

UGUI

Unity Graphical User Interface

控件

Event

ITween

UGUI

简介

UGUI是什么

Unity界面发展史

与OnGUI对比

与NGUI对比

UGUI(1)

基础控件

Canvas

Rect Transform

Image

Text

Button

Input Field

Toggle

Slider

Scrollbar

简介



UGUI是什么

- Unity 图形用户界面 Unity Graphical User Interface
是Unity4.6以上版本新加入的界面显示系统



Unity 界面发展史

排行榜		
序号	姓名	成绩
1	令狐冲	98
2	任我行	59
3	岳不群	30



与OnGUI对比

- 全新的布局系统
Rect Transform Layout Group
- 强大的事件机制
鼠标指针类 拖拽类 点选类 输入类
- 最佳的执行效能
减少了GPU负担



与NGUI对比

- 由NGUI创始人参与开发
- 自适应系统更完善
- 更方便的深度处理
- 淡化了Atlas，直接使用Sprite Packer
- UGUI没有Tween组件
- 开发效率暂时低于NGUI



基础控件



Canvas 画布

- 简介

画布，绘制UI元素的载体，所有UI元素必须在Canvas之下。
UI元素的绘制顺序依赖于层次面板中的顺序。

- 属性

- Render Mode 渲染方式

Screen Space-Overlay 覆盖模式：UI元素将绘制在其他元素之前，且绘制过程独立于场景元素和摄像机设置，画布尺寸由屏幕大小和分辨率决定。

Pixel Perfect 完美像素：若勾选，则会锐化屏幕显示效果。

Sort Order 渲染顺序：在多个Canvas中，值越大越渲染到最上层。

Canvas 画布(续1)

- Screen Space-Camera 摄像机模式：提供UICamera，Canvas对象被绘制在一个与摄像机固定距离的平面上，且绘制效果受摄像机参数的影响。

Render Camera 渲染摄像机。

Plane Distance 平面与摄像机的距离。

Sorting Layer 排序层：通过Edit--Project Settings—Tags and Layers调整Canvas渲染顺序。

- World Space 世界空间模式：画布渲染于世界空间，与场景中其他3D物体性质相同。

Rect Transform 矩形变换

- 简介

派生自Transform，在UGUI控件上替代原有变换组件，表示一个可容纳UI元素的矩形。

- 属性

Pos：控件轴心点相对于自身锚点的位置。

Anchor 锚点：UI元素的4个顶点与锚点的间距保持不变

锚点总是相对于父级，不能超越父物体范围

表示点 --> PosX PosY Width Hight

表示拉伸 --> Left Right Top Bottom

Pivot 轴心点：移动、旋转与缩放都围绕轴心点发生变化，

0,0为左下顶点，1,1为右上顶点

Image 图片

- 属性

Image Type 贴图类型：

Simple 简单

Preserve Aspect 保持贴图原始比例

Set Native Size 将贴图设置为原始尺寸

Sliced 切割

Fill Center 填充中部区域，如果取消中部区域为透明

Tiled 平铺

保持图片原始尺寸，从左下角重复多次填充空白

Filled 填充

可以呈现出从空白到完整填充的过程。

Text 文本标签

- 属性

Character 字符

Font 字体

Font Style 样式

Font Size 大小

Line Spacing 行间距

Rich Text 是否使用富文本样式

Paragraph 段落

Alignment 对齐方式

Horizontal / Vertical Overflow 水平 / 垂直 溢出

Best Fit 大小自适应 范围在Min Size 与 Max Size 之间

Color 颜色

Material 材质

Text 文本标签(续1)

- 富文本语法

`粗体`

`<i>斜体</i>`

`<size=14>字号</size>`

`<color=green>颜色</color>`



Button 按钮

- 属性

Interactable 是否启用交互

Transition 过渡方式

Color Tint 颜色过渡

Normal 正常 Highlighted 高亮 Pressed 点击

Disabled 禁用 Multiplier 颜色倍数

Fade Duration 变化时间

Sprite Swap 精灵切换过渡

Animation 动画过渡

Button 按钮(续2)

- 属性

Navigation

导航

Horizontal

水平导航

Vertical

垂直导航

Automatic

自动导航

Explicit

显式导航

visualize

可视化--显示导航路径

Input Field 输入框

- 属性

Character Limit 字符数量限制

Content Type 内容类型

Standard 标准

Autocorrected 自动验证

Integer Number 数字

Decimal Number 小数

Alphanumeric 字母数字

Name 姓名

Email Address 邮件地址

Password 密码

Pin 仅输入整数，用*隐藏字符

Custom 自定义

Input Field 输入框(续1)

- 属性

Line Type

行类型

Single Line

单行

Multi Line Submit 多行文本，回车提交

Multi Line Newline 多行文本，回车换行

Caret Blink Rate

光标闪烁速度

Selection Color

选择的字符颜色

Toggle 开关

- 属性

Is On

复选框的选中状态

Toggle Transition 状态改变时,是否启动过渡效果

Graphic 切换的背景图片

Group 单选组



Slider 滑块

- 属性

Fill Rect

填充矩形区域

Handle Rect

手柄矩形区域

Direction

手柄方向

Min / Max Value

最小/最大值

Whole Numbers

整数数值

Value

数值

Scrollbar 滚动条

- 属性

Size

手柄大小

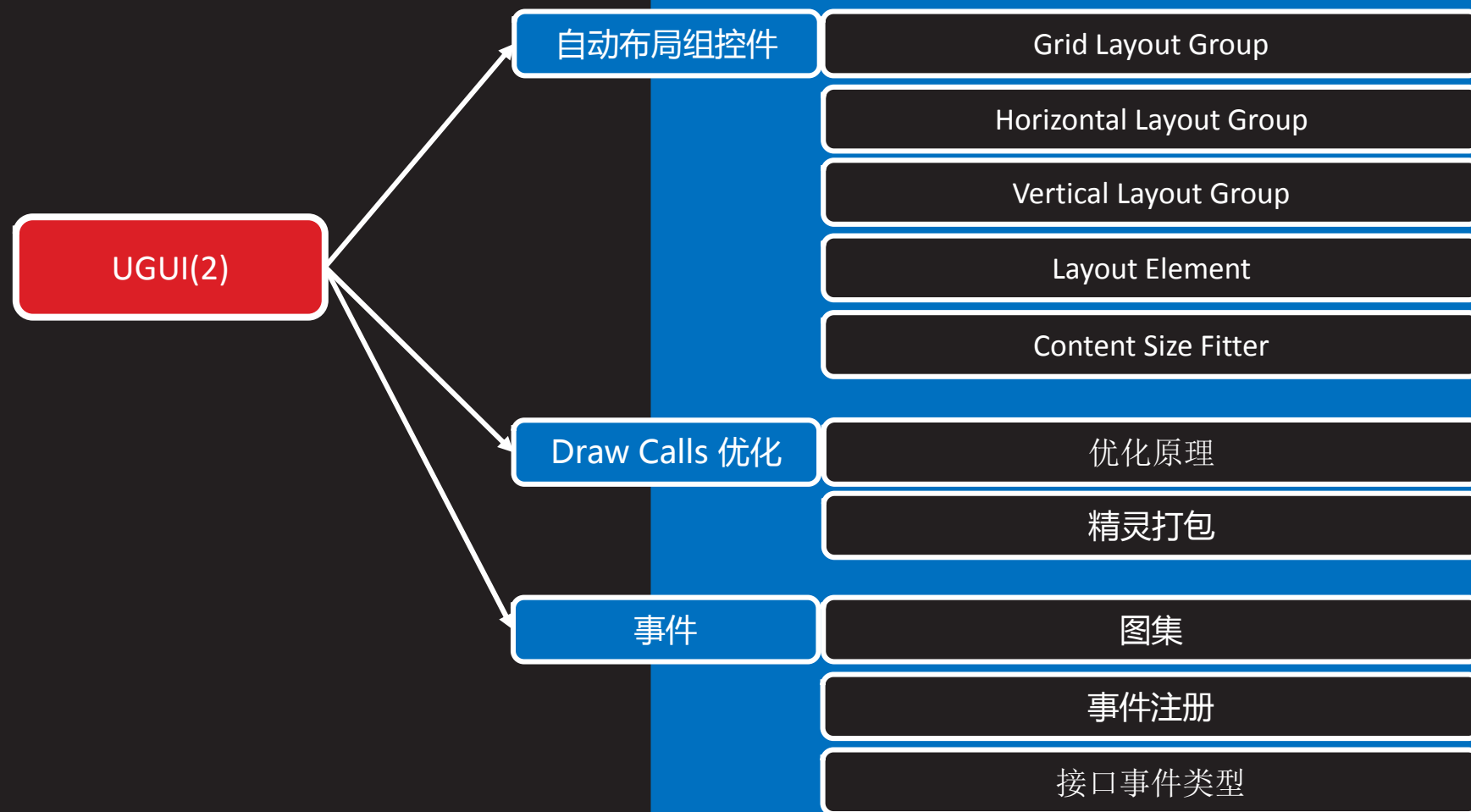
Number of Steps

从开始 滑动 到末尾的步骤

2048游戏---主界面搭建



UGUI



自动布局组控件



Grid Layout Group

- 网格布局组，将子元素以表格形式自动排列。

- 属性

Start Corner 开始角度

Start Axis 开始轴向

Constraint 约束

Flexible 灵活的

Fixed Column Count 固定列数

Fixed Row Count 固定行数

Horizontal Layout Group

- 水平布局组，将子元素按照水平方向自动排列。
- 属性

Padding 填充

Left 左边距

Right 右边距

Top 上边距

Bottom 下边距

Spacing 间隔

Child Alignment 元素的对齐方式

Child Force Expand 元素展开

Vertical Layout Group

- 垂直布局组，将子元素按照垂直方向自动排列。
- 属性

Padding 填充

Left 左边距

Right 右边距

Top 上边距

Bottom 下边距

Spacing 间隔

Child Alignment 元素的对齐方式

Child Force Expand 元素展开

Layout Element

- 布局元素，可以为自动布局组中的子元素指定大小。
- 属性

Ignore Layout

忽略布局

Min Width / Height

最小宽高

Preferred Width / Height

优先宽高(不会超过父物体)

Flexible Width / Height

弹性宽高比例

Content Size Fitter

- 内容大小适配器

根据子元素Layout Element组件自动调整父容器大小。

- 属性

Horizontal Fit	水平适配
Vertical Fit	垂直适配
Unconstrained	无约束
Min Size	最小尺寸
Preferred Size	优先尺寸

2048游戏---4*4方格布局



2048游戏---4*4方格布局

- 脚本控制间距，背景大小决定方格宽高。
- 公式： $4 \times \text{方格} + 5 \times \text{间距} = \text{背景大小}$

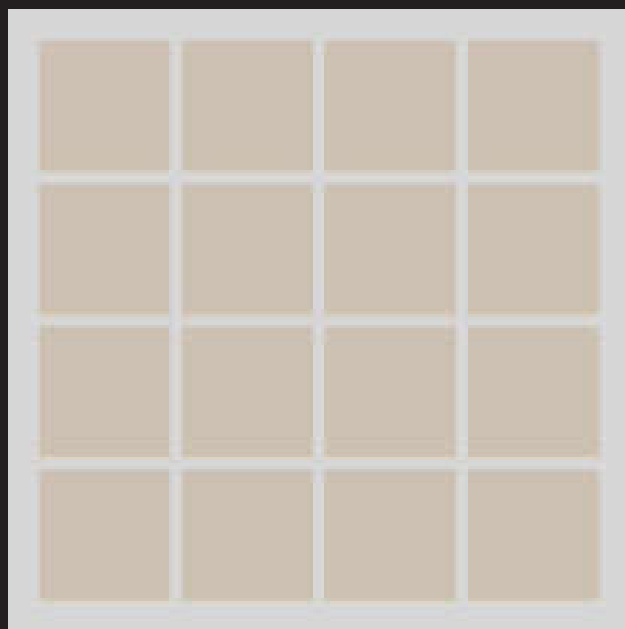


Draw Calls 优化

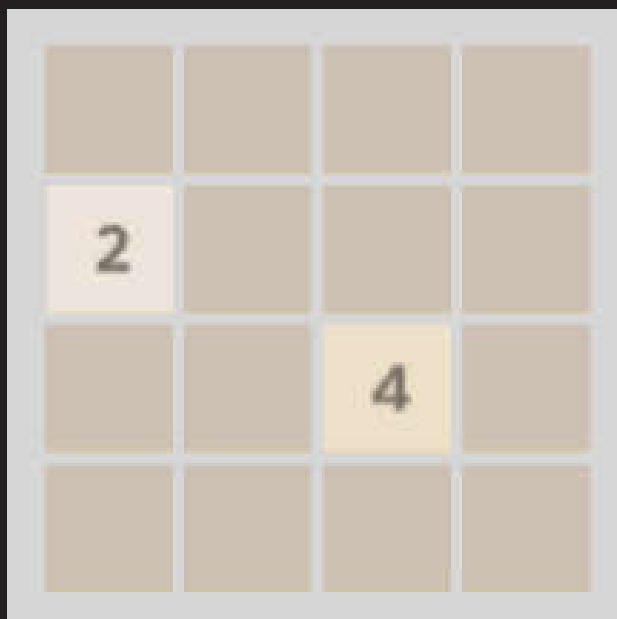


优化原理

- 在界面中默认一张图片一个Draw Calls。
- 同一张图片多次显示仍然为一个Draw Calls。



Draw Calls : 2



Draw Calls : 4

精灵打包

- 做界面时使用小图，而在项目发布时引擎会根据精灵 Packing Tag 自动将小图合并在一张大图集中，从而减少 Draw Calls，减少GPU渲染物体次数。



精灵打包(续1)

- 设置打包方式：

Edit --> Project Settings --> Editor --> Sprite Packer

-- Disabled 表示不启用

-- Enabled For Builds 表示打包时启用

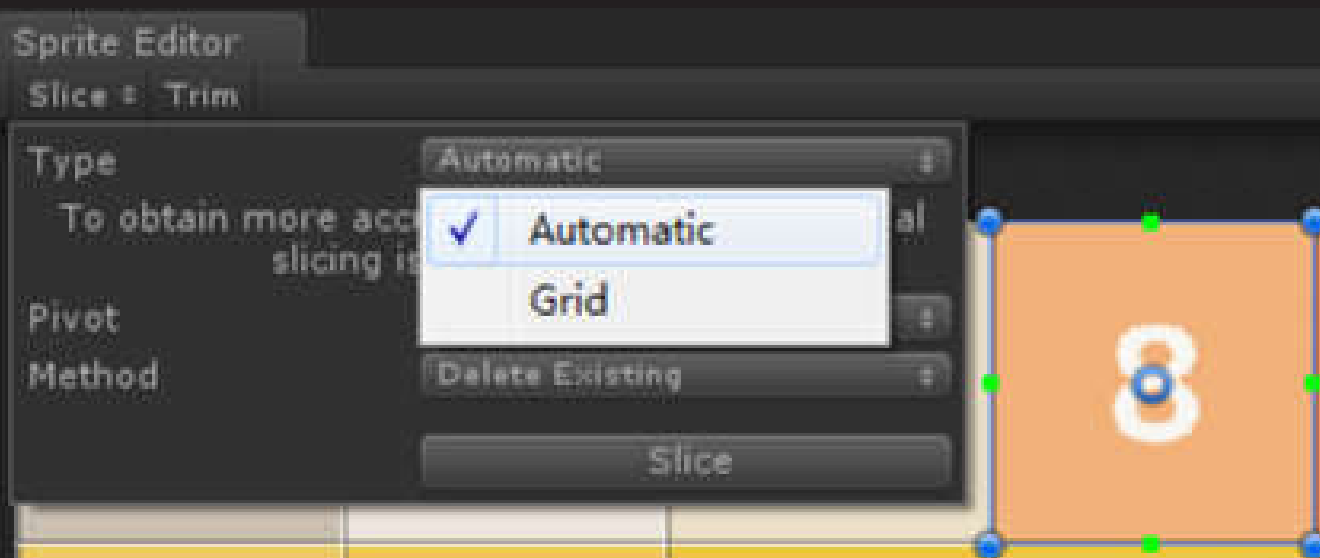
-- Always Enabled 表示永远启用

图集

- 美工制作图片时，尽量将需要在同个界面显示的小图做到一张大图中。
- 在Unity中切割精灵
 - 选中图集，在 Inspector 面板中将Sprite Mode 设置为Multiple。
 - 编辑精灵 Sprite Editor。

图集(续1)

- 三种切割方法：
 - 手动切割；
 - 自动切割；
 - 表格切割；



Event 事件



事件处理流程

- 1. **EventStem** 每帧调用 BaseInputModule 中 Process 方法。
(实现类 : **StandaloneInputModule** / TouchInputModule)
- 2. 计算光标接触的物体 (Graphic) 。
 - Process 方法调用 BaseRaycaster 的 Raycast 方法获取所有 Graphic。
(实现类 : **GraphicRaycaster** / PhysicsRaycaster / Physics2DRaycaster)
 - 通过 **Graphic** 的 IsRaycastLocationValid 方法确定光标选中的 Graphic。
- 3. 通过 ExecuteEvents 引发物体的相关事件。
 - 调用 Execute 方法获取相关接口类型对象 , 再调用其接口方法。

事件注册

- 1.通过编辑器绑定方法
- 2.AddListener
- 3.实现接口
- 4.自定义框架



鼠标指针类

- IPointerEnterHandler
- IPointerExitHandler
- IPointerDownHandler
- IPointerUpHandler
- PointerClickHandler



拖拽类

- IBeginDragHandler
- IDragHandler
- IEndDragHandler
- IDropHandler



点选类

- IUpdateSelectedHandler
- ISelectHandler
- IDeselectHandler



输入类

- IScrollHandler
- IMoveHandler
- ISubmitHandler
- ICancelHandler



2048游戏---获取用户输入的方向

课堂
练习



iTween 动画库



iTween 动画库

- 目的：以最小的投入实现最大的产出，轻松实现各种动画，晃动，旋转，移动以及褪色等等。
- 1.物体移动 MoveTo 2.颜色渐变 ColorTo
- 3.物体淡入淡出 FadeTo
- 4.摄像机淡入淡出 CameraFadeAdd CameraFadeTo
- 5.注视旋转 LookTo 6.物体旋转 RotateTo
- 7.物体缩放 ScaleTo
- 8.晃动 PunchPosition PunchRotation PunchScale
- 9.震动 ShakePosition ShakeRotation ShakeScale



2048游戏---方格特效

课堂
练习

