

# UE中的UI实现

## 什么是UI

UI是“用户界面”(User Interface)的缩写，指的是用户与设备、软件或应用程序进行互动的界面。UI设计的目的是让用户能够直观、便捷地使用系统和功能。U不仅包括屏幕上的按钮、图标、文字、菜单、颜色、布局等视觉元素还涵盖了用户与系统之间的交互方式。

## 游戏中的UI

游戏UI通常需要更注重沉浸感和视觉冲击力，传递信息的同时还要增强玩家的游戏体验。且游戏U更侧重于动态性和交互性，例如生命值、分数、任务进度等信息经常需要实时更新。此外，游戏UI设计常与游戏的整体美术风格、主题和情节紧密结合，以确保其与游戏世界的风格一致，创造更具沉浸感的体验。

## SlateUI与UMG

在虚幻引擎中，U的实现一般有这两种方式

### Slate UI

Slate是虚幻引擎底层的UI框架，用于构建和管理用户界面。用以提供高效、灵活的UI元素构建和渲染。Slate是虚幻引擎中所有UI的基础，很多U的核心功能都建立在Slate之上。Slate框架非常强大，但它的使用相对较复杂，开发者需要编写大量的C++代码。

### UMG控件蓝图

UMG(Unreal Motion Graphics)是虚幻引擎的可视化UI工具，基于Slate构建。它通过可视化的蓝图系统让开发者和设计师可以不需要编写C++代码便捷地构建和管理游戏中的UI界面。同时UMG也提供了较为简单的UI编辑，以方便高效的制作UI界面及动画表现。

简而言之，UMG相当于对Slate进行了易用化封装，提供一个更好的UI设计体验，但UMG底层仍依赖Slate进行UI渲染和交互处理。

## UMG编辑器-设计器

层级视图:用于展示当前正在编辑的UI中的组成元素和层级关系

细节面板:用于调整UI元素的属性(如位置信息，材质，颜色等等)

绑定控件:显示当前绑定到C++变量的蓝图控件或动画

动画轨道：用于编辑控件动画

## UMG编辑器-图表

这里控件蓝图与普通蓝图类似，可以在图表中使用蓝图可视化编程实现功能

## UMG蓝图制作要点

为了呈现同时具备功能性和美观性的游戏U，控件蓝图的制作一般会涉及到以下几点

- 元素展现:通过图片，文字等内容，直观的展现UI需要表述的信息
- 交互体验:使用按钮，滑动条，输入框等交互元素获取玩家输入以达成交互功能
- 布局调整及适配:通过合理的布局以美化U整体效果

- 动画效果:动画效果的实现对于可以进一步提升UI的美术效果!
- 功能实现:以代码驱动UI的整体运行

将控件设置成变量以便使用蓝图代码操作

## 控件事件绑定

preconstruct

construct

OnInitalized

## 通过C++操控UMG控件蓝图

变量绑定:

```
public:
/** testing what bind stands for in UMG editor */
UPROPERTY(BlueprintReadOnly, meta=(Bindwidget))
UTextBlock* TextBlock_TestBinding,
/** another test just for the case of using animation */
UPROPERTY(BlueprintReadOnly, Transient, meta=(BindwidgetAnim))
UWidgetAnimation* TestBindAnim;
/** testing optional bind widget */
UPROPERTY(BlueprintReadOnly, meta=(Bindwidgetoptional))
UTextBlock* TextBlock_TestOptional;
/** another test optional bind widget animation *) */
UPROPERTY(BlueprintReadOnly, Transient, meta=(BindwidgetAnimoptional))
UWidgetAnimation* TestOptionalAnim;
```

在HUD类中创建UMG蓝图并显示到玩家视窗: AddToWiewPort

锚点的控制, 控件填充的控制