# Static

Monday, January 4, 2021 10:28 PM

Stuck in the life of the program. Can be used with variables, classes and methods. Not visible in the Inspector.
**Use Cases:** player score, enemy count. Data stored across the entire game and there is only one of them.

**Remember that static data is in memory for the life of the program, could cause memory issues in large applications.**
**'static' keyword**

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

Unity Script | 2 references
public class Score : MonoBehaviour
{
    // public int score;
    public static int score;
}
```

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

Unity Script | 0 references
public class Test : MonoBehaviour
{
    // private Score _score; Non static variable.
    Unity Message | 0 references
    void Start()
    {
        // _score = GameObject.Find("Score Keeper").GetComponent<Score>(); // Method to access the variable if it's not static.
    }

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            Score.score += 10;
            Debug.Log(Score.score);
        }
    }
}
```

## Enemy Spawn Manager Ex.

**Spawns an enemy and displays the enemy count to the UI.**

**OnEnable():** Called when the object becomes enabled and active.
**OnDisable():** Called when the object becomes disabled and inactive.

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

Unity Script | 3 references
public class SpawnManager : MonoBehaviour
{
    public GameObject enemyPrefab;
    public static int enemyCount;

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            Instantiate(enemyPrefab);
        }
    }
}
```

(field) GameObject SpawnManager

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

Unity Script | 2 references
public class UIManager : MonoBehaviour
{
    public Text activeEnemiesText;

    2 references
    public void UpdateEnemyCount()
    {
        activeEnemiesText.text = "Active Enemies: " + SpawnManager.enemyCount;
    }
}
```

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

Unity Script | 0 references
public class Enemy : MonoBehaviour
{
    private UIManager _UIManager;
    Unity Message | 0 references
    public void OnEnable()
    {
        SpawnManager.enemyCount++;
        _UIManager = GameObject.Find("UI Manager").GetComponent<UIManager>();
        _UIManager.UpdateEnemyCount();
        Die();
    }

    Unity Message | 0 references
    public void OnDisable()
    {
        SpawnManager.enemyCount--;
        _UIManager.UpdateEnemyCount();
    }

    1 reference
    void Die()
    {
        Destroy(this.gameObject, Random.Range(2,5));
    }

}
```

## Static Members vs Instance Members

- Static members live in the life of the program, shared across all instances.
- Instance members are created as copies.

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

6 references
public class Item
{
    string itemName;
    string itemID;

    public static int itemCount;

    4 references
    public Item()
    {
        itemCount++;
    }
}

Unity Script | 0 references
public class StaticTest : MonoBehaviour
{
    Unity Message | 0 references
    void Start()
    {
        Item sword = new Item();
        Item cape = new Item();
        Item bread = new Item();
        Item fish = new Item();

        Debug.Log("Item Count: " + Item.itemCount);
    }

}
```