Game over screen:

-textmeshpro, “game over”

Create a public class under MonoBehaviour

//Start of gameover(GO) screen set up. Use TextMeshPro text element: wrapping "disabled" do prevent stacking.

public TextMeshProUGUI gameOverText;

void Start() {

//Using Boolean to set true so we can set gameObject back to active when conditions are set.

gameOverText.gameObject.SetActive(true); }

//Creating a public void method to set up with sensors to activate GO on conditions of losing.

// under the "public void UpdateScore(int scoreToAdd) :

public void GameOver()

{

gameOverText.gameObject.SetActive(true);

}

private void OnTriggerEnter(Collider other)

{

Destroy(gameObject);

if (!gameObject.CompareTag("Bad")) { gameManager.GameOver(); }

}

// public bool IsGameActive: Stop spawning and score on GO.

public bool isGameActive;

void Start()

{... isGameActive = true; }

public void GameOver() { ... isGameActive = false; }

IEnumerator SpawnTarget() { while isGameActive) { ... }}

private void OnMouseDown() //not sure if we have that script acrtive, so including just in case Ty.

{

if (gameManager.isGameActive) { ... [all function code moved inside] }}

//Adding a 'Restart' button(RB).

//Canvas - UI -> Button (or TMPro) > reactivate GO text to reposition restart button, then deactivate. Adjust font, size etc.

/// 'Restart' button script:

using UnityEngine.SceneManagement;

public void RestartGame() {

SceneManager.LoadScene(SceneManager.GetActiveScene().name); }

//At top of GameManager.cs add "using UnityEngine.UI;" > declare public Button restartButton; > assign in inspector.

//Uncheck "active" on Restart button in inspector > in GameOver function, activate RB.

using UnityEngine.UI;

public Button restartButton;

public void GameOver() { ...

restartButton.gameObject.SetActive(true); }

difficulty scripts: