

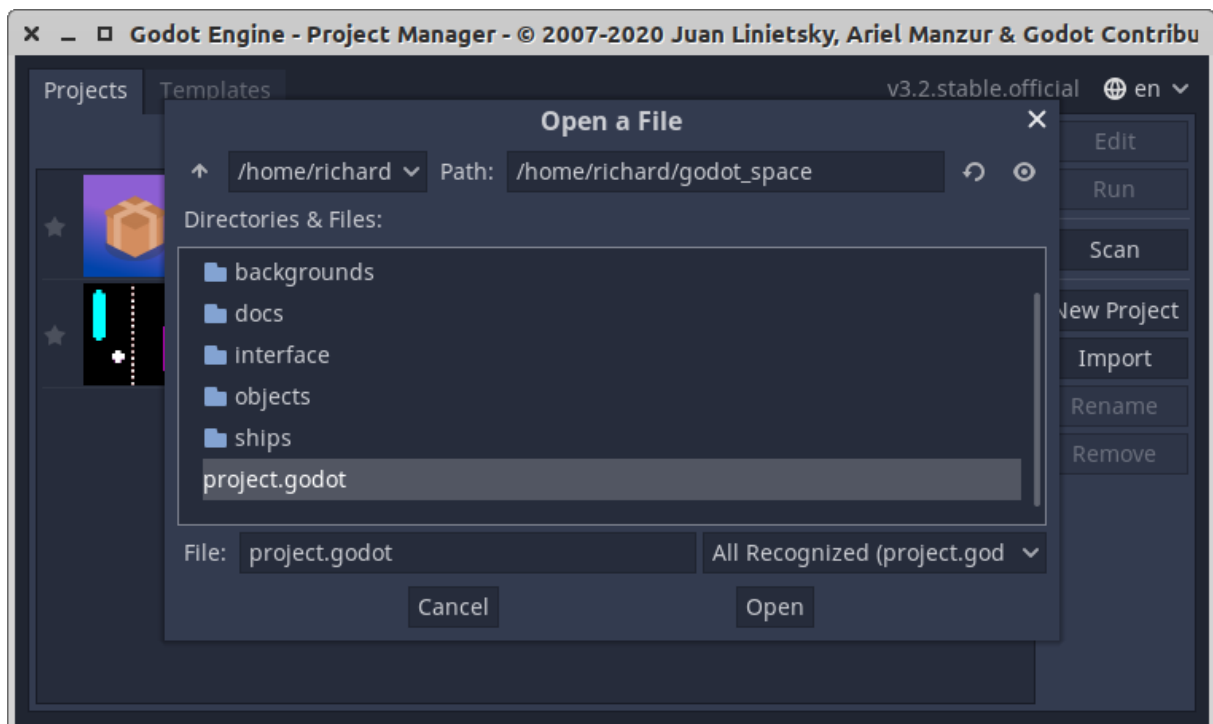
Contents

1	Space Shooter Tutorial	2
1.1	Velocity	3
1.2	Camera	3
1.3	Background	4
1.4	Particle effect	4
1.5	Make enemy move	5
1.6	Add more enemies	5
1.7	Make bullets move	5
1.8	Make bullets collide	6
1.9	Make enemy collide	6
1.10	Make player collide	7
1.11	Optional - Gamepad controls	8
1.12	More types of enemies	8
1.13	Enemy spawner	8
1.14	Enemy randomize	8
1.15	Charge laser	9
1.16	Optional - Touch controls	9
1.17	Title screen	9

1 Space Shooter Tutorial

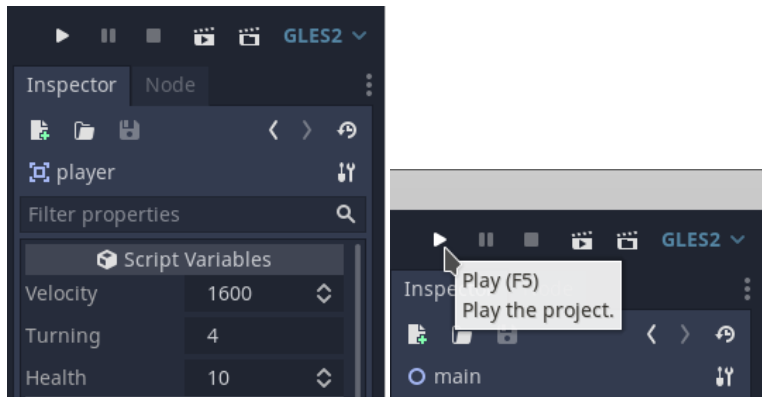
Download the starter project from https://electronstudio.github.io/godot_space/godot_space1.zip.

Unzip it. Open Godot. Import the `project.godot` file.



Run the game. You should have a spaceship sprite that can turn left and right. There is also lighting and a HUD.

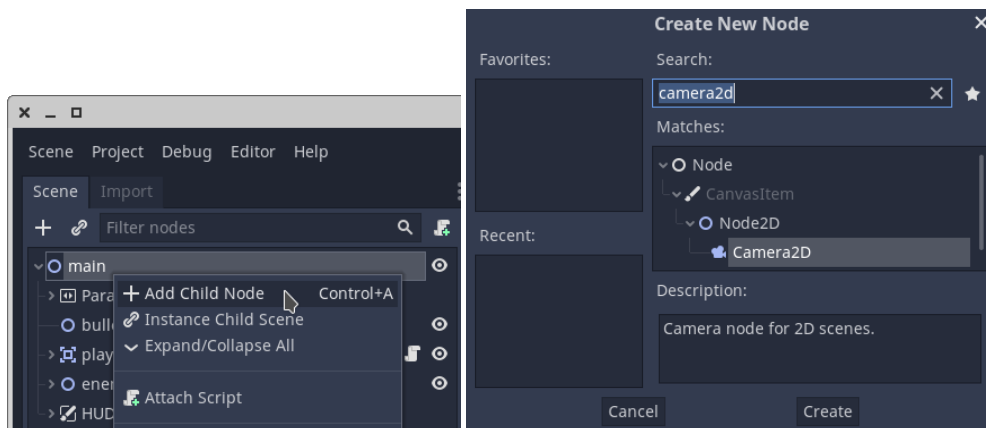
1.1 Velocity



Change the `velocity` of the `player` to 1600 in the node inspector. Run the game.

1.2 Camera

We need a camera to track the player. Add a `Camera2D` node to the `player` node.



In the node inspector set:

- Current: On
- Zoom x: 6
- Zoom y: 6
- Drag Margin:
 - Left: 0
 - Right: 0
 - Top: 0
 - Bottom: 0

1.3 Background

We would like to add another layer to the scrolling background.

1. Add another `ParallaxLayer` node to the `ParallaxBackground` node.
2. Click on the `ParallaxLayer2`.
3. In the node inspector, set:
 - Motion: Mirroring x: 15360
 - Motion: Mirroring y: 15360
4. Find `backgrounds/stars_big_1024.png` in the filesystem (Bottom left of screen).
5. Drag into the scene in the centre of the screen.
6. Click on the `stars_big_1024` sprite node.
7. In the node inspector, under `Node2D Transform` set:
 - Position x: 7680
 - Position y: 7680
 - Scale x: 15
 - Scale y: 15

Run the game to verify it works.

1.4 Particle effect

The player already has a `CPUParticles2D` node made for you. Click on it. In node inspector:

- Emitting: On
- Amount: 50

Experiment with changing these settings. What do they do?

- Lifetime
- Spread
- Gravity
- Velocity
- Color
- Anything else you like

1.5 Make enemy move

Open the `enemy.tscn` scene file by double clicking it.

Right click on the `enemy` node and attach a script. Replace the contents of the script with:

```
1 extends Area2D
2
3 export var VELOCITY = 1000.0
4 export var TURNING = 0.7
5 export var FIRE_RATE = 0.01
6
7 var Bullet = preload(copypaste("res://enemy_bullet.tscn"))
8 onready var player = get_node("/root/main/player")
9
10 func _process(delta):
11     var d = player.position.angle_to_point(position)
12     rotation = Util.rotate_toward(rotation, d, TURNING*delta)
13     position += Vector2.RIGHT.rotated(rotation) * VELOCITY * delta
14
15     if position.distance_to(player.position) > 7000:
16         queue_free()
17
18     if randf() < FIRE_RATE:
19         Bullet.instance().init(self, 3000)
```

1.6 Add more enemies

NOTE: The `Light2D` node under `HUD` covers the whole screen with an invisible object (the light) and that makes it difficult to select other sprites because you always accidentally select the light. I suggest you click the eye icon next to `Light2D` to hide it. But don't forget to unhide it once you have finished positioning your sprites!

In the `main` scene, duplicate the enemy node a few times and try changing the exported variables in the node inspector so they move at different velocities.

Also try changing `Node2D Transform Rotation`.

Test the game again.

1.7 Make bullets move

Try pressing space to shoot bullets. What happens?

We already have scene files for the bullets: `player_bullet.tscn` and `enemy_bullet.tscn`.

They are both attached to the same script file, `bullet.gd`. Double click the file to edit the script and add this function:

```
1 func _process(delta):
2     position += Vector2.RIGHT.rotated(rotationcopypaste) * velocity *
    delta
3     if position.distance_to(player.position) > 5000:
4         queue_free()
```

Test the game again.

1.8 Make bullets collide

1. Open the `player_bullet.tscn` scene file by double clicking it.
2. Click `bullet` node.
3. In the inspector click `Node` at the top to view the signals.
4. Double click the `area_entered` signal.
5. Select `bullet` from the list of nodes to connect to.
6. Click `connect`.
7. Godot will create an empty function for you. Replace it with this:

```
1 func _on_bullet_area_entered(area):
2     queue_free()
```

7. Repeat steps 1-5 for `enemy_bullet.tscn` scene. (No need to edit the script and add the function again, since it's the same script and you already did it.)
8. Test the game again.

1.9 Make enemy collide

1. Open the `enemy.tscn` scene file by double clicking it.
2. Click `enemy` node.
3. In the inspector click `Node` at the top to view the signals.
4. Double click the `area_entered` signal.
5. Select `enemy` from the list of nodes to connect to.

6. Click `connect`.
7. Godot will create an empty function for you. Replace it with this:

```
1 func _on_enemy_area_entered(area):
2     $explosioncoppaste.play()
3     $AnimationPlayer.play("fade")
4     $CollisionPolygon2D.queue_free()
5     $CPUParticles2D.emitting = true
6     $CPUParticles2D.show()
7     player.score += 1
8     get_node("/root/main/HUD/score").text = str(player.score)
9     yield(get_tree().create_timer(1.0), "timeout")
10    queue_free()
```

Test the game again.

1.10 Make player collide

1. Go back to the `main.tscn` scene (should be open as a tab).
2. Click `player` node.
3. In the inspector click `Node` at the top to view the signals.
4. Double click the `area_entered` signal.
5. Select `player` from the list of nodes to connect to.
6. Click `connect`.
7. Godot will create an empty function for you. Replace it with this:

```
1 func _on_player_area_entered(area):
2     health -= 1
3     get_node("../HUD/health").value = health
4     if health <= 0:
5         get_tree().reload_copypastecurrent_scene()
6         $crash_sound.play()
7         modulate = Color(1000, 0, 0, 255)
8         yield(get_tree().create_timer(1.0), "timeout")
9         modulate = Color(1, 1, 1, 255)
```

Test the game again.

1.11 Optional - Gamepad controls

Open `player.gd` script. Delete the `gamepad` function and replace it with this:

```
1 var virtual_stick_direction = Vector2.ZERO
2
3 func gamepad(delta):
4     var input = Vector2(Input.get_joy_axis(0, 0), Input.get_joy_axis(0,
5         1)) + virtual_stick_direction
6     if input.length() > 0.2:
7         var direction = input.angle()
8         rotation = Util.rotate_toward(rotation, direction, turning *
9             delta)
```

1.12 More types of enemies

1.13 Enemy spawner

```
1 extends Timer
2
3 export (PackedScene) var Enemy
4 export var MAX_ENEMIES = 10
5 export var MAX_SCORE = 999999
6 export var MIN_SCORE = 0
7 onready var player = get_node("/root/main/player")
8
9 func _on_enemy_spawner_timeout():
10     if get_child_count() < MAX_ENEMIES && player.score <= MAX_SCORE &&
11         player.score >= MIN_SCORE:
12         var enemy = Enemy.instance()
13         add_child(enemy)
```

1.14 Enemy randomize

```
1 func _ready():
2     position = player.position + Vector2.RIGHT.rotated(rand_range(0, PI
3         *2)) * 5000
4     rotation = player.position.angle_to_point(position)
```


1.15 Charge laser

1.16 Optional - Touch controls

1.17 Title screen