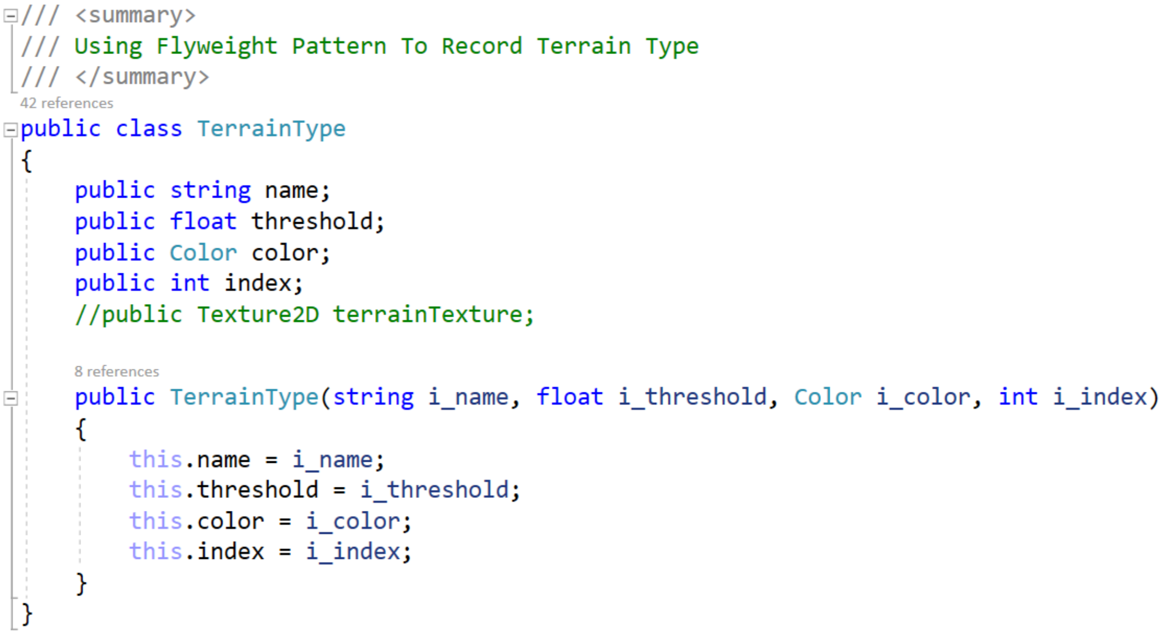
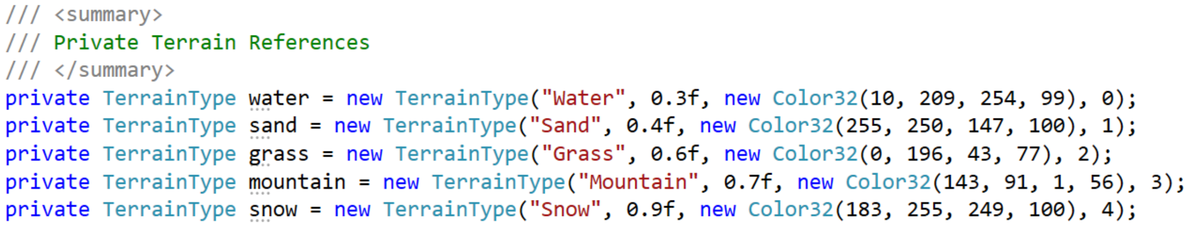
Analyze your usage of the flyweight pattern

How did you implement it?



```
/// <summary>
/// Using Flyweight Pattern To Record Terrain Type
/// </summary>
42 references
public class TerrainType
{
    public string name;
    public float threshold;
    public Color color;
    public int index;
    //public Texture2D terrainTexture;

    8 references
    public TerrainType(string i_name, float i_threshold, Color i_color, int i_index)
    {
        this.name = i_name;
        this.threshold = i_threshold;
        this.color = i_color;
        this.index = i_index;
    }
}
```

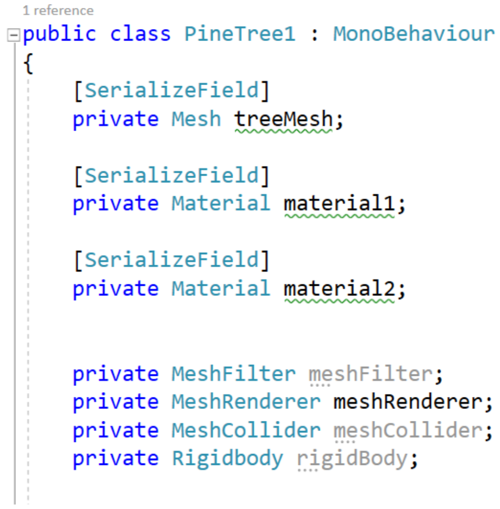


```
/// <summary>
/// Private Terrain References
/// </summary>
private TerrainType water = new TerrainType("Water", 0.3f, new Color32(10, 209, 254, 99), 0);
private TerrainType sand = new TerrainType("Sand", 0.4f, new Color32(255, 250, 147, 100), 1);
private TerrainType grass = new TerrainType("Grass", 0.6f, new Color32(0, 196, 43, 77), 2);
private TerrainType mountain = new TerrainType("Mountain", 0.7f, new Color32(143, 91, 1, 56), 3);
private TerrainType snow = new TerrainType("Snow", 0.9f, new Color32(183, 255, 249, 100), 4);
```

When generating the Terrain and Landscape, I made each terrain type a class and a single instance. Then I can apply these terrain classes to different positions on the land mesh as references. Furthermore, I used a terrain map data structure to store the terrain information of this level. For example, when applying a terrain type to the mesh, also adding it to the data structure. Thus when I try to spawn objects on the map I can get information about the terrain. If I would do it differently, I'm willing to learn more about how unity references objects in memory, and make it more efficient.

Did you use a different implement flyweight elsewhere?

Also, I used Flyweight when generating trees:



```
public class PineTree1 : MonoBehaviour
{
    [SerializeField]
    private Mesh treeMesh;

    [SerializeField]
    private Material material1;

    [SerializeField]
    private Material material2;


    private MeshFilter meshFilter;
    private MeshRenderer meshRenderer;
    private MeshCollider meshCollider;
    private Rigidbody rigidBody;
```

I have to put more information in this one, like the mesh and material. I was also thinking, maybe prefabs in Unity are already using Flyweight, but prefabs are like Copying. However, the materials have an option called "GPU Instancing", which is using the Flyweight as default.

Describe your data importation

The data was inputed from editor or from the UI or in the code. I thought my game didn't have too many types of objects, thus I did not use Prototype. If do it again I would definetly try Prototype Pattern.