

УДК 004.4'22

<https://doi.org/10.36906/AP-2020/08>

ИСПОЛЬЗОВАНИЕ ИГРОВОГО ДВИЖКА UNREAL ENGINE В ПРОИЗВОДСТВЕ ПРОМЫШЛЕННЫХ ПРИЛОЖЕНИЙ

Жалыбин И. О.

Нижневартровский государственный университет

г. Нижневартовск, Россия

Катермина Т. С.

канд. техн. наук

Нижневартровский государственный университет

г. Нижневартовск, Россия

Аннотация. Данная статья показывает возможности продвинутых игровых движков, как мощных инструментов разработки не только игр, а также САПР и других производственных программ. Описаны возможности одного из самых мощных игровых движков и способы применения классов, предоставляемых движком в разработке промышленных приложений.

Ключевые слова: игровой движок; Unreal Engine; среда разработки; САПР.

Производство промышленных программ, в частности САПР на примере которых будет вестись рассуждение, очень длительный и трудоемкий процесс, скорость и качество выполнения которого зависят от глубины изученности автоматизируемых процессов, их количества, а также опыта разработчиков. Но, как и в любой другой профессии, качество рабочего инструмента также оказывает значительное влияние на процесс разработки.

Среда разработки — не столько инструмент, сколько мастерская наполненная инструментами. Существует множество подобных программ, направленных на автоматизацию разработки в целом, содержащих паттерны множества рутинных ситуаций, не фокусируясь на разработке ПО для конкретных целей. Однако существуют среды разработки для огромной индустрии, а именно индустрии создания игр. Может показаться, что игровые движки не самое подходящее средство для реализации рабочих приложений, однако есть множество плюсов в использовании именно этих сред [1, 2].

Положительные стороны использования игровых движков в разработке САПР:

1. Индустрия компьютерных игр молода и активно развивается;
2. Общие мировые практики во многом применимы в игровой индустрии;
3. Многие аппаратные технологии разрабатываются в частности под нужды игровой индустрии (например, с 2018 года в продаже появились видекарты аппаратно поддерживающие трассировку лучей в режиме реального времени);
4. Технологии работы с 3D графикой оптимизируются и дополняются, а фундаментальные операции с 3D, многие математические операции реализованы в классах, предоставляемых движками по умолчанию;
5. Широкие возможности работы с пользовательским интерфейсом;
6. Все популярные движки обладают набором примитивов (куб, сфера, цилиндр и др.), базовыми классами (класс игрока, класс состояния игры, класс состояния игрока);

7. Многие популярные движки предоставляют возможность вести разработку под разные платформы, и если консоль очевидно не подходит под разработку программ разработки, то Linux, OSX, iPhone, Windows, Android платформы подойдут для рабочего приложения как нельзя лучше. Многие движки, в том числе и Unreal Engine.

В пояснение к 6 пункту стоит отметить, что это лишь названия. Реальные функциональные особенности более нейтральны и их вполне возможно использовать для разработок программ отличных от игр, например, с помощью класса игрока можно реализовать парящую камеру, а при помощи класса контроллер игрока можно этой камерой управлять, в наследованном классе останется только выполнить привязку к нужным клавишам. Класс GameMode (прим.: тип игры, режим игры) связывает управление с игроком, а также содержит некоторые другие классы.

Перечисленные плюсы говорят о большом комфорте разработки и возможности сосредоточиться на продукте и его функциях, а не на рутине в виде элементарного кода. Однако в противовес выступают:

1. Большой объем библиотек, входящих в состав программы;
2. Избыточный функционал движка. Многие функции не будут использованы даже в целевых программах (компьютерных играх);
3. Необходимо изучить API используемого движка, который в свою очередь может быть крайне большим;
4. Некоторые технологии требовательны к компьютеру, на котором программа будет исполняться.

Перечисленные плюсы и минусы применимы к большому количеству современных движков. Но эти среды (движки) различаются и порой очень сильно. Связанно это с направленностью движка, предполагаемым масштабом конечного продукта, поддержкой тех или иных функций и технологий, используемым языком программирования. Интересно в поддержке языков программирования движками то, что чем сложнее становится движок, чем он сильнее разрастается, тем тяжелее и дороже поддержка нескольких языков. В качестве примера можно привести историю развития движка Unity, разработчики которого со временем отказались от двух языков: Boo — диалект Python, модификации JavaScript, известной как UnityScript.

Однако существует своего рода исключение, но с некоторыми нюансами. Движок Unreal Engine предоставляет к использованию язык визуального программирования. С точки зрения программиста выглядит это совершенно иначе, чем основной язык движка C++, но в действительности является модификацией C++ скриптов. Unreal Engine предоставляет возможность преобразовать C++ скрипт в Blueprint скрипт. Для этого Существует набор макросов, которые встраиваются в C++ код. Эти макросы позволяют выделить из кода методы и параметры доступные в Blueprint, а также указать модификаторы доступа. Такая возможность позволяет уместить эффективный оптимизированный C++ код в удобную графическую обертку Blueprint. Хотя в действительности Blueprint'ы используются для прототипирования объектов, для отладки C++ методов или для настройки готовых объектов дизайнерами.

Как можно увидеть на рисунке 2, движок Unreal Engine обладает обширными возможностями в создании пользовательских интерфейсов, а также их настройке. Окно Редактора совмещает в себе настройку графического отображения, представленную на рисунках 1 и 2, а также окно Blueprint, где можно настроить логику работы проектируемого интерфейса.



Рис. 1. Пример интерфейса

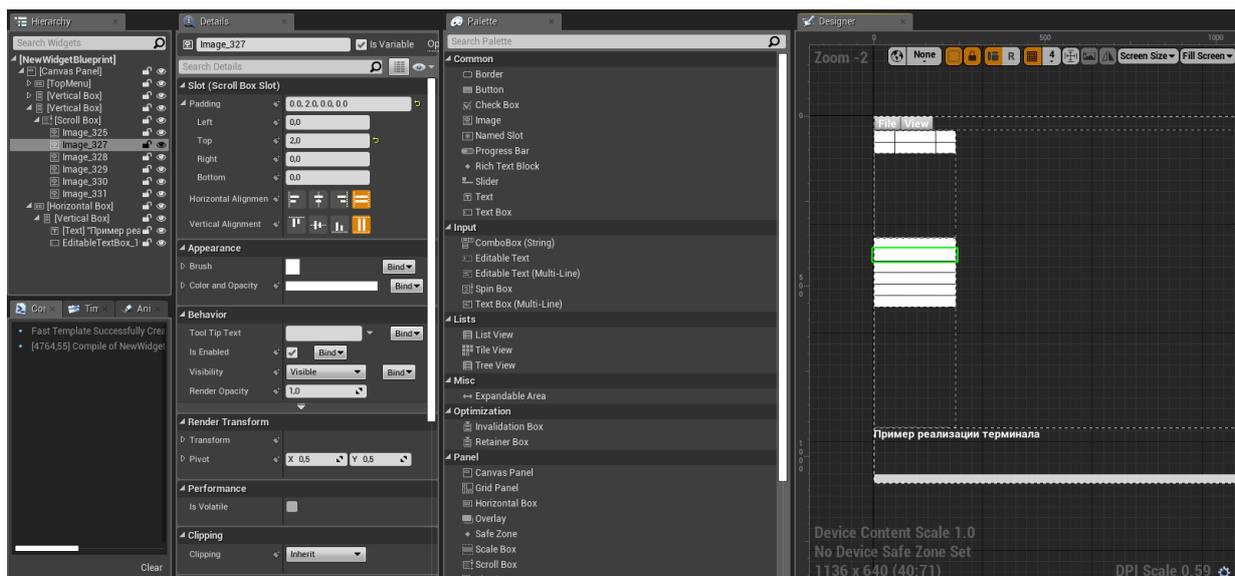


Рис. 2. Основные окна редактора итерфейсов

Одной из особенностей Unreal Engine является возможность использования его исходного кода для дополнения или удаления кода, тем самым позволяя создавать пользовательские версии движка (<https://clck.ru/T7T8s>).

https://www.youtube.com/watch?v=oQf_UuwbOLI — крайне интересная реализация неигрового приложения для создания рекламных или презентационных роликов (<https://www.youtube.com/c/UnrealEngine>). Часть технологий требуют особого оборудования, но рендеринг изображения происходит на движке Unreal Engine. Данная технология способна осуществлять рендер изображения в режиме реального времени, о чем говорится в видео.

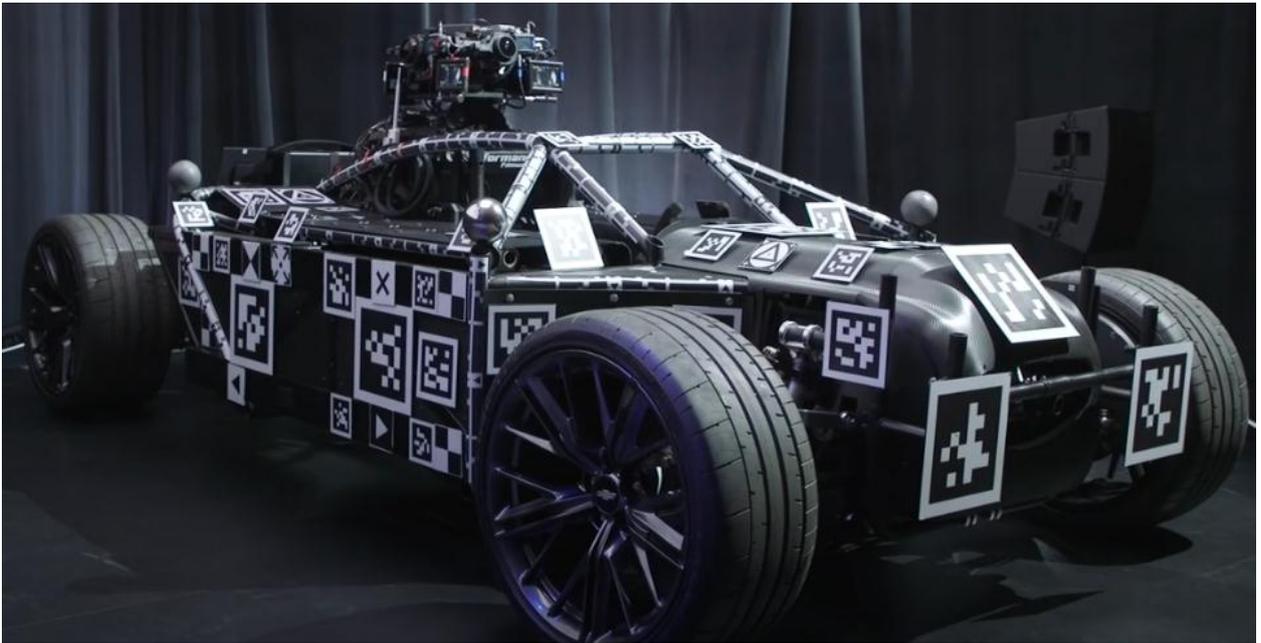


Рис. 3. Модель авто для захвата и съемок

Литература

1. Приемышев А. В. Компьютерная графика в САПР. М.: Лань, 2017. 331 с.
2. Ушаков Д. М. Введение в математические основы САПР. М.: ДМК Пресс, 2015. 208

с.

©Жалыбин И. О., Катермина Т. С., 2020