MINOR PROJECT PLATFORMER GAME

BY- ASHISH DHARVE

UNI. ROLL NO.-22020107002

COLLEGE ROLL NO.- 20222719

GITHUB REPOSITORY

INTRODUCTION

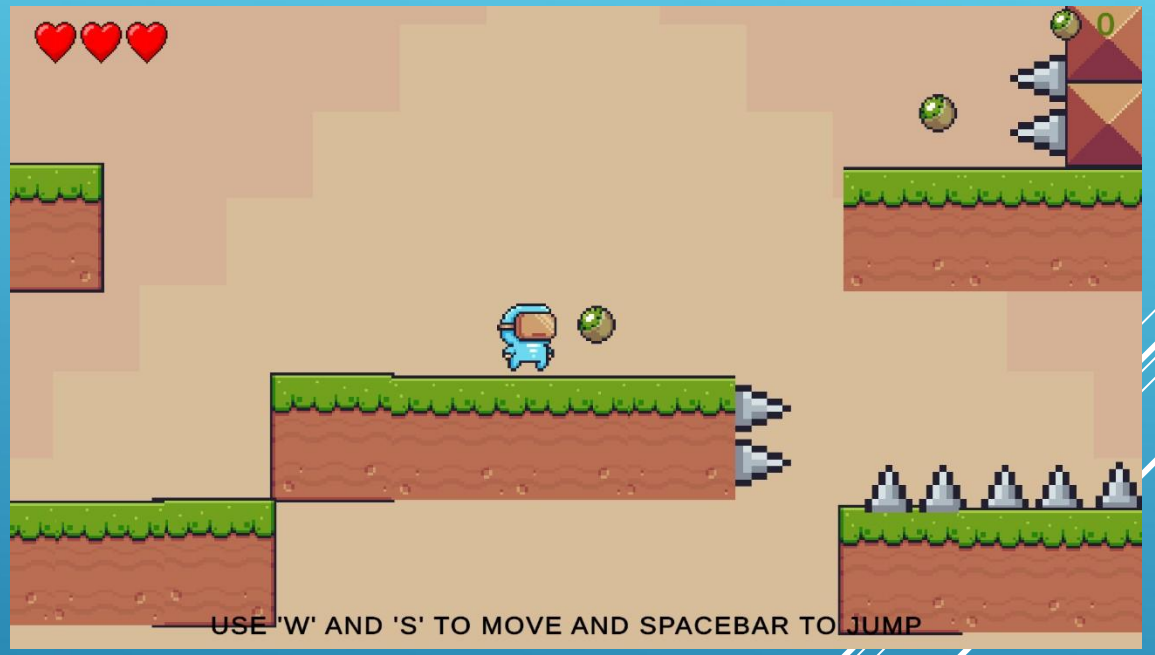
HopBunny is a 2D platformer game developed using Unity, with player controlled character with obstacles filled levels.

Key Elements:

- Player controls and interactions.
- Hazard Mechanics
- Platform Dynamics
- Real Time Physics
- Seamless Animation

OBJECTIVE OF THE GAME

- **Primary Goal:** Provide a fun and engaging 2D platformer experience.
- **Player Goals:**
- Navigate through levels while avoiding traps and hazards.
- Collect as many kiwis as possible to increase score.
- Reach the goal site to clear the level and advance to the next stage.



FEATURES

Player Movement: Walking and Running: Control the player with arrow keys or WASD.

Jumping: Leap between platforms and over obstacles using Spacebar.

MOVING SAW

Level Design: Multiple Levels: Increasing difficulty with each level.

- Moving Platforms: Horizontal, vertical, and rotating platforms.
- Traps and Hazards: Spikes, moving saws, and spikeheads.

Collectibles:

- Kiwis: Gather them to increase your score.

KIWI

UI and HUD:

- Health Bar: Tracks remaining health.

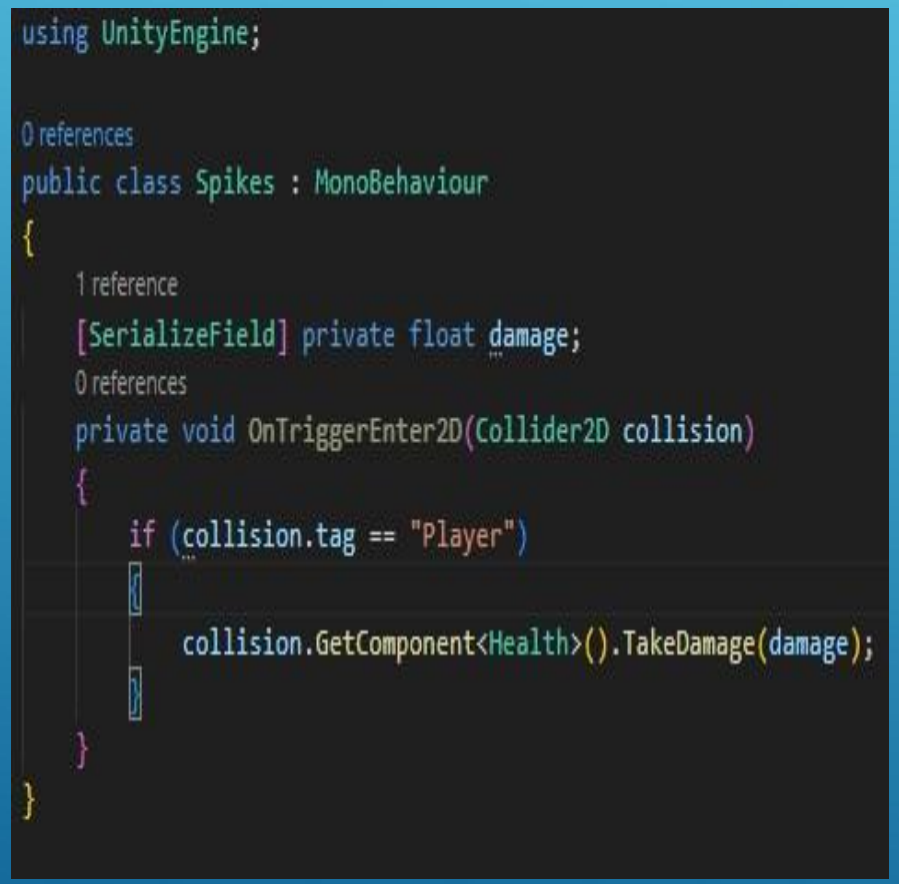
  Score Counter: Displays the player's score.

HEALTHBAR

TECHNOLOGY AND SCRIPTS

Technology Used:**Unity:** Game engine for designing levels, implementing mechanics, and managing assets.

- **C#:** Primary scripting language for game logic and mechanics.
- **Git:** Version control system for collaboration and project management.



```
using UnityEngine;

0 references
public class Spikes : MonoBehaviour
{
    1 reference
    [SerializeField] private float damage;
    0 references
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.tag == "Player")
        {
            collision.GetComponent<Health>().TakeDamage(damage);
        }
    }
}
```

Key Scripts:

- **PlayerController Script:** Handles player movement and interactions.
- **LevelController Script:** Manages level transitions using Unity's SceneManager.
- **Spike Script:** Detects player collision with spikes.

SYSTEM ARCHITECTURE

- **Player:** Controlled by PlayerController script for movement and jumping.
- **Environment:** Platform (static and moving).

Traps and hazards(spikes, saws) managed by respective scripts.

- **UI :** Health Bar to show player's remaining lives.

  Score Counter for collected Kiwis.
- **Game Progression:** Managed by LevelController Script.

HOW TO RUN THE GAME

Prerequisites:

•Unity Hub and Unity version **2022.3.45f1**. or higher.

•Git for cloning the repository.

Steps to Run the Game:

1. Clone the Repository: https://github.com/ashishdharve09/Game_College_Project.git
2. **Open the Project in Unity:**
   - Launch Unity Hub.
   - Click "Add" and select the project folder.
   - Wait for Unity to load the project.

3. Play the Game: Press the "Play" button in the Unity Editor to start.

Alternative :

You can directly play the game without unity, just download this github file : Game

FUTURE SCOPE

- Add more levels with diverse challenges and mechanics.
- Introduce new collectibles and power-ups for enhanced gameplay.
- Implement advanced features such as multiplayer mode or leaderboard integration.
- Optimize the game for mobile and web platforms for broader accessibility.