创建脚本FunctionLibrary，在FunctionLibrary静态类中完成一系列数学函数，使得Graph脚本中可以在不进行实例化的情况下使用FunctionLibrary中的函数并且无法得知其实现。

进行运算时尽量使用乘法而不使用除法，因为除法比乘法更消耗时间和性能。

取绝对值写三目运算符比Mathf.Abs()快2~3倍。

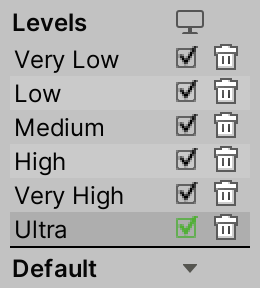
一个条件块序列适用于两个或三个函数，但是当试图支持更多函数时，它很快就变得笨拙。如果我们可以根据某些标准向库请求方法的引用，然后重复调用它，那将会方便得多。可以通过使用委托来获取方法的引用。为库方法（此处是FunctionLibrary类中的方法）定义委托。

委托充当表现层和数据层间的桥梁，有助于提高模块化程度，有助于解耦。

Unity中enum类型会默认转换为下拉框进行选择。

镜头调整：在游戏模式中在场景窗口中找到一个不错的视点，退出游戏模式，然后让游戏的镜头与视点匹配。你可以当Main Camera被选中和场景窗口可见时，通过 GameObject / Align With View 来做到这一点。让它大致沿着XZ对角线向下。

可以调整Direction Light的Rotation获得适宜的光照角度。我们还可以稍微调整阴影的质量。阴影可能已经接受当使用默认渲染管道,但当我们在近距离观察图表时，它们被配置为看远处。



Edit->Project Settings->Quality调节视觉质量等级

在下面的Shadow中可以调整关于阴影距离在阴影性能和精度

URP不使用Quality的这些设置，相反，它的阴影是通过我们的URP资产的检查器配置的。默认情况下，它只渲染一次定向阴影，但Shadows / Max Distance可以减少到10。另外，为了匹配默认渲染管道的标准Ultra质量，启用Shadows / Soft Shadows，并在Lighting下增加Lighting / Main Light / Shadow Resolution到4096。

什么是阴影级联和距离控制?

Unity(以及大多数游戏引擎)会将阴影投射器渲染到纹理中，然后采样它们来创建阴影。这些阴影地图有固定的分辨率。如果它们必须覆盖一大片区域，单个像素也会变大，从而产生块状阴影。

通过减少最大阴影距离，我们减少了阴影贴图必须覆盖的区域，从而提高了阴影质量，但代价是丢失了远处的阴影。

阴影级联通过使用基于距离的多个地图来进一步实现方向光，所以附近的阴影最终会有一个有效的更高分辨率。

在播放模式下视觉撕裂。通过游戏窗口工具栏左侧的第二个下拉菜单，启用\*VSync(Game view only)\*可以防止这种情况发生在游戏窗口中。启用后，新帧的呈现与显示刷新速率同步。这只在同时没有场景窗口可见的情况下可靠地工作。垂直同步是通过质量设置的Other部分为独立的应用程序配置

产生对角波Wave函数：Sin(PI \* (x + z + t)) 最长的一行是x和z相加为0的情况，约为根号2倍的resolution

产生波浪形Multwave是让每个波使用一个维度，两个维度的函数叠加形成这样的波

float y = Sin(PI \* (x + 0.5f \* t));

y += 0.5f \* Sin(2f \* PI \* (z + t));

加上第三个波，沿着XZ轴对角线传播

y += Sin(PI \* (x + z + 0.25f \* t));

叠加的波会以一定周期出现干涉相消的情况

如何形成一个封闭的形状：

通过使用X和Z来定义Y，我们能够创建函数来描述各种各样的曲面，但它们总是与XZ平面相连。没有两个点Y坐标不同，X坐标和Z坐标相同。这意味着我们表面的曲率是有限的。它们的斜率不能变成垂直的，也不能向后折叠。为做到这一点,我们的函数必须不仅输出Y,还有X和Z。