

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДВУХ ИГРОВЫХ ДВИЖКОВ UNITY И UNREAL ENGINE 4

Сравниваются базовые системы движков на конкретных примерах кода проекта Turret. Проект использует версию Unity 5.5.1 и Unreal Engine 4.18

ВВЕДЕНИЕ

Наш доклад об анализе игры "Turret" в жанре «Tower Defense (с англ.—«Башенная защита»», сокращенно TD — название жанра компьютерных стратегических игр, главная задача которого расправиться с наступающими врагами. Игра разработана на двух игровых движках Unity3d и Unreal Engine 4 — это одни из самых популярных игровых движков, доступных на сегодняшний день.

I. ЦЕЛЬ ПРОЕКТА

Продемонстрировать создание простой игры в жанре TD на различных платформах и показать сложности при разработке данных проектов. Сделать сравнительный анализ.

II. КЛЮЧЕВЫЕ ПОНЯТИЯ

В Unreal Engine 4 используется язык программирования C++. В Unity3d в основном C или JavaScript. Какая программа лучше с точки зрения языков программирования — действительно сводится к личным предпочтениям. Unreal Engine 4 имеет решение для людей, которые боятся высокого порога вхождения в C++. Это Blueprint — редактор визуального скриптинга. Технически вам не нужно писать ни одной строки кода. Это очень удобно для создания быстрых прототипов, и с их помощью вы даже можете создавать целые игры. К слову, в Unity есть похожее решение Asset Store, где можно приобрести расширение для редактора под названием Playmaker, которое так же позволяет разрабатывать прототипы игр без написания кода.

В Unity сцена состоит из объектов типа Game Object. Это пустой универсальный объект, к которому добавляются компоненты, реализованные скриптами поведения (MonoBehaviour) и встроенными компонентами движка. Иногда их оставляют пустыми, в качестве объекта-маркера, на месте которого будет создан, например, игровой персонаж или эффект.

В Unreal Engine 4 игровые объекты представляются Actor'ами и их компонентами.

AActor («актер») — это основной класс объекта, который помещается в сцене. Мы можем его создать в игровой сцене (как из редактора, так и кодом), менять его свойства и т. д. Также есть класс, от которого унаследованы все сущности движка: UObject.

В объекте Actor, в отличие от Unity, существует понятие Root Component. Это корневой компонент объекта, к которому крепятся остальные компоненты. В Unity достаточно мышкой перетащить объект, чтобы поменять у него иерархию вложенности. В Unreal это делается через привязку компонентов друг к другу ("attachment").

В Unity мы можем практически в любом месте кода написать GameObject.Instantiate и получим созданный из Prefab'a объект. В Unreal же мы «просим» объект мира (UWorld) создать экземпляр объекта. Создание объекта называется в анриале спавном, от слова spawn. Для этого используется функция World->SpawnActor.

В Unity существуют функции Start, Update и LateUpdate для обновления или начала работы скриптов MonoBehaviour. Их аналоги в Unreal — функции BeginPlay и Tick у Actor'a. У компонентов Actor'a (UActorComponent) для этого существуют функции InitializeComponent и ComponentTick, поэтому нельзя «в один клик» сделать из компонента Actor, и наоборот. Также, в отличие от Unity, Transform есть не у всех компонентов, а только у USceneComponent и унаследованных от него.

III. ВЫВОДЫ

Мы сравнили движки Unity 5 и Unreal Engine 4 на конкретных примерах и проблемах, с которыми вы можете столкнуться, начав работу с движком Unreal. Часть сложностей, присущих Unity, решена в Unreal Engine 4

1. <https://wiki.unrealengine.com>
2. <https://habrahabr.ru/company/mailru/blog/327520/>
3. <https://stfalcon.com/ru/blog/post/unity3d-vs-unreal-engine-4>

Слышова Анастасия Викторовна, студентка кафедры вычислительных методов и программирования БГУИР, anastasiyasl716@gmail.com.

Гапеенка Александра Андреевна, студентка кафедры вычислительных методов и программирования БГУИР, nosliberavit132@gmail.com.

Научный руководитель: Шатилова Ольга Олеговна, ассистент кафедрой вычислительных методов и программирования БГУИР, shatilova@bsuir.by.