Lists

Tuesday, January 5, 2021 1:32 AM

- Similar to an array.
- A data type that accepts multiple data types. Extendable at runtime unlick arrays.
- **Array Ex.** 20 Slot inventory. **List Ex.** Item Database, continually adding items.

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

Unity Script | 0 references
public class GameManagerList : MonoBehaviour
{
    public List<GameObject> enemiesToSpawn = new List<GameObject>();
    public GameObject[] objectsToSpawn = new GameObject[10];

    Unity Message | 0 references
    private void Start()
    {
        // Elements accessed the same way, through an index.
        objectsToSpawn[2].name = "Jacob";
        enemiesToSpawn[2].name = "Kyle";
    }

}

```

Create a List and access it's members randomly

Method 1:

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

Unity Script | 0 references
public class Challenge : MonoBehaviour
{
    public List<string> cNames = new List<string>();

    Unity Message | 0 references
    void Start()
    {
        cNames.Add("Xavier"); // Add an element to a list.
        cNames.Add("Joshua");
        cNames.Add("Rafael");
        cNames.Add("Joe");
        cNames.Add("Myles");

        foreach (var name in cNames)
        {
            Debug.Log(name);
        }

    }

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            cNames.RemoveAt(Random.Range(0, cNames.Count)); // Remove an element from a List using it's index.
            foreach (var name in cNames)
            {
                Debug.Log(name);
            }
        }
    }
}

```

Method 2:

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

Unity Script | 0 references
public class ListChallenge : MonoBehaviour
{
    public List<string> names = new List<string>();

    Unity Message | 0 references
    void Start()
    {
        foreach (var name in names)
        {
            Debug.Log(name);
        }
    }

    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            var nameToRemove = names[Random.Range(0, names.Count)]; // Create a variable to hold a random name taken from the list.
            names.Remove(nameToRemove); // Remove an element from a list using a string.

            foreach (var name in names)
            {
                Debug.Log(name);
            }

            Debug.Log("Removed: " + nameToRemove);
        }
    }
}
```

Ex. Spawn random GameObjects in random positions, then store these spawned GameObjects in a dynamic List. Clear when 10 and change color to Green.

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

Unity Script | 0 references
public class BuildListChallenge : MonoBehaviour
{
    public GameObject[] SpawnList = new GameObject[3];
    public List<GameObject> objectsCreated = new List<GameObject>();

    // Update is called once per frame
    Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && objectsCreated.Count < 10)
        {
            // Select a random object to spawn from the Array.
            GameObject objectToSpawn = SpawnList[Random.Range(0, SpawnList.Length)];

            // Create random x axis and y axis values and assign them to a Vector3.
            int randomPosX = Random.Range(-10, 10);
            int randomPosY = Random.Range(-10, 10);
            Vector3 randomPos = new Vector3(randomPosX, randomPosY, 0);

            // Spawn the randomly selected Object from the Array at the andom osition Vector3 created above &
            // store the spawnedObject in a variable.
            GameObject SpawnedObject = Instantiate(objectToSpawn, randomPos, Quaternion.identity);

            // Add the object created to the List.
            objectsCreated.Add(SpawnedObject);

            if (objectsCreated.Count == 10)
            {
                foreach (var obj in objectsCreated)
                {
                    obj.GetComponent<MeshRenderer>().sharedMaterial.color = Color.green;
                }

                objectsCreated.Clear();
            }

        }

    }
}
```

Item Database Ex.

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

[System.Serializable]
4 references
public class GameItem
{
    public string itemName;
    public int itemID;
    public Sprite icon;
}
```

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

Unity Script | 2 references
public class ItemDatabase : MonoBehaviour
{
    public List<GameItem> itemDatabase = new List<GameItem>();

    1 reference
    public void AddItem(int itemID, Player player)
    {
        foreach (var item in itemDatabase)
        {
            if (item.itemID == itemID)
            {
                Debug.Log("We have a match!");
                player.inventory[0] = item;
                return;
            }
        }
        Debug.Log("Item doesn't exist");
    }

    1 reference
    public void RemoveItem(int itemID, Player player)
    {
        foreach (var item in itemDatabase)
        {
            if (item.itemID == itemID)
            {
                Debug.Log("We have a match!");
                player.inventory[0] = null;
            }
        }

    }
}
```

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

Unity Script | 2 references
public class Player : MonoBehaviour
{
    public GameItem[] inventory =  new GameItem[10];
    private ItemDatabase _itemDatabase;

    Unity Message | 0 references
    private void Start()
    {
        _itemDatabase = GameObject.Find("ItemDB").GetComponent<ItemDatabase>();
    }

    Unity Message | 0 references
    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            _itemDatabase.AddItem(0, this);
        }
        else if (Input.GetKeyDown(KeyCode.R))
        {
            _itemDatabase.RemoveItem(0, this);
        }
    }
}
```