# PROJECT REPORT

**Simple 3D Platformer Game**

Develop a basic 3D platformer game where players control a character to navigate through a level, collect coins, and score points.

Mithun Rajapaksha
mithunsandaruwanabc@gmail.com

# Contents

## Introduction

This project involves the development of a basic 3D platformer game using Unity, aimed at demonstrating skills in game development, C# programming, and problem-solving. This project is submitted as part of the application process for the Research Engineer Internship position at Expernetic LLC. The objective of this assignment is to evaluate my proficiency in using Unity, implementing game mechanics, and overcoming technical challenges in game development.

The goal of this project is to create an engaging and functional 3D platformer game where players control a character to navigate through levels, collect coins, and score points. The game is designed to include smooth player movement, interactive collectibles, and a dynamic user interface displaying the player's score.

## Tools and Technologies

1. **Unity** : Game development platform used to build the 3D environment, implement game mechanics, and manage assets.
2. **C#:** Programming language used for scripting player controls, game logic, and interactions.
3. **Visual Studio**: Integrated development environment (IDE) used for writing and debugging C# scripts.
4. **Mixamo**: Source for free 3D character models and animations.
5. **Kenney.nl**: Source for free 3D models used for coins and other game assets.

## Key Features

1. **Player Moovement:** The character can move in all directions and jump, using smooth and responsive controls implemented via C# scripts.
2. **Collectibles:** Coins are placed throughout the level for the player to collect, increasing their score upon collection.
3. **User Interface:** A dynamic UI element displays the player's current score in real-time, updating as coins are collected.
4. **Level Design:** A thoughtfully designed level with platforms and strategically placed coins to create an engaging gameplay experience.

## Development Process

1. **Learning Unity**

   Referred unity documentation and YouTube tutorials. These resources were invaluable in helping me understand the basics of Unity, 3D rendering, and game development concepts.

2. **Initial Setup**
   1. Scene Selection

   I began by selecting a scene from the Unity Asset Store to serve as the foundation for my game environment. This provided a rich starting point with pre-built assets and textures. I Adjusted the scene as fit for my gaming enviromnment.

   2. Character and Animation

   I chose a character from Mixamo.com, along with the necessary animations for movement and actions. This included animations for walking, running, jumping, and idling.(AJ character)

3. **Animation Transition**

   Creating smooth animation transitions posed a significant challenge. Setting parameters and blending animations required careful adjustments to avoid abrupt or unnatural movements. I spent several days experimenting with different configurations and fine-tuning the Animator Controller in Unity.

4. **Handling Character movements**
5. **Adding collectables**
6. **Create User Interface**
   1. Score Board
   2. Win/Lost message
   3. Timer
   4. Restart Button
7. **Debugging and testing**

   Debugged and tested done throughout the project.

## Challenges Encountered

1. **Lack of Expertise with the Domain**

   Coming into this project, I had no prior experience with Unity or 3D game development. To overcome this, I dedicated a significant amount of time to self-study using online resources. YouTube tutorials and the Unity documentation became my primary sources of learning. This initial learning curve was steep, but it provided a solid foundation to proceed with the project.

2. **Dealing with the unity free assets.**

   Using free assets from the Unity Asset Store posed its own set of challenges. Integrating these assets into the project required understanding how to properly import, configure, and utilize them within the Unity environment. I had to learn to navigate the Asset Store efficiently, choose appropriate assets for my game, and adapt them to fit the project requirements. This involved modifying asset properties and ensuring compatibility with my game design.

3. **Animating the Character Movements**

   Animating the character using Mixamo.com models and animations was a complex task. Creating smooth transitions between different animations (e.g., walking to jumping) required careful adjustment of parameters within the Animator Controller. This process was particularly challenging:

   - Ensuring smooth transitions without jitter or abrupt movements.
   - Setting up appropriate triggers and conditions for animation states.
   - Spending several days experimenting with different configurations to achieve the desired fluidity.

4. **Placing the  main camera object with the player.**

   Aligning the main camera with the player to create a cohesive third-person perspective was another challenge. The camera needed to follow the player smoothly while maintaining an optimal view of the surroundings. This involved:

   - Adjusting the camera's position and rotation relative to the player.
   - Implementing a camera follow script to dynamically adjust the camera's movement based on the player's actions.
   - Ensuring the camera transitions smoothly when the player moves, jumps, or changes direction.

5. **Handling Player Physics**

   Implementing realistic physics for the player character involved configuring the Rigidbody component and managing collisions. This included:

   - Ensuring the player remains stable on different surfaces.
   - Adjusting the Rigidbody properties such as mass, drag, and gravity to achieve natural movement and interactions.
   - Debugging issues where the player might get stuck or exhibit unnatural behavior due to physics calculations.

6. **Making Proper Animation Transitions**

   Creating seamless transitions between different animations was one of the more technically demanding tasks. This required:

   - Setting up parameters within the Animator Controller to manage state changes.
   - Ensuring transitions are triggered correctly based on player inputs and game conditions.

## Overcoming the challenges

1. **Continuous Learning**

   I regularly referred to Unity's official documentation, forums, and community tutorials to gather insights and best practices.

2. **Experimentation and Debugging**

   I used extensive debugging and iterative testing to fine-tune game mechanics and resolve issues. Debug logs were particularly helpful in tracking the state of animations and physics interactions.

   Example: debug code to check the Animator State

```
            void DebugCurrentAnimatorState()
{
    AnimatorStateInfo stateInfo = animator.GetCurrentAnimatorStateInfo(0);
    Debug.Log("Current State: " + stateInfo.fullPathHash);



        if (stateInfo.fullPathHash == idleStateHash)
        {
            Debug.Log("Current State: Idle");
        }
        else if (stateInfo.fullPathHash == walkingStateHash)
        {
```

```
            Debug.Log("Current State: Walking");
        }
        else
        {
            Debug.Log("Current State: Unknown");
        }
    }
```

3. **Incremental Development**

   Breaking down complex tasks into smaller, manageable steps allowed me to focus on one issue at a time and progressively build up the game's functionality.

# References


1. https://youtu.be/N3jOOm--TTg?si=tN7m3Z9CsXYfx4VW
2. https://docs.unity.com/
3. https://www.mixamo.com/#/
4. https://kenney.nl/assets
   s