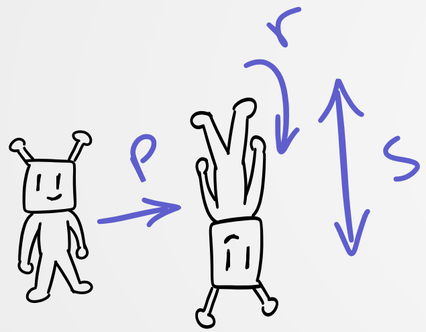
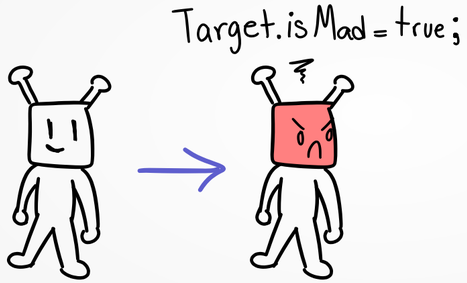
Persistencia

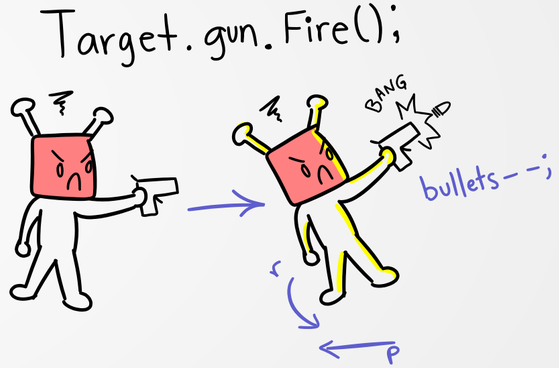
```
Undo.RecordObject(*, message);
```



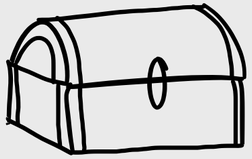
transform



Target



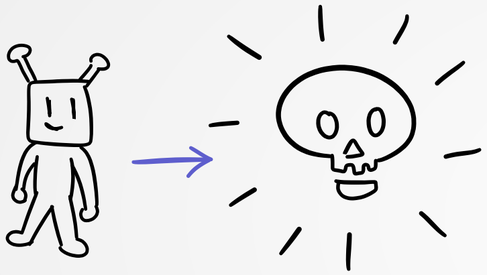
Target.gun; transform;



PERSISTENCIA

No!!

```
Destroy(Target.gameObject);
```



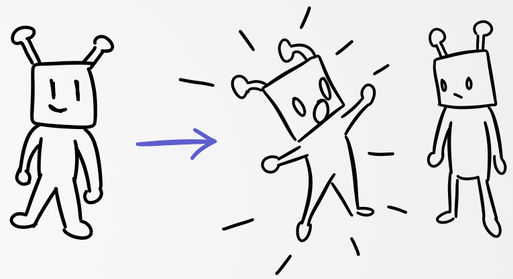
Undo.DestroyObjectImmediate

Util.Destroy(x)

UnityEngine.Object

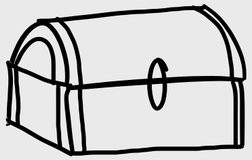
Platform dependant compilation

```
c = Instantiate(Target);
```



Undo.RegisterCreatedObjectUndo(c)

Util.Instantiate



Persistencia

[System.Serializable]

Visible y editable en el inspector!

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

[System.Serializable]
public class Stats {
    public RenewableStat hp;
    public RenewableStat mp;

    public int agility;
    public int perception;
    public int attackPower;
    public float attackRadius;
}
```

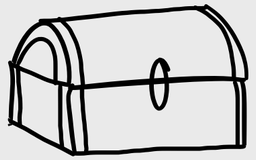
Stats.cs All L6 Git:master (C#//l +5 yas)

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

public class Adventurer : MonoBehaviour {
    public Stats stats;
    public List<GameObject> inventory;

    // ...
}
```

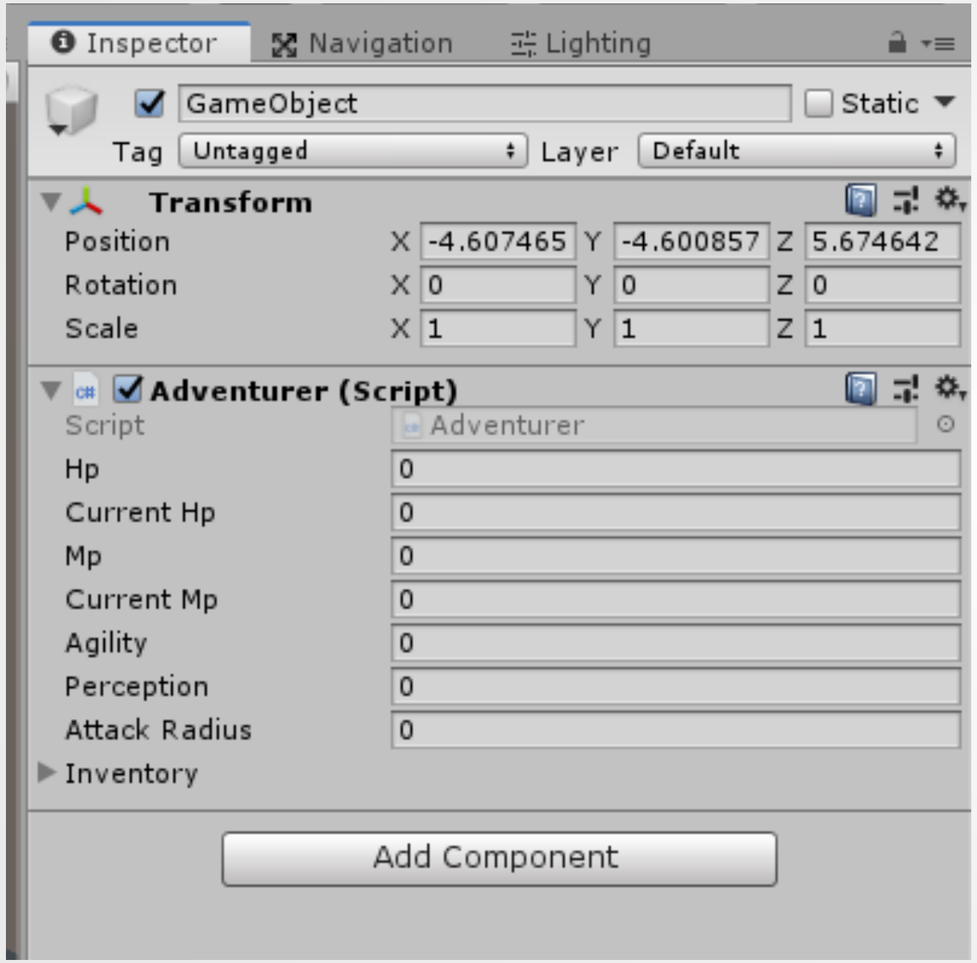
Adventurer.cs All L9 Git:master (C#//l +2 yas)

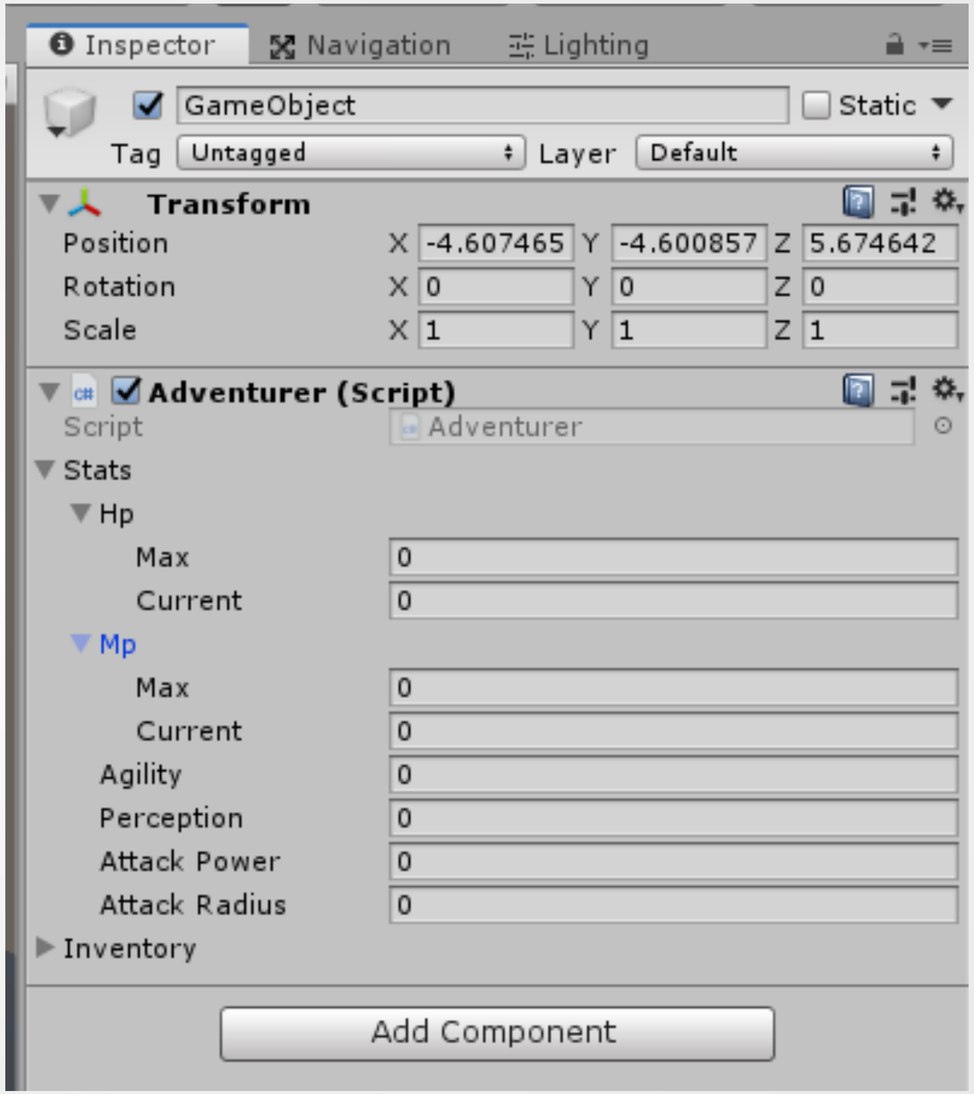
```cs
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

[System.Serializable]
public class RenewableStat {
    public int max = 10;
    public int current = 0;
}
```

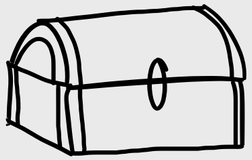
RenewableStat.cs All L1 Git-master (C#//l +5 yas)



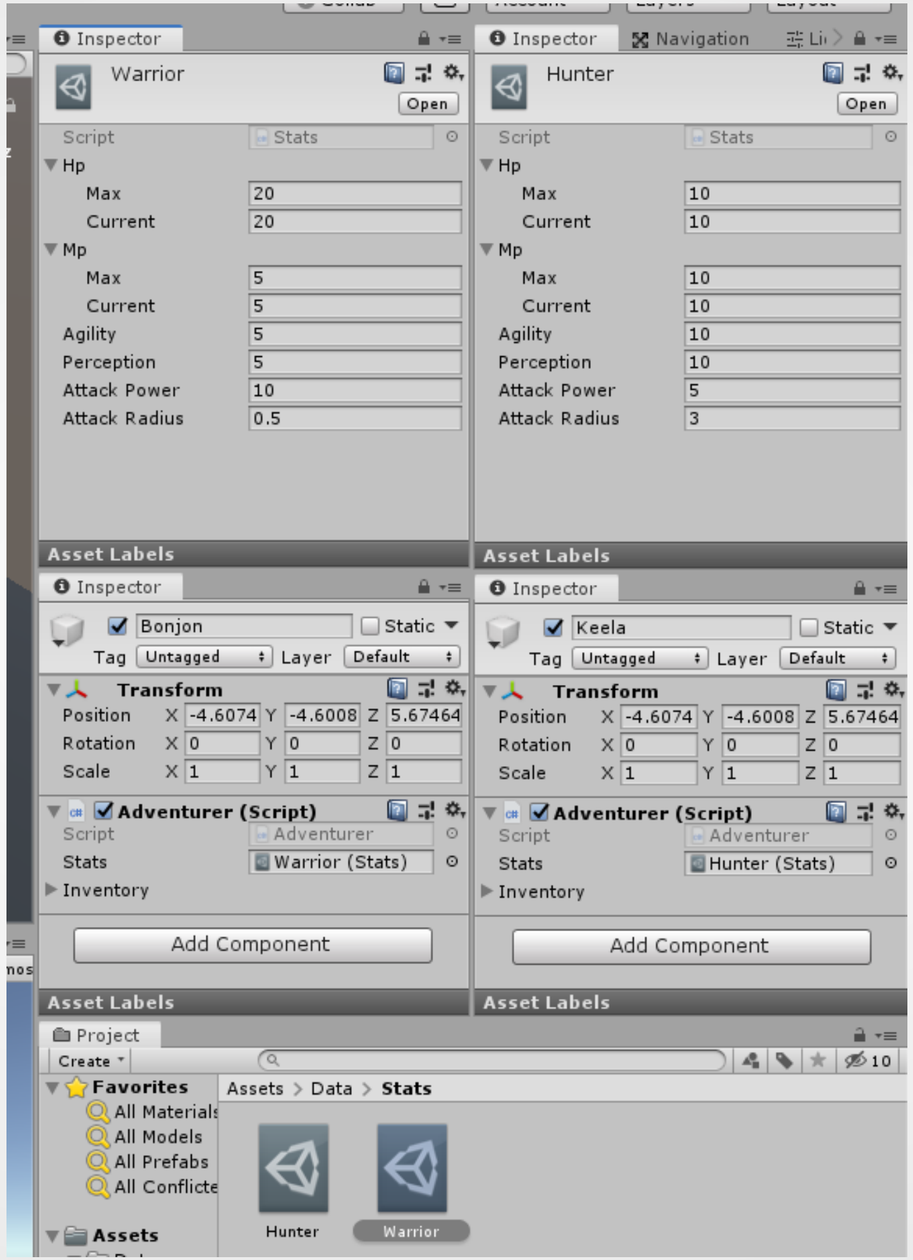
PERSISTENCIA

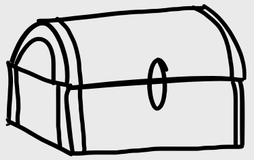






PERSISTENCIA





Persistencia

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

[System.Serializable]
[CreateAssetMenu(menuName = "Stats")]
public class Stats : ScriptableObject {
    public RenewableStat hp;
    public RenewableStat mp;

    public int agility;
    public int perception;
    public int attackPower;
    public float attackRadius;
}
```

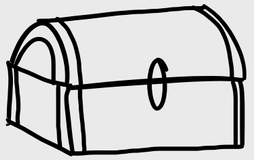
-\--- Stats.cs All L16 Git:master (C#//1 +4 yas)

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

public class Adventurer : MonoBehaviour {
    public Stats stats;
    public List<GameObject> inventory;

    // ...
}
```

-\--- Adventurer.cs All L9 Git:master (C#//1 +2 yas)

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

[System.Serializable]
public class RenewableStat {
    public int max = 10;
    public int current = 0;
}
```

-\--- RenewableStat.cs All L1 Git-master (C#//1 +5 yas)



PERSISTENCIA

Dictionary X

Interface X

private * variables

Si no se ve, no se guarda*

PERSISTENCIA

guarda variables privadas y las muestra en el inspector

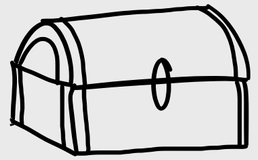
[SerializeField]

- static X
- get{} set{} X
- dictionary X
- interfaces X

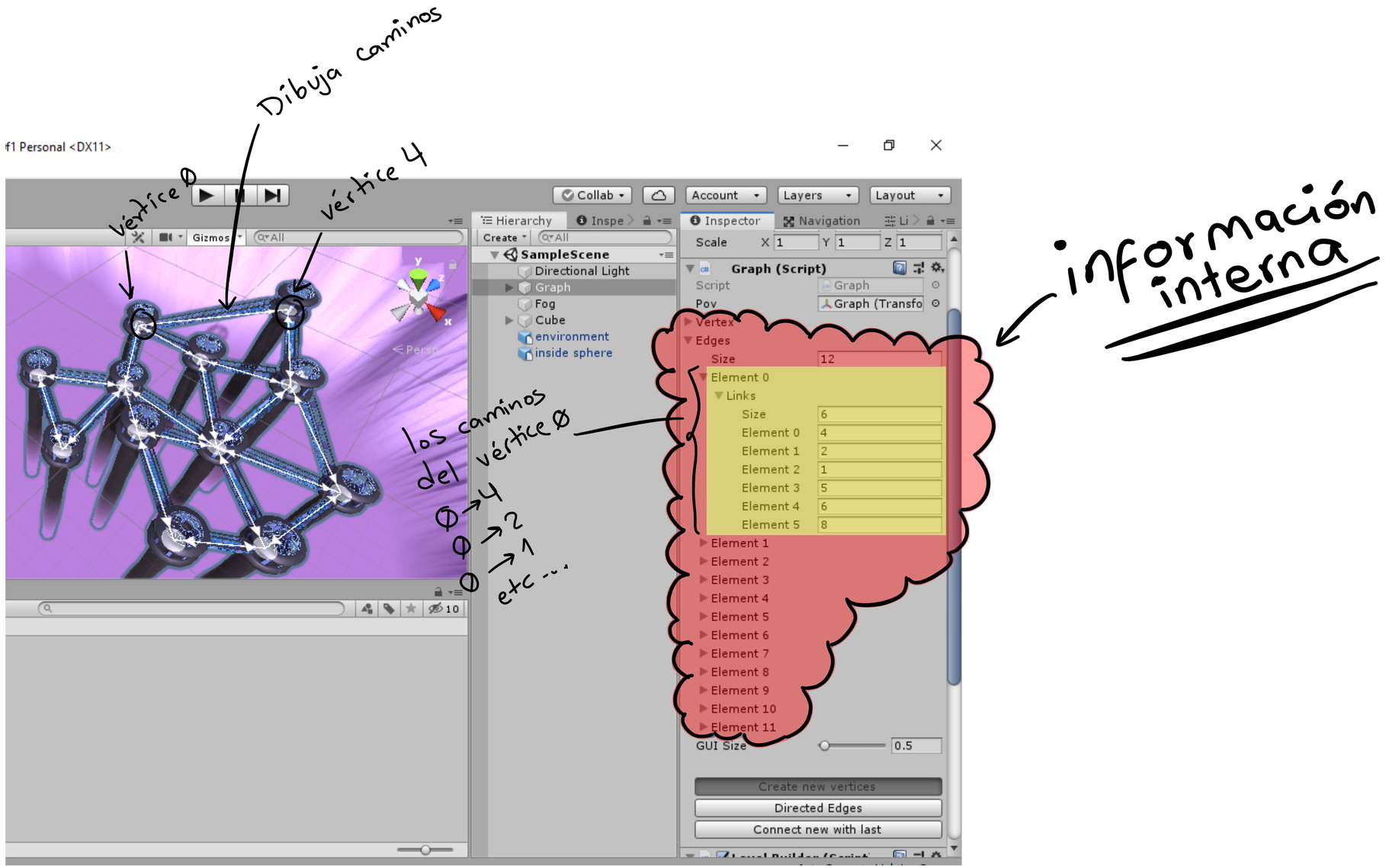
Esconde cosas visibles en el inspector

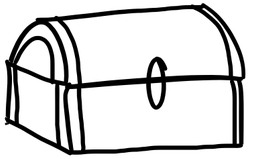
[HideInInspector]

(pero, si eran visibles, las guarda)



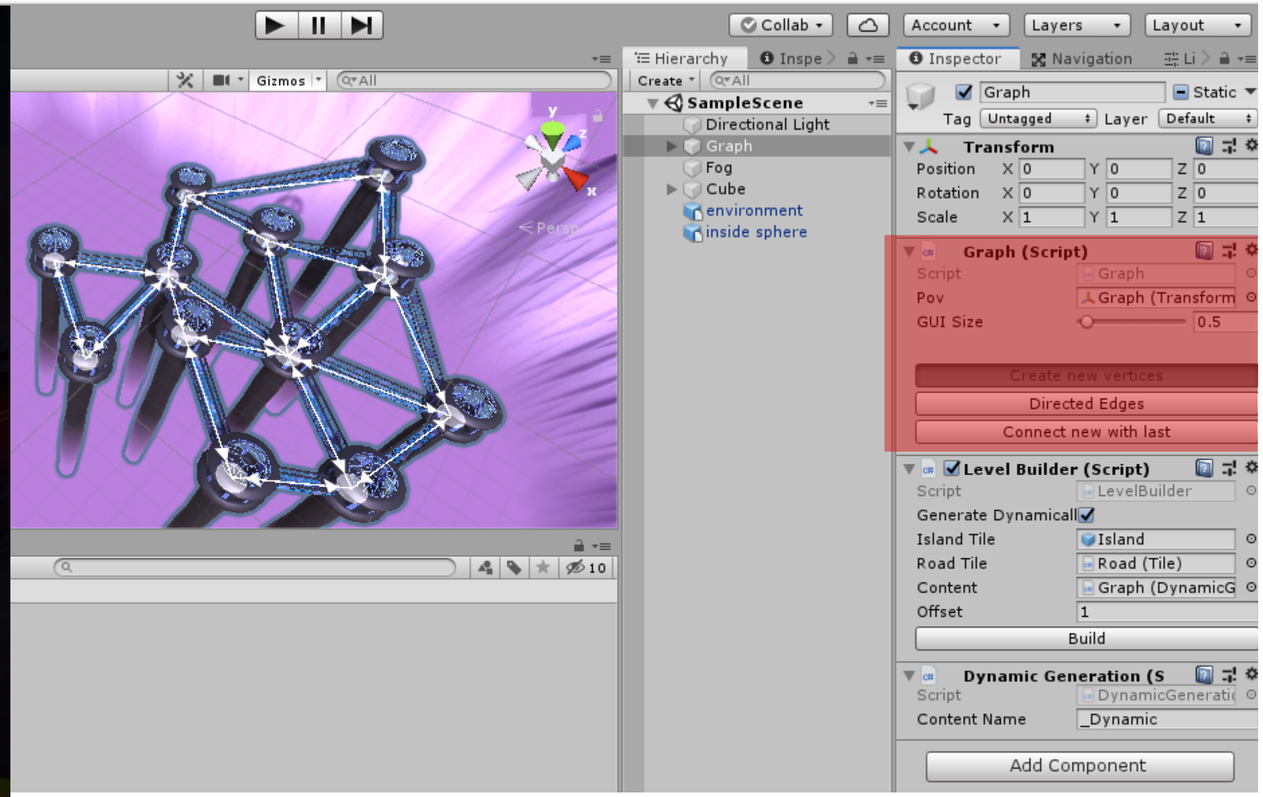
PERSISTENCIA

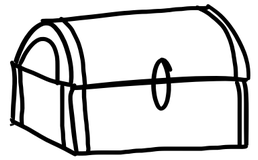




Persistencia

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

public class Graph : MonoBehaviour {
    public event System.Action onGraphModified;

    [SerializeField]
    Transform _pov = null;

    [HideInInspector]
    public List<Vector3> vertex = new List<Vector3>();
    [HideInInspector]
    public List<Edge> edges = new List<Edge>();

    public Vector3 ToWorldPoint (Vector3 point) {
        return PoV.TransformPoint(point);
    }

    public Vector3 ToLocalPoint (Vector3 point) {
        return PoV.InverseTransformPoint(point);
    }

    public void SetVertex (int index, Vector3 pos) {
        vertex[index] = pos;
```

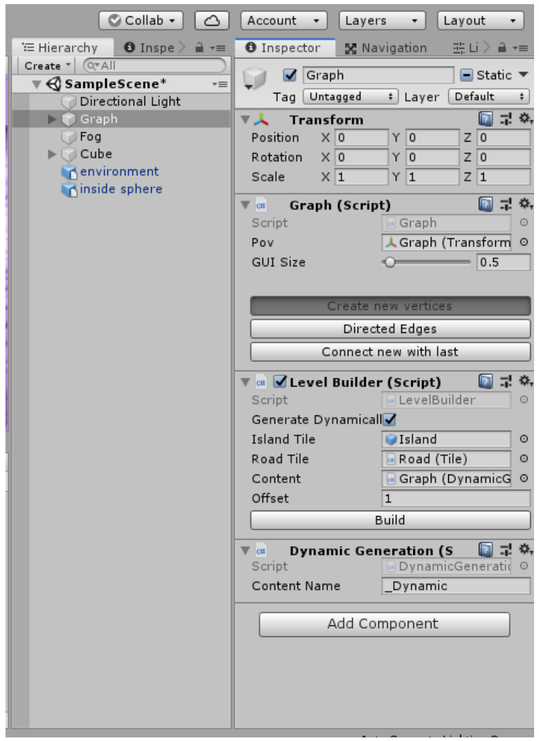




PERSISTENCIA

① Mostrar y Configurar

```
[ExecuteInEditMode]
public class LevelBuilder : MonoBehaviour {
    Graph _g;
    Graph _Graph { get { if (!_g) _g = GetComp

    public bool generateDynamically = false;

    [SerializeField]
    GameObject _islandTile;

    [SerializeField]
    Tile _roadTile;

    [SerializeField]
    DynamicGeneration _content;

    [SerializeField]
    float _offset = 1;

    void OnEnable () {
        _Graph.onGraphModified += GraphModifie
    }

    public void GraphModifiedHandler () {
        if (generateDynamically) {
            Build();
```



② Ocultar y crear prefab

```
[ExecuteInEditMode]
public class LevelBuilder : MonoBehaviour {
    Graph _g;
    Graph _Graph { get { if (!_g) _g = GetComp

    public bool generateDynamically = false;

    [HideInInspector]
    [SerializeField]
    GameObject _islandTile;

    [HideInInspector]
    [SerializeField]
    Tile _roadTile;

    [HideInInspector]
    [SerializeField]
    DynamicGeneration _content;

    [HideInInspector]
    [SerializeField]
    float _offset = 1;

    void OnEnable () {
        _Graph.onGraphModified += GraphModifie
```