

# Cha1-初识神兵：为什么是 Unity

在上一课的内容中，我们一起简要了解了游戏开发的大概流程。

那么问题来了，从上古时代的示波器游戏，到如今的虚拟现实头显游戏，都是使用什么技术开发的呢？

接下来，我们将一起来了解下游戏开发技术的发展简史，特别是游戏引擎的诞生和发展。

个人微信号：iseedo

微信公众号：vrlife

## 004- 游戏引擎的诞生和发展

在正式开始学习 Unity3d 之前，我们需要记住的一个事实是，Unity3d 是一款 3d 游戏引擎，当然我们也可以用它来开发游戏之外的 3D/AR/VR 行业应用。

那么什么是游戏引擎？游戏引擎又是如何诞生的呢？

在电子游戏行业的“石器时代”，游戏开发者的日子非常不好过，对游戏开发者的能力要求相比其它类型软件的开发也更高。以早期的 Atari 2600 游戏机为例，在这个平台上耕耘的游戏开发者必须精通硬件的底层，包括如何合理使用和显示相关的硬件，并熟悉相关操作系统的内核。苹果公司创始人乔布斯的第一份工作就是给 Atari 公司开发游戏。当然，为其它设备平台开发游戏似乎相对容易些，但即便开发者可以无视显示部分的底层架构，内存的限制和调用又成了开发者头上的金箍圈。

一个游戏开发工程师需要全面了解硬件底层的知识，操作系统的内核，3D 数学，C 或者 C++ 语言。总之，在游戏引擎出现之前的远古时代，游戏开发绝对是一个超级硬核的工作，其难度绝对不亚于如今的 AI 人工智能算法工程师。

在整个二十世纪八十年代，真正意义上的游戏引擎并未诞生。尽管如此，也出现了一些用来开发 2D 游戏的类引擎系统，其中就包括了 ASCII 在 1988 年推出的 RPG Maker。

进入二十世纪九十年代，一代游戏大神 John Carmack 创立了 id Software，并把目标投向 3D 游戏的研发。在他的带领下，id Software 接连推出了《德军总部 3D》，《毁灭战士》，《雷神之锤》等惊世之作。在开发这些游戏的过程中，卡马克史无前例的采用了自己独创的 3D 游戏引擎。

以《毁灭战士》为例，其软件架构可以清晰的分为三部分：核心游戏组件（如 3D 图形渲染系统、碰撞检测系统、音效系统等）、美术资源和游戏场景，以及和玩家游戏体验紧密相连的游戏规则。通过使用游戏引擎，游戏开发者得以在游戏核心架构不变的情况下设计自己的游戏画面、人物角色、武器和关卡，也就是所谓的“游戏内容”或“游戏资源”。游戏引擎把和游戏内容本身无关的碰撞检测机制、渲染等独立出来，从而让开发者可以专注于游戏内容和机制的设计。补充一句，3D 游戏引擎之父 John Carmack 于 2014 年正式加入 VR 行业的领军企业 Oculus，并担任 CTO 一职至今。

所以说游戏引擎其实没有那么高深，简单来说，我们可以把它看作用来开发游戏的软件框架。游戏开发者使用游戏引擎开发各种平台的游戏，包括掌机游戏、主机游戏、PC 游戏、手机游戏和 AR/VR 游戏。

在 id Software 的《雷神之锤 III 竞技场》和 Epic Game 于 1998 年推出的 Unreal 游戏中，开发者都采用了类似的理念，也就是将引擎本身和游戏内容区分开来。游戏引擎本身和游戏内容一样开始具备独立的商业价值。在这个时期，高端的商业引擎开始出现，比如 Epic Games 推出的 Unreal Engine（虚幻引擎）。这种高端商业引擎虽然授权费动辄数万甚至数百万美元，但可以让多个公司使用，以快速开发自己的游戏内容。从这个角度来看，游戏引擎推动了整个游戏行业的发展。

随着游戏引擎技术发展的成熟，引擎界面和操作流程对开发者来说也越来越友好，引擎本身的应用领域也越来越广，从传统的游戏开发拓展到其它领域，比如视觉设计、培训、医疗、军事模拟和建筑等等。与此同时，除了传统的游戏主机、掌机和 PC，游戏引擎开始支持更多的设备平台，包括智能手机（Android 手机和 iPhone）、网络浏览器和 VR 设备等等。

此外，大多数的 3D 游戏对 GPU 的资源要求与日俱增，也受限于显卡性能的桎梏。但考虑到提高开发者效率的需求，越来越多的游戏引擎不再需要开发者使用 C 语言或 C++ 语言开发，而是使用更高级的编程语言，如 Java、C#、Python、Lua 等。

但是需要强调的一点是，基于种种原因，如果想在游戏开发领域有更深入的成就，那么对 C++ 语言的掌握还是必不可少的。

很多游戏引擎都提供了一整套的可视化开发工具，以及可重用的软件功能。这些开发工具通常以一种集成开发环境（IDE）的形式提供，让开发者不必从零开始“造轮子”。游戏引擎有时候又被称作“中间件”，给开发者提供了极大的便利和灵活性，可以有效的提高开发效率，降低开发的复杂度，缩减开发成本。

游戏引擎除了把开发工程师从困难之中解救出来，同时也让团队的其他成员可以更好的参与游戏开发过程之中，从原型制作到关卡或场景设计，一些原本只有靠开发工程师写代码才能搞定的工作，如今策划和美术人员也可以轻松参与。

好了，关于游戏引擎的诞生和发展，就先讲到这里。

在下一课的内容中，我们将一起来简单了解一下游戏引擎的基本架构。因为这种基本架构不光适

用于 Unity3d 引擎，也适用于其它类型的引擎，如 UE4，cocos2d-x，CryEngine 等。

我们下一课再见~