ECS简介

上篇文章提了，ECS是Unity DOTS的一部分，即Entity Component System。

ECS为什么有用？

- Extremely performant code (default) (默认写出的代码是极其高效的)
- Easier to read (易读)
- Easier to reuse code (易重用)
- Take advantage of Burst Compiler for High Performance C# (充分利用Burst编译器和HPC#)
- C# Job System (能够利用C# Job System)

总是，ECS是一种更好的组织数据的方式，让我们这些普通程序员也能写出极其高效的代码。

什么是ECS?

ECS基本上就是Unity中一个新的写代码的方式。使用ECS，我们将从传统面向对象编程转为面向数据编程。

当前，以传统或经典方式(Unity 2018以前)使用Unity时，基本上任何东西都是围绕着`GameObject`和`MonoBehaviour`。比方说，想创建一个玩家角色，一般情况下我们会创建一个`GameObject`，然后命名为`Player`。然后，为了使这个玩家具有某些功能，我们会为这个`Player`对象添加各种组件(即各种`MonoBehaviour`)。这些添加的组件负责实现渲染(rendering, 如MeshRenderer)，物理效果(如碰撞)和移动等。

然而!!!使用**ECS**时，我们的游戏会分成三个部分: Entities, Components和Systems。**没有GameObject和MonoBehaviour的概念了**(以兼容模式使用ECS时，还可以部分地使用MonoBehaviour。Unity提供了工具把GameObject转换为Entity)。

Entities 用于把components组织起来。它们很像传统方式中的(轻量级的)GameObject。

Components 只是数据的容器，不包含任何逻辑(即对数据的处理)。

Systems 中包含的是游戏逻辑、行为(behavior)，处理Components中的数据。这意味着一个System负责处理所有具有某些Components的Entities。

现在，使用ECS的情况下，我们要创建一个玩家角色(player)，我们会创建一个Entity，再在这个Entity上放置Components。注意，这些Components唯一的工作就是存储关于一个player的数据，没有任何逻辑。然后，我们创建Components，例如，创建一个render system，来渲染所有的Entities。ECS包已经提供了一些基本的可用的Systems。

Unity在Github上提供了ECS使用例子代码。Youtube上有配套视频(需要科学上网)。