
# Unity基础

欢乐互娱
www.huanle.com
欢乐从这里开始

# 游戏是如何运作起来的

对比普通应用的不同点

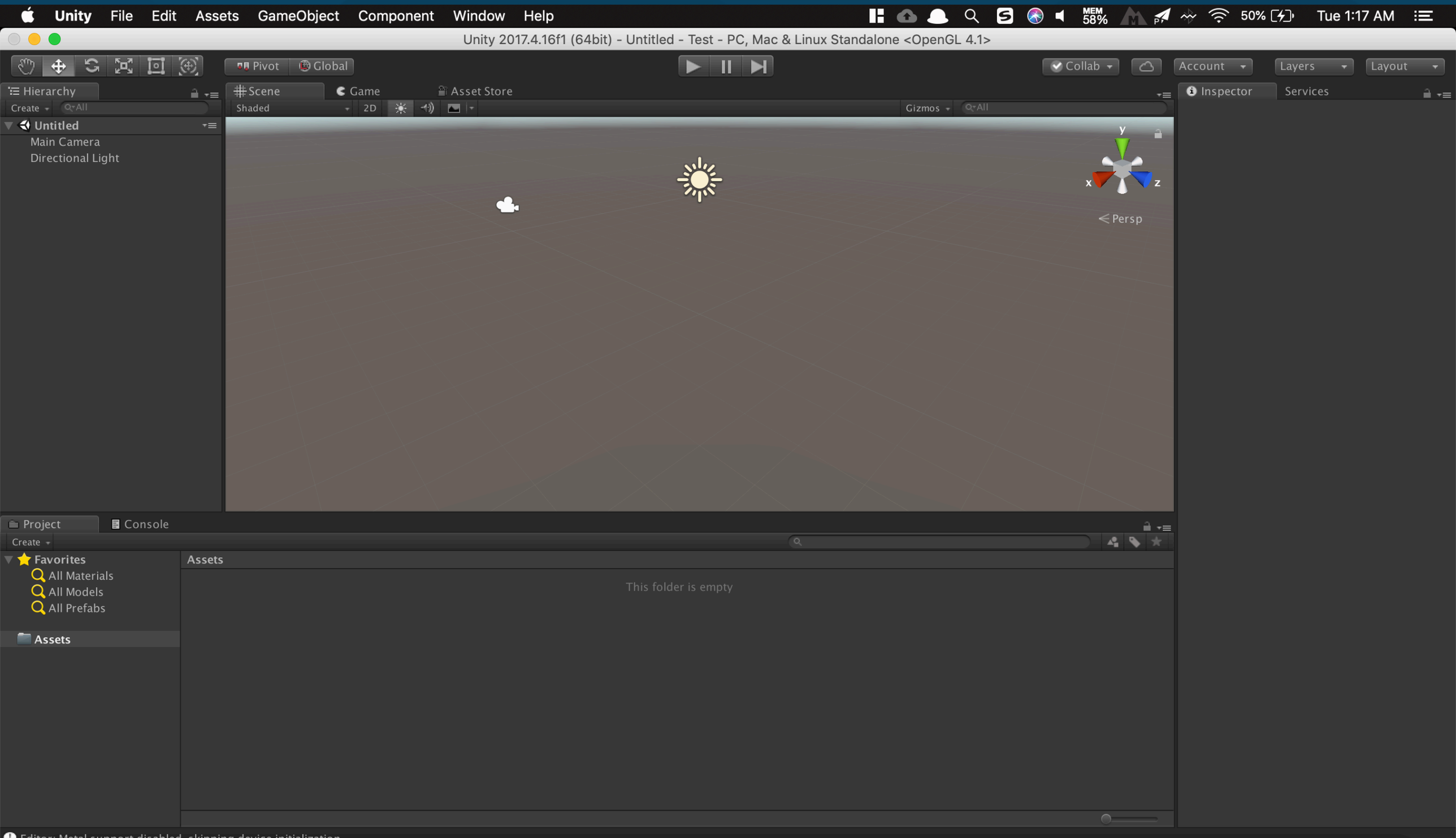
# 应用程序

- 消息队列
- 依次处理消息队列
- 更新视图

# 游戏

- 大循环每帧更新
- 双缓冲绘图

# Unity面板介绍


Unity File Edit Assets GameObject Component Window Help
MEM 58%
50%
Tue 1:17 AM
Unity 2017.4.16f1 (64bit) - Untitled - Test - PC, Mac & Linux Standalone <OpenGL 4.1>
Pivot
Global
Collab
Account
Layers
Layout
Hierarchy
Create
All
Untitled
Main Camera
Directional Light
Scene
Game
Asset Store
Shaded
2D
Gizmos
All
y
x
z
Persp
Inspector
Services
Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
Assets
Assets
This folder is empty
Editor: Metal support disabled, skipping device initialization

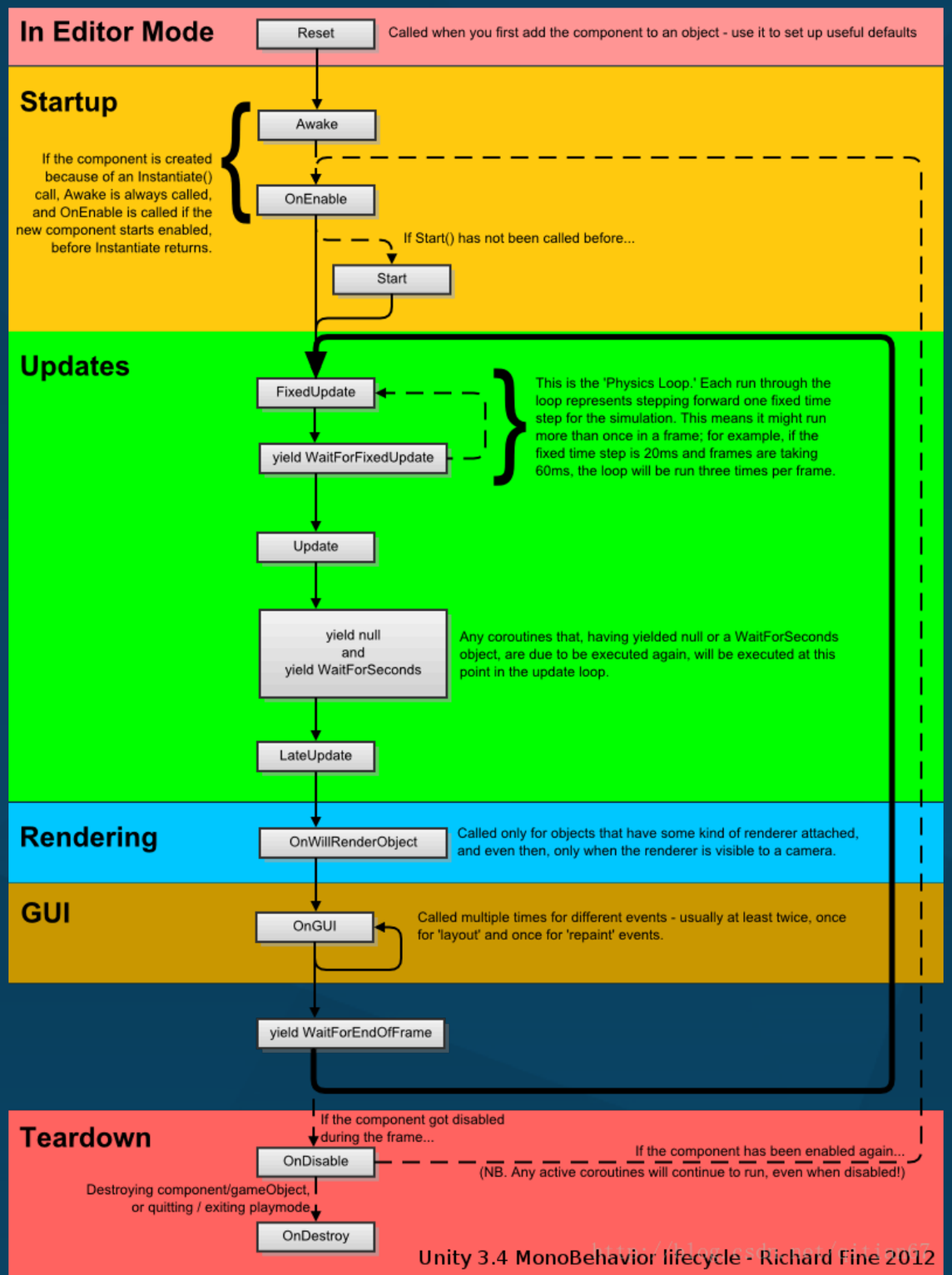
# Unity生命周期的核心

## Monobehavior

# MonoBehavior的生命周期

- Awake
- Start
- Update
- OnDestroy

# MonoBehavior生命周期一览



# 通过transform控制物体

- 区分本地坐标系和世界坐标系
- 小球滚动的demo

# 输入类： Input

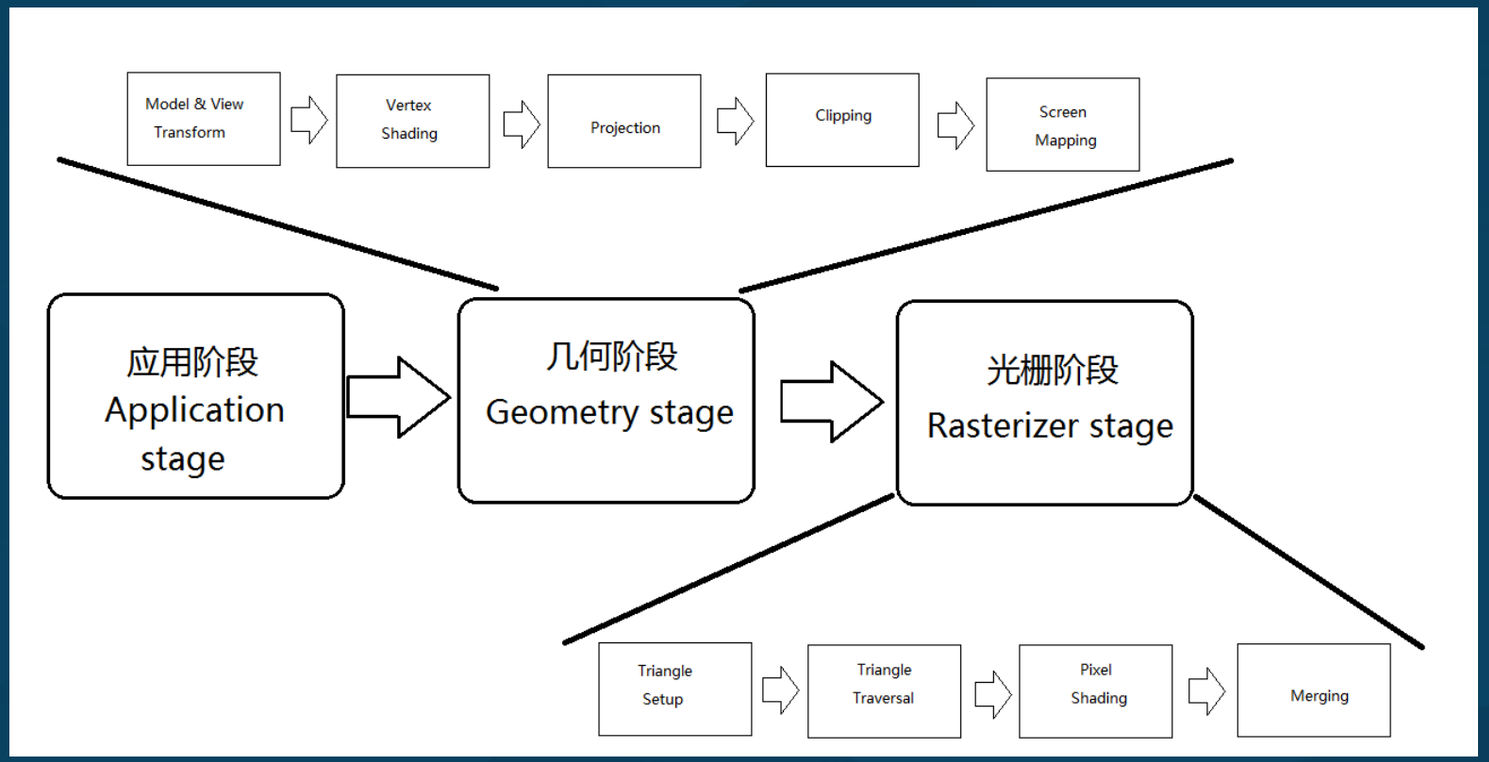
- Input.GetKey
- Input.GetMouseDown
- ...

# 镜头

- 游戏空间有一个主摄像机
- 摄像机可以根据层级拍摄不同的内容
- 可以将多个摄像机的内容组合在一起

# 图形是怎么渲染出来的

- 局部空间坐标
- 世界空间坐标
- 摄像机空间坐标
- 视域空间坐标
- 屏幕空间坐标



# 更多的内容.....

# 扩展： RenderTexture

- 什么是RenderTexture
- 我们可以用RenderTexture做什么

# 灯光

- 在场景中放置灯光
- 平行光
- 点光源
- 聚光灯
- 面光源

# 扩展：光照的烘焙

- 实时计算光照需要花费大量的算力
- 通过烘焙来模拟光照，减少运行时的损耗

# 如何在多帧
# 做同一件事情?

例如实现一个计时器

# 扩展：Unity中的协程

- 在一帧当中做不了太多东西，这个时候怎么办?

# WWW类和UnityWebRequest

- 通过地址来读取信息，本身是异步，通过协程来使用
- 无论是远端还是本地StreamingAssets都可以读取

# 协程的本质：迭代器

- yield return XXX；

# 资源系统介绍

- Resources介绍
- Prefab介绍
- AssetBundle

# Unity的特殊文件夹

- Plugins（用于原生交互）
- Editor（用于编辑器下）
- Resources（用于存放资源，可读不可写）
- StreamingAssets（存放一些资源，可读不可写）
- persistentDataPath（存放本地文件，可读可写）

# 从Resource中加载Prefab

- 我们可以使用代码从Resources中加载资源或者配置文件
- Resources.Load<GameObject>([相对于Resource的路径])

# 试一试：发射小球！

- 第一人称发射小球的demo

# 扩展：AssetBundle与热更新

- 实际项目中的资源管理：Assetbundle
- 文件下载与更新

# AB与Resources比较

- Resources：启动时索引导致游戏开启变慢、无法热更新、不利于文件管理、使用简单易于上手

- AssetBundle：启动时无需索引自行管理、可以热更新、可以根据文件类型管理、上手使用以及管理较复杂

# 游戏中的配置化

- 数据驱动
- 方便策划配置
- 对更新友好

# 使用xml
# 将关卡可配置化

- .NetFramework自带两种xml库
- XmlDocument
- XDocument(可使用C# Linq的特性)

# 自己尝试——改用json进行序列化

- json是一种和xml一样进行序列化的格式，语法更加轻便
- 推荐的第三方C# Json库有：
  - Minijson（只有一个文件非常轻便）
  - Litjson（很多项目都会用，易用性，通用性强）
  - NewtonJson（性能优势最大）

# 多种多样的序列化

- 本地配置
- 网络序列化
- xml、json、sqlite、protobuffer

# 课后目标

- 从零开始的搭建主城
- 一个简单的坦克大战游戏