Project 1

Members: Jeremy Braneke-Hite

1. Scripts/Components:

1.1. PlayerMovement:

- [ ] Responsible for controlling the player's horizontal movement and jump.
- [ ] Attached to the Player GameObject.

1.2. StarPickup:

- [ ] Allows the player to collect star objects in the scene.
- [ ] Attached to each Star GameObject.

2. GameObject Composition:

2.1. Player:

- [ ] Components:
    - [ ] SpriteRenderer: Displays the player sprite.
    - [ ] Rigidbody2D: Enables physics-based movement and collision.
    - [ ] BoxCollider2D: Handles collision detection.
    - [ ] PlayerMovement Script: Controls movement behavior.

2.2. Star:

- [ ] Components:
    - [ ] SpriteRenderer: Displays the star sprite.
    - [ ] BoxCollider2D (set as trigger): Detects when the player overlaps with the star.
    - [ ] StarPickup Script: Defines the pickup behavior.
    - [ ] ObstacleHandler Script: Controls Star Movement

2.3. Ground:

- [ ] Components:

- [ ] SpriteRenderer: Displays the ground sprite.
- [ ] BoxCollider2D: Allows the player to stand on the ground.

2.4. Obstacle:

- [ ] Components:
    - [ ] SpriteRenderer: Displays the obstacle sprite.
    - [ ] BoxCollider2D: Handles collision detection.
    - [ ] ObstacleHandler Script: Controls obstacle movement.

3. GameObjects Using Custom Components:

- [ ] Player uses PlayerMovement.
- [ ] Each Star uses StarPickup.

4. Scenes:

4.1. MainMenuScene:

A simple scene with a title and a button to start the game.

- [ ] GameObjects:
    - [ ] Title Text: Displays the game's title.
    - [ ] Start Button: Click to move to the MainGameScene.

4.2. MainGameScene:

The main gameplay scene.

- [ ] GameObjects:
    - [ ] Player: The user-controlled character.
    - [ ] Stars: Collectible objects.
    - [ ] Ground: Platforms for the player to stand on.

5. Expected Interactions:

Player with Ground: The player can move left or right on the ground.
Player with Star: When the player overlaps with a star, the star is collected and disappears from the scene.

6. Source Code:

ObstacleHandler

```
using UnityEngine;

public class ObstacleHandler : MonoBehaviour
{
    public Vector2 moveDirection = Vector2.left;
    public float moveSpeed = 2.0f;

    private void Update()
    {
        transform.Translate(moveDirection * moveSpeed * Time.deltaTime);
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Player"))
        {


UnityEngine.SceneManagement.SceneManager.LoadScene(UnityEngine.SceneManagement.Sc
eneManager.GetActiveScene().name);
        }
    }
}
```

PlayerMovement

```
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
    private Rigidbody2D rb;
    public float speed = 5.0f;
    public float jumpForce = 5.0f;

    private void Start()
    {
        rb = GetComponent<Rigidbody2D>();
```

```
    }

    private void Update()
    {
        float moveX = Input.GetAxis("Horizontal");
        rb.velocity = new Vector2(moveX * speed, rb.velocity.y);

        if (Input.GetKeyDown(KeyCode.Space))
        {
            rb.velocity = new Vector2(rb.velocity.x, jumpForce);
        }
    }
}
```

StarCollector

```
using UnityEngine;

public class StarCollector : MonoBehaviour
{
    private GameManager gameManager;

    private void Start()
    {
        gameManager = FindObjectOfType<GameManager>();
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Star"))
        {
            Destroy(collision.gameObject);
            gameManager.AddScore(1);
        }
    }
}
```

GameManager

```
using UnityEngine;

public class GameManager : MonoBehaviour
{
    private int score = 0;

    public void AddScore(int points)
    {
        score += points;
        Debug.Log("Score: " + score);
    }


}
```

SceneLoader

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class SceneLoader : MonoBehaviour
{
    public void LoadScene()
    {
        SceneManager.LoadScene("MainGameScene");
    }
}
```