Lesson 3: Collecting Your First Coin

What We're Learning Today

- How collision detection works in games
- How to make objects interact with each other
- How to remove objects from the game world
- How to use functions that get called automatically

What We're Building

By the end of this lesson, your player will be able to walk over coins and collect them. When you touch a coin, it will disappear and print a message to the console confirming you collected it!

Today's New Programming Concept

Collision Detection - This is how games know when two objects touch each other. In Godot, we use special areas that can detect when other objects enter them, like an invisible bubble around our coin.

Part 1: Understanding the Code

Step 1: Look at What We Have

Open your `scripts/coins.gd` file:

```
extends Node

func _on_body_entered(body):
    pass
```

This is a special function that Godot calls automatically when something touches the coin's area. Right now it does nothing (`pass`).

Step 2: Understand Collision Areas

In your coin scene, there's an Area2D node with a CollisionShape2D. This creates an invisible area around your coin:

- When the player enters this area, `_on_body_entered()` gets called
- The `body` parameter tells us what object entered (hopefully our player!)

Step 3: Predict What Should Happen

Think about what we want:

- Player walks into coin area → coin disappears
- We should probably check that it's actually the player who touched the coin

- We should give some feedback that collection happened

Part 2: Learning Collision Detection

How Area Detection Works

```
func _on_body_entered(body):
    # 'body' is whatever object entered our area
    # We need to check if it's the player
    # Then do something (like disappear)
```

The queue_free() Function

`queue_free()` is Godot's way of safely removing an object from the game:

- It marks the object to be deleted
- Godot removes it at a safe time
- The object disappears from the screen

Checking Object Identity

We can check if the touching object is our player:

```
if body.name == "Player":
    # This is definitely our player!
```

Part 3: Building Coin Collection

Step 1: Identify the Player

In your `_on_body_entered` function, you need to check if the object that touched the coin is actually the player.

Hint: Use an if statement to check if `body.name == "Player"` (or whatever your player node is named).

Step 2: Collect the Coin

When the player touches the coin, two things should happen:

1. Print a message to confirm collection
2. Remove the coin from the game

Hints:

- Use `print()` to show a collection message
- Use `queue_free()` to remove the coin (note: call it on `self` since the coin removes itself)

Step 3: Test Your Collection

Run your game and walk over a coin. You should see:

- A message in the console when you collect it
- The coin disappears from the screen
- You can only collect each coin once (because it's gone!)

Part 4: Understanding What Happened

Step 1: The Collision Process

When you walk over a coin, here's what happens:

1. Player enters the coin's Area2D
2. Godot automatically calls `_on_body_entered(body)`
3. Your code checks if it's the player
4. If yes, print message and remove coin

Step 2: Why Check the Player?

Other objects might touch the coin too (like enemies later). We only want the player to collect coins, so we check the name first.

Step 3: The Magic of queue_free()

`queue_free()` is safer than immediately deleting because:

- It waits until Godot is ready
- It prevents crashes from deleting objects mid-process
- It's the "polite" way to remove objects

Part 5: Practice and Testing

Test 1: Basic Collection

Walk your player over different coins in the level:

Expected behavior:

- Console message appears when touching coin
- Coin disappears immediately
- Each coin can only be collected once

Test 2: Selective Collection

Try to understand what happens if other objects existed:

- Only the player should be able to collect coins
- The name check prevents accidental collection

Test 3: Multiple Coins

If there are multiple coins in your level:

- Each works independently
- Collecting one doesn't affect others
- All should have the same collection behavior

Common Problems and Solutions

Problem: Nothing happens when touching coin

Solution: Check that your Area2D node has the signal connected to `_on_body_entered`. In the coin scene, the Area2D should have its `body_entered` signal connected to this function.

Problem: Coin disappears but no message prints

Solution: Your `queue_free()` is working but the name check might be wrong. Print `body.name` to see what name you're actually getting.

Problem: Message prints but coin doesn't disappear

Solution: Make sure you're calling `queue_free()` on the coin itself. Usually this is `queue_free()` or `self.queue_free()`.

Problem: Error when collecting coin

Solution: The signal might not be connected properly. Check the coin scene's Area2D node connections.

Debugging Tips

Add these print statements to understand what's happening:

```
func _on_body_entered(body):
    print("Something touched the coin: ", body.name)
    # Your collection code here
```

This helps you see:

- What objects are touching the coin
- Whether your name check is working
- If the function is being called at all

What We Accomplished

🎉 Excellent work! You just learned:

- **Collision detection** using Area2D and signals
- **Automatic function calls** that respond to game events
- **Object identification** by checking names
- **Object removal** using queue_free()
- **Game interaction** between player and world objects

Understanding Signals (Bonus Concept)

The `_on_body_entered` function is connected to a **signal**. Signals are Godot's way of saying "Hey! Something happened!" When a body enters the area, Godot sends a signal, and our function responds to it.

Next Time Preview

In our next lesson, we'll create dangerous spike traps that hurt the player! We'll learn how to apply the same collision detection concepts but instead of collecting something good, we'll take damage from something bad.