1. 功能介绍

UI上需要的粒子动效，全屏特效等效果，都通过USpeedUI.UEffect.UEffectManager来管理创建、移除。目前支持的UI效果有：粒子动效、全屏特效、UI材质效果、UI模型效果。

1. 名词解释
2. 设计思路

USpeedUI.UEffect.UEffectManager根据创建时的效果类型和效果初始化参数创建对应的效果并返回需要的数据。

1. UEF\_EffectPrefab使用方法

游戏里UI的所有粒子动效都属于这个类型，建议所有UI粒子动效都通过脚本代码来创建，而不是把粒子动效直接挂到View上并将粒子动效和View序列化一起。因为Unity的预制体不支持嵌套，粒子动效与View序列化之后是一个新的预制体，与原粒子动效预制体无关。如果原粒子动效预制体更新的话，同时也需要更新对应的View里面的粒子动效。

* 1. 需要新增预制体效果：
     1. UEffectPrefabType 添加新类型
     2. 从美术那获取新类型对应的预制体文件，并填到m\_dicEffectDataTable里面
     3. 在新预制体里加上脚本UEffectNode
  2. 使用
     1. 创建效果参数对象并填入你需要的参数 var param = new UEffectPrefabParam();
     2. 执行UEffectManager.CreateEffect(UEffectType.UET\_EffectPrefab,ref param);
     3. 手动移除执行 UEffect.DestroyEffect(UEffectType.UET\_EffectPrefab,ref param);
     4. 不主动移除的话，一般会随View的删除而删除。如果创建参数设置AutoDestroy=true时，效果播放结束会自动移除，但前提是该效果非循环效果;

1. UEF\_EffectImage使用方法

UI使用到的全屏效果都走这类型，注意是UI使用的，不是游戏使用的全屏效果。

* 1. 需要新增全屏效果：
     1. 找客户端给全屏效果的类名
  2. 使用
     1. 单独使用

UEffectParamBase param =

new EffectImageParam(typeof(\_eAttachImageEffect:typeof(ImageEffects.BlurAndDownScreen),gameobject));

创建 UEffectManager.CreateEffect(UEffectType.UET\_EffectImage,ref param);

移除 UEffectManager.DestroyEffect(UEffectType.UET\_EffectImage,ref param);

* + 1. View使用

显示/隐藏View的时候使用默认效果ImageEffects.BlurAndDownScreen; View.canUseImageEffect = true;

如果需要在显示/隐藏的时候用另一种效果 View.imageEffect = typeof(另一种效果类);

1. UEF\_EffectMaterial使用方法

UI会用到的一些材质效果，目前还没完善，而且没有地方用到。先不写了，后续补充。

1. UEF\_EffectMesh使用方法

需要在UI上显示模型是，走这个方法，将模型渲染成UI元素，进行使用。

* 1. 编辑器使用方法
     1. 在UI上新建一个GameObject，并挂上UEffectMeshRenderNode脚本。RawImage就是用于显示模型的UI元素。
     2. 设置相机渲染图片尺寸大小，值越大，渲染出的图像越清晰，消耗也越大。不建议超过1920x1080大小。且要注意RawImage的尺寸大小要与渲染图片尺寸大小比例一致，否则会出现图像拉伸情况。
     3. 调整相机的一些属性，目前只列出常用的，其余的只能通过代码实现。
     4. 增加渲染对象，并将要渲染的预制体/场景里的对象拖到这。
     5. 如何判断是否为场景对象，就是观察的对象已经实例化并在场景里面了，不需要再次进行实例化，只有预制体对象才需要进行实例化。
     6. 调整观察对象相对相机的位置。
     7. 需要移除的话只需这个挂有UEffectMeshRenderNode的GameObject删除掉即可。
  2. 脚本使用方法
     1. 初始化UEffectMeshParam所需的参数
     2. 通过UEffectManager.Instance.CreateEffect方法创建UEF\_EffectMesh类型的效果
     3. 如果需要调整观察相机（UEffectMeshCamera）和模型渲染节点(MeshRenderNode)的其余属性，可以在CreateEffect之后通过传入的param参数获取初始化好的UEffectMeshCamera对象，并调整Camera的属性。

UEF\_EffectMesh的观察相机默认是同一管理位置的，使用者也可以将观察相机挂接到某一个Transform上让观察相机随Transform运动。只需获取UEffectMeshCamera并设置

IsAttachToObject=true;AttachObject = 相机要挂接的对象。