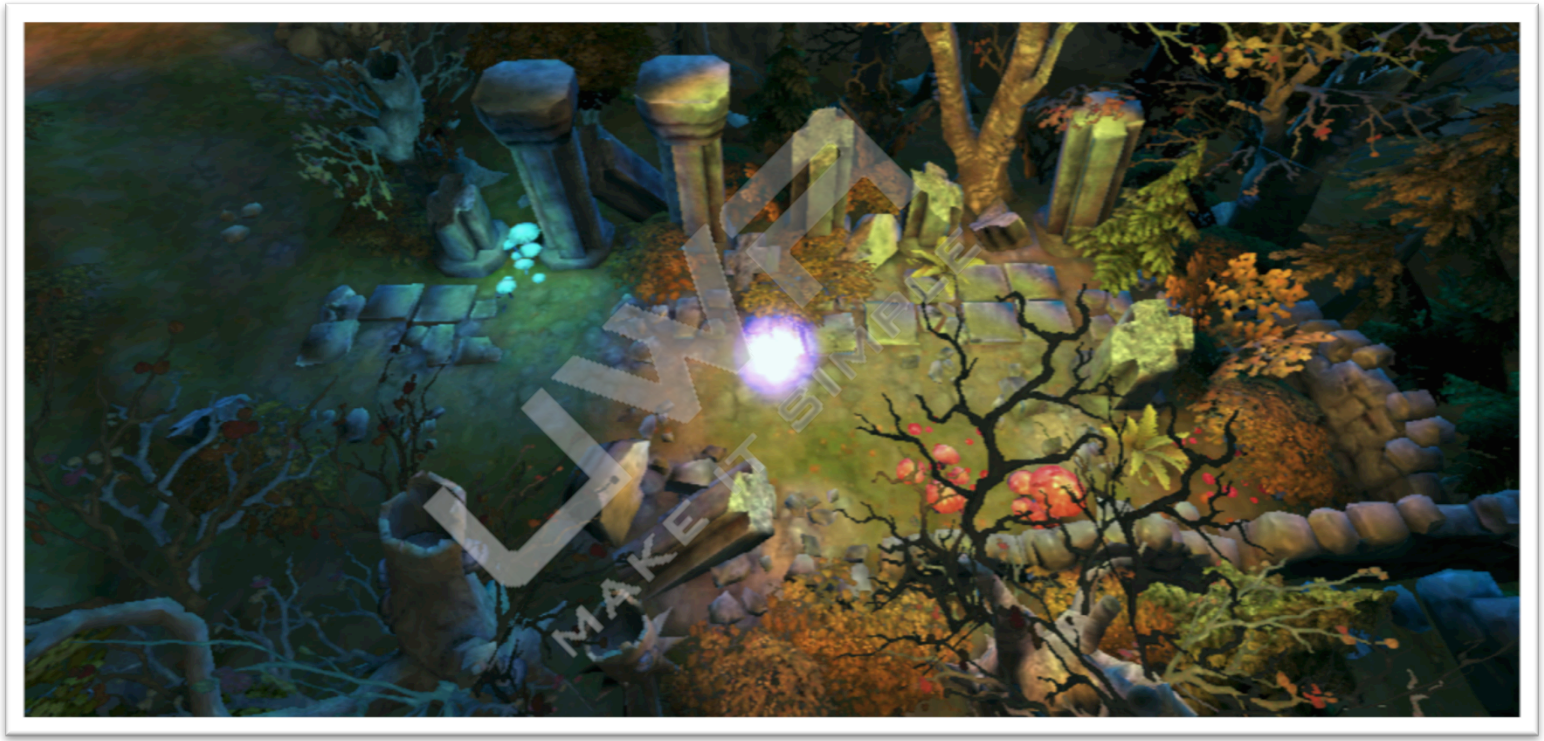
UWA
MAKE IT SIMPLE

大规模场景的资源拆分和动态加载

侑虎科技（上海）有限公司

日期：2016.11.26

Demo演示



概要

- 资源拆分
  - 地形、地表
- 动态加载/卸载
- 打包策略

资源拆分

- 地形资源拆分
  - Unity原始地形有网格尺寸的限制（4096）
  - Mesh地形有面片数限制
  - 降低内存占用，仅需载入部分地形数据
  - 降低渲染面片数，视域体剔除

资源拆分

- 地形资源拆分
  - 制作大规模地形
    - Terrain Composer 2
    - T4M
    - 建模软件…
  - 拆分现有地形
    - Terrain Slicing & Dynamic Loading Kit

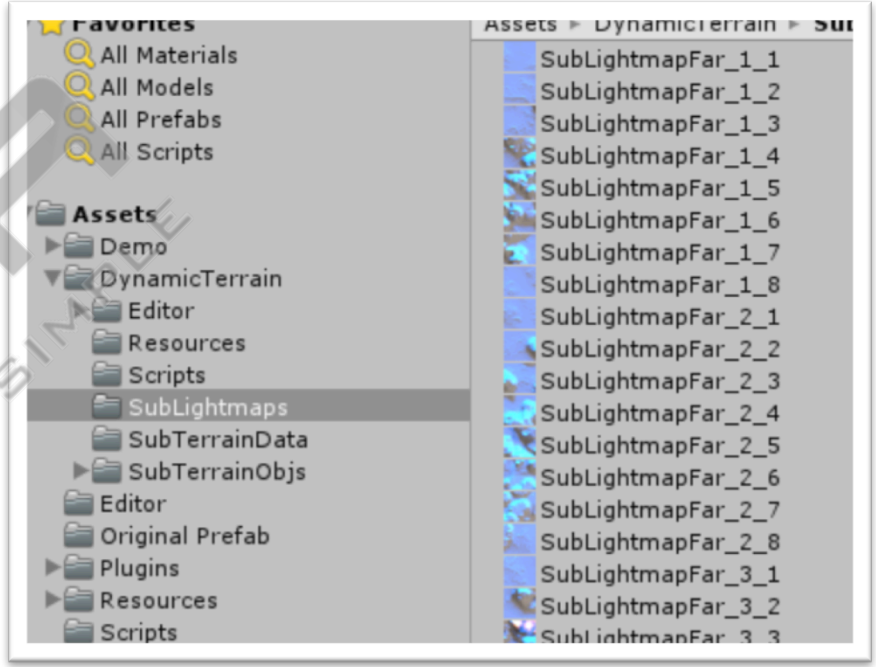
资源拆分

- 拆分现有地形
  - 地形数据

basemapDistance	Heightmap patches beyond basemap distance will use a precomputed low res basemap.
castShadows	Should terrain cast shadows?.
collectDetailPatches	Collect Detail patches from memory.
detailObjectDensity	Density of detail objects.
detailObjectDistance	Detail objects will be displayed up to this distance.
heightmapMaximumLOD	Lets you essentially lower the heightmap resolution used for rendering.
heightmapPixelError	An approximation of how many pixels the terrain will pop in the worst case when switching lod.
lightmapIndex	The index of the lightmap applied to this renderer.
terrainData	The Terrain Data that stores heightmaps, terrain textures, detail meshes and trees.
treeBillboardDistance	Distance from the camera where trees will be rendered as billboards only.
treeCrossFadeLength	Total distance delta that trees will use to transition from billboard orientation to mesh orientation.
treeDistance	The maximum distance at which trees are rendered.
treeMaximumFullLODCount	Maximum number of trees rendered at full LOD.

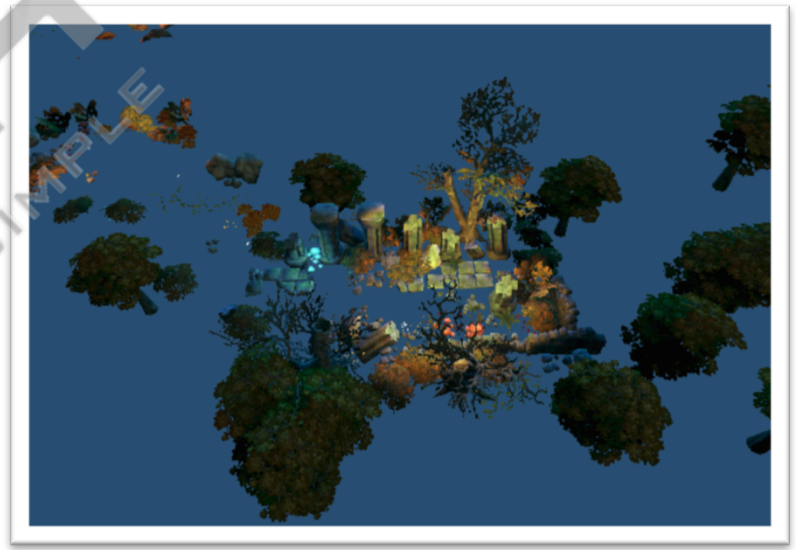
资源拆分

- 拆分现有地形
  - Lightmap
    - 拆分后重新烘焙
    - 切分烘焙好的Lightmap
      - exr 格式，FreeImage，



资源拆分

- 地表资源拆分
  - 切割跨地形的大模型
  - 按地形块分组



动态加载

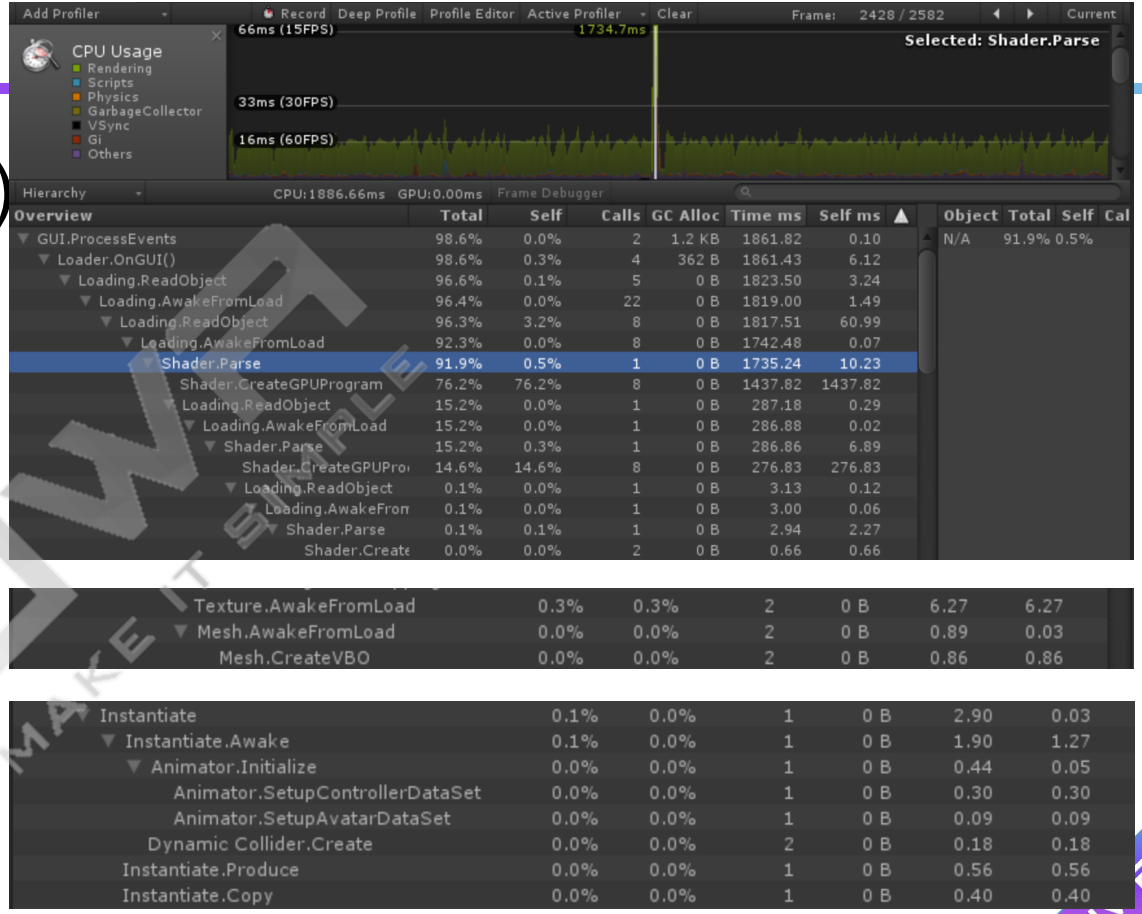
- 关键在于：流畅
  - 卡顿分析

动态加载

- 关键在于：流畅
  - Instantiate 前自动加载未加载的引用资源
    - Shader(Fallback)
    - Texture
    - Mesh
    - AnimationClip

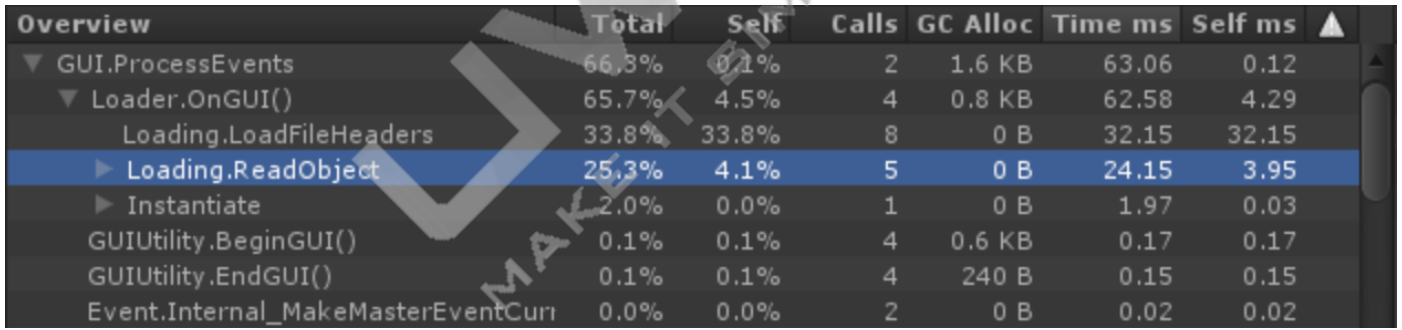
动态加载

- Shader(Fallback)
- Texture
- Mesh
- AnimationClip



动态加载

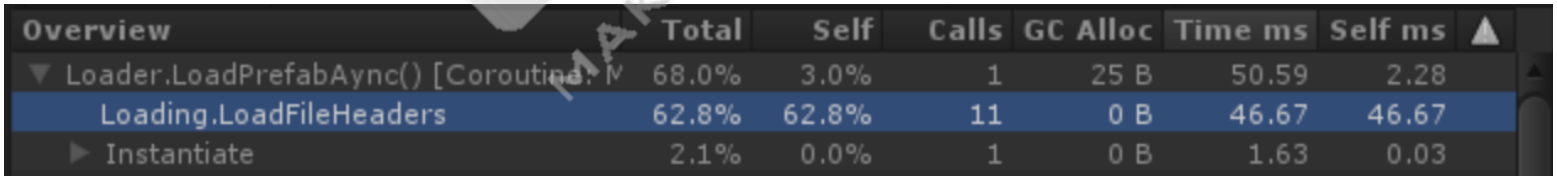
- 预加载资源
  - Shader(Fallback)
  - Texture



Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
GUI.ProcessEvents	66.8%	0.1%	2	1.6 KB	63.06	0.12
Loader.OnGUI()	65.7%	4.5%	4	0.8 KB	62.58	4.29
Loading.LoadFileHeaders	33.8%	33.8%	8	0 B	32.15	32.15
Loading.ReadObject	25.3%	4.1%	5	0 B	24.15	3.95
Instantiate	2.0%	0.0%	1	0 B	1.97	0.03
GUIUtility.BeginGUI()	0.1%	0.1%	4	0.6 KB	0.17	0.17
GUIUtility.EndGUI()	0.1%	0.1%	4	240 B	0.15	0.15
Event.Internal_MakeMasterEventCurr	0.0%	0.0%	2	0 B	0.02	0.02

动态加载

- 预加载资源异步加载
  - Resources.LoadAsync
  - AssetBundle.LoadAsync
  - ……



Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
Loader.LoadPrefabAync() [Coroutine]	68.0%	3.0%	1	25 B	50.59	2.28
Loading.LoadFileHeaders	62.8%	62.8%	11	0 B	46.67	46.67
Instantiate	2.1%	0.0%	1	0 B	1.63	0.03

动态加载

- 实例化引起的序列化操作(Loading.LoadFileHeaders)
  - 避免一次性实例化过多的粒子系统（预加载）
  - 避免层级复杂，组件Awake过多
  - 尝试拆分 Prefab，流式 Instantiate

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
▼ Loader.LoadPrefabAync() [Coroutine:	53.4%	29.8%	1	139 B	28.95	16.17
Loading.LoadFileHeaders	21.4%	21.4%	2	0 B	11.63	11.63
▶ Instantiate	2.1%	0.0%	1	0 B	1.14	0.03

动态加载

- 加载策略（Assetbundle）
  - Shared 包常驻内存
    - 大纹理等资源采用 LoadFromCacheOrDownload
    - Material/Mesh等可采用 new WWW
  - Prefab 包动态加载 Loadsync + unload(false)
    - 采用 new WWW

参考资料 http://blog.uwa4d.com/archives/ABTheory.html

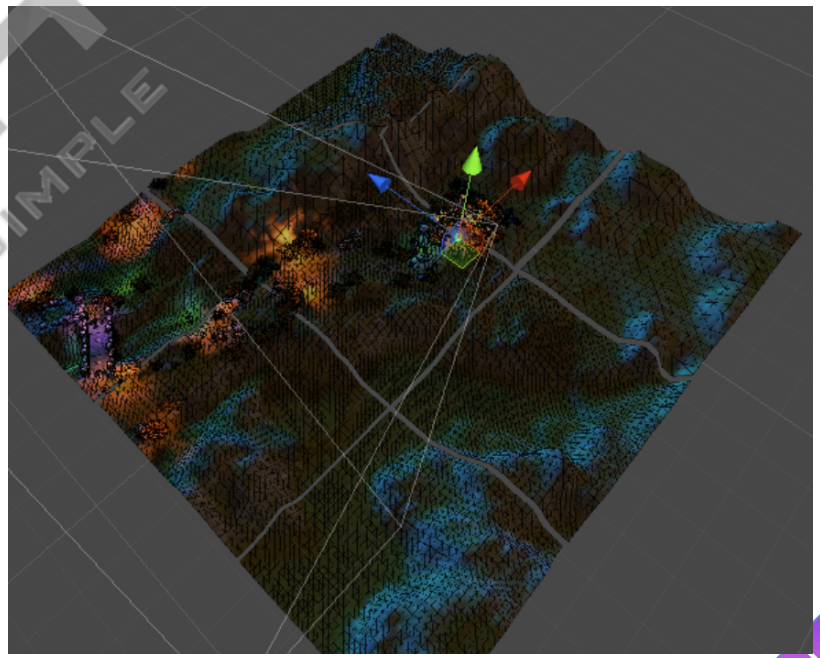
动态加载

- 卸载策略（AssetBundle）
  - Prefab 包
    - GameObject 可通过 Destroy 来销毁
    - TerrainData、Object 等可通过 Resources.UnloadAsset 来进行卸载
  - Shared 包
    - 建议在确认不使用或切换场景时进行卸载
    - 切换场景时调用 UnloadUnusedAssets 来卸载 Texture、Mesh 等加载的共享资源

参考资料 http://blog.uwa4d.com/archives/ABTheory.html

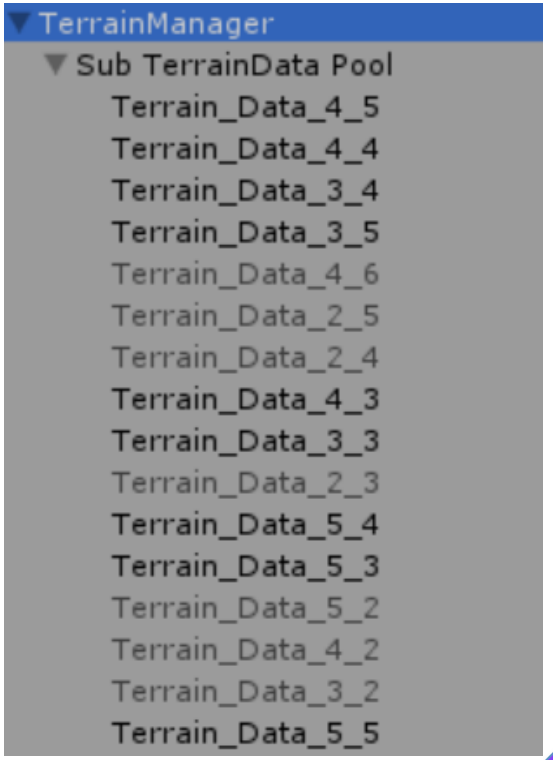
动态加载

- 注意事项
  - 加载方式
    - 九宫格
      - 适用于非自由视角



动态加载

- 注意事项
  - 加载方式
    - 建立缓冲池，防止反复实例化。



动态加载

- 注意事项
  - 加载方式
    - Load vs LoadAsync
      - 进场景预加载，推荐Load
      - 游戏中，推荐LoadAsync

动态加载

- 注意事项
  - Lightmap 动态加载
    - LightmapSettings
    - lightmapIndex/lightmapScaleOffset
    - Shader Stripping

动态加载

- 注意事项
  - Terrain 动态加载
    - LightmapId
    - Terrain.SetNeighbors

动态加载

- 注意事项
  - 动态 Static Batching（StaticBatchingUtility）
    - 运行时CPU/堆内存开销较大
    - 优先推荐手动拼合
    - 推荐分组拼合

动态加载

- 注意事项
  - 防止资源泄漏
    - 运行时创建
      - new Material、Mesh
      - Material set
    - AssetBundle重复加载
      - 共享资源
    - 隐形资源
      - TerrainData/AlphaMap

动态加载

- 注意事项
  - 防止资源泄漏
    - 运行时创建
      - new Material、Mesh
      - Material set

```
public Material dynamicMaterial { get { return mDynamicMat; } }
```

```
if (mMaterial != null)
{
    mDynamicMat = new Material(mMaterial);
    mDynamicMat.name = "[NGUI] " + mMaterial.name;
    mDynamicMat.hideFlags = HideFlags.DontSave | HideFlags.NotEditable;
    mDynamicMat.CopyPropertiesFromMaterial(mMaterial);
```

```
    NGUITools.DestroyImmediate(mDynamicMat);
    mDynamicMat = null;
```

动态加载

- 注意事项
  - 防止资源泄漏
    - AssetBundle重复加载
      - 共享资源
        - Texture、Mesh、Font
        - Prefab

动态加载

- 注意事项
  - 防止资源泄漏
    - 隐形资源
      - TerrainData/AlphaMap

```
            if (_loadedTerrainDataAssets.ContainsKey(dataName))
            {
#if !UNITY_5
                Texture2D[] splatMaps = _alphamapTextures.GetValue(_loadedTerrainDataAssets[dataName], null) as Texture2D[];
#else
                Texture2D[] splatMaps = _loadedTerrainDataAssets[dataName].alphamapTextures;
#endif
                if (splatMaps != null)
                {
                    foreach (Texture2D splatMap in splatMaps)
                    {
                        Resources.UnloadAsset(splatMap);
                    }
                }
```

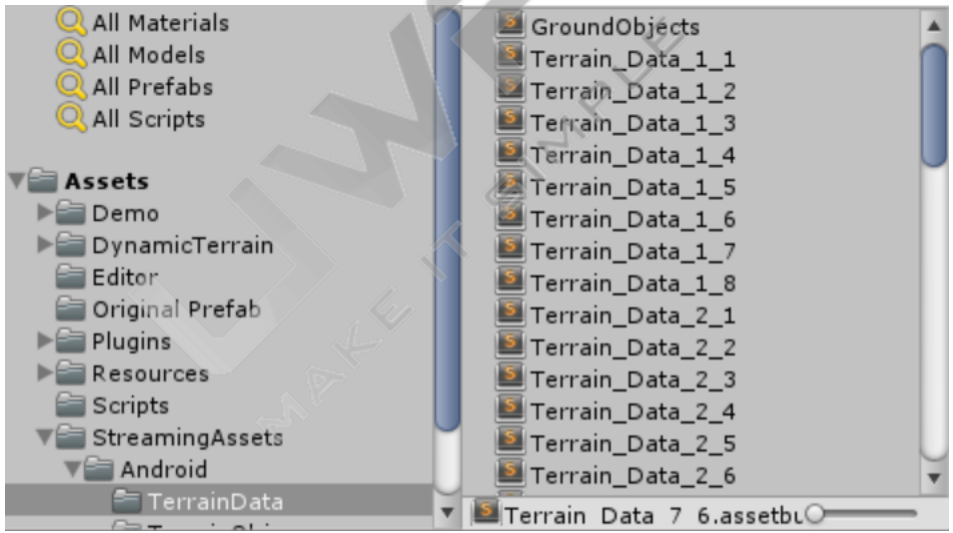
打包策略

- 地形资源

```
List<Object> groundObjs = new List<Object>();
groundObjs.AddRange(Resources.LoadAll(BundleConfig.GroundObjectsPath, typeof(Texture2D)));
groundObjs.AddRange(Resources.LoadAll(BundleConfig.GroundObjectsPath, typeof(GameObject)));

BuildPipeline.PushAssetDependencies();

string sharedBundlePath = BundleConfig.BundleRootFolder + GameConfig.TerrainDataBundlesRelativeFolder +
                    "/GroundObjects" + GameConfig.BundleExt;
BuildPipeline.BuildAssetBundle(groundObjs[0], groundObjs.ToArray(), sharedBundlePath,
    BundleConfig.BundleBuildOptions, BundleConfig.BundleTarget);

BuildPipeline.PushAssetDependencies();

string [] terrainDataPaths = Directory.GetFiles (TerrainSlicerConfig.SubTerrainDataAssetsFolder, "*.asset");
foreach (string terrainDataPath in terrainDataPaths)
{
    Object terrainDataObj = AssetDatabase.LoadAssetAtPath(terrainDataPath, typeof(TerrainData));
    if(terrainDataObj == null) continue;

    Object[] objs = {terrainDataObj};
    string bundlePath = BundleConfig.BundleRootFolder + GameConfig.TerrainDataBundlesRelativeFolder +
        "/" + terrainDataObj.name + GameConfig.BundleExt;
    BuildPipeline.BuildAssetBundle(objs[0], objs, bundlePath, BundleConfig.BundleBuildOptions, BundleConfig.BundleTarget);
}

BuildPipeline.PopAssetDependencies();
BuildPipeline.PopAssetDependencies();
```

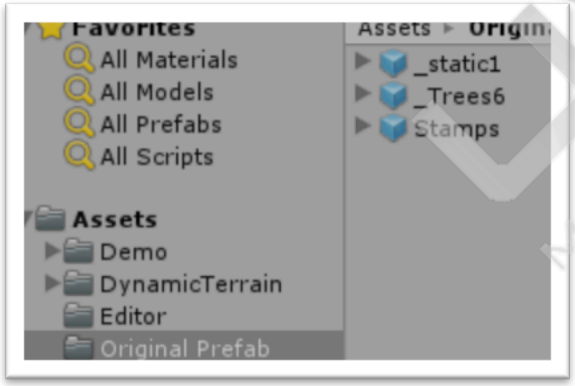
打包策略

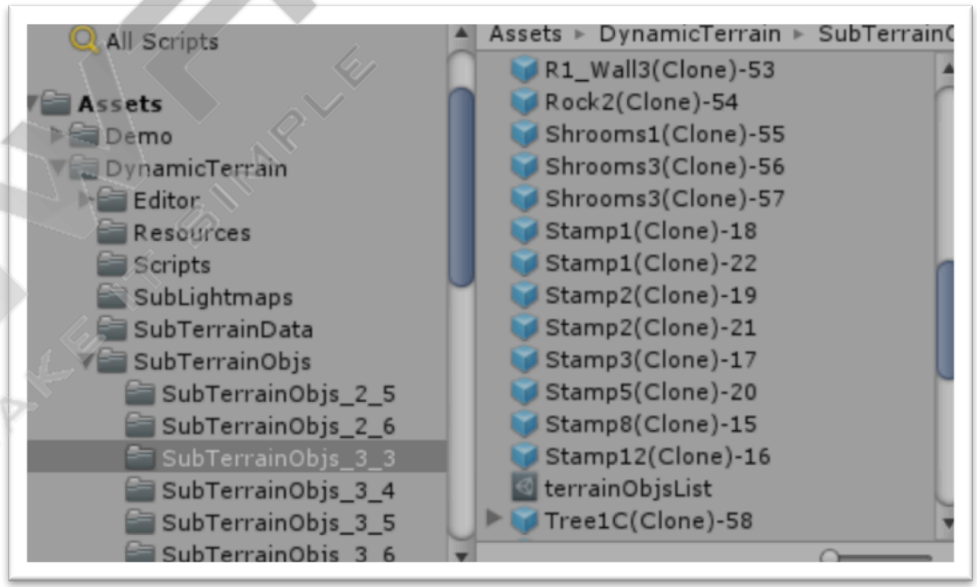
- 地形资源


All Materials
All Models
All Prefabs
All Scripts
Assets
Demo
DynamicTerrain
Editor
Original Prefab
Plugins
Resources
Scripts
StreamingAssets
Android
TerrainData
GroundObjects
Terrain_Data_1_1
Terrain_Data_1_2
Terrain_Data_1_3
Terrain_Data_1_4
Terrain_Data_1_5
Terrain_Data_1_6
Terrain_Data_1_7
Terrain_Data_1_8
Terrain_Data_2_1
Terrain_Data_2_2
Terrain_Data_2_3
Terrain_Data_2_4
Terrain_Data_2_5
Terrain_Data_2_6

打包策略

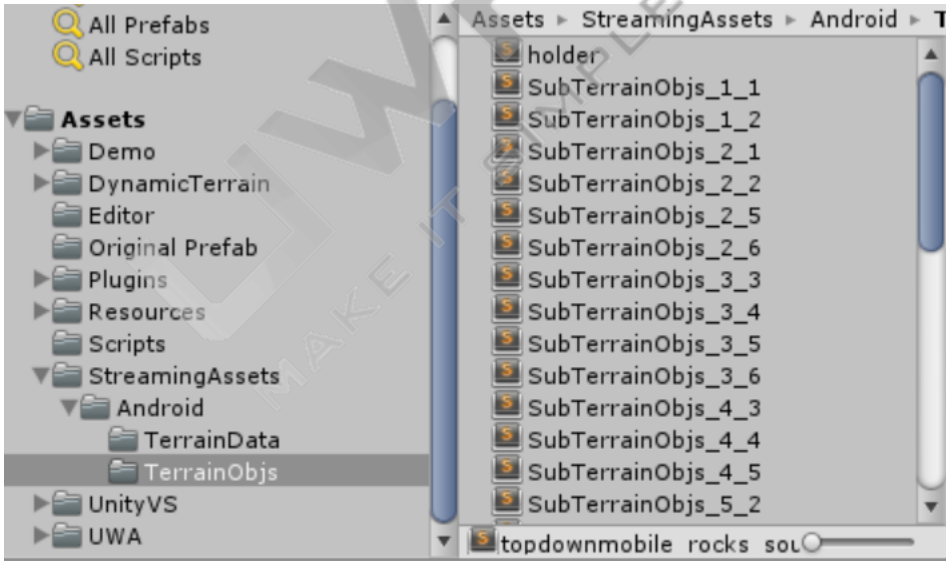
- 地表资源
  - 复杂资源拆分
    - “流式“实例化





打包策略

- 地表资源
  - 按地形块分组打包



www.uwa4d.com