

Lesson 1 Quick Reference Guide

Building a 2D Platform Game in Godot

Duration: 40 minutes

Godot Version: 4.5

What We're Building

A simple platformer where:

- Player moves left and right with arrow keys
 - Player falls with gravity and lands on platform
 - Player can fall off edges
 - Touching the goal box advances to next level
 - _____
-

Key Vocabulary

- **Node:** Building block of Godot (like a LEGO brick)
 - **Scene:** Collection of nodes working together
 - **Script:** Code that controls behavior
 - **Signal:** Event notification (like a doorbell)
 - **Collision:** When two physics objects touch
 - **Velocity:** Speed and direction of movement
 - **Delta:** Time since last frame (keeps movement smooth)
-

Project Structure

Files Created:

```
Year9_Platformer/
├── level_1.tscn      ← Main gameplay scene
├── level_2.tscn      ← Victory screen scene
└── player.gd         ← Player movement script
```

Nodes in level_1.tscn:

```
Node2D (root)
├── Platform (StaticBody2D)
│   ├── CollisionShape2D (RectangleShape2D: 400×50)
│   └── ColorRect (Size: 400×50, Position: -200,-25)

├── Player (CharacterBody2D) [SCRIPT ATTACHED]
│   ├── CollisionShape2D (RectangleShape2D: 32×48)
│   └── ColorRect (Size: 32×48, Position: -16,-24)

└── GoalBox (Area2D) [SIGNAL CONNECTED]
    ├── CollisionShape2D (RectangleShape2D: 64×64)
    └── ColorRect (Size: 64×64, Position: -32,-32)
```

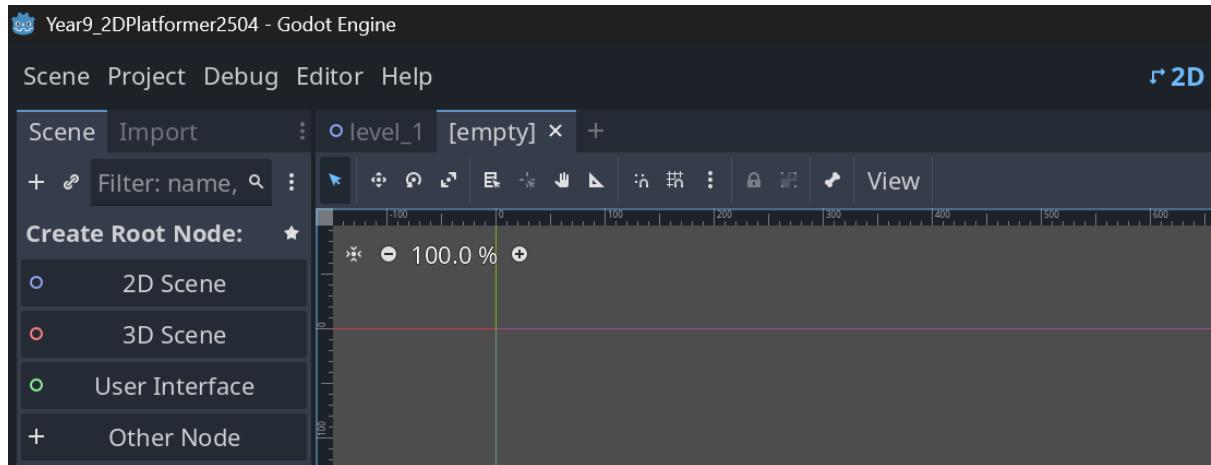
Build Steps Summary

1. Create Project (2 min)

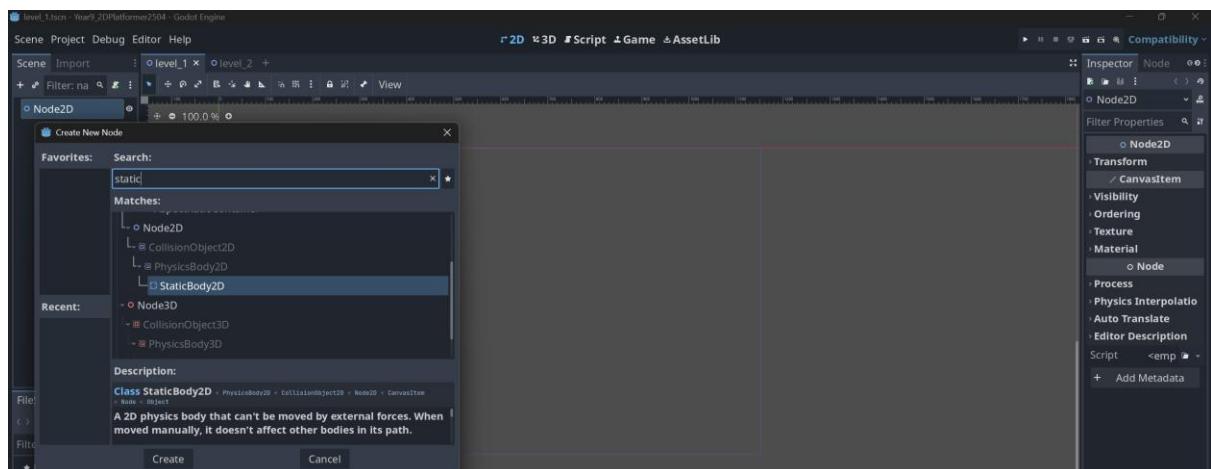
- New Project → Name: Year9_Platformer
- Renderer: Compatibility
- Save location: Desktop or Documents

2. Create level_1.tscn and level_2.tscn

1. Create Root Node → 2D Scene
2. Save as level_1
3. Click + next to level_1 to create level_2



3. Build Platform (3 min)



1. Select Other Node, Create StaticBody2D → Rename to Platform
2. Add child: CollisionShape2D → Shape: RectangleShape2D (Use scale button to size)



3. Add child: ColorRect → Size: use scale button, Position: use move button, Color:

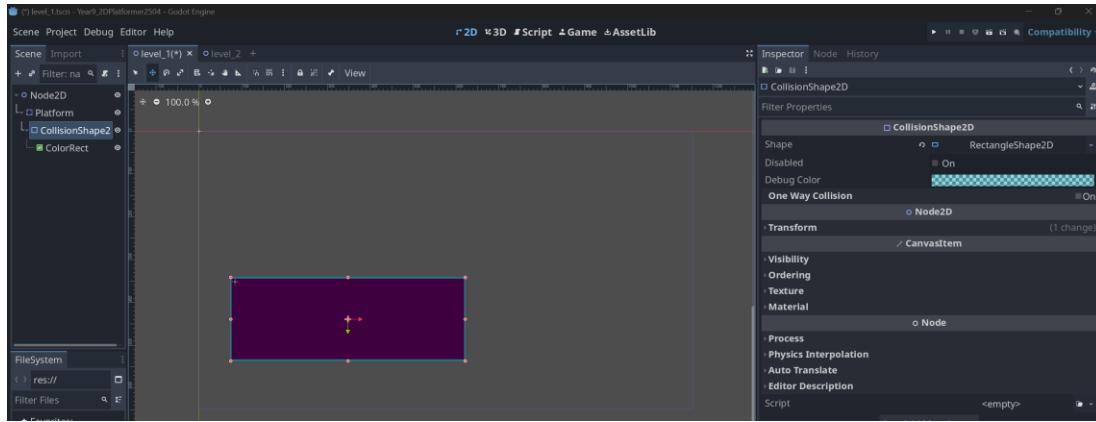
4. Position Platform: move with move button

Hints:

+ here to Create Platforms etc

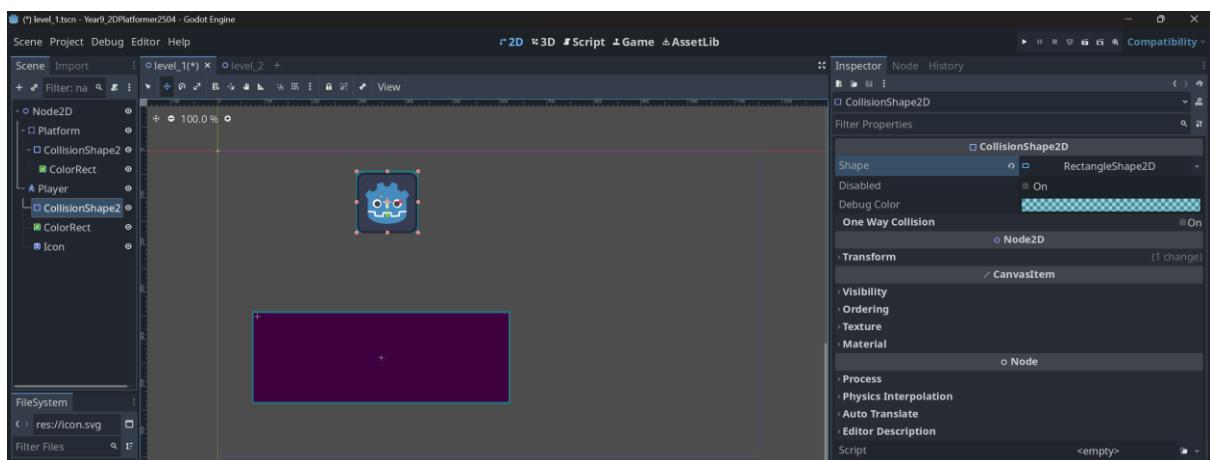
+ here to create scenes

Shape needs to be filled in

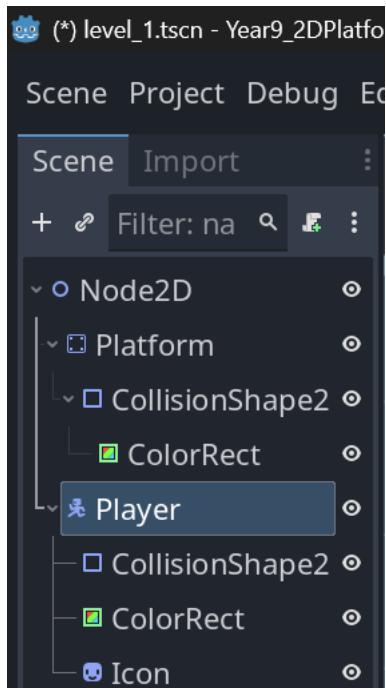


4. Build Player (5 min)

1. Create CharacterBody2D → Rename to Player
2. Add child: CollisionShape2D → Shape: RectangleShape2D
3. Add child: ColorRect → Size: resize using tools, Color: use inspector
you can use the Godot Icon as well
4. Position Player: use move tool



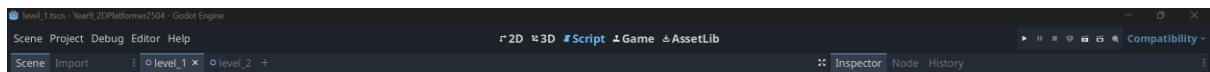
5. Attach Script (2 min)



1. Select Player → Click "Attach Script" icon – it is below Debug and next to Filter
2. Keep defaults (Template: CharacterBody2D)
3. Click Create

6. Test the game

1. Select “Run Specific Scene” (top right in window – near Compatibility)



7. Modify Script (5 min)

Replace ALL code with:

```
extends CharacterBody2D
```

```
# === MOVEMENT CONSTANTS ===  
  
const SPEED = 300.0  
  
const JUMP_VELOCITY = -400.0 # For Lesson 2!
```

```

# === MAIN PHYSICS FUNCTION ===

func _physics_process(delta):

    # --- APPLY GRAVITY ---

    if not is_on_floor():

        velocity += get_gravity() * delta


    # --- JUMP CODE (DISABLED FOR LESSON 1) ---

    # if Input.is_action_just_pressed("ui_accept") and is_on_floor():

    #     velocity.y = JUMP_VELOCITY


    # --- GET PLAYER INPUT ---

    var direction = Input.get_axis("ui_left", "ui_right")


    # --- MOVE THE PLAYER ---

    if direction:

        velocity.x = direction * SPEED

    else:

        velocity.x = move_toward(velocity.x, 0, SPEED)


    # --- EXECUTE THE MOVEMENT ---

    move_and_slide()


# === GOAL DETECTION SETUP ===

func _ready():

    var goal_box = get_parent().get_node("GoalBox")

    goal_box.body_entered.connect(_on_goal_box_body_entered)

```

```
# === GOAL COLLISION RESPONSE ===

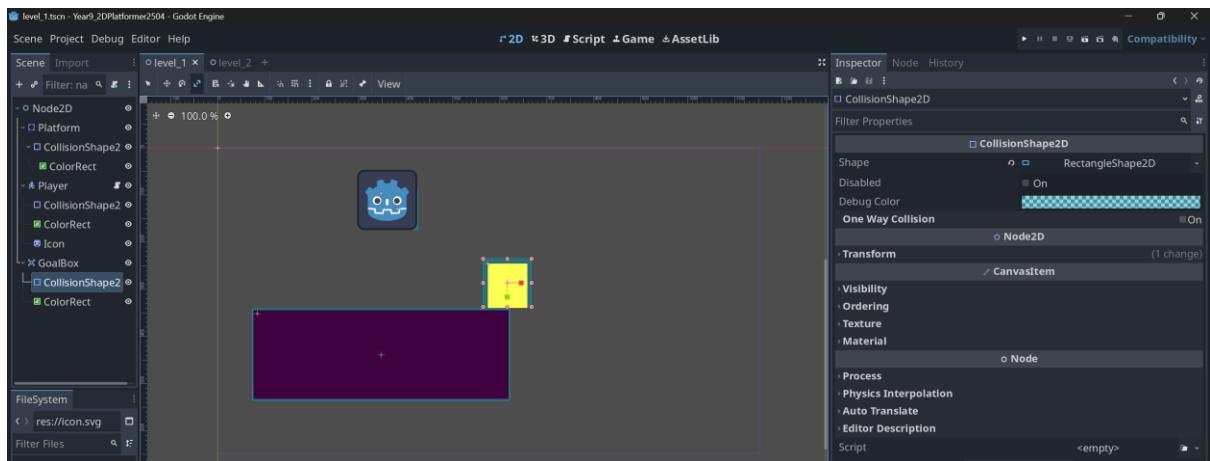
func _on_goal_box_body_entered(body):
    if body == self:
        print("Level Complete! Great job!")
        get_tree().change_scene_to_file("res://level_2.tscn")
```

Save: **Ctrl+S** (Windows) / **Cmd+S** (Mac)

6. Test Movement (2 min)

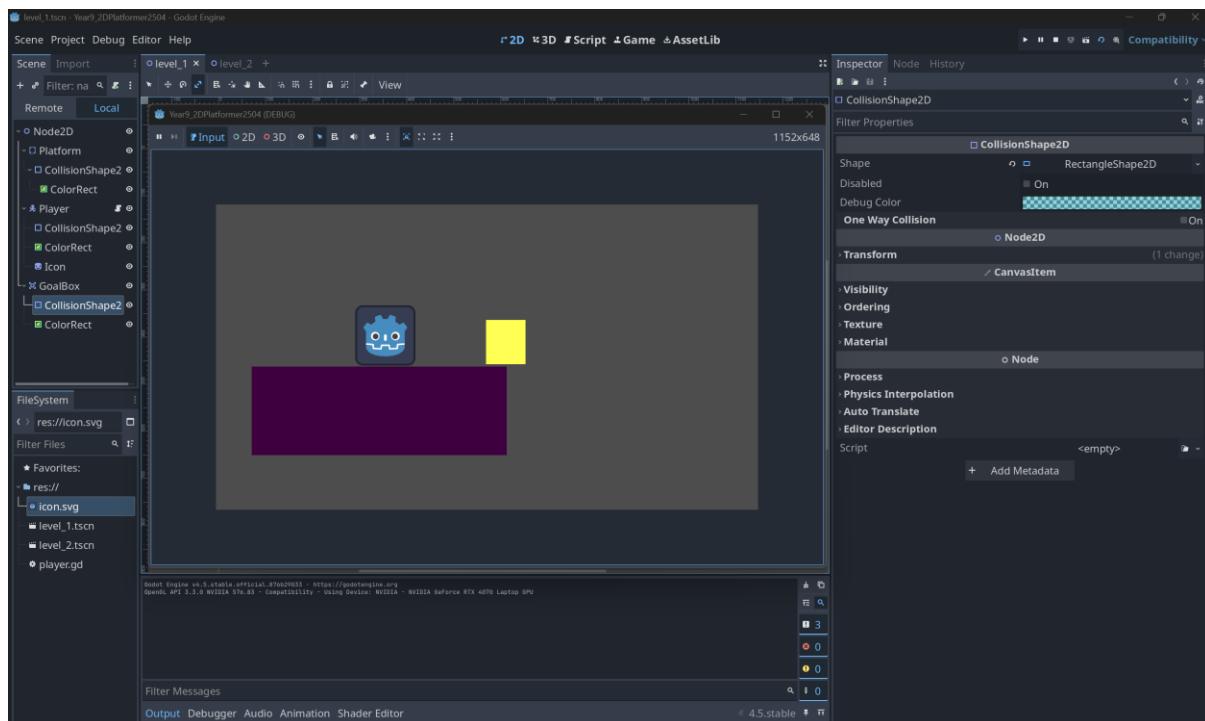
- Press **F5** to run
- Test: Left/Right arrows move, player falls and lands
- Press **F8** to stop

7. Build GoalBox (3 min)



1. Create Area2D → Rename to GoalBox
2. Add child: CollisionShape2D → Shape: RectangleShape2D (64×64)
3. Add child: ColorRect → Size: 64×64, Position: -32,-32, Color: green/gold
4. Position GoalBox: X=900, Y=490

9. Final Test (2 min)



- Press **F5**
- Walk to goal box
- Check "Level Complete!" in Output panel
- Verify scene changes to Level 2



💡 Key Concepts

Godot is Like LEGO

- **Nodes** = Individual LEGO bricks (each has a purpose)
- **Scene** = A completed LEGO model (made of many bricks)
- **Scene Tree** = Instructions showing how bricks connect
- **Script** = Special instructions that make your model move
- **Project** = Your entire LEGO collection

Node Type vs Node Name

GoalBox (Area2D)

↑ ↑

NAME TYPE

- **Type** = What it is (Area2D, CharacterBody2D, StaticBody2D)
- **Name** = What you call it (GoalBox, Player, Platform)

Parent-Child Relationships

- **Parent Node:** Has children below it (Platform, Player, GoalBox)
- **Child Node:** Belongs to a parent (CollisionShape2D, ColorRect)
- Children move with their parent!

🎮 Controls

Key	Action
Left Arrow	Move left
Right Arrow	Move right
Spacebar	Jump (disabled in Lesson 1)
F5	Run game
F8	Stop game
Ctrl+S / Cmd+S	Save

🐛 Common Problems & Fixes

Problem	Cause	Solution
Player falls through platform	Missing collision shapes	Both need CollisionShape2D
Can't see player/platform	Wrong ColorRect settings	Check size matches collision shape
Arrow keys don't work	Script not attached	Look for 📁 icon on Player node
"Node not found: GoalBox"	GoalBox wrong name/location	Check Scene tree - GoalBox at same level as Player
Game crashes at goal	level_2.tscn missing	Create level_2.tscn scene
Code errors	Typing mistakes	Check indentation, colons, spelling

Success Checklist

Before finishing, verify:

- Project saved in known location
 - level_1.tscn has Platform, Player, GoalBox
 - Player has script attached ( icon visible)
 - Left/Right arrows move player
 - Player falls and lands on platform
 - Player falls off edges
 - Touching goal shows "Level Complete!"
 - Scene changes to Level 2
 - All files saved (Ctrl+S)
-

Challenge Extensions

If you finish early, try:

1. **Speed Test:** Change SPEED to 500 or 150 - what feels best?
 2. **More Platforms:** Duplicate Platform (Ctrl+D) and create stairs
 3. **Custom Colors:** Make your game unique with different colors
 4. **Danger Zone:** Create a red box that restarts the level
 5. **Comments:** Add your own code comments explaining what you learned
-

What's Next: Lesson 2 Preview

Coming next lesson:

-  Uncomment the jump code
-  Learn about velocity and negative Y values
-  Make jumping feel amazing
-  Experiment with jump height and gravity

Think about: What games have really satisfying jumping?