

大规模场景的资源拆分和 动态加载

张鑫

Unity Technologies China



报告大纲

- 资源拆分
 - 资源拆分的必要性
 - 地形资源拆分
 - 地表资源拆分
 - Assetbundle打包
- 动态加载
 - 资源动态加载的优点
 - 动态加载方法
 - 资源缓冲池的介绍
 - 资源的异步加载



资源拆分

- 资源拆分的必要性？
 - 利于分块加载，减小内存占用
 - 减小客户端的大小
 - 便于资源的异步加载
 - 便于资源的动态更新



资源拆分

- 地形资源拆分



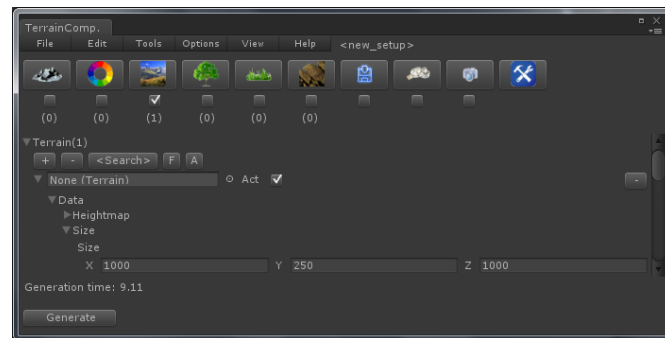
资源拆分

- 地形资源拆分
 - Unity原始地形有网格尺寸的限制
 - 地形拆分后，每次仅需载入部分地形数据



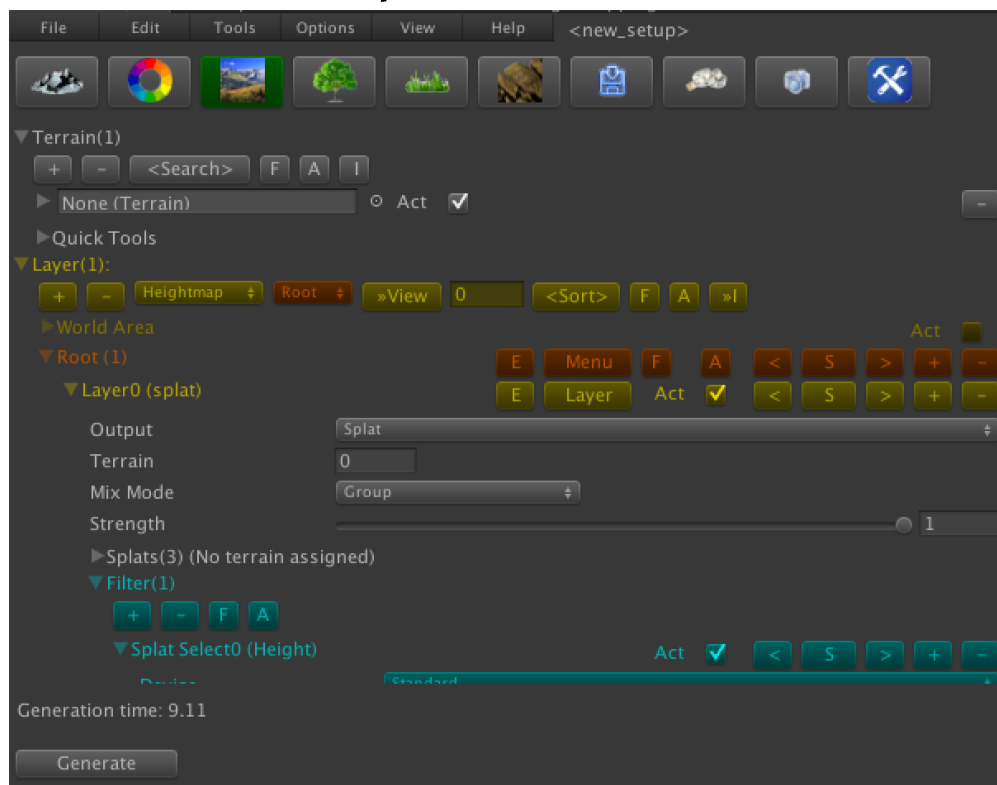
资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 遵循Unity基本地形机制
 - 将设计师的手动操作变为自动
 - 生成复杂地形高度数据
 - 地形自动分块
 - 随机生成Splat混合纹理
 - 随机布置草及树
 - 对于任何分块均可手动调整



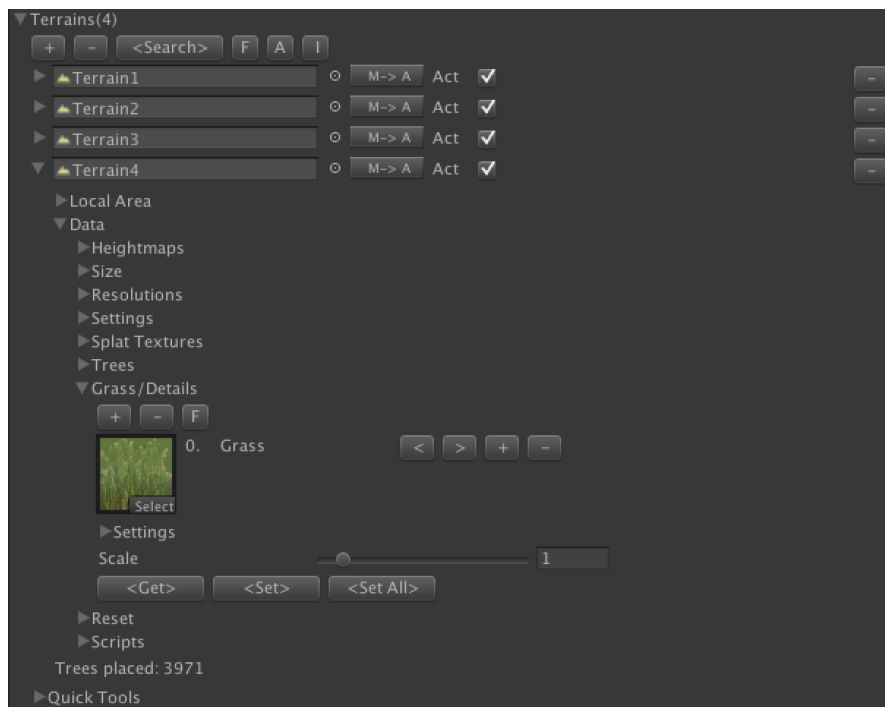
资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件



资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 地形资源编辑界面



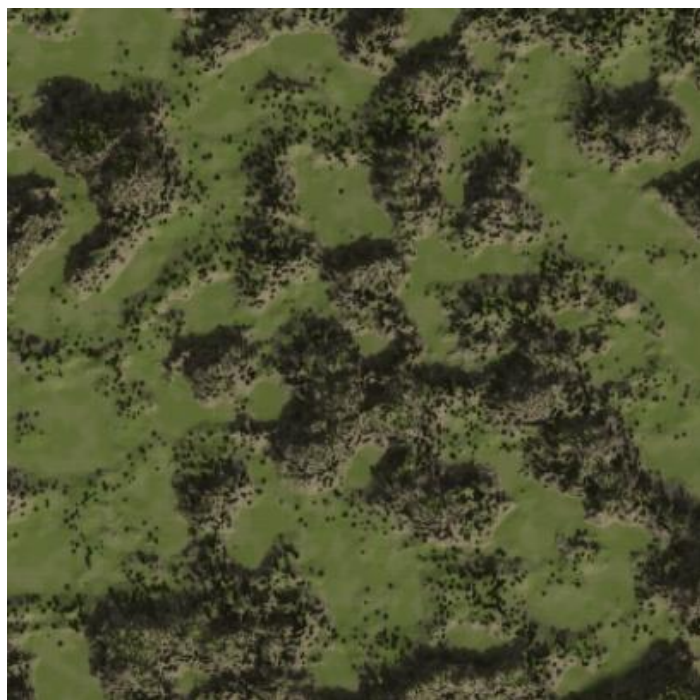
资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 地形细节编辑界面



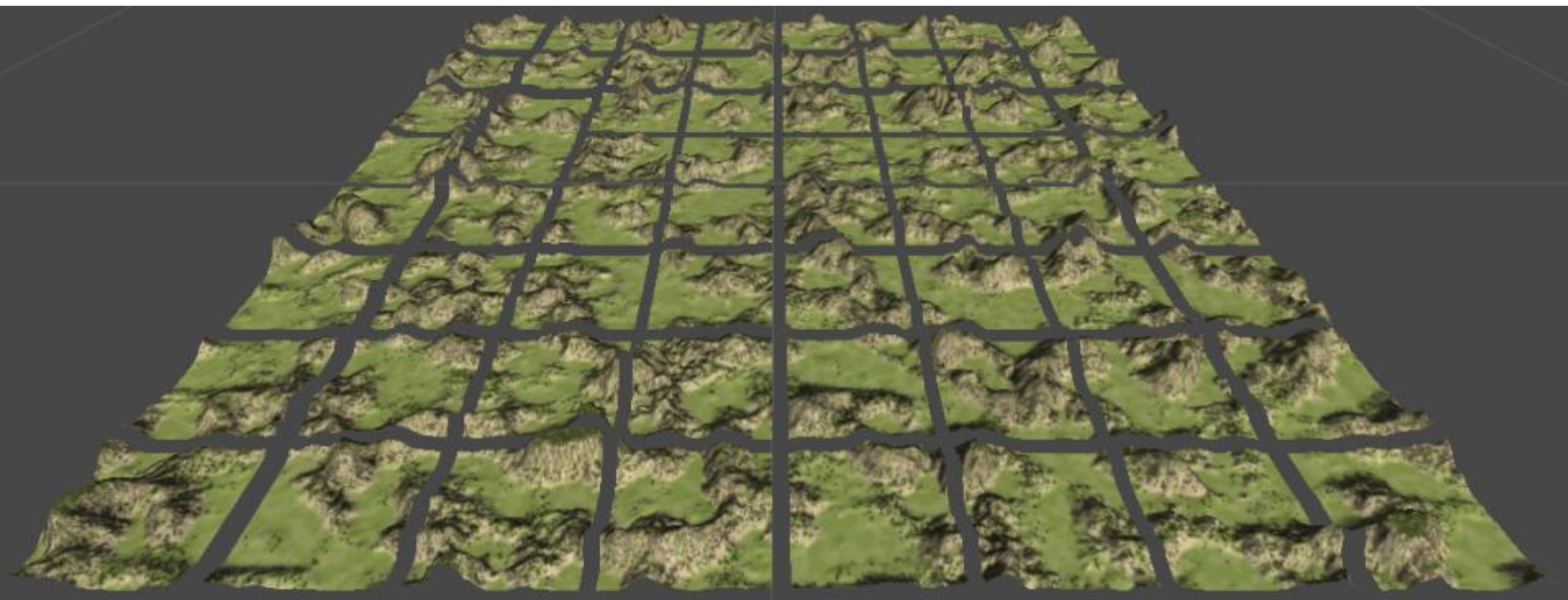
资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 3 x 3 地形



资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 8 x 8 地形



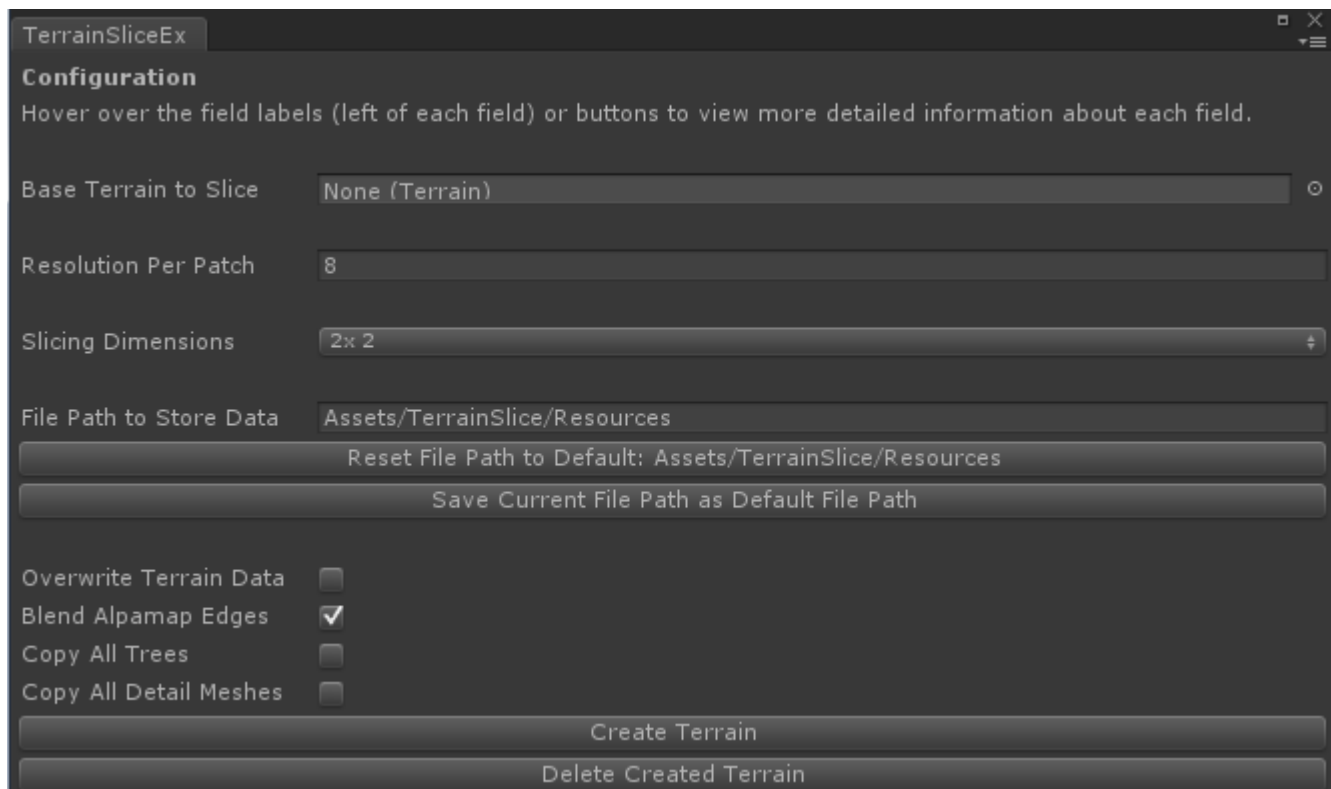
资源拆分

- 地形资源拆分
 - 使用Terrain Composer插件
 - 适用于
 - 从无到有自动生成地形
 - 不适用于
 - 针对某个已制作好的Unity地形进行拆分



资源拆分

- 地形资源拆分
 - 开发Slice Terrain插件



资源拆分

- 地形资源拆分
 - 深入解析Slice Terrain
 - 获得原始地形数据

Variables

alphamapHeight	Height of the alpha map.
alphamapLayers	Number of alpha map layers.
alphamapResolution	Resolution of the alpha map.
alphamapWidth	Width of the alpha map.
baseMapResolution	Resolution of the base map used for rendering far patches on the terrain.
detailHeight	Detail height of the TerrainData.
detailPrototypes	Contains the detail texture/meshes that the terrain has.
detailResolution	Detail Resolution of the TerrainData.
detailWidth	Detail width of the TerrainData.
heightmapHeight	Height of the terrain in samples (Read Only).
heightmapResolution	Resolution of the heightmap.
heightmapScale	The size of each heightmap sample.
heightmapWidth	Width of the terrain in samples (Read Only).
size	The total size in world units of the terrain.
splatPrototypes	Splat texture used by the terrain.
treeInstances	Contains the current trees placed in the terrain.
treePrototypes	The list of tree prototypes this are the ones available in the inspector.
wavingGrassAmount	Amount of waving grass in the terrain.
wavingGrassSpeed	Speed of the waving grass.
wavingGrassStrength	Strength of the waving grass in the terrain.
wavingGrassTint	Color of the waving grass that the terrain has.



资源拆分

- 地形资源拆分
 - 深入解析Slice Terrain
 - 创建新地形数据



资源拆分

- 地形资源拆分
 - 深入解析Slice Terrain
 - 拷贝地形数据
 - TerrainData
 - 地形高度
 - 地形分辨率
 - 地形混合贴图
 - 地形植被数据
 - 地形其他参数
 - 保存数据到JSON配置文件



资源拆分

- 地形资源拆分

- 深入解析Slice Terrain

- Lightmap数据处理

因为Unity地形供LightMap使用的uv2无法更改，致使创建的新的小块地形无法使用原有地形的LightMap

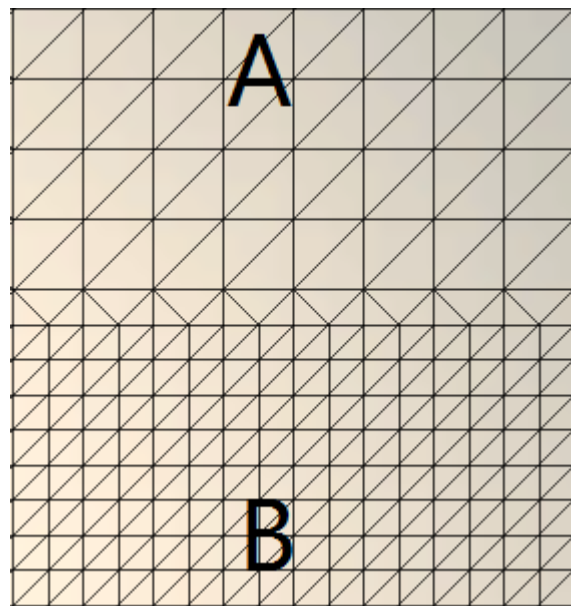
- 拆分后重新烘焙

- 切分烘焙好的Lightmap



资源拆分

- 地形资源拆分
 - 深入解析Slice Terrain
 - 设置邻接地形块
 - 需要解决不同LOD地块之间网格接缝缝合的问题
 - 如何设置
 - Unity提供了API进行该设置
`Terrain.SetNeighbors`



资源拆分

- 地表资源拆分



资源拆分

- 地表资源拆分
 - 按照地形块来拆分地表资源

```
[MenuItem("Assetbundle/Split GameObjects for Terrain")]
static void Excute()
{
    GameObject[] gos = Selection.gameObjects;
    if (gos.Length == 0)
    {
        Debug.Log("Please select gameobjects in the scene!");
        return;
    }

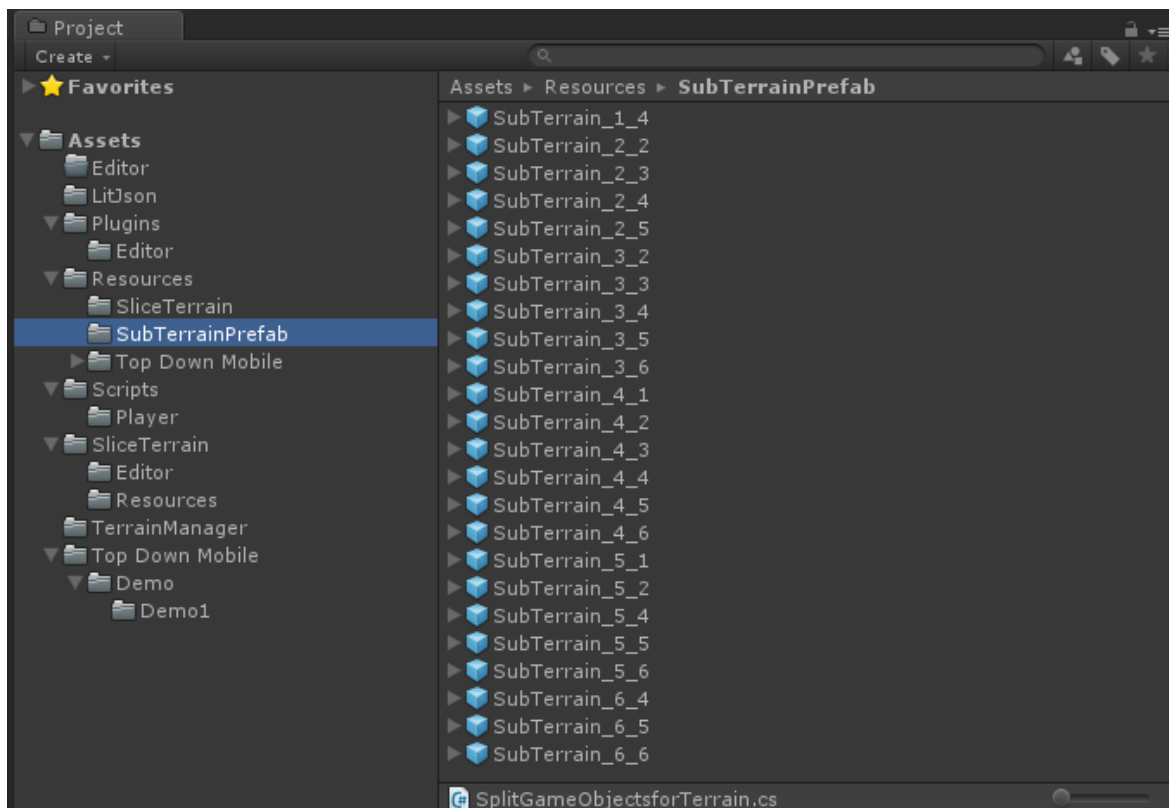
    ObtainTerrainInfo();

    for(int i = 0; i < gos.Length; i++)
    {
        SplitGameObject(gos[i]);
    }
}
```



资源拆分

- 地表资源拆分
 - 按照地形块来拆分地表资源



资源打包

- Assetbundle制作



资源打包

- Assetbundle制作
 - 对地形分块文件进行打包

```
[MenuItem("Assetbundle/Create Assetbundles for Terrain")]
static void Excute()
{
    Object[] terrainObjects = Resources.LoadAll(terrainPath, typeof(TerrainData));
    Object[] groundObjects = Resources.LoadAll(groundTexturePath, typeof(Texture2D));

    BuildPipeline.PushAssetDependencies();
    string groundABPath;

    groundABPath = "Assetbundles/Ground/GroundTexture.assetbundle";
    BuildPipeline.BuildAssetBundle(groundObjects[0], groundObjects, groundABPath,
        BuildAssetBundleOptions.CollectDependencies|BuildAssetBundleOptions.DeterministicAssetBundle);

    BuildPipeline.PushAssetDependencies();
    string terrainABPath;
    for(int i = 0; i < terrainObjects.Length; i++)
    {
        objs[0] = terrainObjects[i];
        terrainABPath = "Assetbundles/" + terrainObjects[i].name + ".assetbundle";
        Debug.Log(terrainABPath);
        BuildPipeline.BuildAssetBundle(objs[0], objs, terrainABPath,
            BuildAssetBundleOptions.CollectDependencies|BuildAssetBundleOptions.DeterministicAssetBundle);
    }
    BuildPipeline.PopAssetDependencies();
    BuildPipeline.PopAssetDependencies();
}
```



资源打包

- **Assetbundle制作**
 - 对地表资源文件进行打包
 - 根据依赖关系进行打包，方法同上



资源打包

- Assetbundle制作

- 注意事项

- 注意发布平台

- 如果在Android上运行，需要Android格式的AssetBundle

- 注意打包大小

- 每个Assetbundle文件尽量控制在1M以下
 - 过大者需预先加载



资源打包

- Assetbundle制作

- 注意事项

- 地形草无法显示的问题

- 在以AssetBundle的形式导入地形资源的时候，通常会发现草无法显示
 - 需要将草纹理的“Read/Write Enabled”设置为true，再将草纹理打包为AssetBundle文件，这样即可解决该问题



资源动态加载



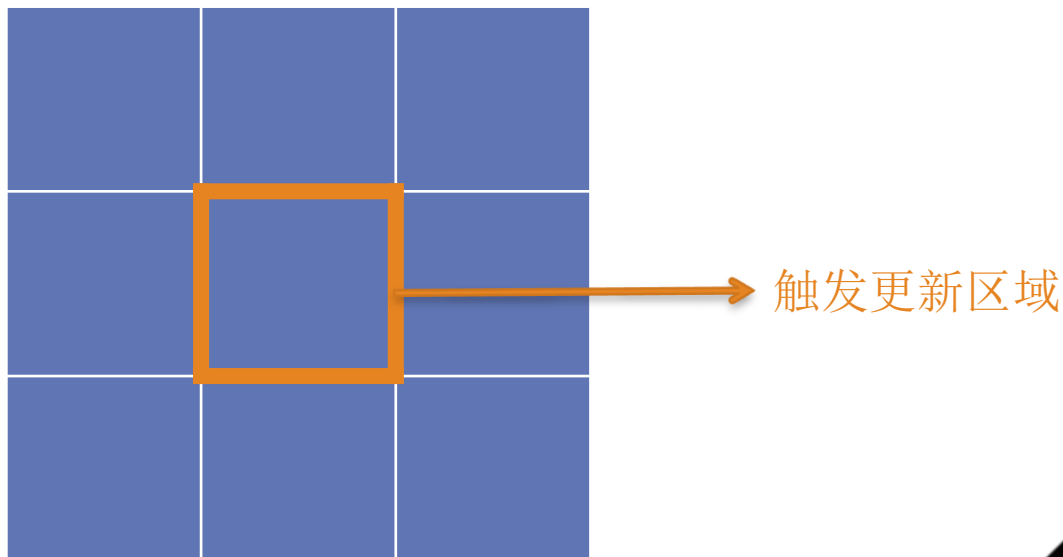
资源动态加载

- 优点
 - 避免一次性导入整块的场景数据
 - 仅加载角色周围的一些场景数据
 - 提高了加载速度并减少内存资源的消耗
 - 利用Cache机制，重用资源



资源动态加载

- 九宫格加载方法
 - 根据角色的位置，每次最多只显示9块场景数据
 - 当角色移动到“触发更新区域”时，更新需显示的场景块



资源动态加载

- 资源缓冲池
 - 一次性载入所有游戏资源是不现实的
 - 但如果仅载入当前使用的资源，又会频繁进行Instantiate和Destroy操作，势必影响游戏运行的流畅性



资源缓冲池（Object Pool）

资源动态加载

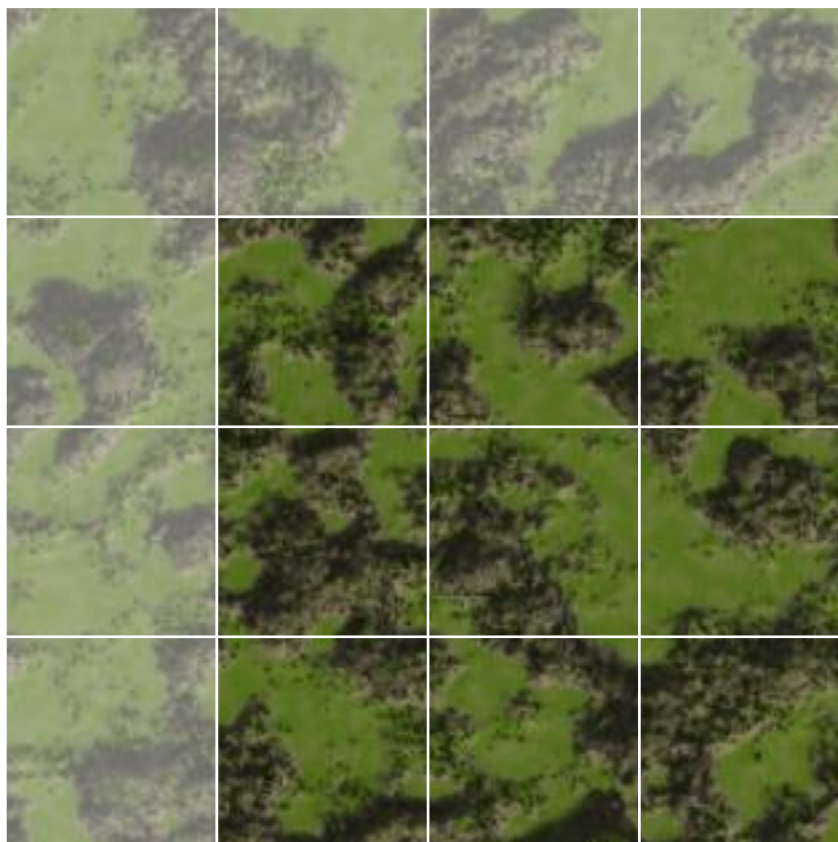
- 资源缓冲池
 - 创建一个固定大小的资源缓冲池
 - 新加载的场景资源首先放入池中
 - 如果池满了，清空最常不使用的资源
 - 从池中获取需要显示的场景资源



资源动态加载

- 4×4的资源缓冲池

New Terrain4
New Terrain5
New Terrain6
New Terrain7
New Terrain12
New Terrain13
New Terrain14
New Terrain15
New Terrain20
New Terrain21
New Terrain22
New Terrain23
New Terrain28
New Terrain29
New Terrain30
New Terrain31



资源动态加载

- 异步加载
 - 目前Unity推荐开发者使用www方法来下载和加载资源文件（Assetbundle文件）
- 使用WWW方式加载
 - www加载在Unity中是异步执行的
 - 需要用yield return返回
 - 资源加载代码通常会放在一个协程函数里，并通过调用StartCoroutine执行



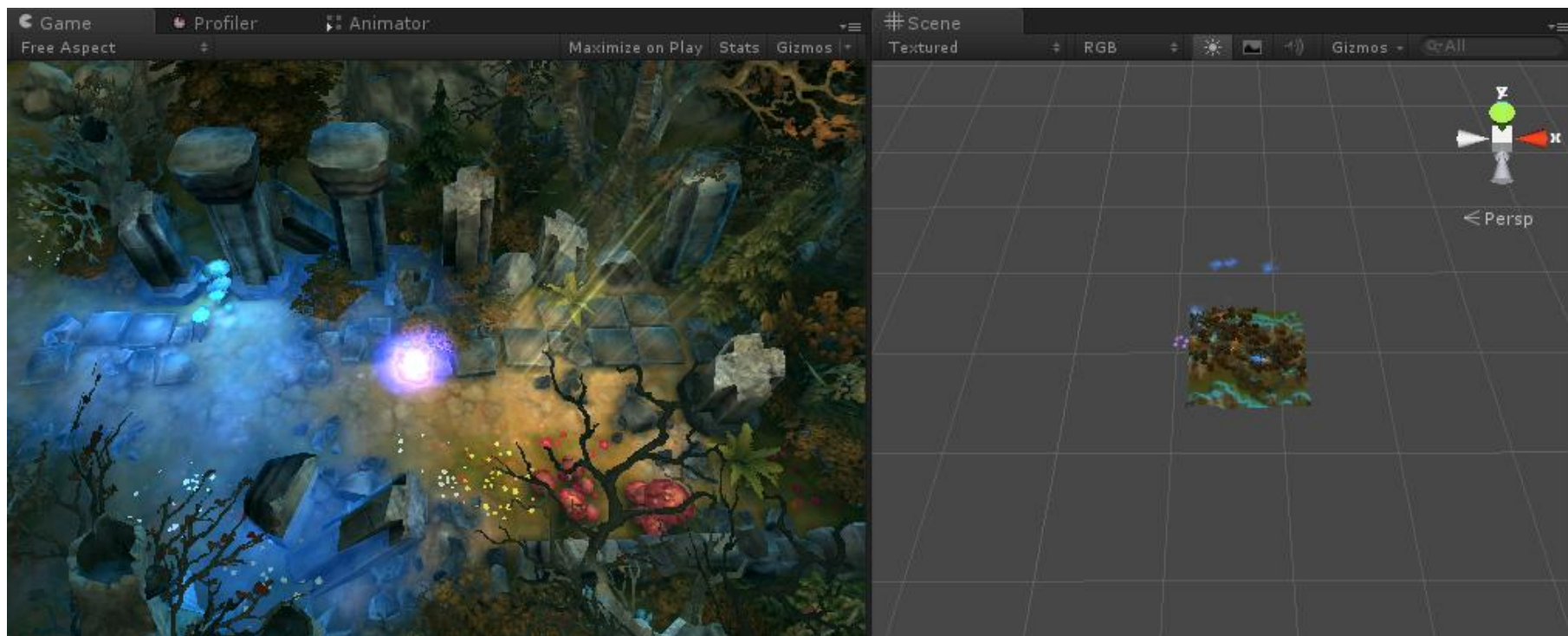
资源动态加载

- 异步加载
 - 协程的使用
 - 每个资源使用一个协程进行逐步加载
加载速度稍慢、内存占用量小
 - 使用一个协程逐步加载每个资源
加载速度稍慢、内存占用量小
 - 使用一个协程同时加载多个资源
加载速度快、内存占用量很大



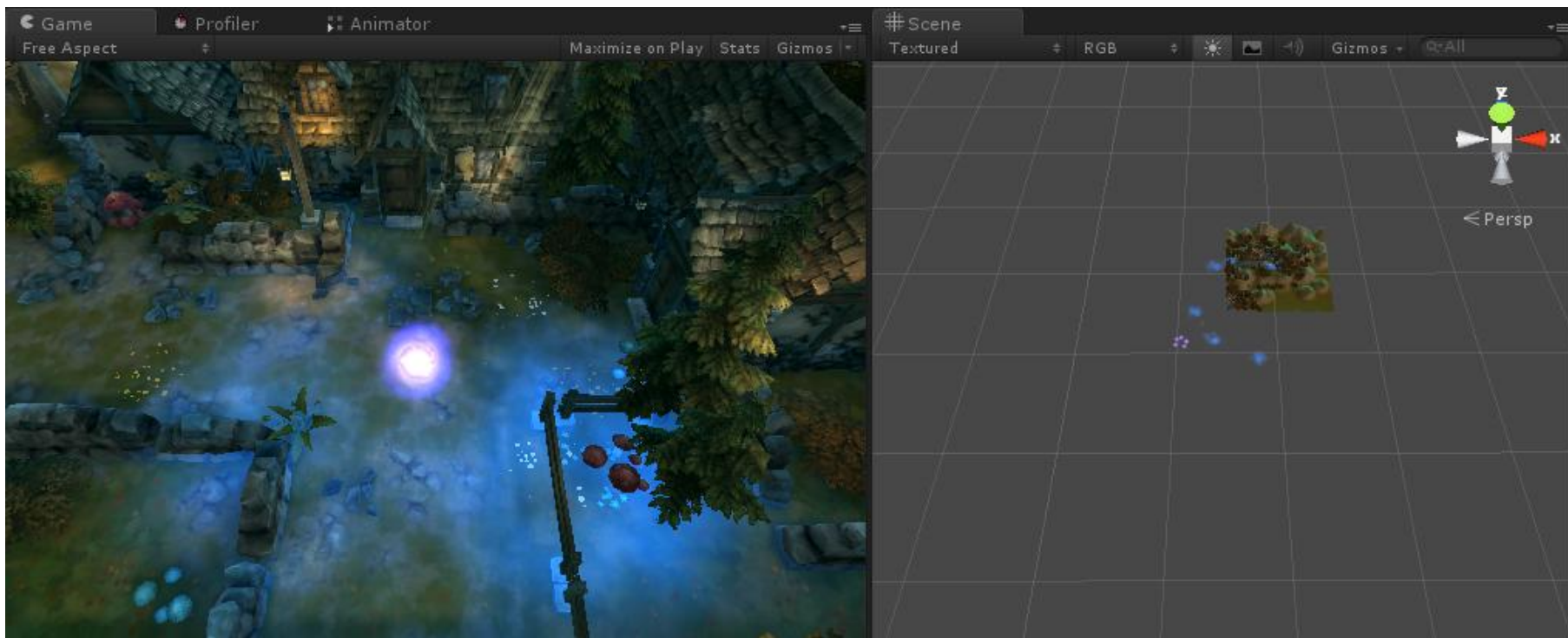
资源动态加载

- 动态加载效果展示



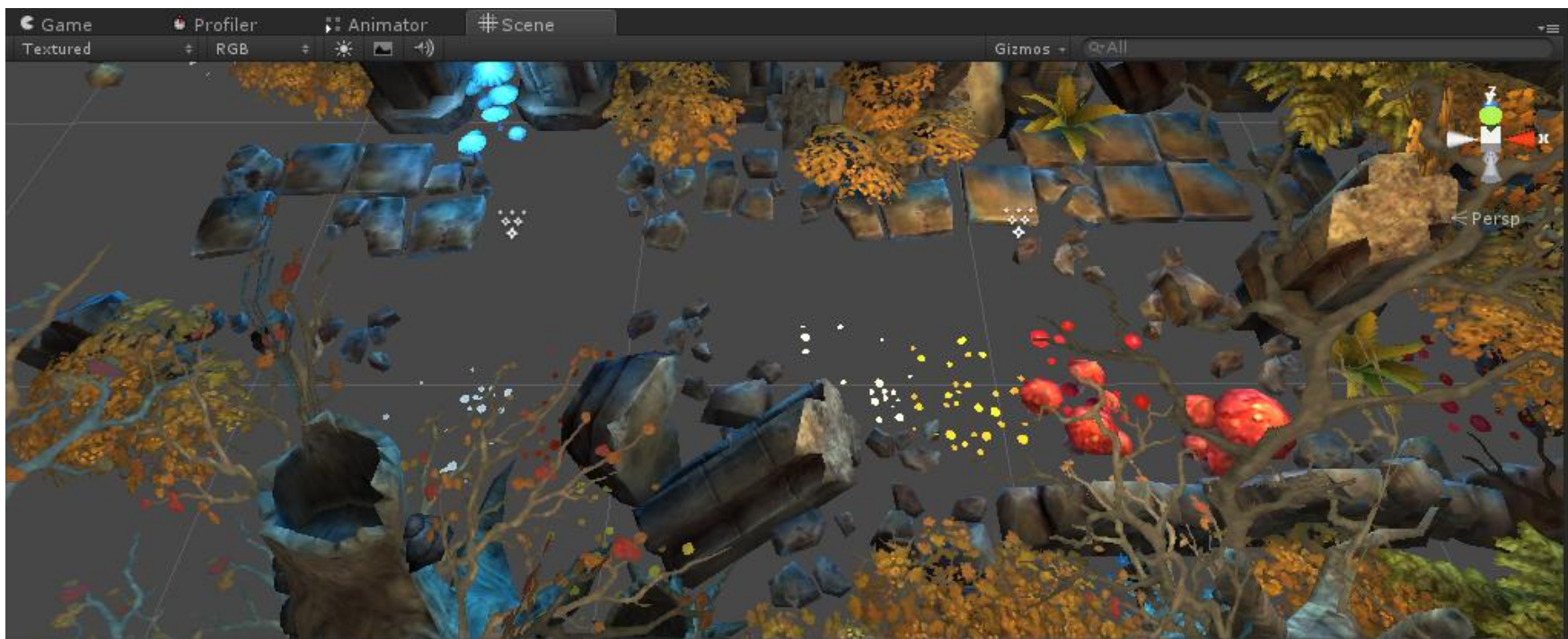
资源动态加载

- 动态加载效果展示



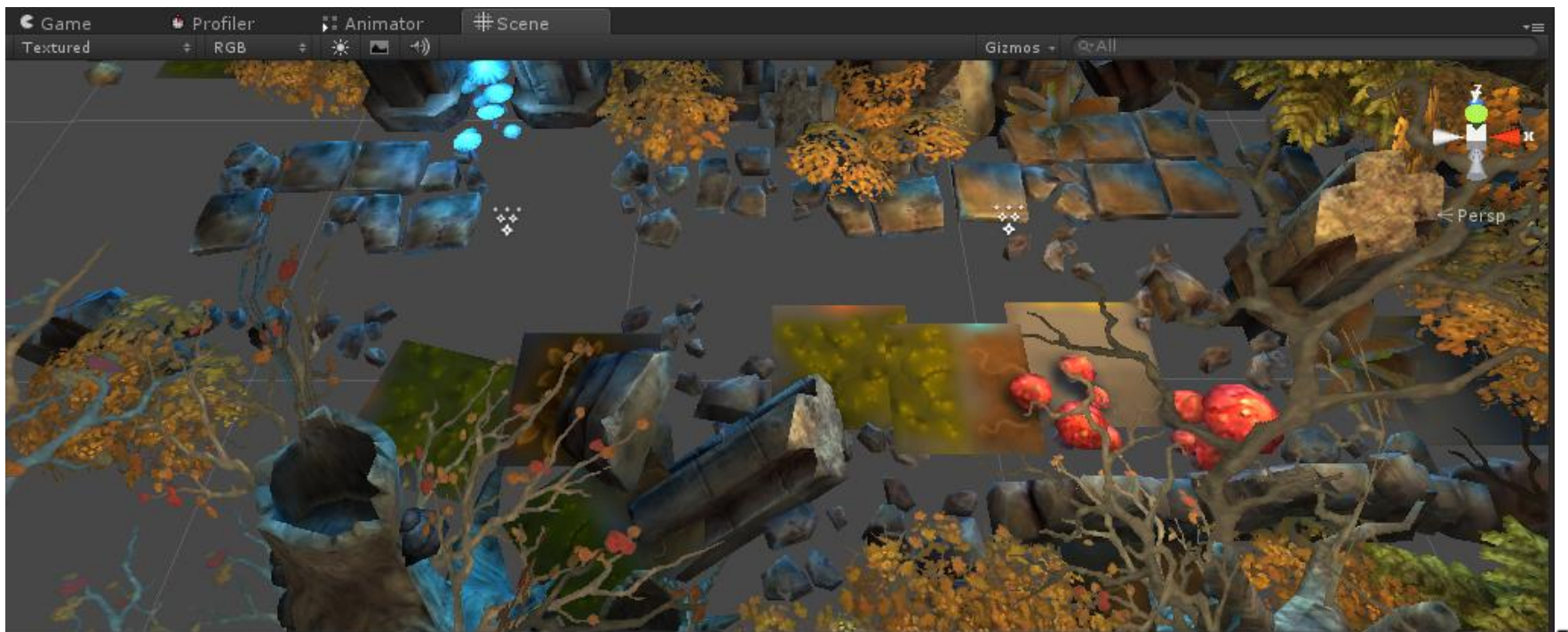
资源动态加载

- 资源增量更新
 - 初始Stamp纹理



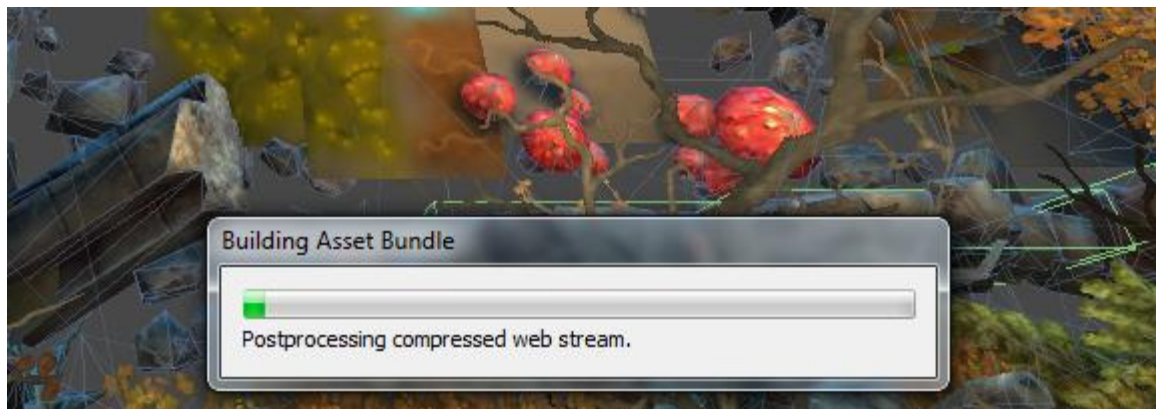
资源动态加载

- 资源增量更新
 - 初始Stamp纹理



资源动态加载

- 资源增量更新
 - 重新制作Assetbundle文件



Name

- ☐ Buildings_mat.unity3d
- ☐ Buildings_mesh.unity3d
- ☐ holder.assetbundle
- ☐ Nature_mat.unity3d
- ☐ Nature_mesh.unity3d
- ☐ Rocks_mat.unity3d
- ☐ Rocks_mesh.unity3d
- ☐ Ruins_mat.unity3d
- ☐ Ruins_mesh.unity3d
- ☒ Stamps_mat.unity3d
- ☐ Stamps_mesh.unity3d
- ☐ SubTerrain_2_5.unity3d
- ☐ SubTerrain_2_6.unity3d
- ☐ SubTerrain_3_3.unity3d
- ☐ SubTerrain_3_4.unity3d
- ☐ SubTerrain_3_5.unity3d
- ☐ SubTerrain_3_6.unity3d
- ☐ SubTerrain_4_3.unity3d
- ☒ SubTerrain_4_4.unity3d
- ☐ SubTerrain_4_5.unity3d
- ☐ SubTerrain_5_2.unity3d
- ☐ SubTerrain_5_3.unity3d

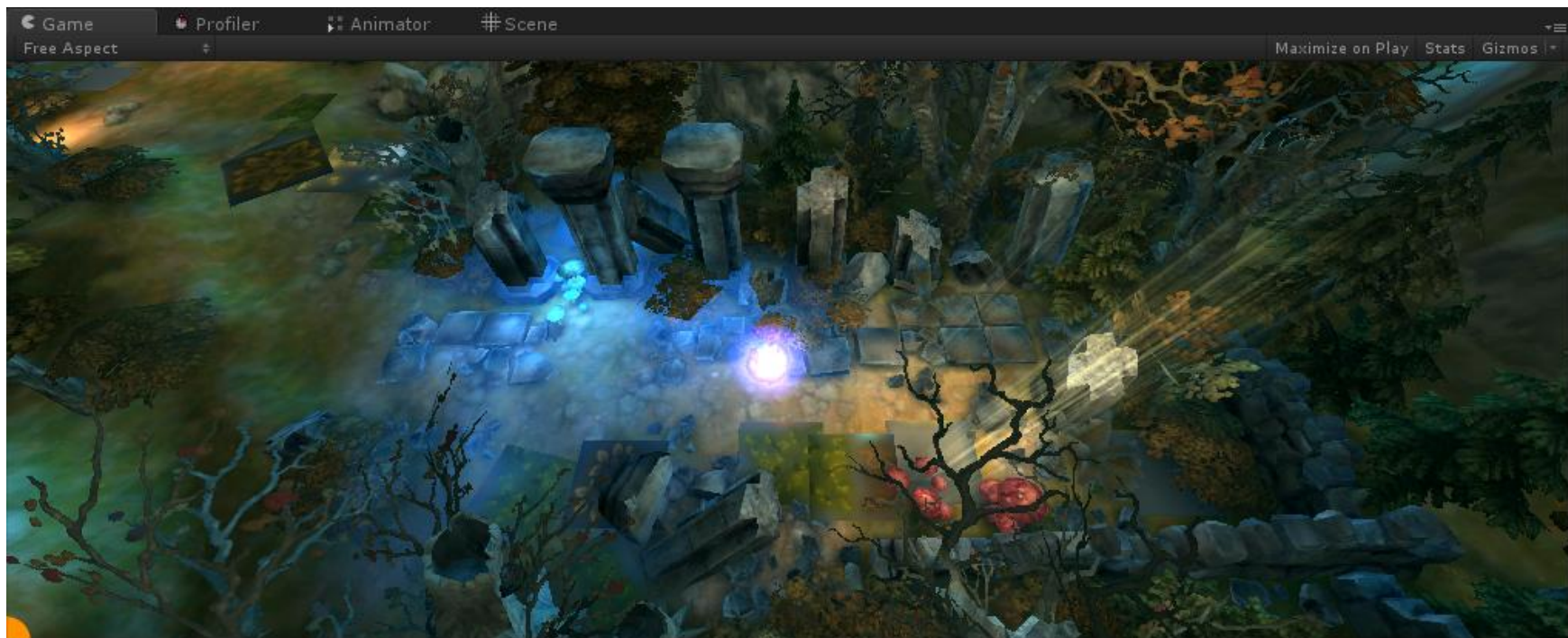
资源动态加载

- 资源增量更新
 - 重新制作Assetbundle文件
 - 仅将更新后的Stamps_mat Assetbundle文件拷贝至服务器，并覆盖对应文件即可
 - 将WWW.LoadfromCacheorDownload函数中加载该Assetbundle的版本号+1



资源动态加载

- 资源增量更新
 - 重新制作Assetbundle文件



谢谢

xin@unity3d.com

<http://china.unity3d.com>