# 对游戏引擎架构的理解

## 游戏引擎的功用

从实际意义上思考，游戏引擎的目的是使游戏开发的过程更加有效率，而我们可以从以下几方面观察。

### 减少重覆操作

游戏引擎分析并整合了大多数电子游戏共有的原素，并提取其核心构架理念，成为了游戏引擎的核心。这其实是计算机科学中的面向对象概念，我们将不同个体整合成一个大类，不同个体共享部分的基础设定，而每次创作或更改都只需专注于个体独有的因素，感少对基础设定的重覆操作。

### 模塊化設計

游戏开发的过程是将设计师的想法转换成可游玩的游戏，模塊化的設計更貼近人類的思維，能方便在游戏中落实设计师的想法，也能更好的管理游戏项目的不同部分。

### 减少编程需求

在商业引擎（Unity 及 Unreal）有 Playermaker 和 Blueprint 等组件减少实现游戏逻辑的过程中对编程的需求，这降低了游戏开发的门坎，也是近年独立游戏产量大增的主因之一。同时编程的本质就是寻找事物的通用逻辑规律，以机器实现自动化的过程，因此游戏引擎的去代码化，或者说偏往高级编程语言（high level programming language）是合理的发展方向。

## 游戏引擎的主要架构

### 核心构架-游戏逻辑（Game Logic）

游戏逻辑主要是通过编写逻辑代码以实现游戏玩法，上述的 Playermaker 和 Blueprint 便是可视化的逻辑编写工具。电子游戏的操作，进程和大量其他元素都需要通过游戏逻辑来实现，而目前比较复杂或是独特的玩法都难以通过可视化工具来实现，因此大多主流游戏制作中仍需要编程。

### 核心构架-渲染引擎（Rendering engine）

电子游戏和传统游戏（例如桌游）的最大分是动态画面表现，而这是通过渲染引擎来实现。无论是 2D 还是 3D 游戏都是有渲染过程的，而这涉及到 2D 图形或是 3D 模形，以及控制他们动态表现的程序代码。当然游戏引擎在大多时候都只是负责整合和渲染游戏中的图形表现，图形和代码的创作和编辑一般是使用专门的软件工具（各种 3D 图形软件和 Visual Studio），然后再导入至游戏引擎中。

### 核心构架-音频引擎（Audio Engine）

音频引擎控制游戏中的音乐，语音和其他声音（环境音，特别音效等）的表现，音频内容大多也是以游戏引擎以外的专门软件制作或录制的。音频引擎的作中就是在将音频资源导入到游戏中，并在合适的时机在游戏里播放。

核心构架-物理引擎（Physics Engine）

物理引擎主要负责在游戏中模拟各种物理法则的表现，常见的有物体碰撞，加速度，地心引力，反作用力等。物理引擎能使游戏中的部分动态表现更符合人类认知，增强玩家浸沉。商业引擎（Unity 及 Unreal）一般内置有常规的物理法则，用家一般不用自己编写，但在高质量游戏中便可能需要以代码实现更真实或符合设定的物理法则表现。

核心构架-人工智能（Artificial intelligence）

说到人工智能，大家想到的在游戏中的表现应该都与 NPC 有关。实际上，游戏人工智能除了操控 NPC 以外，也会用于自动寻路（Path finding），复杂的行为模拟和决策（Decision Making 相关，State Machine 或 behavior tree 等）。