



唐老狮系列教程

# 渲染管线——应用阶段

WELCOME  
TO THE  
UNITY  
SPECIALTY COURSE  
STUDY

版权所有：唐老狮 tpandme@163.com



# 唐老狮系列教程-渲染管线 应用阶段

## 知识回顾



# 唐老狮系列教程-渲染管线 应用阶段

## 知识回顾

渲染管线（流水线）就是将数据分阶段的变为屏幕图像的过程

其中

数据就是我们在游戏场景中放置的模型、光源、摄像机等等内容的数据

阶段就是渲染管线中的三个阶段

应用阶段——>几何阶段——>光栅化阶段

通过这三个阶段对数据的处理，最终我们就能够在屏幕上看见图像



# 唐老狮系列教程-渲染管线 应用阶段

## 知识必备





# 唐老狮系列教程-渲染管线 应用阶段

## 知识必备——CPU和GPU

CPU：中央处理器，负责算数运算、逻辑操作、数据传输等通用计算任务，同时还管理和调度计算机的资源（游戏中—游戏逻辑处理）

GPU：图形处理器，是专门用于图形和并行计算的处理器

显卡就是搭载GPU的硬件设备，显卡包含一个或多个GPU芯片，还包含显存（用于存储图像数据）、显示接口、视频解码器等等（游戏中—渲染相关处理）

CPU主要处理操作系统管理、程序执行、通用计算等等

GPU主要处理图形渲染、图像处理等等



# 唐老狮系列教程-渲染管线 应用阶段

## 应用阶段主要做什么



# 唐老狮系列教程-渲染管线 应用阶段

## 渲染管线 在应用阶段主要做什么



渲染管线的应用阶段中大部分的内容都和渲染无关（比如：游戏逻辑处理、动画更新、物理模拟、场景管理等等）  
当应用阶段完成后，后面的几何阶段以及光栅化阶段将开始处理和图形渲染相关的数据和操作。

那么应用阶段为什么会归纳到渲染管线中呢？

那是因为应用阶段为渲染管线的后续提供了最重要的内容——**数据**

**应用阶段主导者是CPU**，在这一阶段，我们将**渲染需要用到的数据**传递给GPU用于后续的两个阶段的处理





# 唐老狮系列教程-渲染管线 应用阶段

## 应用阶段为渲染准备了些什么





# 唐老狮系列教程-渲染管线 应用阶段

## 应用阶段为渲染准备了些什么？

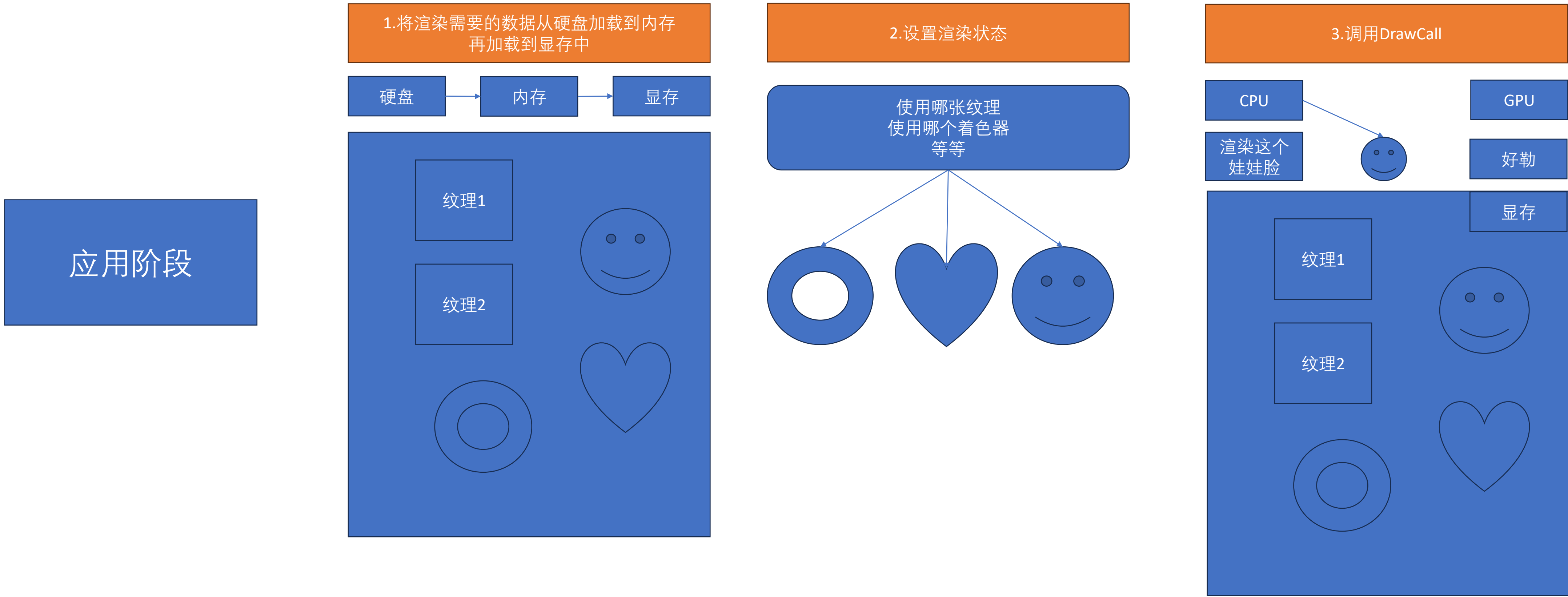


- 1.把不可见的物体数据剔除
- 2.准备好模型相关数据（顶点、法线、切线、贴图、着色器等等）
- 3.将数据加载到显存中
- 4.设置渲染状态（设置网格需要使用那个着色器、材质、光源属性等等）
- 5.调用DrawCall（CPU通知GPU使用相关的数据和渲染状态进行渲染）



# 唐老狮系列教程-渲染管线 应用阶段

## 应用阶段为渲染准备了些什么？





# 唐老狮系列教程-渲染管线 应用阶段

## 应用阶段为渲染准备了些什么？

渲染管线（流水线）中的**应用阶段**

主要是**CPU主导的阶段**

**它为渲染完成的最主要的工作就是提供后续的渲染数据**

比如：

顶点、法线、切线、纹理坐标、变换矩阵、材质属性等等

在应用阶段中，我们主要就是按照Unity的规则进行游戏开发即可

我们需要注意的就是关于DrawCall的优化





# 唐老狮系列教程-渲染管线 应用阶段

## | DrawCall



# 唐老狮系列教程-渲染管线 应用阶段

## 关于DrawCall



我们之前在讲解UI相关知识点时，提到了DrawCall，也知道它会影响我们游戏的性能

一次DrawCall其实就是CPU命令GPU进行渲染的命令

为什么DrawCall多了会影响性能呢？

主要的性能的瓶颈是CPU造成的

每次调用DrawCall之前，CPU需要向GPU发送很多内容，包括数据、状态、命令等等。

如果DrawCall过多，CPU就会把大量的时间花费在提交DrawCall上，造成CPU过载，让玩家感受到卡顿



# 唐老狮系列教程-渲染管线 应用阶段

## 如何减少DrawCall



使用批处理，可以有效的减少DrawCall，从而提升性能表现

- 1.合并网格（可以将静态物体合并网格）
  - 2.共用材质（在不同网格之间共用一种材质）
  - 3.合并图集（2D游戏和UI中，可以将多张图片合并为一张大图）
- 等等





# 唐老狮系列教程-渲染管线 应用阶段

## | 总结



# 唐老狮系列教程-渲染管线 应用阶段

## 总结

渲染管线（流水线）中的**应用阶段**

主要是**CPU主导的阶段**

**它为渲染完成的最主要的工作就是提供后续的渲染数据**

比如：

顶点、法线、切线、纹理坐标、变换矩阵、材质属性等等

在应用阶段中，我们主要就是按照Unity的规则进行游戏开发即可

DrawCall过多时，性能瓶颈是由CPU造成的，我们可以用批处理技术优化它



# 唐老狮系列教程

Thank

谢谢您的聆听