



Building simple systems - High Scores & Saving data

Task 1. Edit AddScore script

Explanation

- When the player dies, the scene changes to the GameOver scene
- We want to make sure the zombies that the player did not kill will not give the player score when the scene changes
- To do this we need to make a couple of minor changes to the AddScore script

Do this

- In the **Project view**, open the **AddScore** Script

Do this

- **REMOVE** the highlighted lines of code to your script

```
using UnityEngine;

public class AddScore : MonoBehaviour {

    public delegate void SendScore(int theScore);
    public static event SendScore OnSendScore;

    public int score = 10;

    private void OnDestroy() {
        if(OnSendScore != null) {
            OnSendScore(score);
        }
    }
}
```

Do this

- **ADD** the highlighted lines of code to your script

```
using UnityEngine;

public class AddScore : MonoBehaviour {

    public delegate void SendScore(int theScore);
    public static event SendScore OnSendScore;

    public int score = 10;
    private bool scoreSent = false;

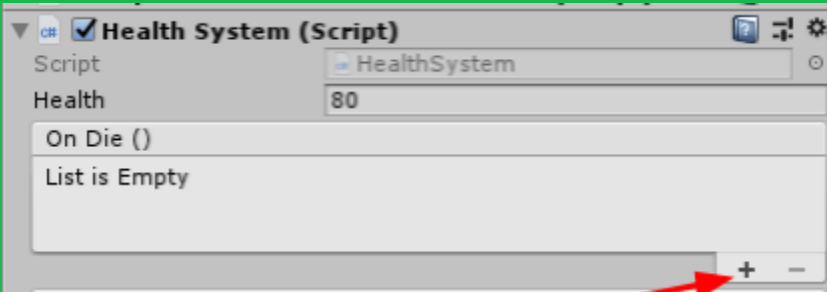
    public void OnAddScore() {
        if(OnSendScore != null) {
            if(!scoreSent) {
                scoreSent = true;
                OnSendScore(score);
            }
        }
    }
}
```

Do this

- In the **Project view**, open the **Zombie** prefab

Do this

- In the **Inspector**, add a listener to the **On Die** event on the zombies **HealthSystem**



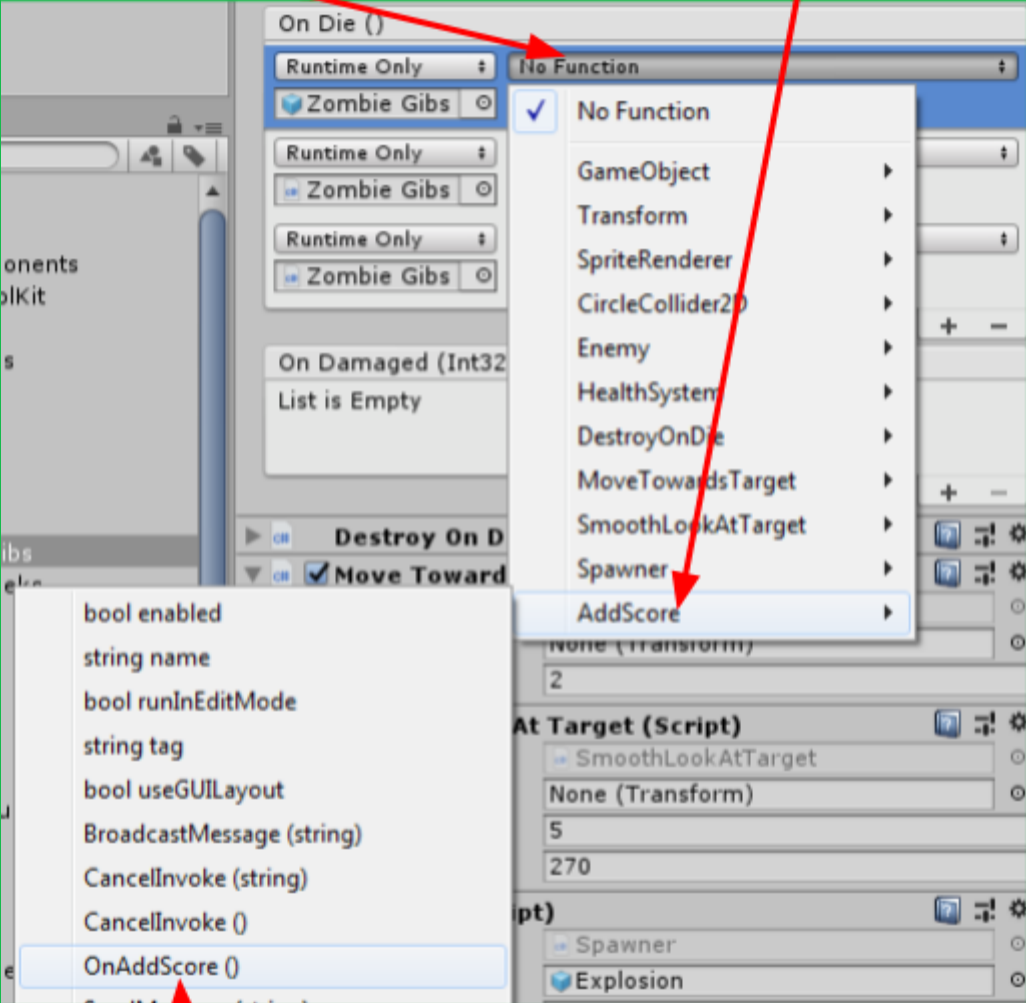
Add a listener to the On Die event

Do this

- Drag the zombie from the **Project view** onto the inlet of the **On Die** event

Click the function dropdown

Select AddScore



Select OnAddScore

Task 2. Edit GameUI script

Explanation

- To get our player to the highscore table when the player dies, we will save the current score to PlayerPrefs
- PlayerPrefs will save our score to disc and allow us to access it in other scenes, like the Game Over scene

Do this

- In the **Project view**, open the **GameUI** Script

Do this

- **ADD** the highlighted lines of code to your script

```
using UnityEngine;
using UnityEngine.UI;

public class GameUI : MonoBehaviour {

    public Slider healthBar;
    public Text scoreText;

    public int playerScore = 0;

    private void OnEnable() {
        Player.OnUpdateHealth += UpdateHealthBar;
        AddScore.OnSendScore += UpdateScore;
    }

    private void OnDisable() {
        Player.OnUpdateHealth -= UpdateHealthBar;
        AddScore.OnSendScore -= UpdateScore;
        PlayerPrefs.SetInt("Score", playerScore);
    }

    private void UpdateHealthBar(int health) {
        healthBar.value = health;
    }

    private void UpdateScore(int theScore) {
        playerScore += theScore;
        scoreText.text = "SCORE: " + playerScore.ToString();
    }
}
```

Useful links

- More information about **PlayerPrefs.SetInt** [PlayerPrefs.SetInt - Scripting reference](#)

Task 3. Create HighScoreUI script

Explanation

- When the Game Over scene loads we want to display all the high scores

Do this

- In the **Project view**, create a new C# script in the **Scripts** folder
- Name the script **HighScoreUI**
- Double click the the **HighScoreUI** script to open for editing

Do this

- Type out this code into your script file
- Make sure your code is **EXACTLY** the same!

```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

[System.Serializable]
public struct HighScores
{
    public List<int> scores;
}

public class HighScoreUI : MonoBehaviour {
    public Text highScoreText;
    public HighScores highScore;
    private int totalHighScores = 5;

    public void Start()
    {
        int score = PlayerPrefs.GetInt("Score");
        string s = PlayerPrefs.GetString("HighScores");

        if (string.IsNullOrEmpty(s)) {
            highScore = new HighScores();
            highScore.scores = new List<int>();
        } else {
            highScore = JsonUtility.FromJson<HighScores>(s);
        }

        if (highScore.scores.Count < totalHighScores) {
            int amount = totalHighScores - highScore.scores.Count;
            for (int i = 0; i < amount; i++) {
                highScore.scores.Add(0);
            }
        }

        if (score > highScore.scores[totalHighScores - 1]) {
            highScore.scores[totalHighScores - 1] = score;
        }

        highScore.scores.Sort();
        highScore.scores.Reverse(0, totalHighScores);

        highScoreText.text = "HIGH SCORES\n";

        for (int i = 0; i < totalHighScores; i++) {
            if(highScore.scores[i] == score) {
                highScoreText.text += "<color=#FF0000FF>" + (i + 1).ToString() + ". " +
highScore.scores[i].ToString() + "</color>\n";
            } else {
                highScoreText.text += (i + 1).ToString() + ". " + highScore.scores[i].ToString() + "\n";
            }
        }

        string scoresJSON = JsonUtility.ToJson(highScore);
        PlayerPrefs.SetString("HighScores", scoresJSON);
    }
}
```

Useful links

- | | |
|---|---|
| • More information about PlayerPrefs.SetString | PlayerPrefs.SetString - Scripting reference |
| • More information about PlayerPrefs.GetInt | PlayerPrefs.GetInt - Scripting reference |
| • More information about JsonUtility.FromJson | JsonUtility.FromJson - Scripting reference |
| • More information about JsonUtility.ToJson | JsonUtility.ToJson Scripting reference |
| • More information about string IsNullOrEmpty | string IsNullOrEmpty - Scripting reference |
| • More information about struct | Struct Scripting reference |

Task 4. Connect the player die event to the GameManager EndGame method

Explanation

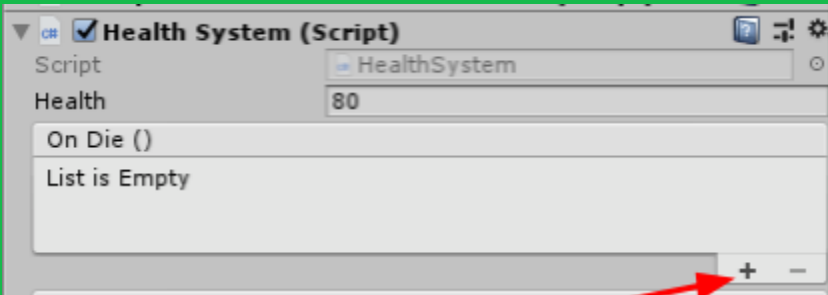
- From a previous tutorial you may have a **GameManager** script in your scene to load the **Game**, **Main Menu** and **Game Over** scenes
- We will use the **GameManager** in our **Game** scene to load the **Game Over** scene when the player dies

Check this

- Check you have a **GameManager** script on a GameObject in your **Game** scene
- If not put a **GameManager** script on a GameObject in your **Game** scene

Do this

- Select the **Player** GameObject in the **Hierarchy**
- In the **Inspector**, add a listener to the **On Die** event on the players **HealthSystem**

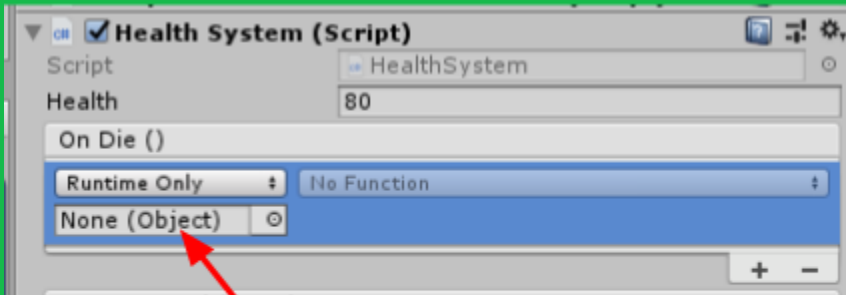
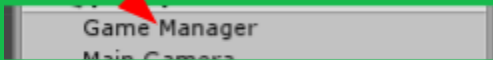


Add a listener to the On Die event

Do this

- In the **Hierarchy**, drag the GameObject with the **GameManager** onto the inlet of the **On Die** event

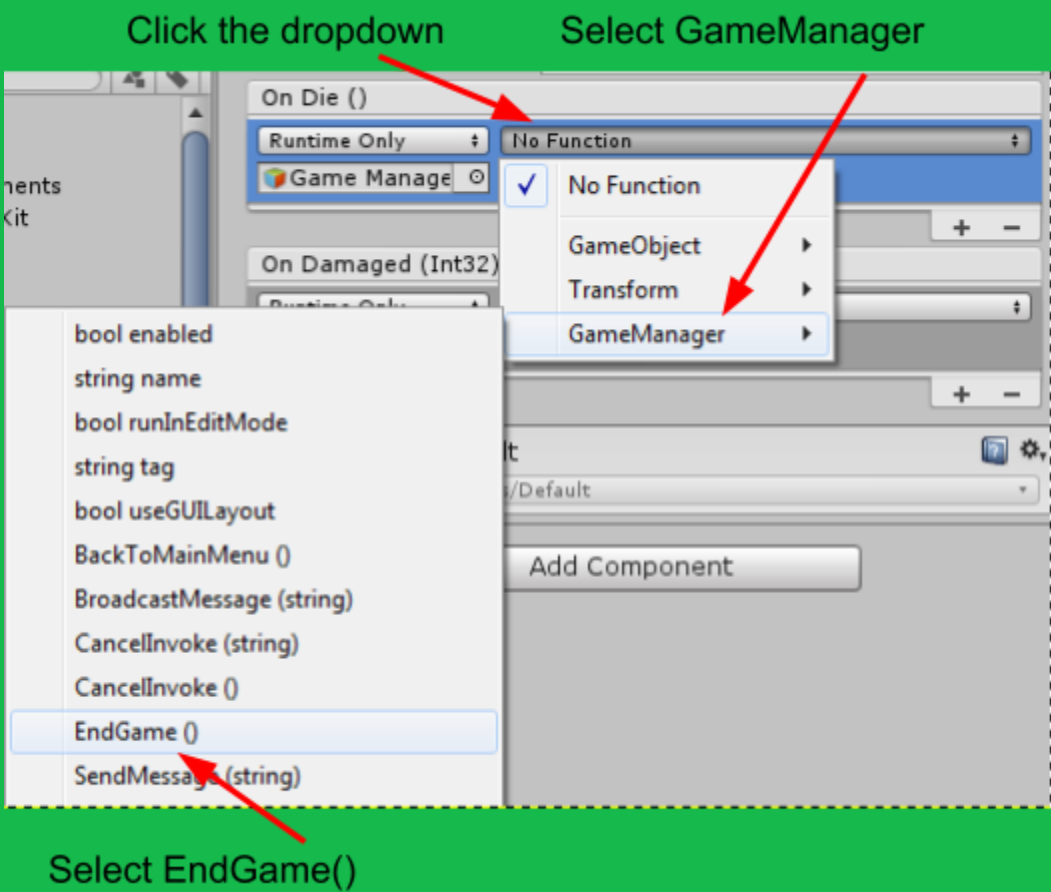
Drag this...



...onto here

Do this

- Click the dropdown on the **On Die** event
- Select **GameManager > EndGame**



Task 5. Set up HighScoreUI in the GameOver scene

Explanation

- We want to show high scores when the Game Over scene loads
- We need to add the **HighScoreUI** script to a Canvas (UI) GameObject in the **Game Over** scene

Do this

- In the **Project view**, open your **Game Over** scene

Do this

- In the **Hierarchy**, select the **Canvas** GameObject
- Add the **HighScoreUI** script to the **Canvas** GameObject

Do this

- In the **Hierarchy**, select the **Canvas** GameObject
- Click the **Create** button on the **Hierarchy**
- Select **UI > Text**
- Rename the **Text** GameObject to **High Scores**

Check this

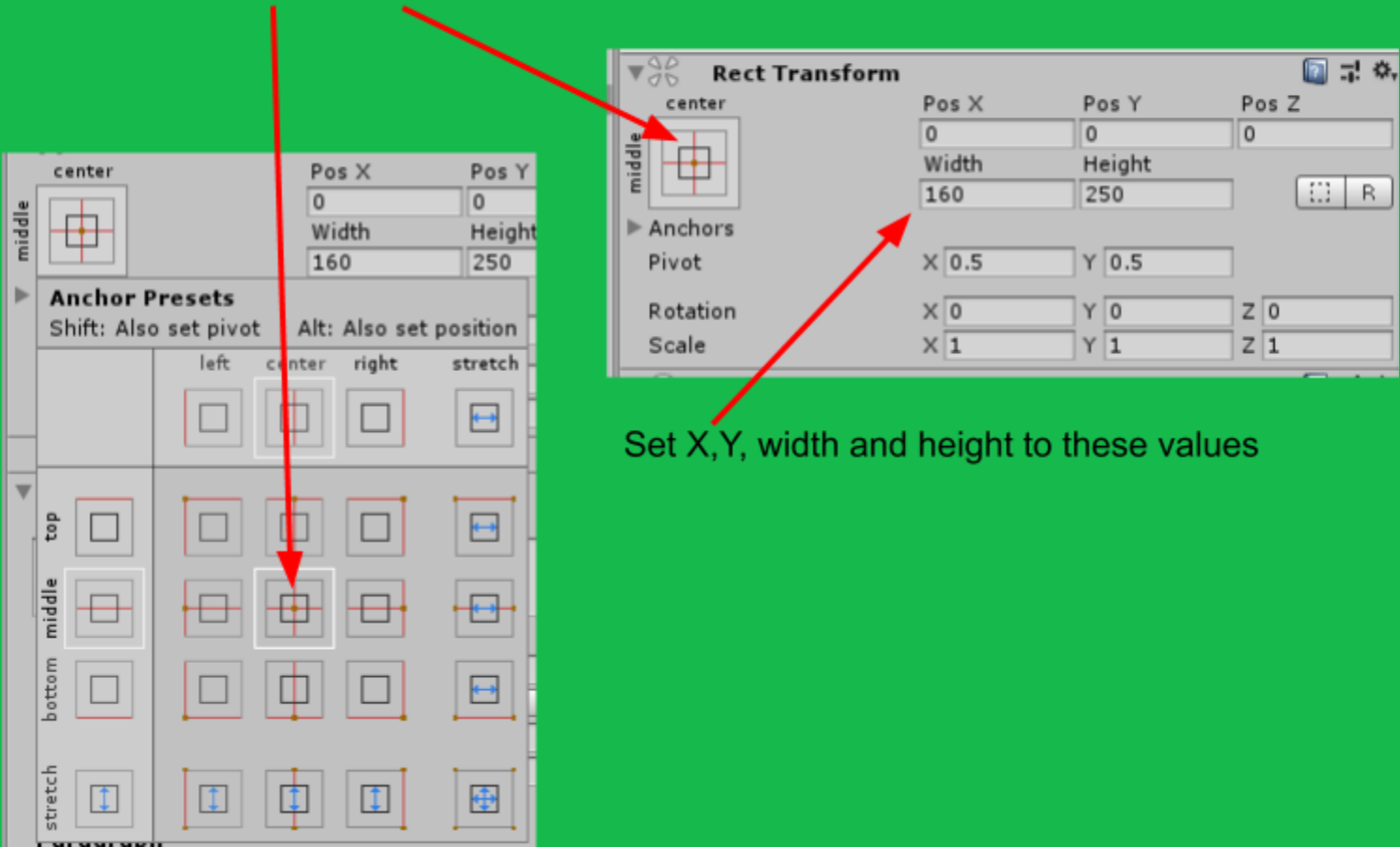
- Check your **High Scores** GameObject is a child of the **Canvas** GameObject



Do this

- In the **Hierarchy**, select the **High Scores** GameObject
- Make the following changes to the **Rect Transform** component in the **Inspector**

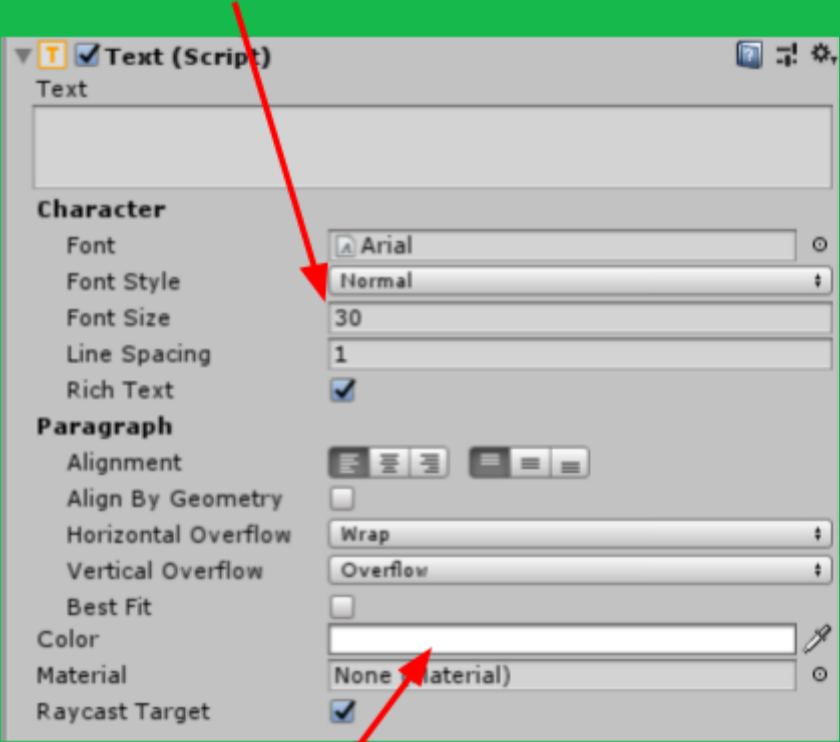
Click and set to middle center



Do this

- In the **Hierarchy**, select the **High Scores** GameObject
- Make the following changes to the **Text** component in the **Inspector**

Set size to 30



Change colour to white

Do this

- In the **Hierarchy**, select the **Canvas** GameObject
- Drag the **High Scores** GameObject onto the **High Score Text** inlet on the **HighScoreUI** component

Drag this...

CanvasHigh Scores

High Score UI (Script)

ScriptHighScoreUI

High Score TextHigh Scores (Text)

...onto here

Check this

- Check your High Scores work by testing the game!