2a Adding a Fly-squito - Banana Mayhem

Nicholas & Aikens

July 26, 2025

1 FlyMovement.cs

Purpose: Chases the player and triggers a "death" (scene reload) on contact.

1. In `Start()`, find and cache the `Transform` of the GameObject tagged "Player".
2. In `Update()`, compute the normalized direction vector toward the player and translate the fly each frame by `speed * Time.deltaTime`.
3. In `OnTriggerEnter2D()`, detect collision with the player and reload the active scene.

```
// FlyMovement.cs
using UnityEngine;
using UnityEngine.SceneManagement;

[RequireComponent(typeof(Rigidbody2D))]
public class FlyMovement : MonoBehaviour
{
        [Tooltip("Chase␣speed␣in␣units␣per␣second")]
        public float speed = 3f;

        private Transform player;

        void Start()
        {
                // Find the Player by tag in the scene
                GameObject playerGO = GameObject.
                   FindWithTag("Player");
                if (playerGO != null)
                player = playerGO.transform;
                else
                Debug.LogError("No␣GameObject␣with␣tag␣'
                   Player'␣found.");
        }

        void Update()
        {
                if (player == null) return;

                // Compute direction toward player
                Vector2 direction = (player.position -
                   transform.position).normalized;
                // Move the fly
                transform.Translate(direction * speed *
                   Time.deltaTime);
        }

        void OnTriggerEnter2D(Collider2D other)
        {
                if (other.CompareTag("Player"))
                {
                        // Reload current scene => player
                           dies
                        SceneManager.LoadScene(
                        SceneManager.GetActiveScene().
                           buildIndex);
                }
        }
}
```

2 FlySpawner.cs

Purpose: Instantiates one fly at each designated spawn point when the scene starts.

1. Expose a `GameObject flyPrefab` field for the Fly prefab.
2. Expose a `Transform[]` array of spawn points.
3. In `Start()`, loop through each spawn point and `Instantiate` the prefab at its position.
4. Log a warning if no prefab or spawn points are assigned.

```
// FlySpawner.cs
using UnityEngine;

public class FlySpawner : MonoBehaviour
{
        [Tooltip("Assign␣your␣Fly␣prefab␣here")]
        public GameObject flyPrefab;

        [Tooltip("These␣Transforms␣mark␣where␣flies␣will␣
           appear")]
        public Transform[] spawnPoints;

        void Start()
        {
                if (flyPrefab == null || spawnPoints.
                   Length == 0)
                {
                        Debug.LogWarning(
                        "FlySpawner␣needs␣a␣prefab␣and␣at␣
                           least␣one␣spawn␣point.");
                        return;
```

```
                                }

                                foreach (Transform point in spawnPoints)
                                {
                                        Instantiate(flyPrefab, point.
                                           position, Quaternion.identity);
                                }
                        }
                }
```

3 Scene Setup

1. **Player GameObject**:
   - Must be tagged `"Player"`.
   - Should have a `Collider2D` (e.g., `CircleCollider2D`) and a `Rigidbody2D`.
2. **Fly Prefab**:
   - Add your fly sprite as a GameObject.
   - Attach a `CircleCollider2D` (set `Is Trigger` = true).
   - Attach a `Rigidbody2D` (Body Type = `Kinematic`, Gravity Scale = 0).
   - Attach the `FlyMovement.cs` script.
   - Tag the prefab (or its instances) as `"Enemy"` if desired.
3. **Spawn Points**:
   - Create empty GameObjects at positions where you want flies to appear.
   - Assign these to the `spawnPoints` array on your `FlySpawner` component.
4. **Fly Spawner**:
   - Create an empty GameObject named `"FlySpawner"`.
   - Attach the `FlySpawner.cs` script.
   - Drag in the Fly prefab and populate the spawn points.
5. **Testing**:
   - Press Play. Flies should home in on the player.
   - On contact, the scene reloads, simulating player death.