

# SAST 2023 Unity Tutorial

Lecture 1

# Contents

- Unity简介
- Unity界面
- Unity基础:GameObject
- Unity基础: C#基础与脚本

# 什么是Unity

- Unity 是由 Unity Technologies 公司开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具。
- Unity 可以运行在 Windows 和 MacOS X 下，可将游戏发布至 Windows、Mac、Wii、iPhone、WebGL（需要 HTML5）和 Android 等平台。也可以凭借 Unity Web Player 插件发布网页游戏，支持 Mac 和 Windows 平台的网页浏览，是一个全面整合的**专业游戏引擎**。

# Unity的优势

- 免费易用
- 支持多平台
- 功能强大
- 社区完善，资源丰富

# Unity能做什么？

- 游戏开发 (\_\_\_\_\_)!



# Unity能做什么？

- 游戏开发
- 工业仿真
- 虚拟现实

# 本系列Unity课程都会讲什么？

- Lecture 1: Unity的基础概念； C#基础
- Lecture 2: Unity的坐标、材质； Unity中的C#
- Lecture 3: Canvas&UI； Unity的场景； Unity的动画
- Lecture 4: C# 进阶； 在Unity中完成项目
- 有关作业：
  - 我们会提供一个半成品demo仓库（一个塔防游戏），每次作业需要你向其中添加一些内容；最终，我们可以完成一个完整的Unity3D游戏。

# 学习完本系列Unity课程，我可以？

- ~~画大饼时间~~
- 上手Unity引擎
- 了解Unity开发的基本逻辑
- 学习Unity的基础概念



# Unity的下载与安装

- 参考课前准备

# 认识Unity

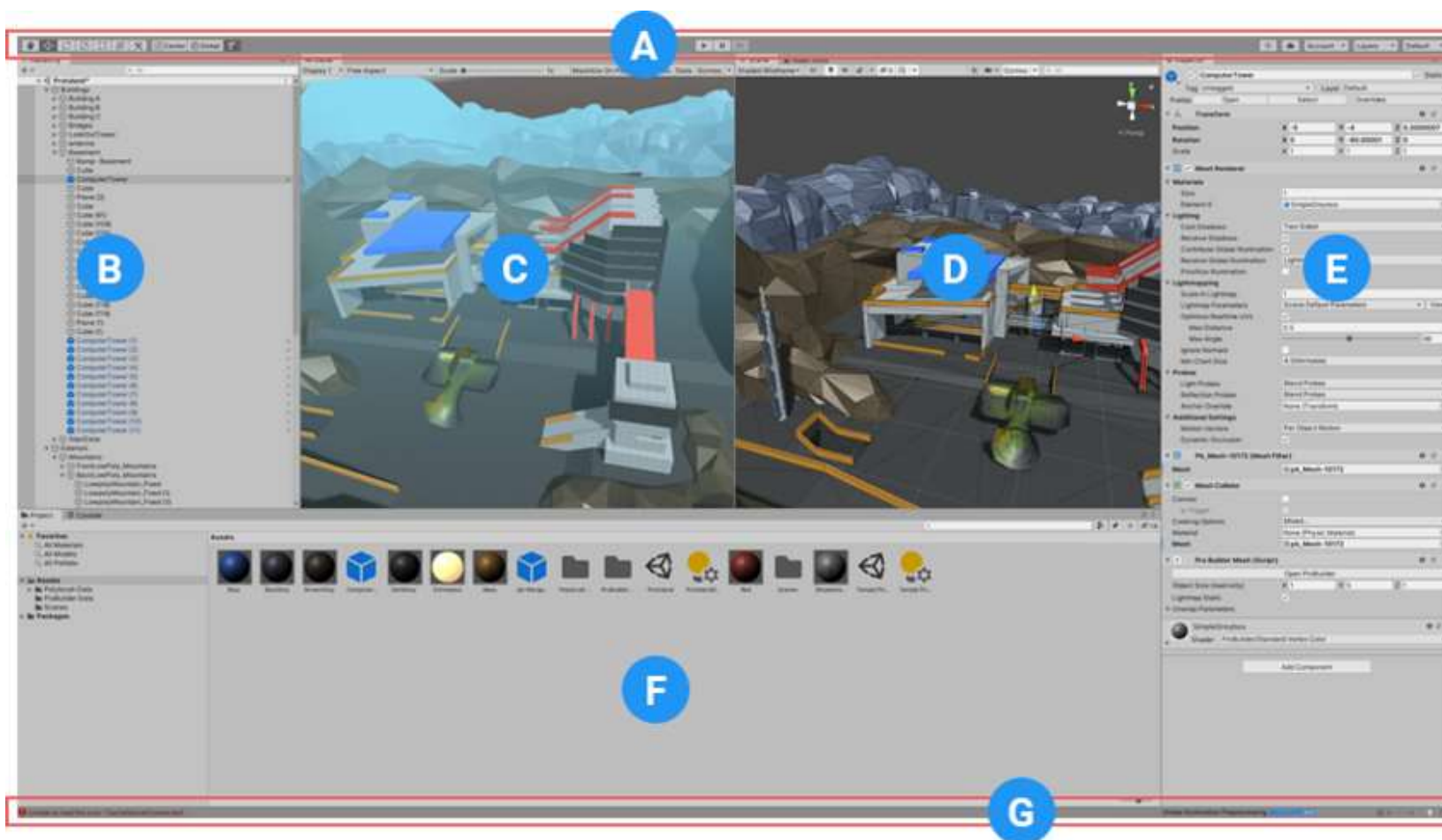
- Unity更像一款**IDE** (vscode,pycharm…)
- 采取讲课+演示的方式
- 打开你的Unity一起操作!

# 认识Unity——创建项目

- 创建一个项目是制作游戏的第一步
- demonstration

# 认识Unity——界面





(A) 工具栏提供最基本的工作功能。左侧包含用于操作 Scene 视图及其中游戏对象的基本工具。中间是播放、暂停和步进控制工具。右侧的按钮用于访问 Unity Collaborate、Unity 云服务和 Unity 帐户，然后是层可见性菜单，最后是 Editor 布局菜单（提供一些备选的 Editor 窗口布局，并允许保存自定义布局）。



(B) Hierarchy 窗口是场景中每个游戏对象的分层文本表示形式。场景中的每一项都在层级视图中有一个条目，因此这两个窗口本质上相互关联。层级视图显示了游戏对象之间相互连接的结构。

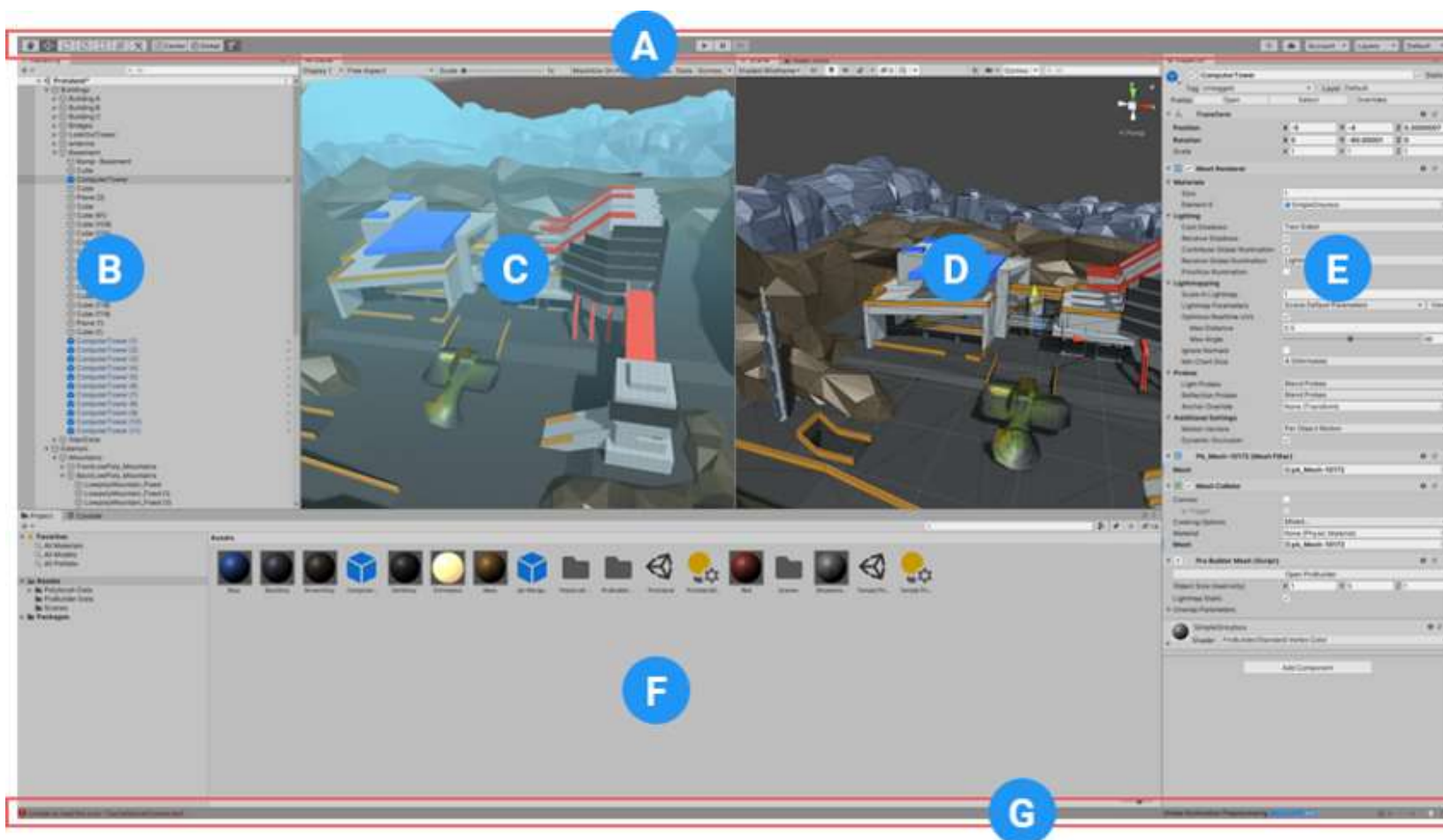


- (C) Game 视图通过场景摄像机模拟最终渲染的游戏的外观效果。单击 Play 按钮时，模拟开始。
- (D) Scene 视图可用于直观导航和编辑场景。根据正在处理的项目类型，Scene 视图可显示 3D 或 2D 透视图。

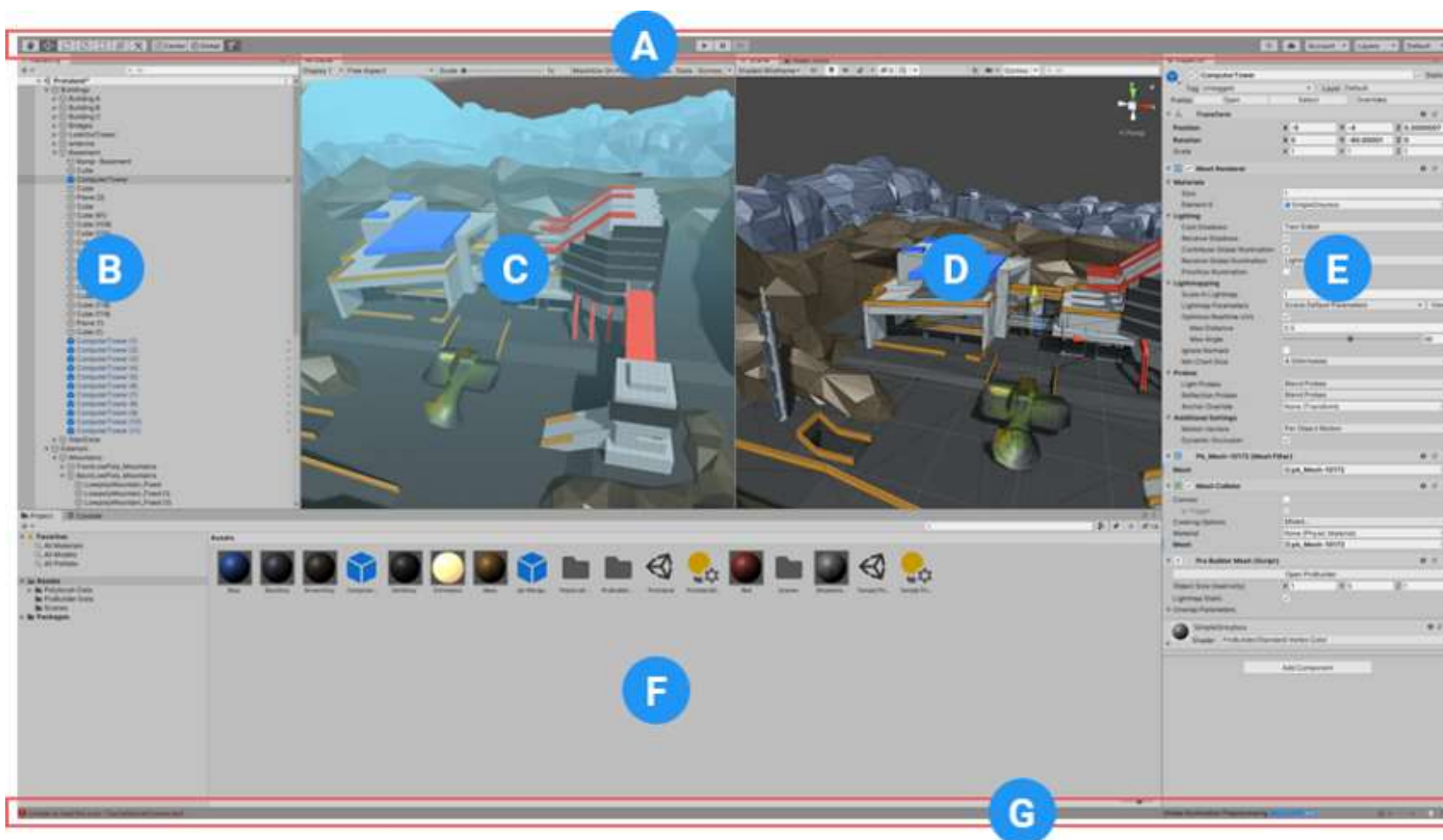


(E) Inspector 窗口可用于查看和编辑当前所选游戏对象的所有属性。由于不同类型的游戏对象具有不同的属性集，因此在您每次选择不同游戏对象时，Inspector 窗口的布局和内容也会变化。





(F) Project 窗口显示可在项目中使用的资源库。将资源导入到项目中时，这些资源将显示在此处。



(G) 状态栏提供有关各种 Unity 进程的通知，以及对相关工具和设置的快速访问。

# 认识Unity——导出项目

- 假设我们已经做完了一个游戏——(等下我们不是什么都还没做吗)——，该怎么把它从项目导出为可玩的游戏呢？
- Unity可以将项目导出到各种平台
- demonstration

# 游戏的组成——GameObject

- GameObject是 Unity Editor 中**最重要的概念**。
- 游戏中的每个对象（从角色和可收集物品到光源、摄像机和特效）都是GameObject。
- 可以向GameObject添加Component以实现功能。

# 创建一个GameObject

- demonstration

# Component

- 有了GameObject, 怎么实现游戏中各种各样的特性?
- 向GameObject添加Component (组件) !
- Component 添加到 GameObject, 为GameObject实现各种功能。  
从另一个角度来看, GameObject实质上是Component的容器,  
是包含一些Component的集合。

# Component

- 一些常见的Component:
  - Transform: 定义空间坐标、旋转和缩放
  - Camera: 为GameObject附加相机
  - Collider: 处理碰撞
  - Renderer: 处理物体的渲染
  - Animator: 动画
  - Script: 脚本
  - ...

# Component : Transform

- 游戏对象始终附加一个Transform组件（表示位置和方向），并且无法删除此组件。可以使用 Editor 的 Component 菜单或通过脚本来添加为对象提供功能的其他组件。
- Transform用于存储游戏对象的位置、旋转、缩放和父子化状态，因此非常重要。游戏对象始终附加一个变换组件，无法删除Transform或创建没有Transform的游戏对象。
- 更多的Component将会在后续课程中涉及



# 游戏的组成——Prefab

- 如果一个游戏中有1000个同样的敌人，手动创建1000个？
- 在游戏运行过程中需要创建的物体（e.g.子弹）该怎么做？
- 使用Prefab（预制件）！

# 游戏的组成——Prefab

- Unity 的预制件系统允许创建、配置和存储游戏对象及其所有组件、属性值和子游戏对象作为可重用资源。预制件资源充当模板，在此模板的基础之上可以在场景中创建新的预制件实例。

# 创建一个Prefab

- 将一个游戏对象从 Hierarchy 窗口拖入 Project 窗口。
- demonstration

# 实例化一个Prefab

- 有了敌人的预制件 我该怎么从Prefab创建一个GameObject?
- 在Unity界面中创建：从 Project 视图拖动到 Hierarchy 或 Scene 视图，在 Editor 中创建预制件资源的实例。
- demonstration
- 从脚本创建Prefab：很快介绍!

# 实例化一个Prefab

- 有了敌人的预制件 我该怎么从Prefab创建一个GameObject?
- 在Unity界面中创建：从 Project 视图拖动到 Hierarchy 或 Scene 视图，在 Editor 中创建预制件资源的实例。
- demonstration
- 从脚本创建：很快介绍!

# 从3D模型创建GameObject&Prefab

- Unity只内置了几种GameObject（球、正方体...）
- 很多时候需要从3D模型（.blend,.fbx文件）创建Prefab&GameObject
- 直接拖拽进Unity即可！ demonstration
- 有关3D模型的更多知识（Mesh、Material...）将在后续课程中涉及

# Component : Script

- 为GameObject添加控制与逻辑： 添加Component : Script
- Script（脚本）可用于：
  - 响应玩家的输入并安排游戏过程中应发生的事件。
  - 创建图形效果
  - 控制对象的物理行为
  - 为游戏中的角色实现自定义的 AI 系统。
  - ...

# C# basic

- Unity中的Script基于C#
- C# 是一个现代的、通用的、面向对象的编程语言
- C# 的构想十分接近于传统高级语言 C 和 C++，是一门面向对象的编程语言，但是它与 Java 非常相似



# C# 与 .NET 架构

- .NET 架构是微软在 2001 年推出的，基于虚拟机运行的框架
- C# 和 VB .NET 都是基于这个框架运行的
- 可以通过 NuGET 商店下载不同的插件

# C# basic

- [微软C#文档](#)
- [菜鸟教程](#)

# Component : Script

- 整个游戏的灵魂
- 一个脚本文件只有且必须有一个继承 `Monobehavior` 的类
- 也可以继承其他类和接口，把不同特征实现
  - 类和接口：抽象类可以把部分函数实现，子类只可以一个父类；接口没有实现，子类可以实现多个接口
  - OOP 万恶之源（悲）
- 对于所有直接挂接在 `GameObject` 上的脚本，都需要继承 `Monobehavior`
  - 都需要带上 `Start()` `Update()` 等功能
  - 可以分别利用 `Awake()` `OnDestroy()` 指定被挂接的 `GameObject` 的生成、销毁时候的行为，类比一个类的构造、析构函数

# Component : Script: 创建脚本

- 可以从 Project 面板左上方的 Create 菜单新建脚本，也可以通过从主菜单选择 Assets > Create > C# Script 来新建脚本。
- demonstration

# Component : Script : MonoBehaviour

- MonoBehaviour包含几个方法:Start() Update() OnDestroy()
- Update() 每帧调用一次, 用于处理游戏对象的帧更新。
- Start() 在游戏开始之前 (即第一次调用 Update() 函数之前), Unity 将调用Start()函数
- OnDestroy(): 使用Destroy()删除GameObject, 删除时将调用OnDestroy()
- 对象的构造由编辑器处理, 不需要也不能为脚本定义构造函数

# Component : Script : MonoBehaviour

- MonoBehaviour中一些实用的方法：具体参数参见[Unity手册](#)
- 鼠标相关：
  - `OnMouseDown()`:在collider上按下鼠标时除法
  - `OnMouseEnter()`:鼠标进入collider时调用
  - `OnMouseOver()`:鼠标停留在collider上时
  - `OnMouseExit()`:鼠标移开collider上时调用

# Component : Script : 变量和 Inspector

- Unity脚本中有几个比较重要常用的类:
  - GameObject
  - Transform
  - Vector2 Vector3
  - Object
  - ...
- Demonstration

# Component : Script : 变量和 Inspector

- 脚本是组件的一种
- 因此脚本中的变量可以在Unity界面中进行绑定和编辑
- Demonstration
- Inspector中的变量名相较脚本中有细微差别:[Unity官方文档](#)



# GameObject 的创建和销毁

- `Instantiate(gameObjectTemplate, position, orientation)` 创建物件
  - `gameObjectTemplate`: 此前创建的 Prefab
  - `Position`: 生成的绝对位置 (绝对与相对的概念留在第二次课介绍)
  - `Orientation`: 生成的方向, 以四元数表示
  - 返回值: 物件实例
- `Destroy(gameObject[, timeout])` 销毁物件
  - `gameObject`: 待销毁的物件实例
  - `Timeout`: 可选, 销毁时延
  - `void`

# Component : Script : 实例化预制件

- 在游戏进行过程中需要生成GameObject (e.g. 子弹,敌人...)
- 使用脚本的Instantiate()方法
- Demonstration

# Unity 坐标系统初步

- 回忆： Unity 用 Transform 控制坐标：
  - 以 Transform 控制一般部件的位置和方向
  - 以 RectTransform 控制 UI 相关元素的位置和方向
- Transform 类有两个成员：
  - **Vector3 Position**: 表示某个GameObject 的绝对坐标
    - 就是单精度浮点数 (float) 三元组
    - 坐标也可以“调参”
  - **Quaternion Direction**: 表示某个物件的朝向
    - 对于需要朝向的场景，先统一用 `Quaternion.Identity`, 后续课程将有更全面的介绍

# Component : Script : 获取GameObject

- 在游戏进行过程中, 可能需要实时获取游戏物体(e.g. 防御塔需要知道可以攻击哪些敌人)
- 如何使用脚本获取某个GameObject?
  - `GameObject.gameObject` 获取脚本实例被挂接的物件
  - `GameObject.find("name")` 按物件名称 (路径) 获取某个物件
  - `GameObject.FindGameObjectsWithTag ()` 通过标签获取某个物体
  - `transform.parent()` 获取父物件的transform组件
  - `transform.GetChild(index)` 获取下标为index的子组件
  - `transform.FindChild("name")` 获取子组件
  - ...
- Demonstration

# Component : Script : 控制Component

- 如何通过脚本控制GameObject绑定的component?
- 实际上, Unity自带的component本质上都是Script (脚本)
- 在Inspector中绑定
- `AddComponent<T>()` 对某个物件挂接脚本
- `GetComponent<T>()` 获取某个挂接在物件上的脚本实例的引用
- Demonstration

# 课程小结

- Unity简介
- Unity界面
- Unity基础:GameObject
- Unity基础: C#基础与脚本

# 作业1

- 作业内容:
  - 创建GameObject
  - 创建Prefab
  - 使用脚本实现点击地图放置防御塔
  - 使用脚本实现生成怪物
- 我们会提供详细的提示与帮助
- 除此之外，可以自行添加任何你喜欢的内容!
- 作业仓库随后发布! ~~—(在做子在做子)—~~

Thanks!

Any questions?