



唐老狮系列教程

# 透明必备知识点——重要知识回顾

WELCOME  
TO THE  
UNITY  
SPECIALTY COURSE  
STUDY

版权所有：唐老狮 tpandme@163.com



# 唐老狮系列教程-重要知识回顾

## | 主要讲解内容



# 唐老狮系列教程-重要知识回顾

## 主要讲解内容

1. 回顾 渲染标签 — 渲染队列
2. 回顾 渲染状态 — 深度缓冲
3. 回顾 渲染状态 — 深度测试
4. 回顾 渲染状态 — 混合方式



# 唐老狮系列教程-重要知识回顾

## | 渲染队列





# 唐老狮系列教程-重要知识回顾

## 渲染标签 — 渲染队列

主要作用：  
确定物体的渲染顺序

Tags{ "Queue" = "标签值" }

常用Unity预先定义好的渲染队列标签值：

- 1.Background(背景)(队列号:1000) 最早被渲染的物体的队列，一般用来渲染天空盒或者背景
- 2.Geometry(几何)(队列号:2000) 不透明的几何体通常使用该队列，当没有声明渲染队列时，Unity会默认使用这个队列
- 3.AlphaTest(透明测试)(队列号:2450) 有透明通道的，需要进行Alpha测试的几何体会使用该队列
- 4.Transparent(透明的)(队列号:3000) 该队列中几何体按照由远到近的顺序进行绘制，半透明物体的渲染队列，所有进行透明混合的几何体都应该使用该队列
- 5.Overlay(覆盖)(队列号:4000) 用是放在最后渲染的队列，于叠加渲染的效果，比如镜头光晕等

### 6.自定义队列

基于Unity预先定义好的这些渲染队列标签来进行加减运算来定义自己的渲染队列

比如：

Tags{ "Queue" = "Geometry+1" } 代表的队列号就是 2001

Tags{ "Queue" = "Transparent-1" } 代表的队列号就是 2999

自定义队列在一些特殊情况下，特别有用

比如 一些水的渲染 想要在不透明物体之后，半透明物体之前进行渲染，就可以自定义

Tags{ "Queue" = "Background" }

Tags{ "Queue" = "Geometry" }

Tags{ "Queue" = "AlphaTest" }

Tags{ "Queue" = "Transparent" }

Tags{ "Queue" = "Overlay" }



# 唐老狮系列教程-重要知识回顾

## | 深度缓冲



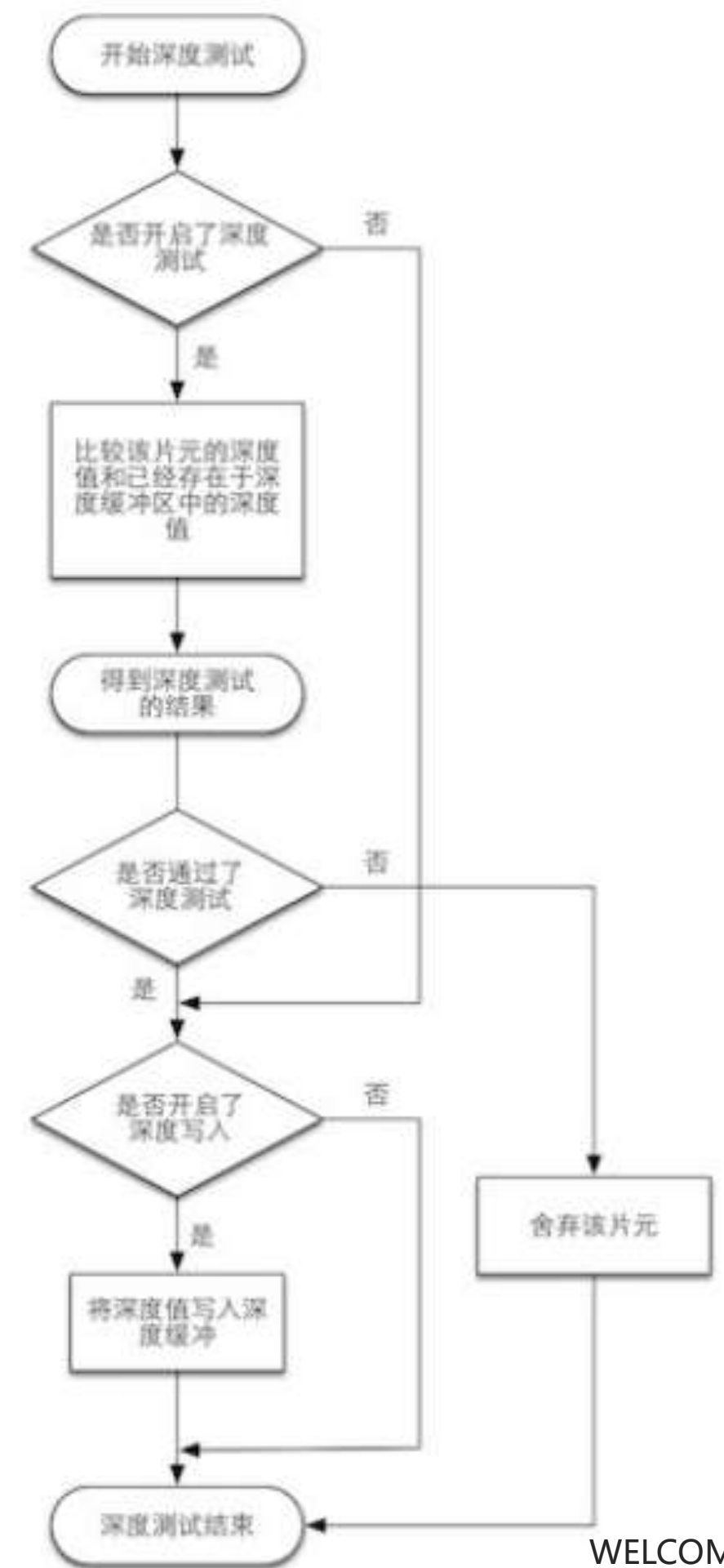
# 唐老狮系列教程-重要知识回顾

## 渲染状态 — 深度缓冲（深度写入）

深度缓冲是一个与屏幕像素对应的缓冲区，用于存储每个像素的深度值（距离相机的距离）  
在渲染场景之前，深度缓冲被初始化为最大深度值，表示所有像素都在相机之外

ZWrite On 写入深度缓冲  
ZWrite Off 不写入深度缓冲  
不设置的话，默认为写入

一般情况下，我们在做透明等特殊效果时，会设置为不写入



WELCOME  
TO THE  
UNITY  
SPECIALTY COURSE  
STUDY





# 唐老狮系列教程-重要知识回顾

## | 深度测试





# 唐老师系列教程-重要知识回顾

## 渲染状态 — 深度测试

深度测试的主要目的是确保在渲染时，像素按照正确的深度（距离相机的距离）顺序进行绘制，从而创建正确的遮挡关系和透视效果

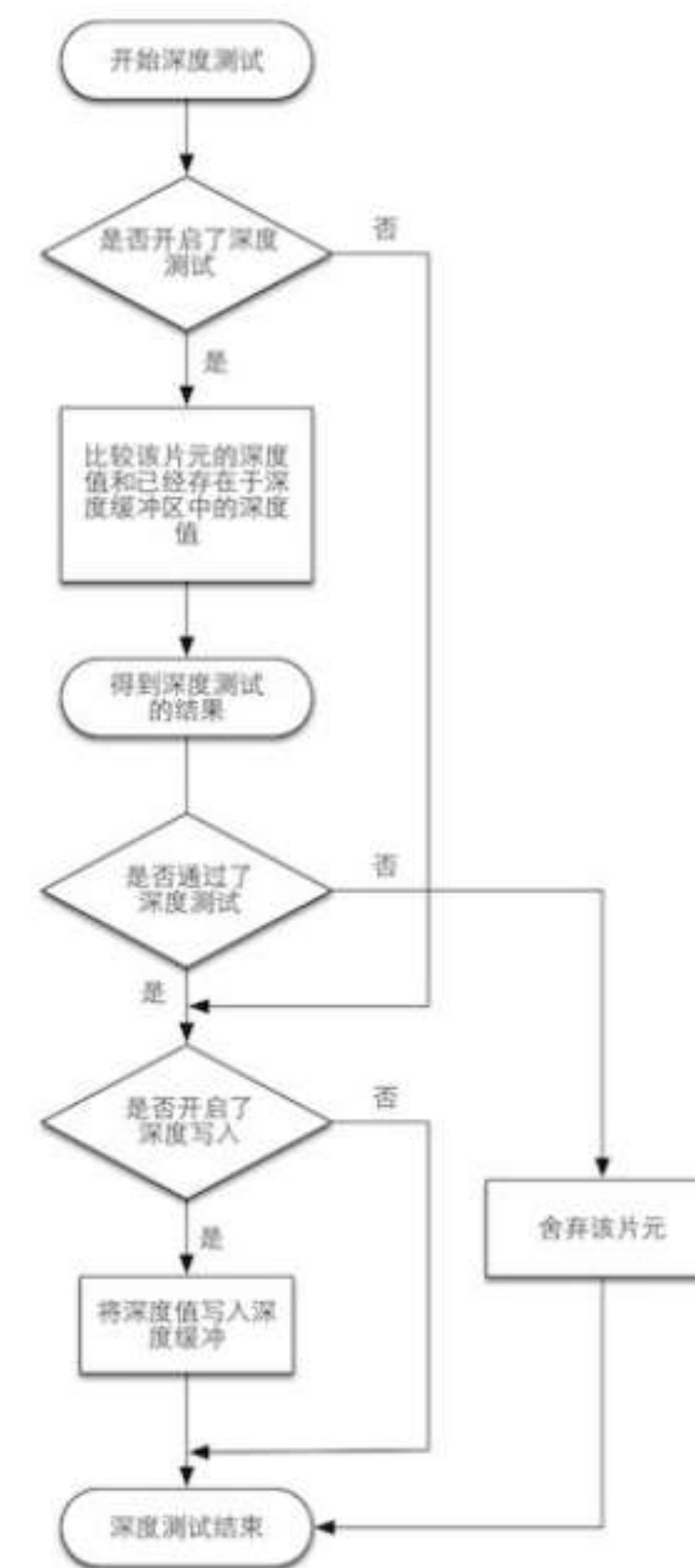
在渲染场景之前，深度缓冲被初始化为最大深度值，表示所有像素都在相机之外。

在渲染过程中，对于每个像素，深度测试会将当前像素的深度值与深度缓冲中对应位置的值进行比较。一般情况下

- 1.如果当前像素的深度值小于深度缓冲中的值，说明当前像素在其他物体之前，它会被绘制，并更新深度缓冲。
- 2.如果当前像素的深度值大于等于深度缓冲中的值，说明当前像素在其他物体之后，它会被丢弃，不会被绘制，并保持深度缓冲不变。

ZTest Less	小于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest Greater	大于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest LEqual	小于等于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest GEqual	大于等于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest Equal	等于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest NotEqual	不等于当前深度缓冲中的值，就通过测试，写入到深度缓冲中
ZTest Always	始终通过深度测试写入深度缓冲中

不设置的话，默认为LEqual 小于等于





# 唐老狮系列教程-重要知识回顾

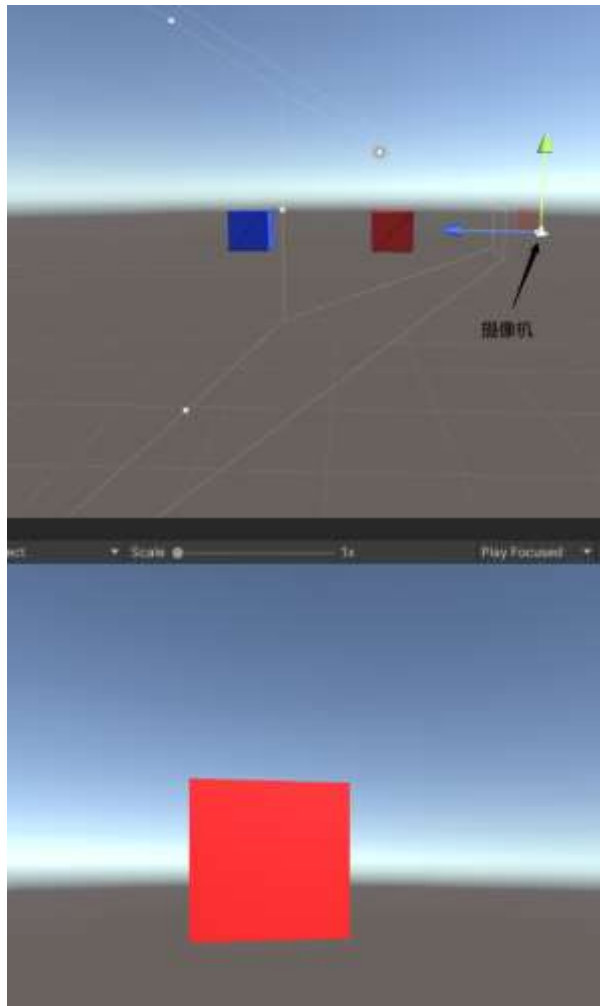
## | 举例说明





# 唐老狮系列教程-重要知识回顾

## 举例说明



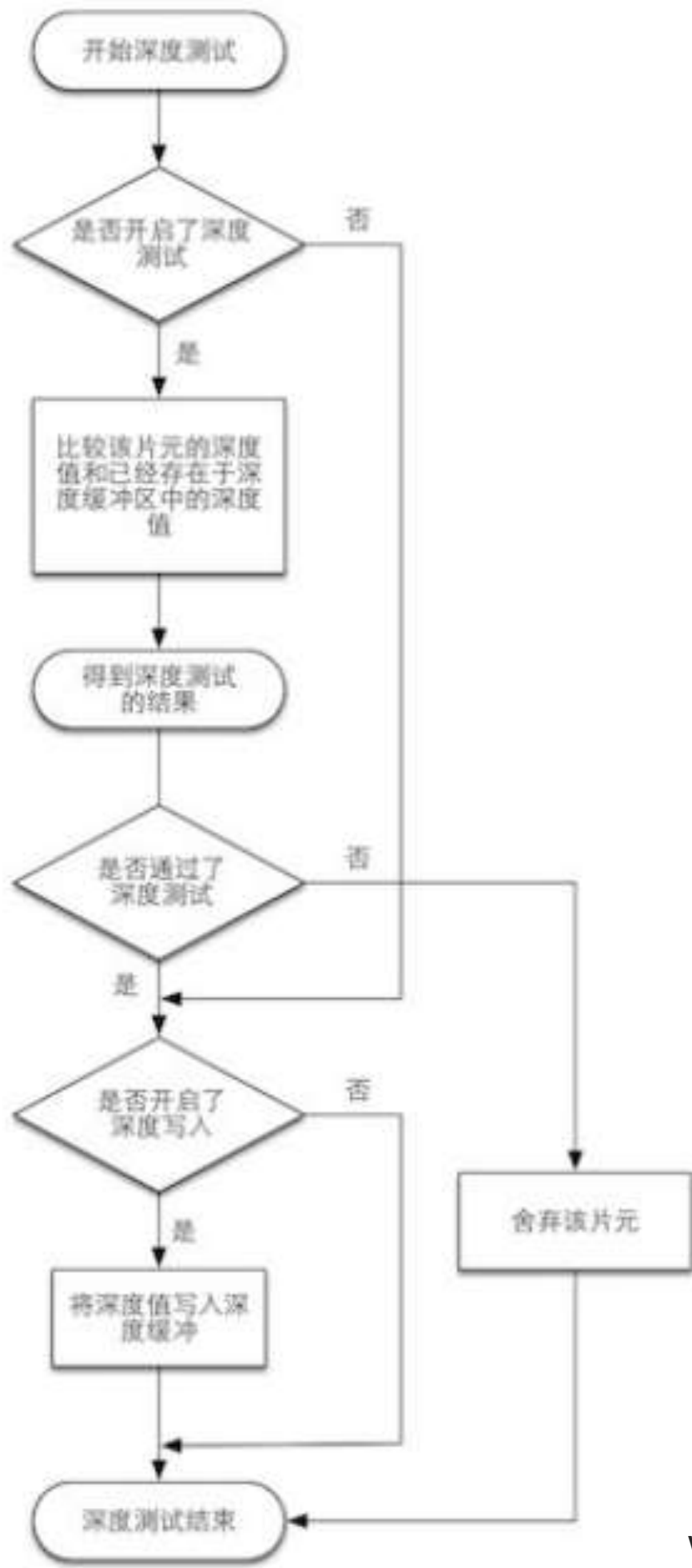
物体	渲染队列	深度
远处的天空盒	1000	认为无限大
蓝色物体	3200	假设为150
红色物体	2800	假设为100

通过渲染队列来看，渲染顺序：天空盒 -> 红色物体 -> 绿色物体

渲染机制：默认的深度比较都是小于比较

	深度缓冲区	颜色缓冲区	说明
天空盒深度<缓冲区默认值无限大	无限大	天空盒颜色	深度测试通过，将天空盒深度写入到深度缓冲区，将天空盒颜色写入到颜色缓冲区中
红色物体的深度（100）<无限大	100	红色	深度测试通过，将红色物体的深度写入到深度缓冲区中，将红色写入到颜色缓冲区中
蓝色物体的深度（150）>100	100	红色	深度测试不通过，深度不写入，颜色也不写入

最终颜色缓冲区中的颜色是红色，所以屏幕上中心像素显示的是红色

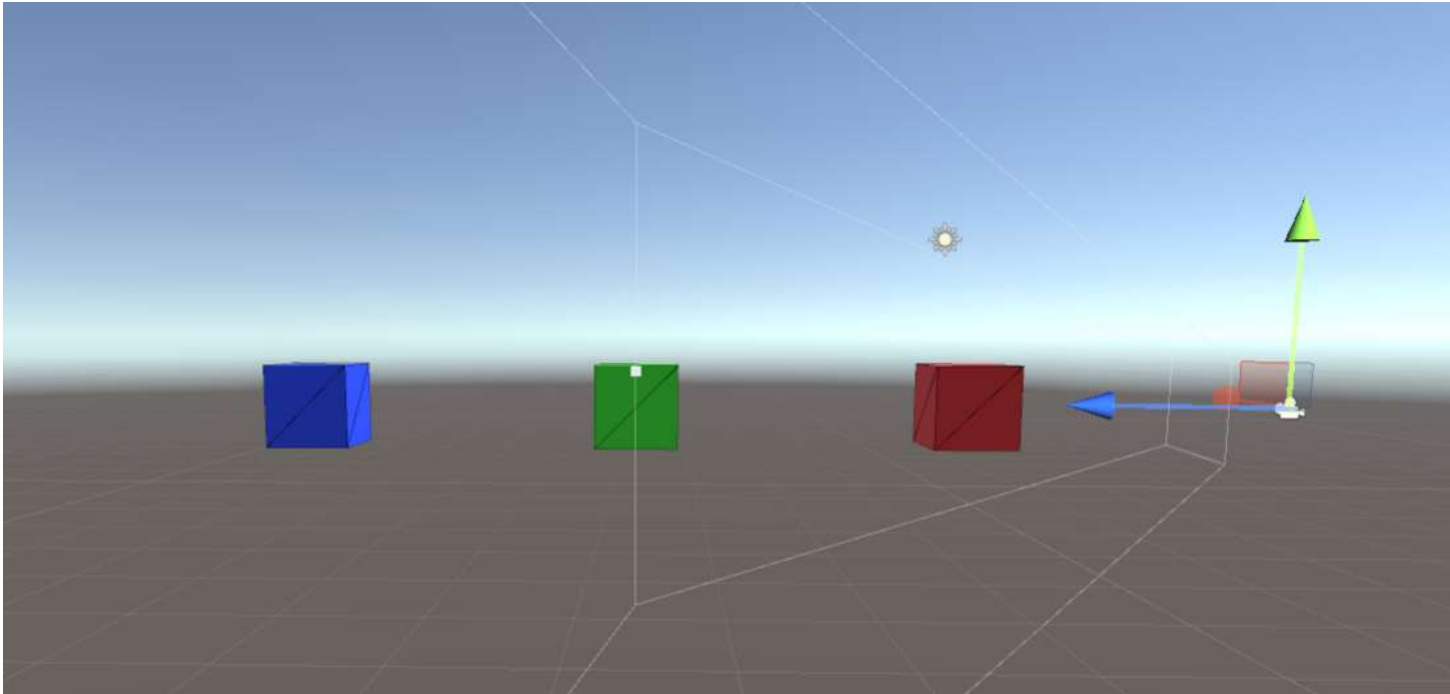






# 唐老狮系列教程-重要知识回顾

## 举例说明

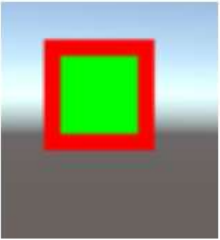


物体	渲染队列	深度	测试方式	深度写入
天空盒	1000	无限大		
蓝色物体	3000	假设为200	>	关闭
绿色物体	3200	假设为150	>	开启
红色物体	2800	假设为100	Always	开启

根据渲染队列值，比较的顺序（值小的先渲染）：  
天空盒（1000）->红色（2800）->蓝色（3000）->绿色（3200）

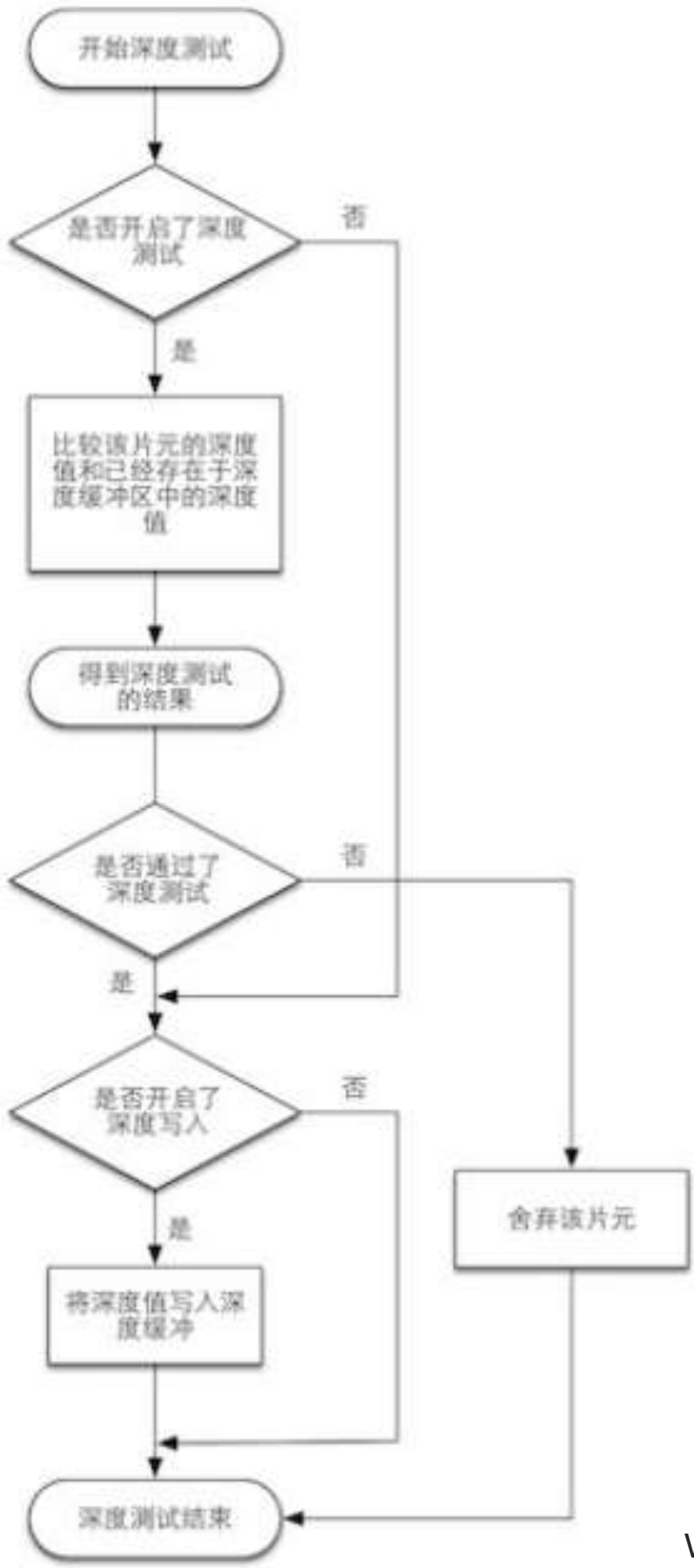
比较	比较结果	深度缓冲区	颜色缓冲区	说明
天空盒	通过	无限大	天空盒颜色	
红色 Always	通过	100	红色	将红色的深度写入了深度缓冲区中，红色写入颜色缓冲区中
蓝色 > 100	通过	100	蓝色	深度写入关闭的，蓝色深度不写入，蓝色写入颜色缓冲区中
绿色 > 100	通过	150	绿色	绿色深度写入深度缓冲区，绿色写入颜色缓冲区

因此最终在Game窗口中央显示的颜色应该是绿色，如果是透视摄像机，可能会看到绿色边缘有红色



ZTest Greater      大于当前深度缓冲中的值，就通过测试，写入到深度缓冲中

ZTest Always      始终通过深度测试写入深度缓冲中





# 唐老狮系列教程-重要知识回顾

## | 混合方式





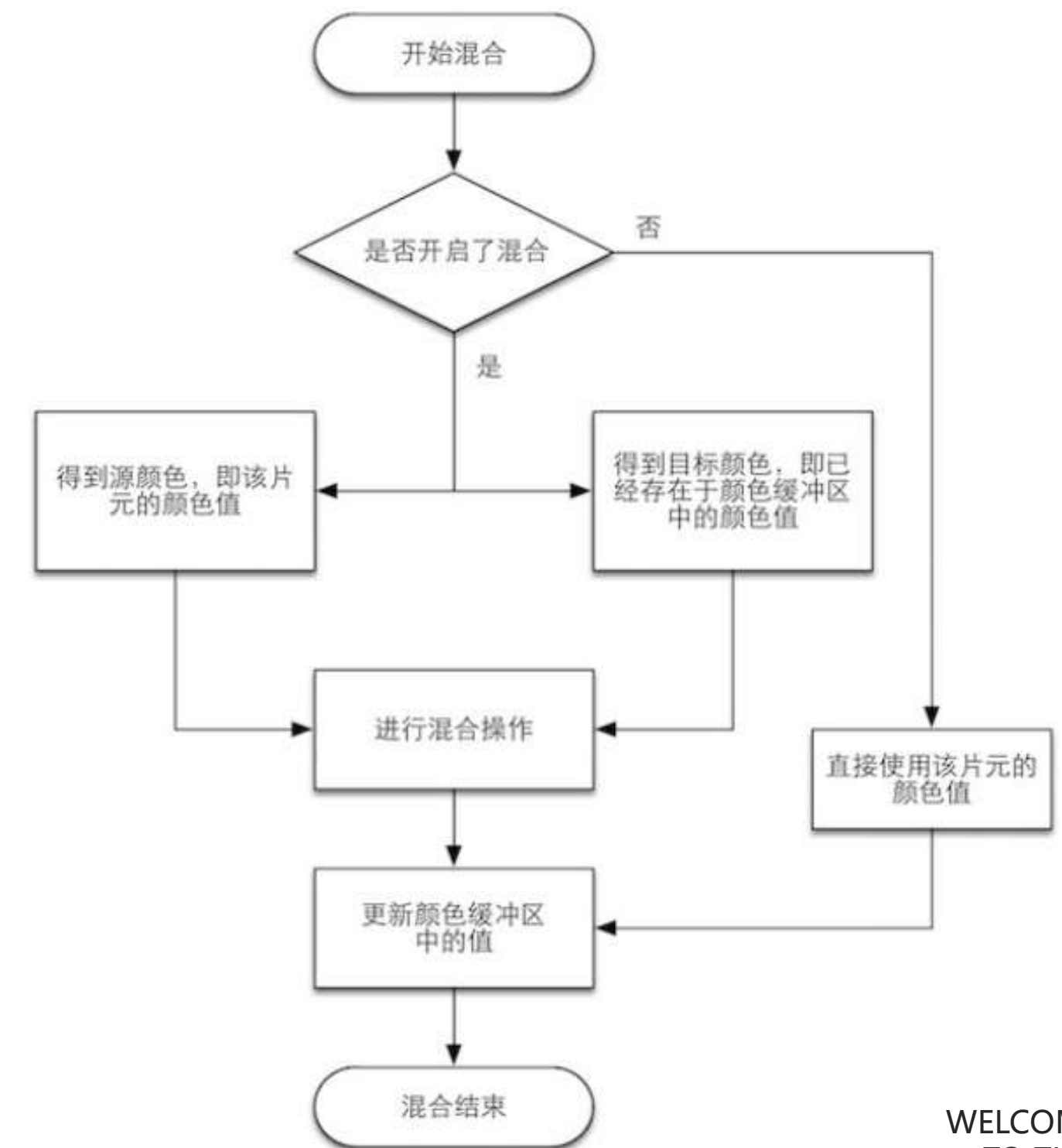
# 唐老狮系列教程-重要知识回顾

## 渲染状态 — 混合方式

设置渲染图像的混合方式（多种颜色叠加混合，比如透明、半透明效果和遮挡的物体进行颜色混合）

Blend One One	线性减淡
Blend SrcAlpha OneMinusSrcAlpha	正常透明混合
Blend OneMinusDstColor One	滤色
Blend DstColor Zero	正片叠底
Blend DstColor SrcColor	X光片效果
Blend One OneMinusSrcAlpha	透明度混合
等等	
不设置的话，默认不会进行混合	

一般情况下，我们需要多种颜色叠加渲染时，就需要设置混合方式，具体情况具体处理







# 唐老狮系列教程-重要知识回顾

## | 总结



# 唐老狮系列教程-重要知识回顾

## 为什么要回顾

在Unity Shader中实现半透明效果的基本原理就是  
利用多个物体或背景的颜色进行混合计算呈现出类似半透明效果  
这节课回顾的这些知识点都会影响我们达到这个最终目的  
因此我们对他们进行了巩固



# 唐老狮系列教程-重要知识回顾

## 主要讲解内容

### 1. 回顾 渲染标签 — 渲染队列

确定物体的渲染顺序

### 2. 回顾 渲染状态 — 深度缓冲

记录像素的深度值，用于之后进行比较，决定片元(颜色)的去留

### 3. 回顾 渲染状态 — 深度测试

将当前片元深度值和深度缓冲中深度值进行比较，决定片元(颜色)去留

### 4. 回顾 渲染状态 — 混合方式

将通过深度测试的片元颜色和颜色缓冲区中的颜色按指定算法进行混合

并更新颜色缓冲区





# 唐老狮系列教程

Thank  
感谢您的聆听