



唐老狮系列教程

# 透明必备知识点——渲染顺序的重要性



# 唐老狮系列教程-渲染顺序的重要性

## | 主要讲解内容



# 唐老狮系列教程-渲染顺序的重要性

## 主要讲解内容

1. 深度测试和深度写入带来的好处
2. 透明混合为什么需要关闭深度写入
3. 关闭深度写入带来的问题
4. 如何解决渲染顺序可能带来的问题



# 唐老狮系列教程-渲染顺序的重要性

## 深度测试和深度写入带来的好处





# 唐老狮系列教程-渲染顺序的重要性

## 深度测试和深度写入带来的好处

有了深度测试和深度写入发挥作用  
让我们不需要关心不透明物体的渲染顺序

比如

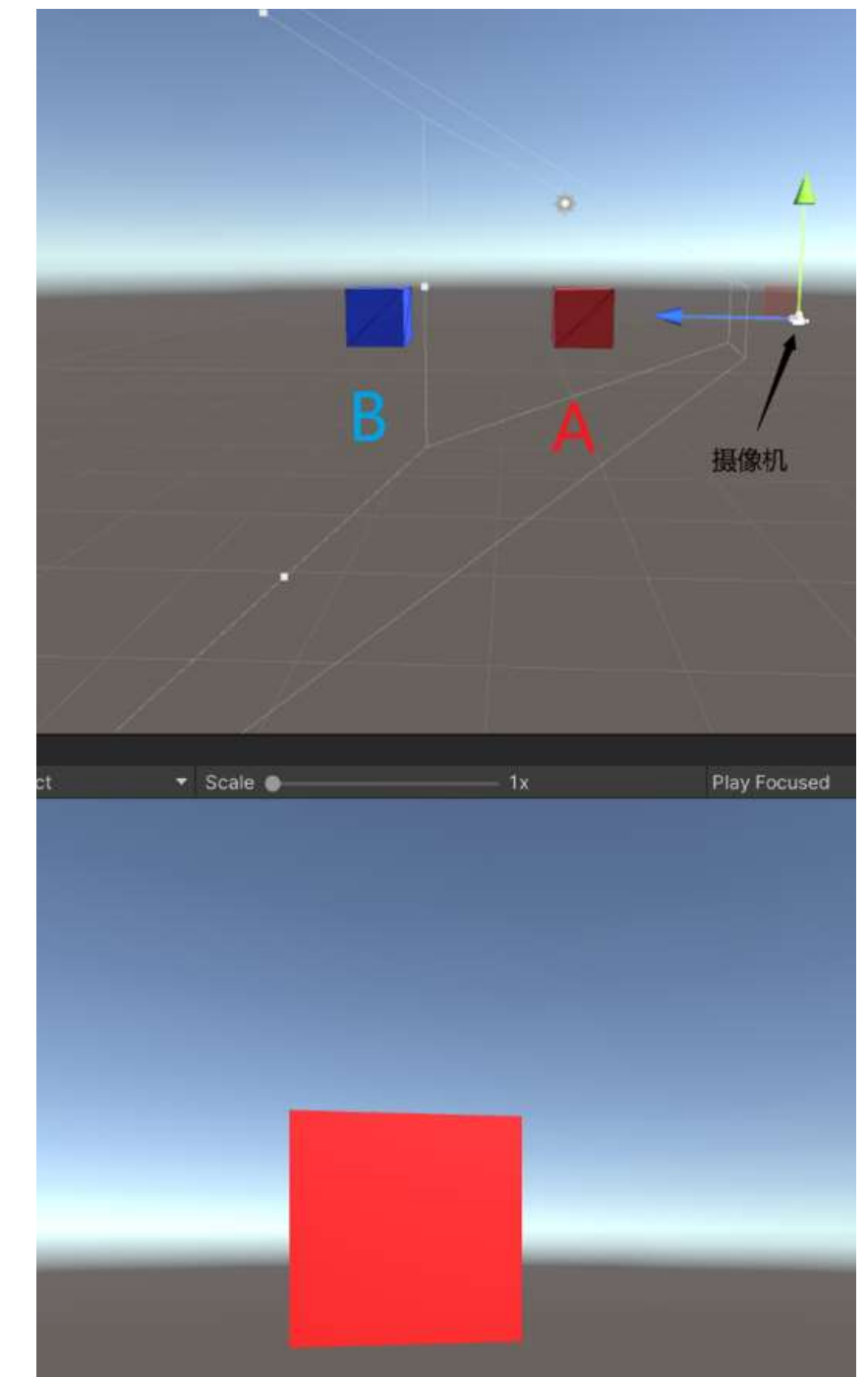
一个物体A 挡住了 物体B

即使底层逻辑中 先渲染A，后渲染B，我们也不用担心 B的颜色会把A覆盖

因为在进行深度测试时，远处的B的深度值无法通过深度测试

因为它的深度会比已经写入深度缓冲中A的深度值大

重合处的片元会被丢弃，颜色自然就不会写入，最终重叠处渲染出来的会是A的颜色





# 唐老狮系列教程-渲染顺序的重要性

## 深度测试和深度写入带来的好处

我们之前写的所有Shader都没有刻意的去设置  
深度测试(默认小于判断)、深度写入(默认开启)、  
混合模式(默认不混合)、渲染队列(默认几何Geometry队列) 相关内容  
是因为对于不透明的物体来说,使用默认设置就能够得到正确的渲染效果  
但目前我们将要学习的透明相关知识,就需要对他们进行改变  
其中,最最最重要的改变就是:

**处理透明混合时,需要关闭深度写入**



# 唐老狮系列教程-渲染顺序的重要性

## | 透明混合为什么需要关闭深度写入





# 唐老狮系列教程-渲染顺序的重要性

## 透明混合为什么需要关闭深度写入

在图形学中模拟出 现实世界的半透明效果是通过将多个颜色进行混合计算呈现出来的

举例：

对于右图来说

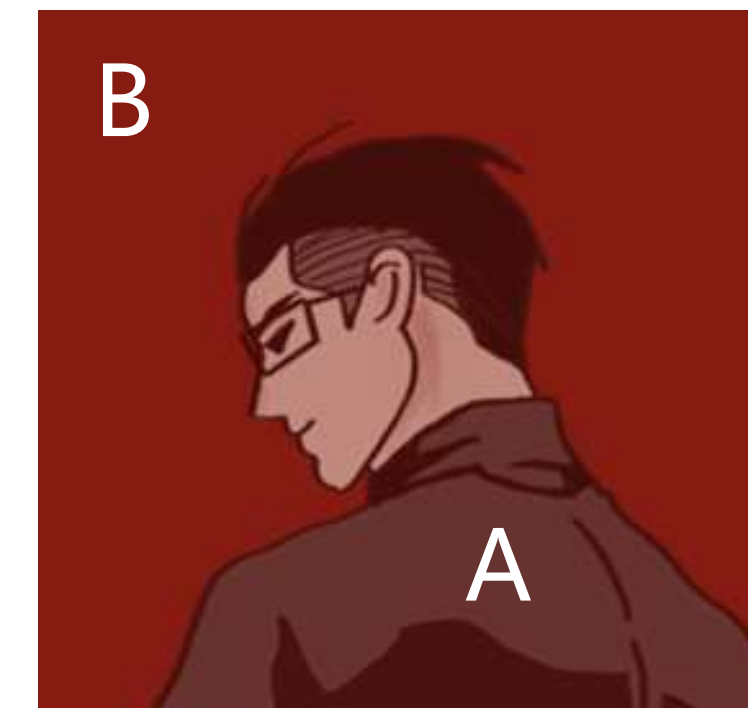
如果前方的A对象不关闭深度写入

那么B被A遮挡的部分会在深度测试时被舍弃（因为A离摄像机更近深度更小）

那么我们就无法得到被遮挡部分的颜色，也就无法进行颜色的混合计算

因此我们必须关闭透明物体的深度写入

让其不会导致被遮挡物体无法通过深度测试从而无法进行颜色混合







# 唐老狮系列教程-渲染顺序的重要性

## | 关闭深度写入带来的问题



# 唐老狮系列教程-渲染顺序的重要性

## 关闭深度写入带来的问题

若关闭深度写入，那么物体的渲染顺序就会变得非常的重要

谁先被渲染，谁后被渲染都会影响最终的呈现效果

我们举例说明

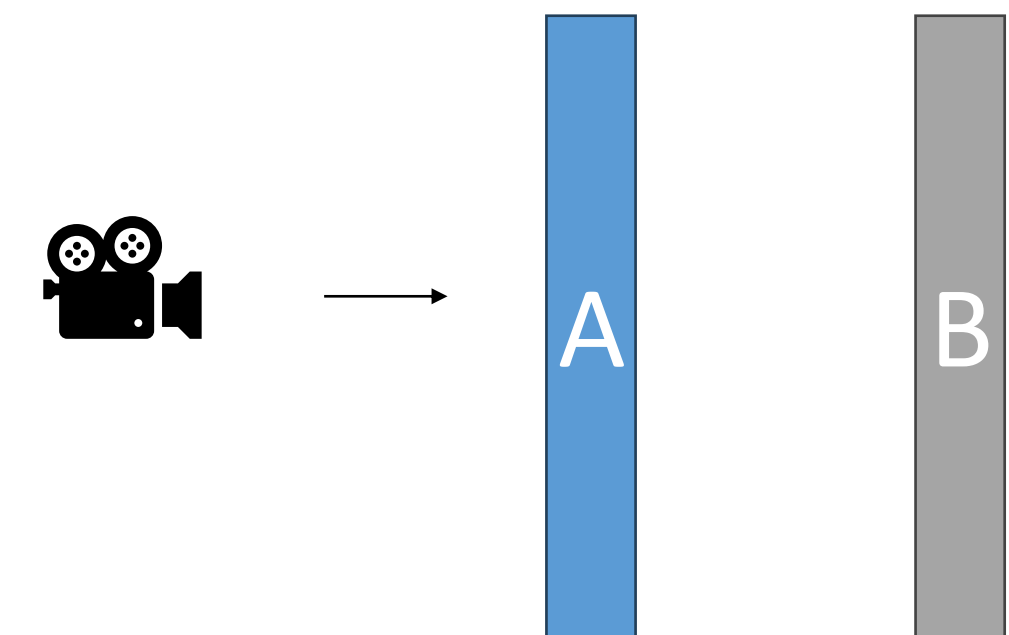
事例一

两个物体 A为半透明物体 B为不透明物体

两种可能的渲染顺序

**第一种：先渲染B，再渲染A，可以得到正确的半透明效果**

**第二种：先渲染A，再渲染B，只会看到B的颜色**



从这个例子我们看出，想要得到正确的半透明混合效果

需要先渲染不透明物体，再渲染透明物体



# 唐老狮系列教程-渲染顺序的重要性

## 关闭深度写入带来的问题

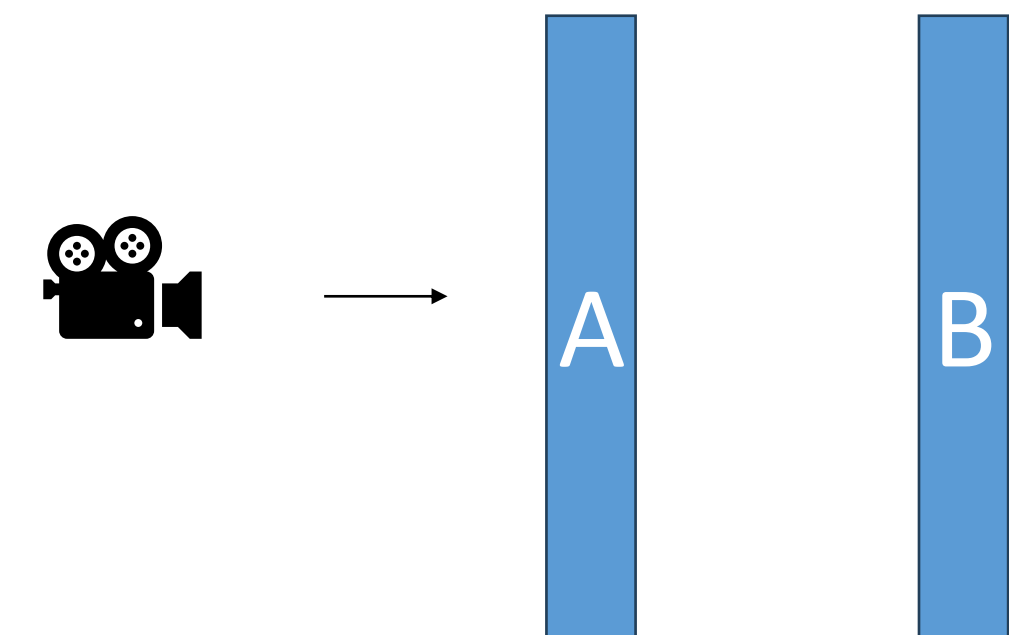
事例二

两个物体 A B 都为半透明物体

两种可能的渲染顺序

**第一种：先渲染B，再渲染A，得到正确的半透明效果**

**第二种：先渲染A，再渲染B，得到错误的半透明效果（B在前）**



从这个例子我们看出，想要得到正确的半透明混合效果

半透明物体之间也需要正确的渲染顺序才能得到正确结果





# 唐老狮系列教程-渲染顺序的重要性

## 关闭深度写入带来的问题

通过这两个事例我们知道

当关闭了深度写入后带来的最大问题是

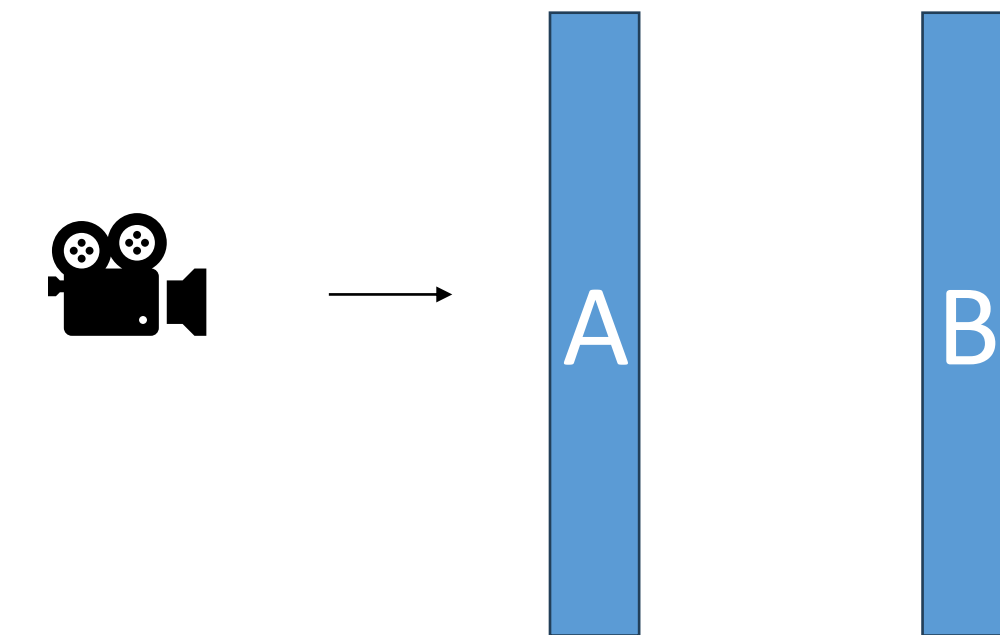
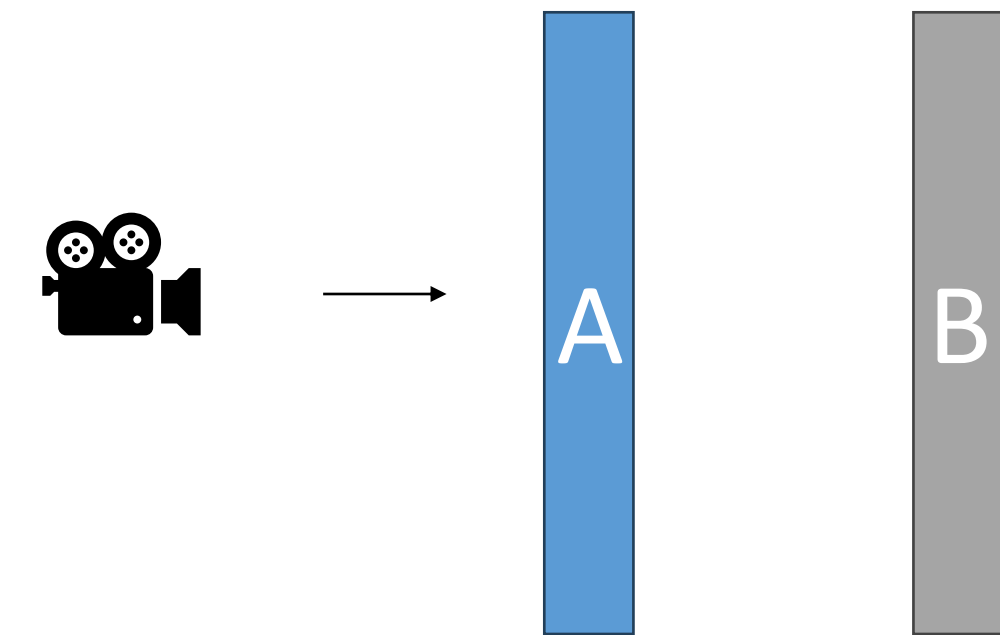
**物体的渲染顺序会直接影响最终的渲染效果**

如果先渲染没有开启深度写入的前面的物体

之后渲染后面的物体就会将前面物体的颜色覆盖

或出现错误前后关系表现

那么接下来我们就来看，应该如何来解决该问题





# 唐老狮系列教程-渲染顺序的重要性

## 如何解决渲染顺序带来的问题



# 唐老狮系列教程-渲染顺序的重要性

## 如何解决渲染顺序带来的问题

为了解决渲染顺序带来的问题

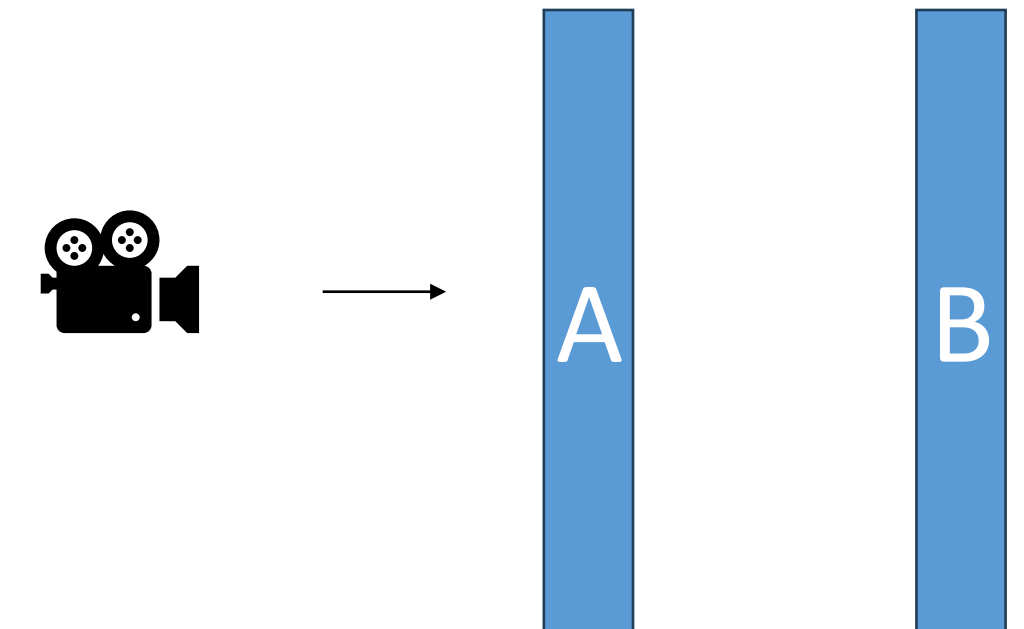
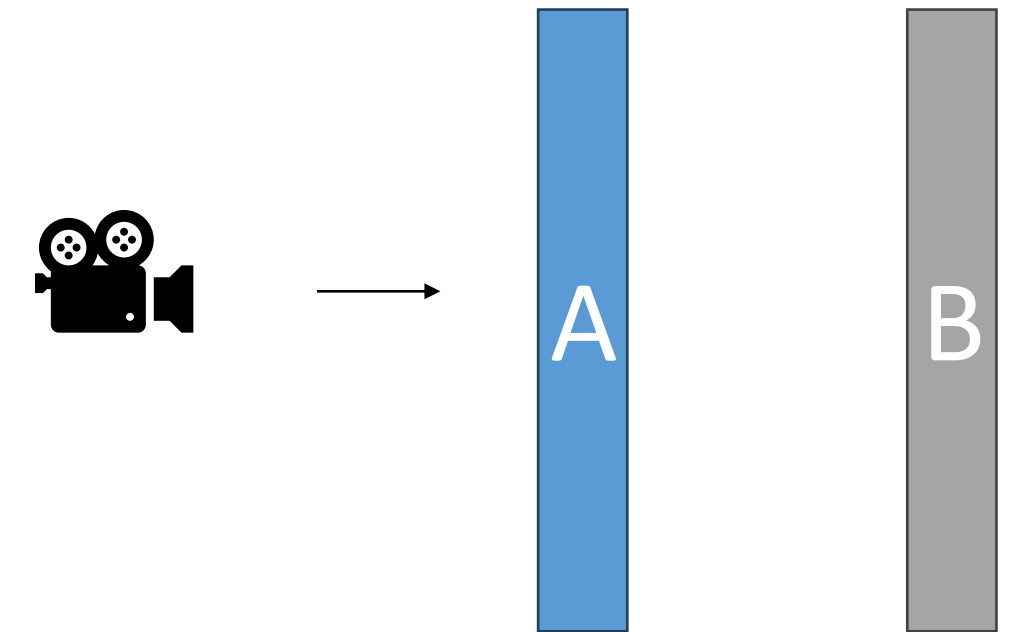
大部分的渲染引擎采取的方式都是先对场景上的物体进行排序，然后再渲染  
常规方法为：

### 1.先渲染所有不透明物体

开启它们的深度测试和深度写入

### 2.将半透明物体按照离摄像机的远近进行排序，然后从后往前的顺序渲染它们

开启深度测试关闭深度写入



这样做的目的是保证靠后的物体颜色能先直接进入颜色缓冲区或混合后进入颜色缓冲区

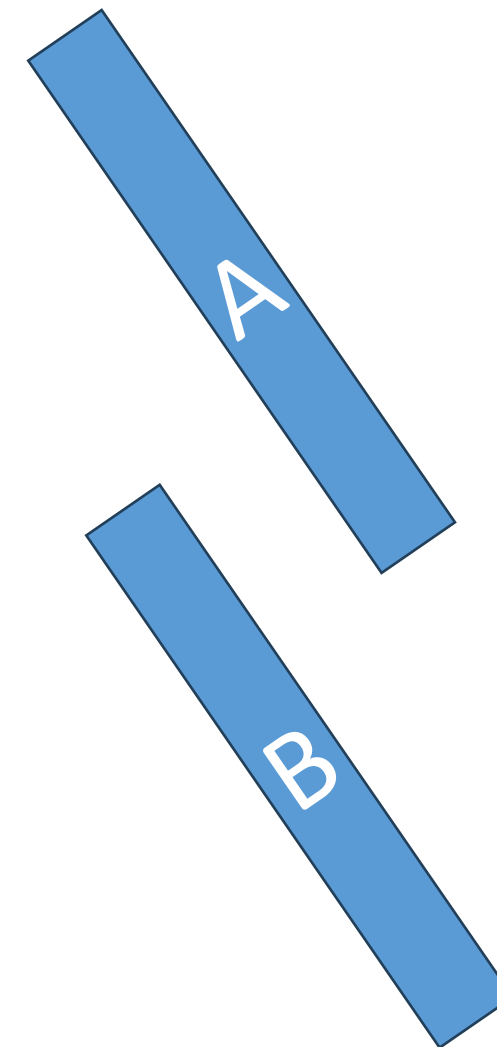
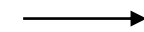
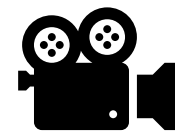
就不会出现之前两个事例当中由于先渲染前面的物体导致结果出现问题





# 唐老狮系列教程-渲染顺序的重要性

## 特殊情况



常规方法：

**1.先渲染所有不透明物体**

开启它们的深度测试和深度写入

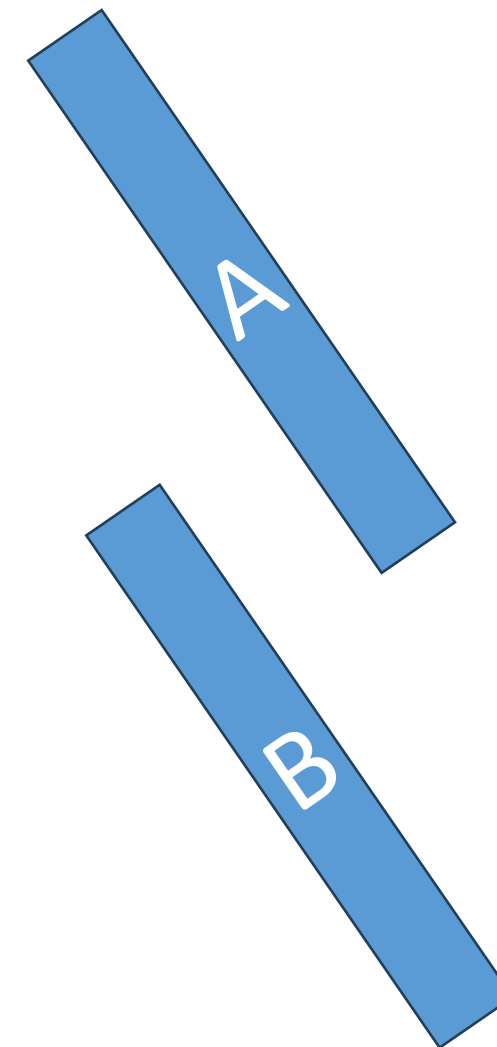
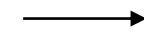
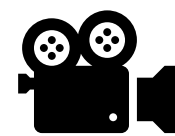
**2.将半透明物体按照离摄像机的远近进行排序，然后从后往前的顺序渲染它们**

开启深度测试关闭深度写入 **(排序是基于物体的，而不是片元的)**



# 唐老狮系列教程-渲染顺序的重要性

## 解决方案



- 1.忍一忍，接受部分细节的小瑕疵
- 2.从模型入手：拆分模型
- 3.使用开启深度写入的半透明效果来模拟半透明（性能消耗较高）



# 唐老狮系列教程-渲染顺序的重要性

## 如何解决渲染顺序带来的问题

引擎底层

### 1.先渲染所有不透明物体

开启它们的深度测试和深度写入

### 2.将半透明物体按照离摄像机的远近进行排序，然后从后往前的顺序渲染它们

开启深度测试关闭深度写入

遇到特殊情况时

### 1.忍一忍，接受部分细节的小瑕疵

### 2.从模型入手：拆分模型

### 3.使用开启深度写入的半透明效果来模拟半透明（性能消耗较高）





# 唐老狮系列教程-渲染顺序的重要性

## | 总结



# 唐老狮系列教程-渲染顺序的重要性

## 主要讲解内容

### 1. 深度测试和深度写入带来的好处

无需关心不透明物体的渲染顺序

### 2. 透明混合为什么需要关闭深度写入

能够得到背后物体的颜色进行颜色混合计算

### 3. 关闭深度写入带来的问题

物体的渲染顺序会影响最终的渲染效果

### 4. 如何解决渲染顺序可能带来的问题

引擎底层指定了通用的渲染顺序规则，特殊情况时可以根据我们的需求选择性的进行处理



# 唐老狮系列教程

Thank  
感谢您的聆听