# HUD

UI屏幕的坐标系，左上角是原点，想右下角进行延展

### DrawHUD入口函数

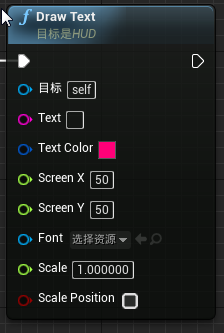


Size X 返回的是屏幕的分辨率宽度

Size Y 返回的是屏幕分辨率高度

### 常用的绘制节点

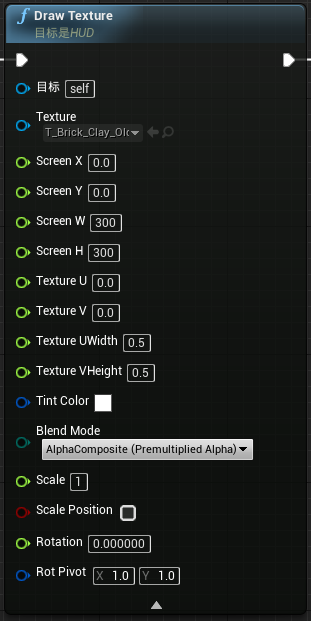
* 绘制文字



Screen X和Y标记了绘制文字在屏幕中的位置

Scale Position 勾选后，缩放信息将会影响位置信息，例如原位置50，50，缩放3倍，将变成150，150

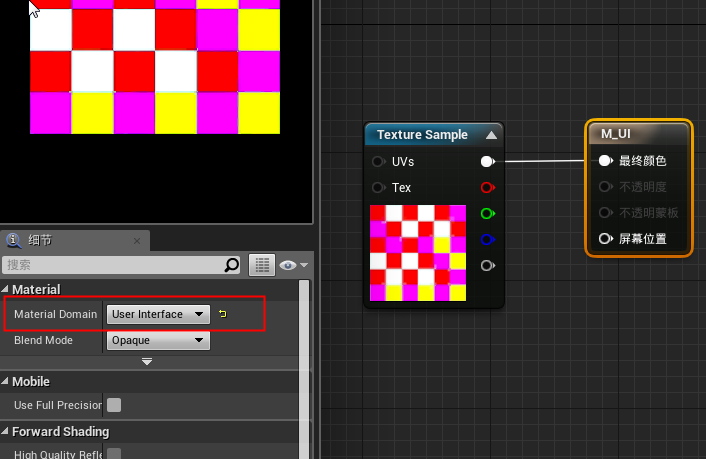
* Draw Texture



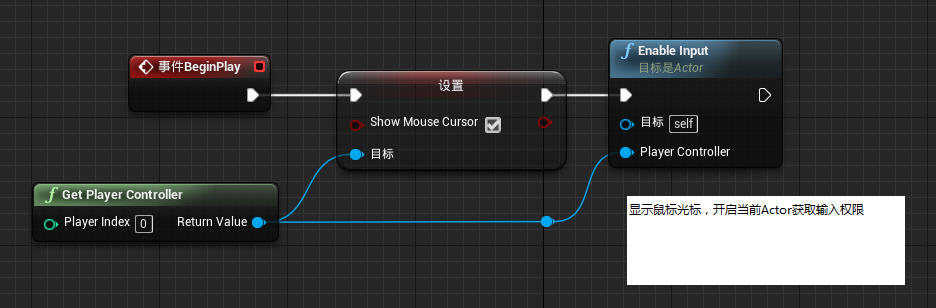
参数解释：

* Texture 贴图绘制源
* Screen X Y绘制贴图到屏幕上的坐标
* Screen W H 绘制贴图在屏幕中的尺寸
* Texture UV 绘制贴图纹理UV的起始点坐标
* Texture UWidth VHeight 绘制贴图纹理UV宽度和高度
* Tint Color 叠加色
* Blend Mode 混合模式，决定了绘制贴图和场景的融合方式。默认使用半透明。
  + Opaque 无半透明信息绘制，使用不透明绘制
  + Masked 遮罩绘制，贴图中的Alpha值将只有两个结果，如果小于1，则等于0，如果等于1， 则等于1
  + Tran 透明模式，将贴图中的透明信息进行展示。
  + Additive 叠加模式，类似PS中的线性减淡，不会进行暗化，所有像素都将被加到一起，黑色由于是0，将直接被渲染为底图，表现为透明
  + Modulate 调制模式，类似PS中的正片叠底，表现为色调变暗。
  + Alpha Composite 有点像叠加，但是比叠加更细腻，叠加在背景图是白色的情况下将会出现颜色过曝，导致细节丢失。而AC不会出现此问题。
  + Scale 贴图绘制缩放
  + Scale Position 缩放时坐标信息是否跟随缩放
  + Rotation 贴图旋转 正值顺时针旋转 负值逆时针旋转
  + Rot Pivot 锚点位置，左上角是0，0右下角是1.1，例如，如果是0.5，0.5则表示旋转参考点在图片正中心。
* Draw Material

绘制材质，注意只有材质球的材质域是User Interface的材质球才可以在UI中被绘制



### 如何在屏幕上显示鼠标并获取输入权限



### 基于HUD的点击响应输入事件

**必须开启Enable Click Event，在PlayerController**

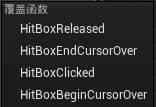


* 节点AddHitBox



参数描述

* + Position 响应框起始点
  + Size 响应框大小
  + In Name 响应框名称，主要用来区分响应框
  + Consumes Input 是否组织点击事件传递，如果勾选，响应框重叠是将只有最上层响应框获取到输入事件
  + Priority 响应框优先级排序，0就加入到队列的末尾，1则加入到队列顶端
* 接收响应逻辑节点



* + 当鼠标在响应区域中松开左键时，被调用
  + 当鼠标移出响应区域时调用
  + 当鼠标在响应区域中点击左键时，调用
  + 当鼠标移入响应区域时么被调用

### 在HUD中如何进行屏幕空间坐标和世界空间坐标转换

以下节点只能在HUD的DrawHUD中调用

* Project



将空间坐标（3D）转为屏幕坐标

* Deproject



将屏幕坐标转投到空间坐标，基本参照源是摄像机

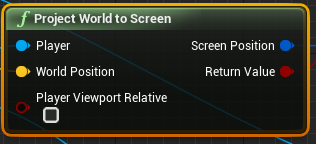
### 更好用的空间坐标转换

在PlayerController里面

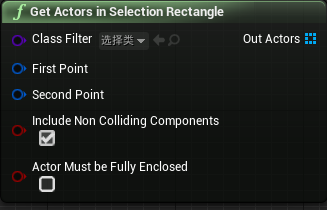
* 屏幕坐标转空间坐标



* 世界坐标转屏幕坐标



### HUD中框选目标函数



* + First Point 框选起始点
  + Second Point 框选中节点
  + 是否选择那些没有碰撞组件的物体 默认是勾选
  + 框选目标是否必须完全被选中。例如，如果勾选此项，被选中的目标必须完全处于选择框内，才会被选取。