Unity Lab Worksheet 2

SWE402: Game Programming

Objective: Create a simple collectibles game featuring a character with physics, background audio, collectible items with rotation and VFX feedback, and an ambient environment built from prefabs.

Part 1: Scene & Character Setup

■ 1.1 Create a New Scene

Create a new 3D scene (File → New Scene → Basic 3D). Save it with a descriptive name in your project's Scenes folder.

■ 1.2 Build Your Character

Create a character using 3D primitives (GameObject → 3D Object).

Hint: Parent multiple primitives under an empty GameObject to group them as one character.

■ 1.3 Add Physics Components

Select your character's root GameObject and add:

- **Rigidbody** component — enables physics simulation (gravity, forces)
- **Collider** component — defines physical boundaries (Box, Sphere, or Capsule Collider)

✓ **Checkpoint 1:** Character visible in scene with Rigidbody & Collider attached

Part 2: Audio Setup

■ 2.1 Add Background Music

Create an empty GameObject named 'BackgroundMusic'. Add an **Audio Source** component and assign an audio clip. Enable **Loop** and **Play On Awake**.

Hint: Keep Spatial Blend at 0 (2D) for background music so volume stays constant.

■ 2.2 Add Ambient Sounds

For extra immersion, add additional Audio Sources with environmental sounds positioned in 3D space (set Spatial Blend to 1 for 3D audio).

Part 3: Environment Design

■ 3.1 Create an Ambient Scene

Build an environment using existing prefabs from your project **Prefabs** folder. Add a ground plane, decorative objects, lighting, and any other elements that fit your theme. **Be creative!**

✓ **Checkpoint 2:** Scene has background music playing and creative environment setup

Part 4: Character Movement Script

■ 4.1 Create PlayerController Script

Create a new C# script called 'PlayerController'. Attach it to your character.

■ 4.2 Implement Movement

Your script should read input (WASD or Arrow keys) and move the character using the Rigidbody. Include public variables for **speed** and **rotationSpeed** so you can adjust them in the Inspector.

Key concepts: Input.GetAxis("Horizontal"/"Vertical"), Rigidbody.AddForce() or Rigidbody.MovePosition(), FixedUpdate() for physics.

Part 5: Collectible System

■ 5.1 Create a Collectible Object

Create a simple 3D object (Sphere, Cube, or custom shape) to serve as your collectible. Add a **Collider** with **Is Trigger** enabled.

■ 5.2 Create Collectible Script

Create a new C# script called 'Collectible'. The script should:

- **Rotate** the object continuously (use Transform.Rotate in Update)
- **Detect** when the player enters the trigger (OnTriggerEnter)
- **Spawn VFX** when collected (Instantiate a particle prefab)
- **Destroy** the collectible after collection

Hint: Use a public GameObject variable to reference your VFX prefab in the Inspector.

■ 5.3 Add VFX Prefab

Create or import a Particle System for the collection effect. Save it as a prefab and assign it to your Collectible script.

Part 6: Prefab & Scene Population

■ 6.1 Create Collectible Prefab

Drag your completed collectible from the Hierarchy into the Project window (Prefabs folder) to create a prefab.

■ 6.2 Populate the Scene

Place multiple instances of your collectible prefab throughout the scene. Position them creatively to encourage exploration!

Hint: Hold Ctrl while dragging to snap objects to the grid.

✓ **Final Checkpoint:** Character moves with input, collectibles rotate, VFX plays on collection, collectibles disappear when gathered