# 虚幻引擎概述

虚幻引擎4经过1988年虚幻引擎第一带开发以来，为使用实时开发的业内人员开发的完整开发工具。从企业应用和游戏体验到高品质的个人电脑、主机、移动端、VR及AR游戏，虚幻引擎4能提供从项目初步开发到项目最终发行所需的所有功能，在和寒霜引擎以及现在在手游端流行的Unity等同类产品中优势明显。

## 实时逼真渲染

虚幻引擎基于物理的渲染技术，支持最先进的DirectX12渲染语言，高级动态阴影选项高级动态阴影选项、屏幕空间反射以及光照通道，物理模型通道等强大功能可以实现电影特效级别的视觉效果，让人感觉更加真实。

## 引擎公布C++源代码

与其他游戏开发引擎不同，虚幻引擎向开发者开放了引擎的所有源代码，这意味着每一个开发者都能按照自己的想法修改引擎，更有助于开发者们的创意性设计和引擎自身的优化。并且C++语言和C#，Java等语言相比较，虽然逻辑上比较复杂，但是它的优点是有更高的执行效率，能够让引擎更流畅地运行，实现更为强大的功能。

## 蓝图：可视化编程

虚幻引擎有对设计师更加友好的蓝图可视化脚本，即使没有编程经验的人也可以快速制作出一个项目雏形。蓝图可以用来构建游戏世界中模型，规定模型的逻辑运算，创建交互，可以这么理解：蓝图就是经过简化版本的C++，它同样有变量，函数，结构体等定义，但是这些基础的内容以及逻辑之间的连接，我们不需要像编程一样用代码来完成，而是可以通过拖拽一些控件，来组成我们的游戏框架。

## 稳健的多人框架

从1988年以来，虚幻引擎多人框架已通过众多平台以及不同游戏类型的考验，制作过众多业内顶尖的多人游戏体验。虚幻引擎推出的“开箱即用”型客户端/服务器端结构不但具有扩展性，而且久经考验，能够使任何项目的多人组件“立等可用”。

## 电影级后期处理效果

虚幻引擎的后期处理能够能让开发者优雅地调整场景的外观和感觉。动动指尖就能轻松获得电影级的效果，包括环境立方体贴图、环境遮挡、光溢出、颜色分级、景深、人眼适应、镜头光晕、光束、随机采样抗锯齿和色调映射等众多实用功能。

## 灵活的材质编辑器

虚幻引擎4的材质编辑器采用基础物理的着色技术，与蓝图相结合，可以让开发者更加直观地感受到颜色的生成，便于我们的修改和调试，其中包括颜色，混合，光照，粗糙等一系列材质属性都可以由开发者自己定义，给予开发者对于模型和角色外观和感觉的强大掌控力。使用以节点为基础的直观工作流程快速创建多种细腻逼真的材质贴图，像素级别的材质贴图和数字化的材质参数能创作出任何我们想要的风格。

## 包罗万象的动画套件

通过虚幻的网格体以及动画编辑工具，开发者能够完全地自定义角色。动画工具中强大的功能有状态机、混合空间、逆向运动学和由动作驱动的物理特性。我们可以使用动画蓝图实时更改，通过即时预览动作，配套外在动作捕捉套件，就能创造出形象逼真的人物动画。

## 完整的VR模式编辑器

在虚幻引擎的帮助下，我们可以通过VR头盔和手柄，来完成在VR模式下的游戏编辑。不同的是用手柄取代了鼠标移动，用扳机取代了鼠标点击。在VR编辑器中还有配套的操作面板，让我们的编辑过程更加轻松，大大增加了开发者身临其境的体验和开发过程中的乐趣，虚幻引擎是目前引擎中最稳定、功能最完整、最实用的VR开发工具。

## 专为VR、AR及XR而生

得益于Epic与全球顶尖的硬件及软件厂商间紧密的合作，虚幻引擎能为虚拟现实及增强现实体验的创作者提供最高品质的解决方案。通过与最流行的各大平台实现本地集成，以及前向渲染、多采样抗锯齿以及实例化双目绘制，以及单视场远景渲染等优化手段，UE4能够在无损性能的前提下制作出高品质的成果。此外，Epic也帮助推动了由Khronos Group的OpenXR发起的VR及AR标准化潮流。

## 植被与地形

我们可以使用地形系统创建广阔的，独特的世界环境。 这归功于虚幻引擎地貌系统强立的LOD系统和高效的内存分配，我们可以创建比前几代引擎大出几个数量级的地形。使用Landscape Grass功能用不同类型的花草覆盖庞大的游戏世界，并可以使用植被工具高效地绘制摆放灌木、岩石、树木及其他户外物体。

## 虚幻音频系统

使用革命性的功能提升我们项目的音频水准，包括实时合成、动态DSP效果以及物理音频传播模型。Valve已与Epic合作，将Steam Audio插件集成到了UE4中。Steam Audio为基于物理的音频空间化和传播提供了跨平台解决方案，能够使VR中的音频更加真实（体验版版已随UE4.16版一同推出）。

## 内容浏览器

使用虚幻引擎4的内容浏览器来导入、组织、搜索、添加标签、过滤及修改虚幻编辑器中的大量游戏资源。拖曳资源并将其直接放入屏幕以创建属于我们的世界。创建资源集以用于个人工作，或者和其他开发者共享。

## 商城生态圈

虚幻商城拥有数以千计的高品质资源及插件，这些引擎可以省去我们制作模型的时间，从而专心的投入到游戏的制作过程中，此外具有动画等高级功能的模型可加速我们的制作流程并为项目带来新的功能，可以通过商城获得全新的环境、角色、动画、纹理、道具、声音及视觉效果、音轨、蓝图集成插件、辅助工具以及完整的初学者资源。

## 无限可扩展性

我们可以使用模块化插件系统将自己所需的几乎任何功能集成到虚幻引擎项目中。自由的资源访问功能可以让开发者创建并使用属于自己的中间件工具包。此外，虚幻商城以及GitHub社区以及国内外的众多开发者论坛也提供了无数UE4工具以及插件包供我们下载学习。

正是由于虚幻引擎的无限拓展性，才能够在它上完成许多本不属于“游戏引擎”范畴的功能。

# 虚幻蓝图

蓝图（可视化脚本）是虚幻特有的编程脚本系统，它的主要观念是，在虚幻引擎编辑器中，使用基于变量或者函数而创造的节点增加游戏可玩元素。就像其他的游戏脚本，蓝图的用法也是通过定义在游戏世界中的模型对象或者类来为游戏对象增加脚本从而控制游戏行为。在我们开发过程中，即使一个娴熟的C++程序员，也不可避免会用到由蓝图构造的对象，在引擎中，这类对象也会被直接成为”BluePirnt”。

蓝图系统非常灵活，因为正如上面所提到的，它为编程人员提供了代码中所需要的所有概念，如类，变量，函数，结构体，流程判断（循环，序列），并且在虚幻引擎中C++与蓝图也被很好的结合起来，这一点我会在虚幻引擎C++与蓝图中详细介绍。这样与C++的兼容功能，可以使开发人员更加高效地使用蓝图填补一些无需使用编程就可以完成的游戏框架搭建或者是其他游戏行为。

蓝图的形式是通过在不同的节点（代表变量，函数）来完成游戏流程的控制。

蓝图通过各种样式的节点来代表不同的概念，同时在编辑器的属性设置面板轻松更改蓝图节点的属性（如更改变量类型，增加函数输入，输出）

## 关卡蓝图

对于游戏中的每个关卡，都有一个关卡蓝图、它的作用是控制当前关卡中的对象，控制游戏流程，也可以用来设置过场动画，并可以管理类似载入，关卡检查以及其他关卡相关的功能。关卡蓝图还可以与关卡中的其他蓝图（主要是蓝图类）进行交互，比如读取蓝图类中的变量，触发蓝图类中的功能事件。

## 蓝图类

蓝图类非常适合处理交互式对象，比如游戏场景中常见的门，灯光，可操作的物体以及可更改的景观，如门的蓝图类中可以包含空间重叠时间，门开关动画，开关音频，更改材质，与玩家交互等功能。

但是蓝图类中包含的事件既可以有蓝图类本身出发，也可以放在关卡蓝图中由关卡中其他流程事件来触发。设置好蓝图类后，我们只需把它放入游戏场景，就像编程中的“实例化”，我们就可以自由使用蓝图类中的功能了，并且在蓝图类中的更改会影响每一个场景中的实例对象。