# Unity 编辑器

## IMGUI

传统GUI步骤

1. 声明控件变量
2. 绑定控件
3. 响应控件事件

如：Unity UI系统

立即模式GUI步骤

1. 直接调用控件函数（内部自动处理绘制、点击等）

缺点：

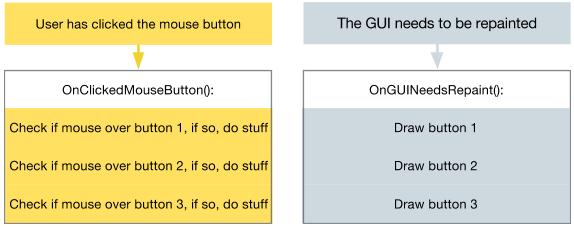
* 作为游戏UI低效率（每帧调用多次）
* 逻辑操作困难（没有控件变量）

优点：

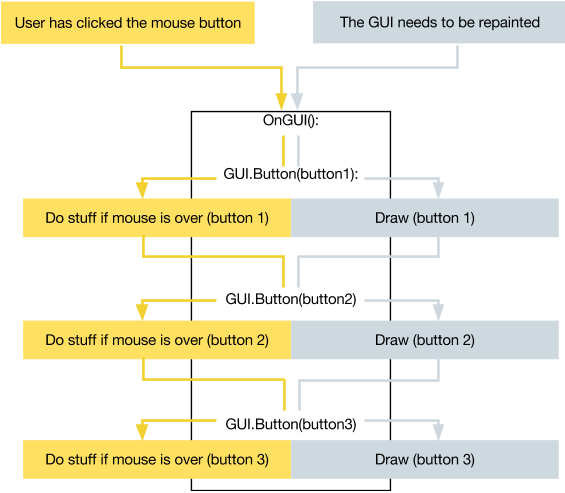
* 简单粗暴，不需要多余的代码
* 作为编辑器UI非常便利
* 非实时非全局的重绘

## 事件

对于传统GUI点击事件和绘制分开处理：

****

IMGUI统一处理：



在GUI.Button里面使用Event.current来判断当前按钮事件，比较常用的事件类型：

* EventType.MouseDown 鼠标按下
* EventType.MouseUp 鼠标抬起
* EventType.KeyDown 键盘按下
* EventType.KeyUp 键盘抬起
* EventType.Repaint 重绘控件

用控件ID来识别不同控件。

## 自定义控件

### 自定义滑动条

带有颜色的[0,1]滑动条，类似血条表现，当1的时候表示满血状态，当0的时候表示被击杀状态。

函数签名：

* Rect 绘制矩形区域
* Float 滑动条值
* GUIStyle 绘制的风格

返回值：

* 通过拖动滑动条设置的新值

完整函数签名：

|  |  |
| --- | --- |
|  | public static float CustomHPSlider(Rect controlRect, float value, GUIStyle style) |

* 示例：CustomHPSliderDemo

焦点类型FocusType：

* Native 只在Windows平台可以获取键盘焦点，如按钮等；
* Keyboard 所有平台键盘焦点，文本控件使用；
* Passive 不接收任何键盘焦点，即不接收按键事件；

当前鼠标捕获的控件ID GUIUtility.hotControl：

鼠标按下去的时候，捕获当前位置的控件，设置hotControl为控件ID，这样其他控件就会被过滤鼠标事件。抬起来的时候，释放捕获的控件，设置hotControl为0。

GUI.changed = true; 标记改变

才能被EditorGUI.BeginChangeCheck()和EditorGUI.BeginChangeCheck()检测到控件值改变。

避免设置GUI.changed = false，可能会隐藏前一个控件值的变化。

Event.current.Use(); 标记事件已被使用

自动设置事件类型EventType.Used，其他控件不再被响应。

控件需要保留更多信息的情况

IMGUI提供一个简单的存储系统“State Object”来关联控件。定义类存储值，关联控件ID，IMGUI会负责构造。不会被序列化，只能用于短暂生命期使用。

### 自定义闪烁按钮

创建一个按钮，当按下超过2秒的时候，会闪红。

实现这个需要跟踪按钮的按下时间，通过“State Object”状态对象来实现。

* 示例：FlashingButtonDemo

## GUIStyle风格样式

GUI元素界面属性，如字体、文本颜色、布局属性等。自动绘制不同情况下的样式，如鼠标滑动、键盘焦点、禁用状态，点中状态等。

|  |  |
| --- | --- |
|  | Draw(Rect position, GUIContent content, int controlID) |

内部通过传入的控件ID，来获取控件的状态。

获取GUIStyle四种方式：

* 动态构造 new GUIStyle()，手动赋值；
* 直接使用内建风格（EditorStyles）；
* 基于当前风格修改new GUIStyle(existingStyle)，比较常克隆使用EditorStyles；
* 从GUISkin皮肤类获取，如内建的GUI.skin.button；

GUISkin本质上是GUIStyle的集合。也是资产文件，可以保存在工程里面。每个皮肤文件都包括标准控件样式和自定义样式，自定义样式通过名称来获取。

* 示例：CustomSkinDemo

## 位置布局

自动计算布局，不再需要传入矩形参数。触发计算布局方法：

* GUILayoutUtility.GetRect()
* GUILayout.BeginHorizonal / Vertical
* GUILayout.EndHorizontal / Vertical

这些方法会响应EventType.Layout事件，构建控件布局树，构建完成就可以知道每个控件的真实宽高。

GUILayoutOption布局选项，限定控件布局。如限定高度30；水平扩展填充等。

* GUILayout.Width / GUILayout.Height 固定宽高
* GUILayout.MinWidth / GUILayout.MinHeight 最小宽高
* GUILayout.MaxWidth / GUILayout.MaxHeight 最大宽高
* GUILayout.ExpandWidth / GUILayout.ExpandHeight 水平垂直扩展
* 示例：CustomHPSliderLayoutDemo