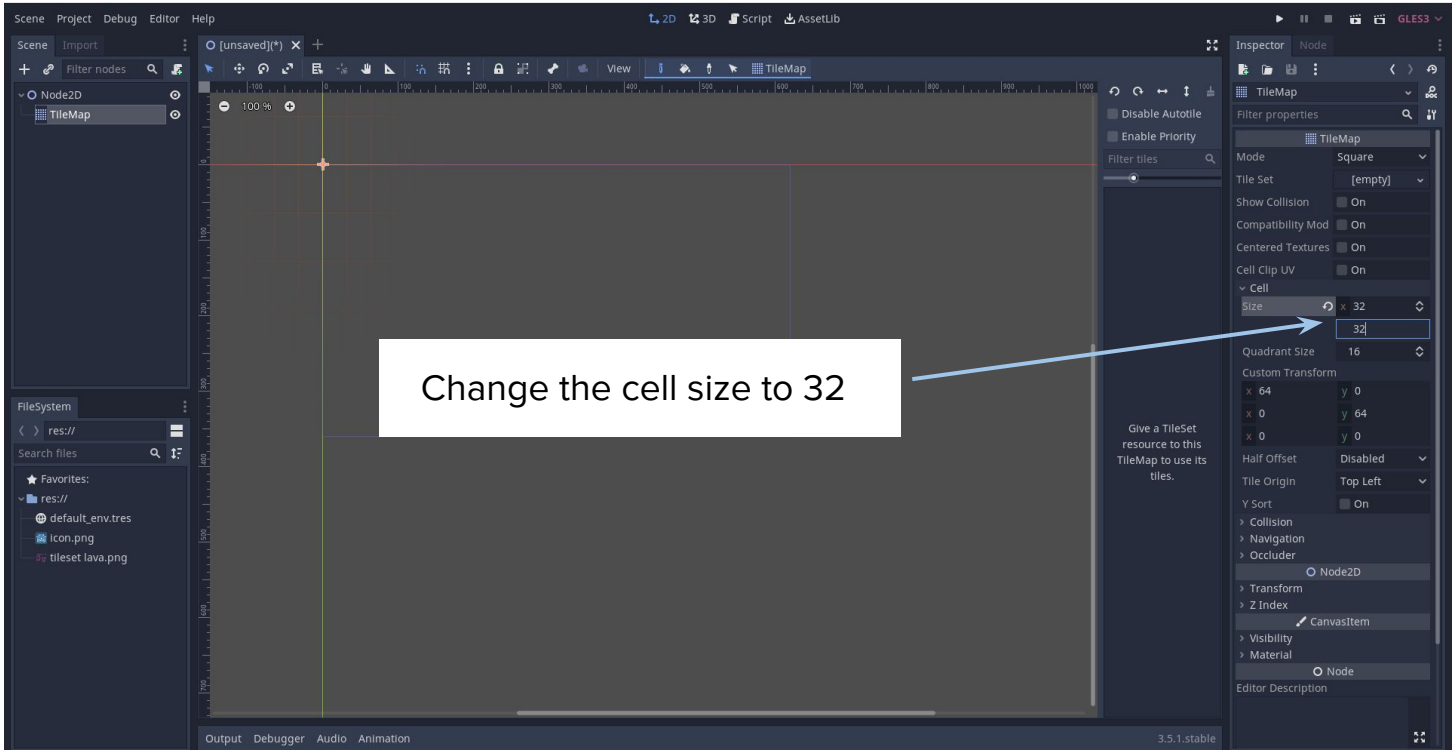
Automatic Tiling



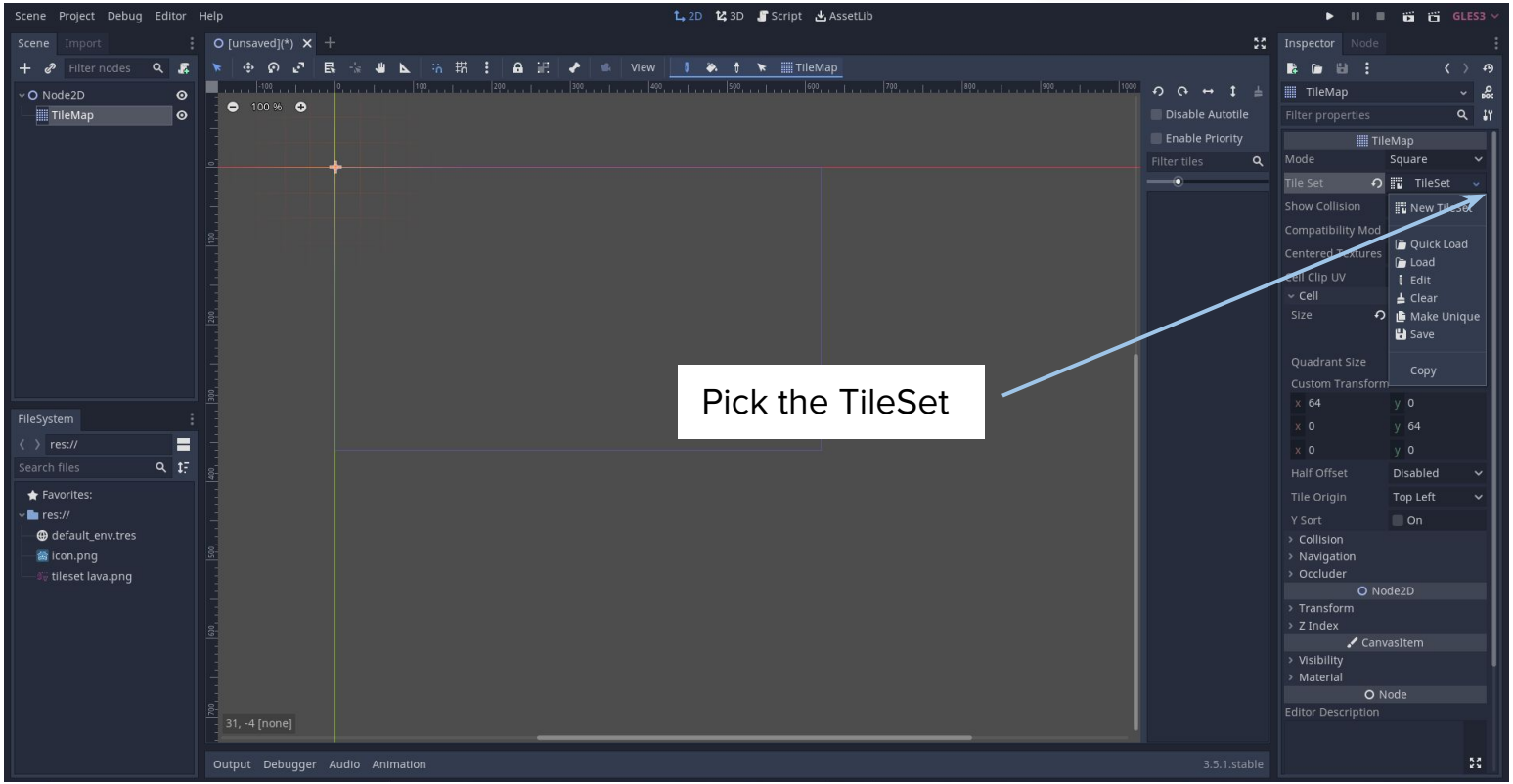
Automatic Tiling



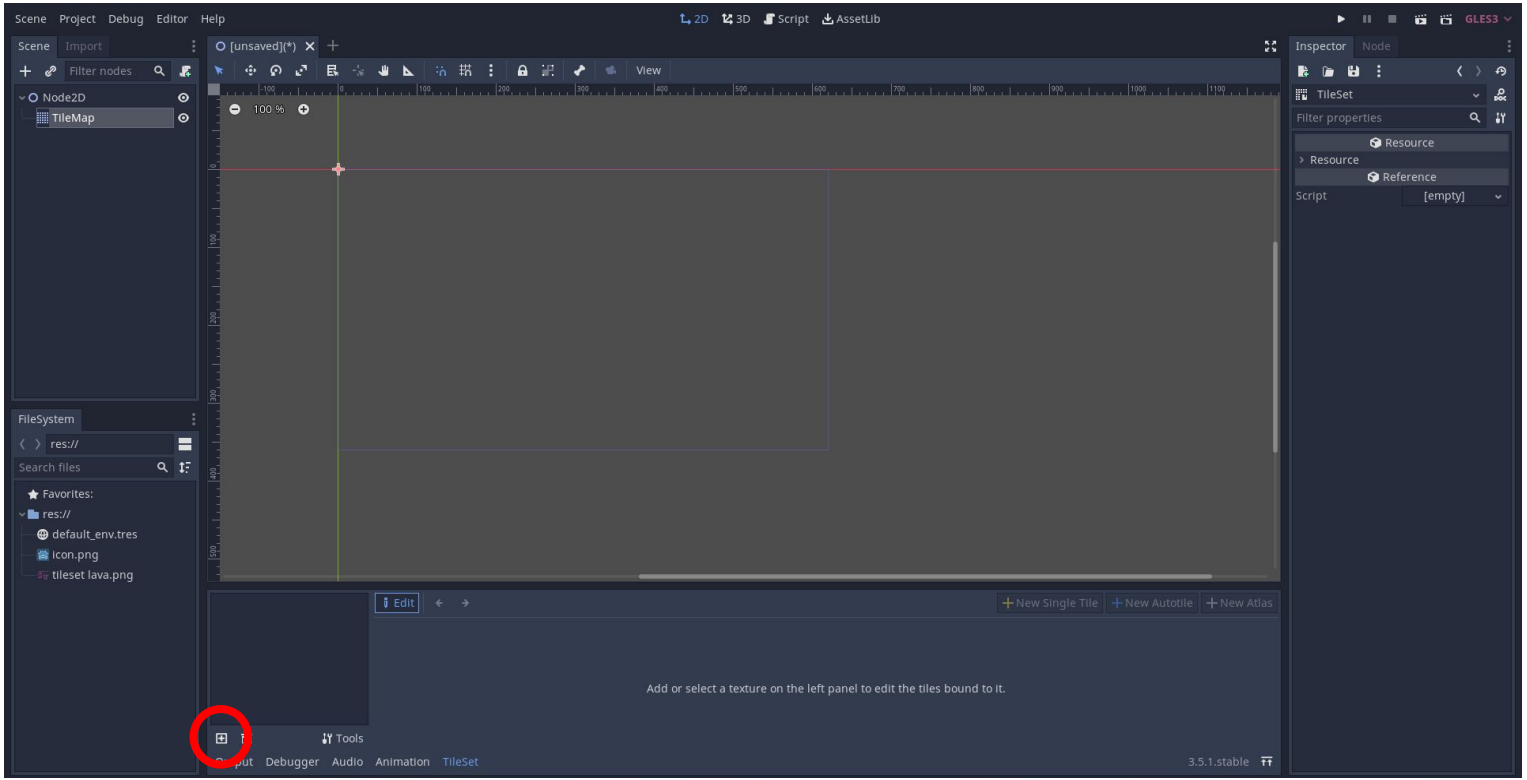
Automatic Tiling



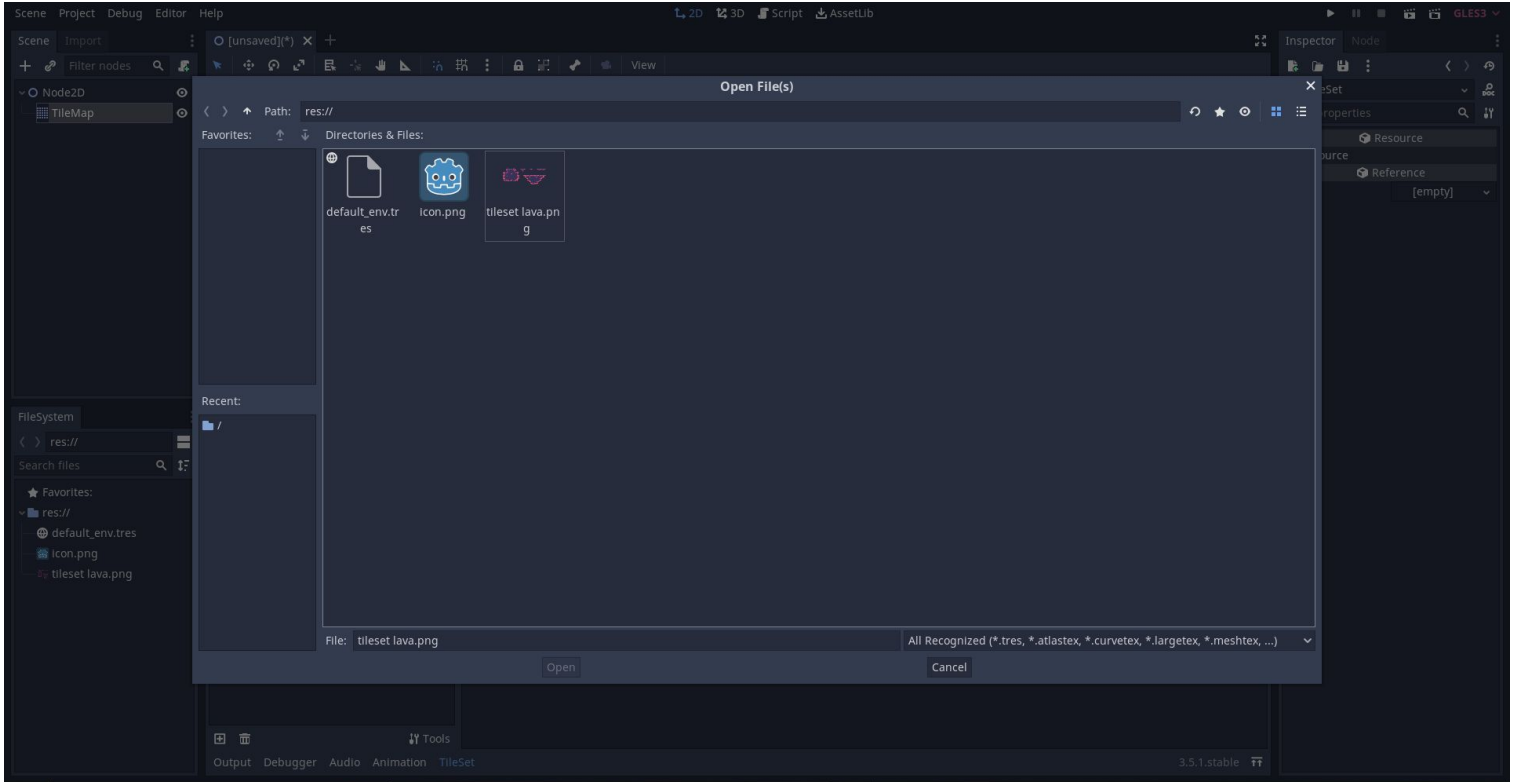
Automatic Tiling



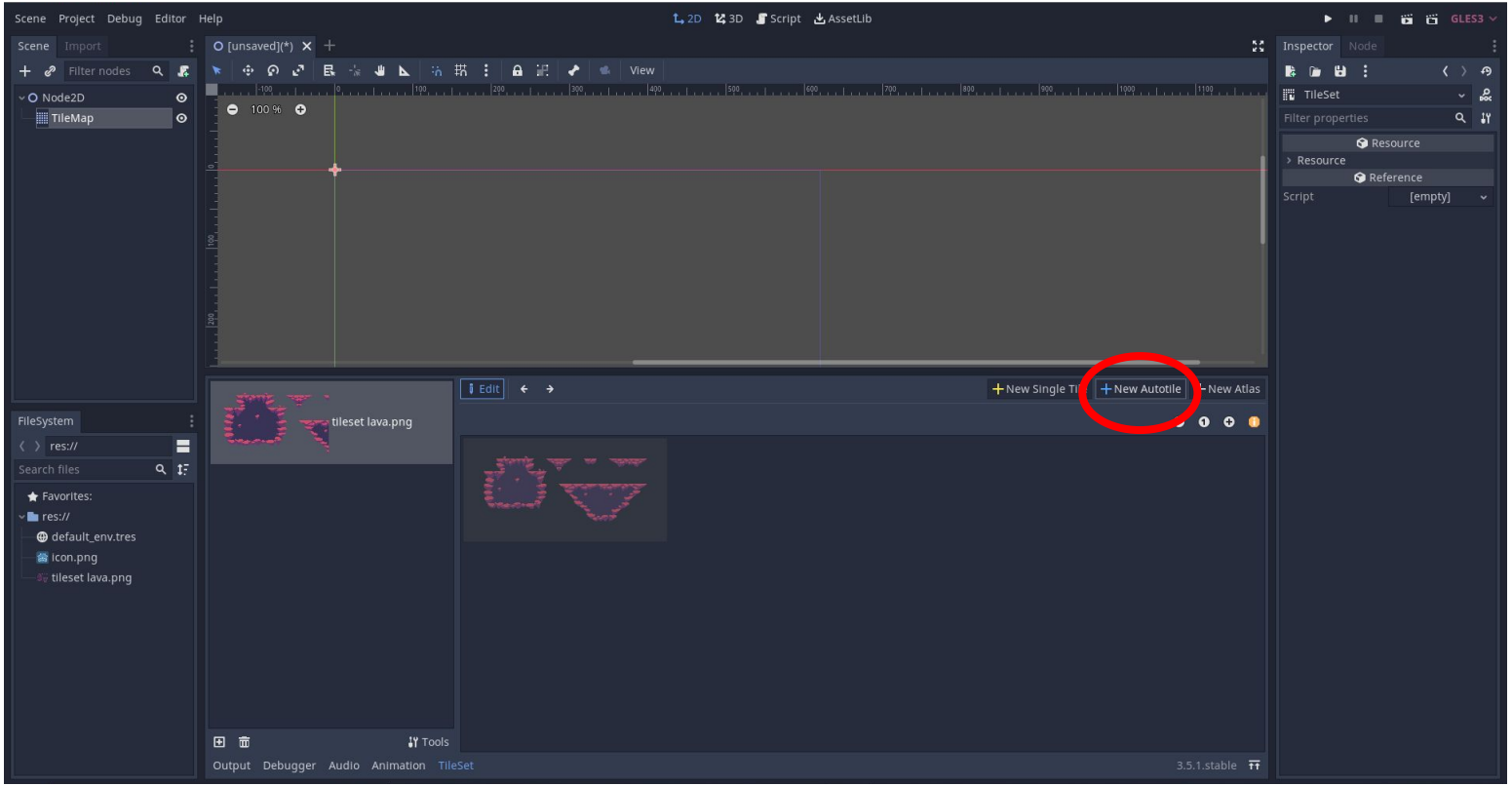
Your page should look like this



Pick your TileMap

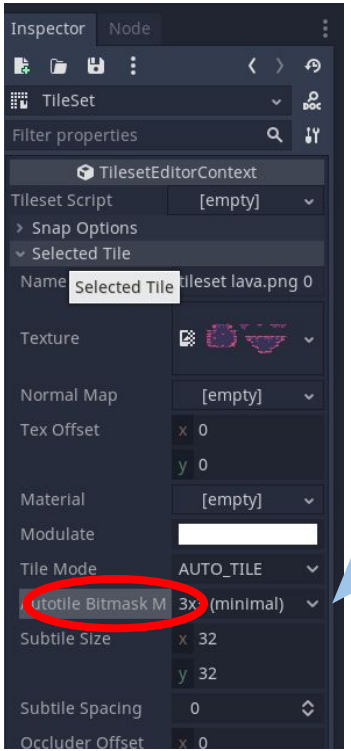


Automatic Tiling



Enable snap and show grid

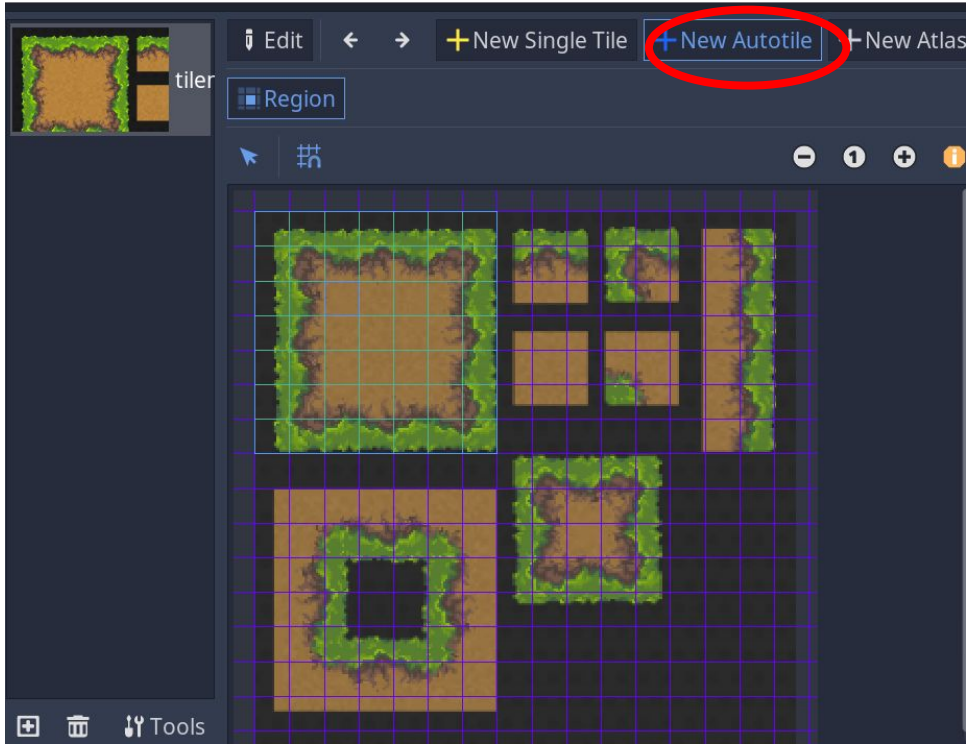
Highlight your tile map



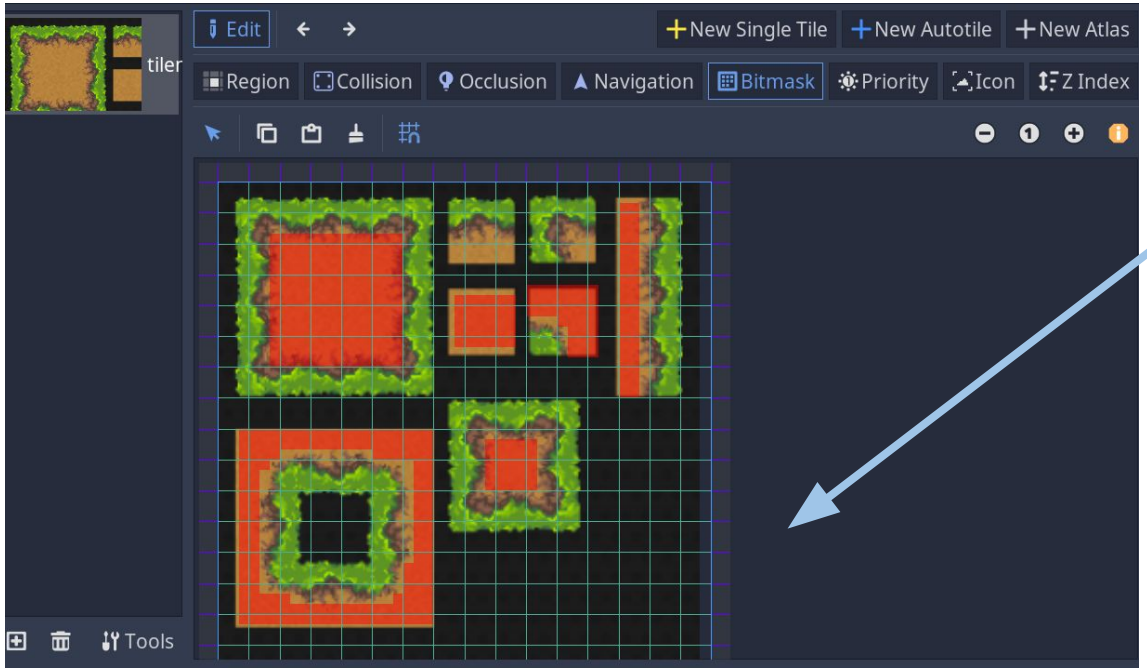
Change the autotile Bitmask

meet™

Pick the Bitmask



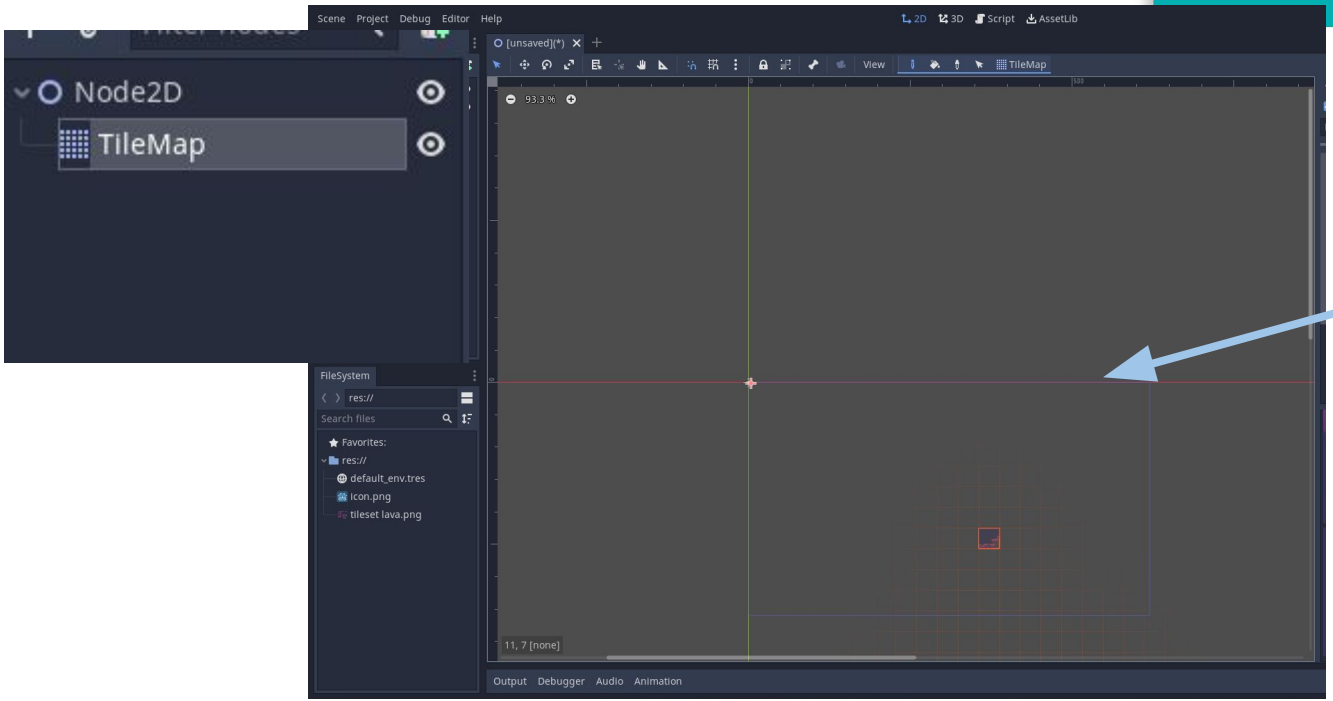
Set the Bitmask for your AutoTile



Right click = erase
Left click = select

meet™

Click on your tile map

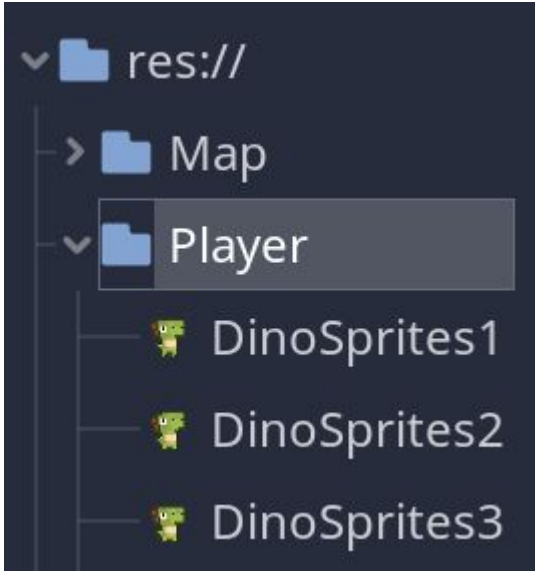


Right click = erase
Left click = create

meet™

Importing your player

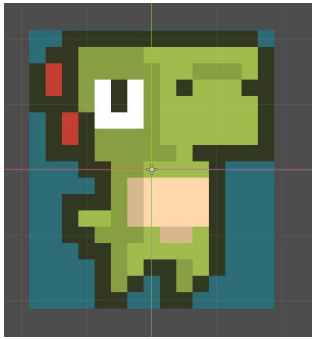
- Go the the linked folder
- Choose your character
- Create a Player folder in your Files Dock
- Download and drag to your Player folder



Folder

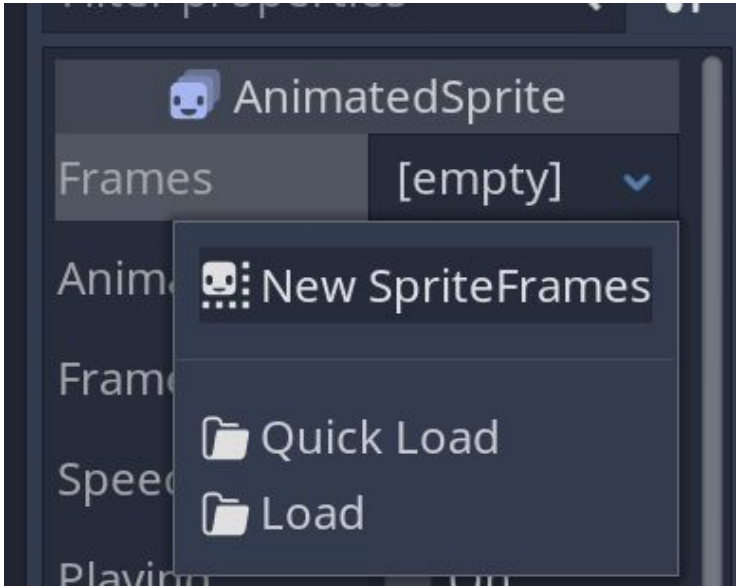
Importing your player

- Go to the Scene Dock
- Add a “KinematicBody2D” node
- Attach an “AnimatedSprite” node
- Attach a “CollisionShape2D”
- In the inspector click on shape and choose the rectangle
- Scale your rectangle according to your player



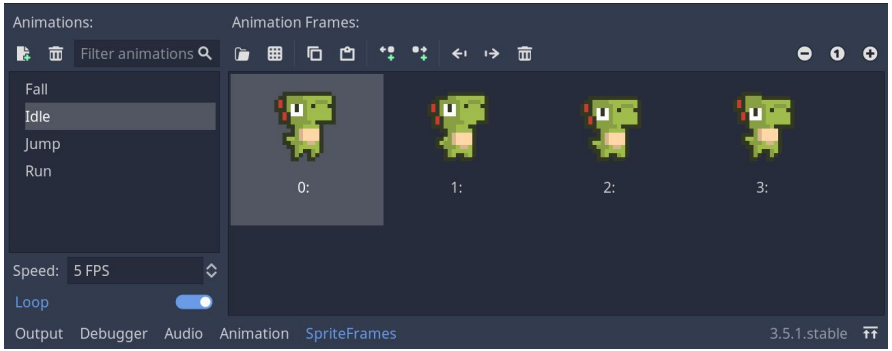
Player animation

- Click on your Animated Sprite
- Go to the Inspector
- Click on Frames
- Click New SpriteFrames



Player animation

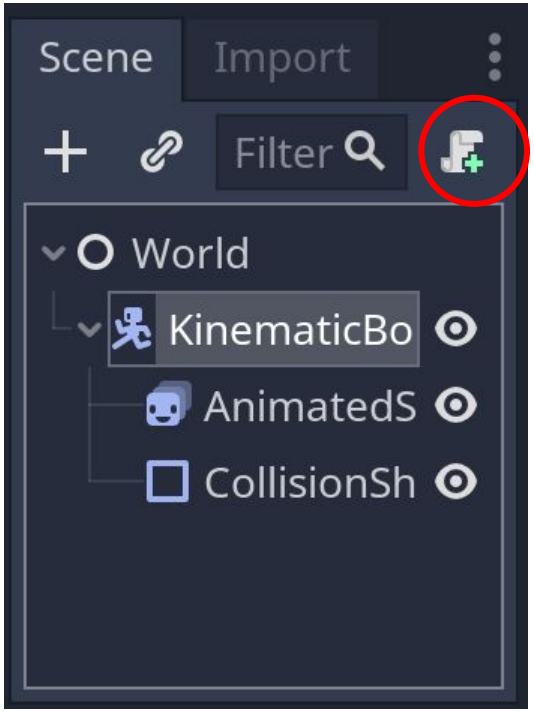
- Click on the AnimatedSprite in the Scene Dock
- Rename to idle (double click)
- Drag your frames
- Add more cases accordingly (Run, Jump, Fall, Idle)



Player movement

- Click on KinematicBody2D
- Attach a script

Now let's get to *Coding!*



Coding player movement

Defining the key variables:

```
extends KinematicBody2D   #What were currently programming


const GRAVITY = 1200      #Setting up Gravity const

const SPEED = 200         #Setting up Speed const

const JUMP_HEIGHT = 200   #Setting up Jump const
```

Coding player movement

```
func _physics_process(delta):
    motion.y += GRAVITY*delta  #Setting up our gravity

    if Input.is_action_pressed(“ui_right”):

        #Code to complete

    elif Input.is_action_pressed(“ui_left”):

        #Code to complete

    else:

        #Code to complete
```

meet™

Coding player movement

```
if is_on_floor():
    if Input.is_action_pressed("ui_up"):
        motion.y = JUMP_HEIGHT

else:
    $AnimatedSprite.play("Jump")

if motion.y > 0 and is_on_floor() == false:
    $AnimatedSprite.play("Fall")


motion = move_and_slide(motion, UP)
```

Godot Dictionary

GUI → Graphical User Interface.

Tiles → tile are small, regular-shaped images that are used to create floors,backgrounds..

Delta → delta can be used to refer to the difference in a variable or data over time.

Vector2() → are used to represent points or positions with the parameters X and Y.

is_on_floor() → a built in function that checks if any of our colluding elements -is on the floor-.

Move_and_slide() → automatically calculates frame-based movement using delta

minimum requirements

- Have a tile map .
- Have at least one functioning sprite .
- Add collision shapes!
- Your sprite should be able to move(left,right), and should be able to jump(add gravity).
- Go crazy!!!

meet™

BOUNSES

- CREATE MULTIPLE SCENES.
- MOVING BETWEEN LEVELS.
- CREATING YOUR OWN SPRITE.
- ADD AUTOMATIC TILEMAP
- Make the sprite move using (W,A,S,D).
- Make the sprite move and rotate using the mouse.
- Think of other mechanics to make your game harder (time limit, moving objects)

Work Time!

tinyurl.com/3rs77hmt