РАЗДЕЛ I. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Андреев К.А., Горячкин Б.С.

Сравнительный анализ инструментов для разработки VR-приложений для мобильных устройств

*Московский государственный технический университет им. Н. Э. Баумана
(Россия, Москва)*

doi: 10.18411/lj-04-2021-01

Аннотация

В настоящее время появляется все больше оборудования и технологий для работы с виртуальной реальностью. В связи с возрастающей доступностью виртуальной реальности для пользователей, возрастает и популярность разработки VR-приложений.

Данная статья посвящена рассмотрению и сравнению популярных инструментов для разработки приложений виртуальной реальности. Проведен сравнительный анализ двух наиболее используемых 3d-движков для создания VR-приложений, включающий выделение и расчет наиболее важных параметров движков. Разработаны экспериментальные VR-продукты для оценки производительности и процесса разработки вышеупомянутых инструментов, а также для выбора оптимального варианта.

Ключевые слова. Виртуальная реальность, смартфон, Android, 3d-движок, инструментарий, средство разработки, VR-приложение.

Abstract

Currently, there are more and more equipment and technologies for working with virtual reality. Due to the increasing availability of virtual reality for users, the popularity of developing VR applications is also increasing.

This article is devoted to the review and comparison of popular tools for developing virtual reality applications. A comparative analysis of the two most used 3D engines for creating VR applications is carried out, including the selection and calculation of the most important parameters of the engines. Experimental VR products have been developed to evaluate the performance and development process of the above-mentioned tools, as well as to select the optimal option.

Key words. Virtual reality, smartphone, Android, 3d engine, toolkit, development tool, VR application.

Введение

Не секрет, что в области разработки программного обеспечения существует множество инструментов для упрощения и ускорения процесса разработки и проектирования, таких как библиотеки, фреймворки и целые наборы средств разработки. Но что насчет виртуальной реальности? Существуют ли средства для разработки VR-приложений? Что они из себя представляют, каковы их особенности и какие именно из них стоит использовать для конкретных целей?

Крупные VR-приложения, которыми в основном являются игры или околоигровые приложения, разрабатываются для использования со специальными устройствами, позволяющими пользователю взаимодействовать с виртуальной реальностью – шлемами виртуальной реальности. Полноценные шлемы виртуальной реальности, стоимость которых превышает стоимость бюджетного игрового ПК, позволяют пользователю не только просматривать, но и полностью взаимодействовать с виртуальной средой – двигаться, трогать, перемещать предметы, и при этом у

пользователя присутствует чувство реалистичности происходящего. Однако не все пользователи могут позволить себе столь дорогой опыт взаимодействия с виртуальной средой [1].

Дорогие VR-шлемы – не единственные устройства, способные предоставить пользователю опыт взаимодействия с виртуальным окружением. Попробовать на себе виртуальное окружение можно в том числе и с помощью обычного смартфона и шлема-адаптера. Еще в 2014 году компания Google представила альтернативу полноценному VR-шлему, сделанную из картона и стекла, стоимость которой не превышала 10\$. После этого многие компании начали производство недорогих шлемов-адаптеров для смартфонов, таких как Daydream View или Samsung Gear VR [2].

Анализ средств разработки мобильных VR-приложений

Различные средства для разработки приложений имеют различную степень воздействия на процесс разработки. Так, например, небольшая библиотека, реализующую конкретную нужную разработчику для проекта функцию, просто применяется как готовое решение и встраивается в уже имеющийся проект. Фреймворки же являются комплексными инструментами, которые предлагают широкий набор возможностей и функционала, однако в таком случае архитектура проекта должна быть построена вокруг выбранного фреймворка. Крупными фреймворками для создания VR и вообще 3d приложений являются 3d движки, которые будут рассмотрены далее.

Когда речь идет о мобильной разработке, то следует понимать, что создание мультиплатформенного продукта существенно расширит потенциальную пользовательскую аудиторию. Из этого следует, что наилучшим решением будет использование такого инструментария, который легко позволит создавать продукт сразу для обеих самых популярных мобильных платформ – Android и IOS. Самыми известными и используемыми движками для создания 3d-приложений, в том числе и виртуальной реальности, являются Unity3d и Unreal Engine.

История и описание

Движок Unity3d впервые появился в 2005 году. С тех пор он стал самым популярным для разработки как 3d, так и 2d игр (60% всех приложений виртуальной и дополненной реальности, а также около 50% всех мобильных игр создано с помощью Unity) [4].

Unreal Engine 4 имеет более длинную историю, чем Unity, он был создан еще в 1998 году. С тех пор он и его модификации были много раз использованы при создании игр AAA-уровня (высокобюджетные компьютерные игры, рассчитанные на массовую аудиторию и требующие огромных затрат на разработку), таких как Unreal Tournament и Deus Ex, поэтому он считается более «профессиональным» инструментом для проектов с высоким бюджетом [3].

Инфраструктура

Виртуальную среду, с которой будет взаимодействовать конечный пользователь продукта необходимо наполнить виртуальным содержимым – виртуальными 3d-объектами, которые будет видеть пользователь. Причем для придания динамичности и реалистичности окружению, недостаточно лишь одной 3d-модели, необходимы текстуры, логика поведения, свойства, взаимодействия и другие составляющие объекта. Конечно, модель можно создать и самостоятельно, используя 3d-редактор, вроде 3dsMax или Blender, затем нарисовать текстуры и прописать логику поведения с помощью скриптов, но для желающих попробовать себя в VR такие сложные действия могут быть недоступны или слишком ресурсозатратны. Решение этой проблемы достаточно простое – использовать уже готовые объекты с моделями, текстурами, содержимым и т.д. Именно такие модели, а также множество других ресурсов, такие как скрипты, модели поведения физики, звуковое сопровождение, шейдеры, элементы графического интерфейса и т.д. и можно приобрести в Asset store [4]. В магазинах

ресурсов сравниваемых движков присутствуют сотни бесплатных и тысячи платных готовых решений для VR-разработки, а также множество общих ресурсов для разработки в целом. В обоих магазинах количество платных решений существенно больше количества бесплатных, однако в магазине Unity бесплатных решений все же больше, чем в магазине Unreal Engine.

Документация и сообщество

Для создания продукта с помощью движка необходимо для начала разобраться с тем, как с этим конкретным движком можно работать – какие функции, опции возможности он предоставляет, как ими пользоваться, как управлять элементами интерфейса редактора, как использовать настройки и т.д. Для этого существует официальная документация, которая описывает, как именно, следует работать с определенными функциями движка.

Документация Unity имеет большое количество примеров и пояснений и в целом более дружелюбна к новичкам по сравнению с документацией Unreal Engine. Помимо документации, которая, безусловно, является опорой для обучения разработки, программисты обычно посещают курсы или занятия по интересующим их технологиям и изучают официальные форумы и другие источники новой информации или решения вопросов. В целом, курсов, посвященных Unity, больше, чем курсов по Unreal Engine, причем, среди них есть не только курсы для новичков, но и продвинутые занятия, в том числе и курсы по VR.

Важной частью обучения и работы с инструментарием является взаимодействие с сообществом разработчиков, которое может помочь решить возникшую проблему или разъяснить некоторые непонятные моменты. О том, что сообщество Unity более активно, чем сообщество Unreal Engine, можно судить по большему количеству открытых тем и ответов на них на официальном форуме. Из этого можно сделать вывод о том, что начинающий разработчик Unity с большей вероятностью найдет решение своих проблем на форуме или с большей вероятностью получит ответ на пост с описанием проблемы, чем разработчик Unreal.

Программирование

Следующее, о чем стоит сказать – это о самом процессе программирования с использованием этих инструментов. Основным языком программирования в Unity является C# в то время как в Unreal Engine используются C++ и специальный язык визуального программирования Blueprint. C++ более сложен в освоении для новичков, чем C#, который часто используется для написания десктопных приложений для Windows и веб-логики, однако это не является проблемой в случае, если разработчик уже знаком с основами. Визуальный язык Blueprint использует узловую структуру и позволяет очень наглядно создавать логику поведения объектов, однако визуальное ориентирование может быть сильно затруднено при усложнении логики, сопровождающем расширение проекта [5].

Стоит отметить, что оба движка имеют открытый исходный код, но если исходный код Unreal Engine можно модифицировать, то в случае с Unity, для этого необходима специальная лицензия. Маленьким студиям и независимым разработчикам не стоит беспокоиться относительно модификации исходного кода, так как даже малейшие изменения, внесенные туда могут привести к серьезным ошибкам, поэтому обычно они стараются этого избежать, чему весьма способствует тот факт, что практически все нужные и необходимые функции для удовлетворения их потребностей движок предоставляет по умолчанию или посредством проверенных модификаций и плагинов. С другой стороны, крупные компании могут предпочесть использование Unreal Engine, чтобы в случае возникновения необходимости добавления чего-либо в исходный код движка, у них не возникло с этим особых проблем.

Определение критичных характеристик инструментов VR-разработок

Вышеприведенное текстовое описание должно было помочь читателю сформировать некоторое представление об особенностях Unity и Unreal Engine, однако для того, чтобы выбрать тот или иной движок для своего проекта в конкретной ситуации, необходимо провести объективный сравнительный анализ. Сначала необходимо определить наиболее важные параметры для сравнения движков.

Остановимся на наиболее важных параметрах с пояснением выбора, а также методиками расчета и поиска.

1. Охват форматов 3d-моделей

Важным фактором работы с 3d-движком являются форматы файлов, которые он поддерживает. 3d-модели обычно создаются в 3d-редакторах или импортируются в проект как готовые решения. Чем больше форматов движок поддерживает, тем более вероятно, что имеющаяся у разработчика готовая модель подойдет для нового проекта. Охват форматов 3d-моделей можно определить с помощью выражения, приведенного ниже (см. (1)). Значение параметра является суммой произведения коэффициентов распространенности для каждого формата на коэффициент поддержки по умолчанию (поддержка движком определенного формата только с помощью дополнительных модулей дает меньше «баллов» к общей сумме).

$$K_1 = \sum_{i=1}^N r_i * v_i \quad (1)$$

где:

K_1 – охват форматов 3d-моделей,

r_i – коэффициент распространенности i -го формата файла,

v_i – коэффициент поддержки формата по умолчанию,

N – максимальное число рассматриваемых форматов файлов.

Коэффициент r_i может быть определен путем поиска по расширению файлов 3d-моделей в крупнейшей современной облачной системе управления версиями Github. путем глобального поиска по github.com с установкой фильтра на расширение файла; тогда количество файлов соответствует количеству результатов запроса.

На гистограмме на рис. 1 приведено распределение количества файлов в логарифмическом масштабе.

Сравнение распространенности форматов файлов

Коэффициенты распространенности, рассчитанные (коэффициент был высчитан как нормализованное значение от количества файлов для каждого формата) на основе полученных данных, приведены в табл. 1.

Таблица 1

формат	r_i
fbx	1,00
obj	0,99
stl	0,85
dae	0,84
blend	0,79
step	0,78
dxs	0,73
3ds	0,70
x3d	0,63

Коэффициент поддержки формата по умолчанию $v1$, может быть определен следующим образом:

- 1, если формат поддерживается по умолчанию,
- 0.5, если формат поддерживается посредством встраиваемых плагинов,
- 0.25, если формат требует конвертации сторонними средствами,
- 0, если формат не поддерживается.

2. Возможности работы с виртуальным окружением

Чем больше возможностей для работы с виртуальным окружением предоставляет движок, тем более интерактивным, реалистичным и достоверным становится возможным создать виртуальную среду продукта. Возможности работы с виртуальным окружением можно определить с помощью выражения (2).

$$K_2 = \frac{f + \frac{f'}{2}}{f_{max}} \quad (2)$$

где

K_2 – возможности работы с виртуальным окружением,

f – количество поддерживаемых движком функций работы с виртуальным окружением из следующего списка:

- отслеживание положения головы пользователя;
- отслеживание положения глаз пользователя;
- отслеживание положения рук пользователя (с использованием дополнительных контроллеров);
- использование камеры устройства для корректировки динамического освещения;
- поддержка фотограмметрии (трехмерного фотосканирования для реалистичного воссоздания виртуального окружения);
- поддержка Google VR сервисов;
- поддержка HDR,

f' – количество поддерживаемых движком функций работы с виртуальным окружением из вышеуказанного списка, доступных только после установки дополнительных плагинов,

f_{max} – общее количество перечисленных функций.

Данная формула отражает то, какая часть функций из общего списка функций, поддерживается соответствующим движком. Если функция поддерживается по умолчанию, то она получает 1 балл для суммы. Если функция поддерживается только с помощью дополнительных модулей, то она получают 0.5 балла для суммы. Если функция не поддерживается, она получает 0 баллов для суммы. Затем сумма делится на общее

количество функций. Таким образом, можно произвести количественную оценку по данному параметру.

3. *Возможности работы со светом*

Большее количество опций работы с освещением позволит создать более реалистичную и приятную для пользователя виртуальную среду. Возможности работы со светом можно определить с помощью выражения (3). Это выражение математически соответствует выражению (2) и отражает то, какая часть опций из общего списка опций, поддерживается соответствующим движком. Если опция поддерживается по умолчанию, то она получает 1 балл для суммы. Если опция поддерживается только с помощью дополнительных модулей, то она получает 0.5 балла для суммы. Если опция не поддерживается, она получает 0 баллов для суммы. Затем сумма делится на общее количество опций.

$$K_3 = \frac{l + \frac{l'}{2}}{l_{max}} \quad (3)$$

где

K_3 – возможности работы со светом,

l – количество поддерживаемых движком опций работы со светом из следующего списка:

- источник света типа Directional light (имитирует солнечный свет);
- источник света типа Point Light (имитирует точечный источник света);
- источник света типа Area Light (имитирует источник света, имеющий площадь);
- источник света типа Ambient Light (Заполняющий свет, не имеющий источника);
- источник света типа Spot Light (имитирует источник, испускающий свет из одной точки, но направленно, подобно прожектору);
- источник света типа Sky (имитирует источник, применяющий фон сцены как общее освещение);
- динамическое затенение в углах;
- обработка рассеянного света,

l' – количество поддерживаемых движком опций работы со светом из вышеуказанного списка, доступных только после установки дополнительных плагинов,

l_{max} – общее количество перечисленных опций.

4. *Обучающий материал и готовые решения*

Ранее, в текстовом сравнении движков, были упомянуты параметры, связанные с обучением работе с движком и ускорением разработки путем использования готовых решений, а также решением возникающих в процессе разработки проблем путем поиска решений на форумах. В качестве параметров сравнения были выбраны наиболее существенные для процесса разработки, а именно:

- количество обучающих курсов,
- количество тем на официальном форуме,
- количество готовых решений (платных и бесплатных) в официальных магазинах ресурсов,
- количество репозиторий с проектами, связанными с движком.

Один из возможных вариантов оценки параметра количества обучающих курсов – осуществление поиска по официальному portalу онлайн-академии Udemу по запросам «unreal engine» и «unity3d».

Для оценки параметра количества тем на форуме возможно осуществление поиска открытых тем на официальном англоязычном форуме Unity в разделе «AR/VR Discussion» (общее количество тем указано внизу страницы) и официальном англоязычном форуме Unreal Engine в разделе «VR and AR Development» (общее

количество тем может быть рассчитано путем умножения количества страниц на количество тем на странице).

Чтобы оценить количество бесплатных и платных готовых решений можно осуществить поиск в официальном магазине ресурсов Unity (используются фильтры «3D» и «Android», для бесплатных решений также «Free Assets») и официальном магазине ресурсов Unreal Engine (используется фильтр «Android», для бесплатных решений также «Max Price - Free»). Общее количество можно увидеть около навигации по страницам.

Оценка параметра количества репозиторий на github может быть осуществлена посредством поиска в крупнейшей современной облачной системе управления версиями Github по запросам «Unity VR» и «Unreal Engine VR» соответственно.

Для расчета соответствующих параметров в проведенном в данной статье эксперименте и последующем анализе были использованы методы, указанные выше.

Выбор приоритетных инструментов VR разработок для различных категорий пользователей

Для наглядности сравнения и облегчения принятия решения относительно степени пригодности одного из рассматриваемых инструментов к использованию в конкретном проекте или ситуации, были определены 2 типовых случая (табл. 6), соответствующих различной квалификации исполнителя проекта и, соответственно, требованиям к нему.

Выделенные категории пользователей

Первый случай олицетворяет разработчика, не имеющего достаточного опыта работы с 3d-движками, чтобы пытаться сразу реализовать сложный проект, предполагающий продвинутые взаимодействия с виртуальной средой.

Второй случай олицетворяет немного более опытного разработчика, который должен иметь опыт работы с 3d-графикой, уже создал несколько проектов с использованием других инструментов и получил не только базовые знания и умения обращения с ними (во время работы над своим первым проектом), но и общее представление о работе с 3d-графикой (во время работы над другим проектом, уже зная основы благодаря опыту с первым) и готов к освоению нового инструментария.

Таблица 2

Номер случая	Квалификация разработчика	Требования к проекту
1	Разработчик имеет опыт программирования, но имеет малый (0-1 проект) опыт работы с 3d-движками	Базовые возможности взаимодействия с виртуальной средой, низкая стоимость готовых решений
2	Разработчик имеет знания и умения работы с 3d-графикой и опыт работы с 3d-движками (>2 проектов), но до этого не работал с VR/Unity/UE	Продвинутые возможности взаимодействия с виртуальной средой, качественная обработка света, высокое качество готовых решений

В 1 случае неопытный разработчик понимает, что ему еще только предстоит разобраться в основах работы с 3d-графикой, поэтому не выдвигает серьезных требований к качеству и функционалу проекта, вместо этого такой пользователь предпочтет иметь доступ к большему числу готовых (бесплатных) решений, так как это освободит массу времени и позволит сосредоточиться на непосредственно освоении архитектуры движка, а также подробную и дружелюбную к новичкам документацию.

В 2 случае разработчик уже имеет опыт работы с 3d-движками и уже представляет, как работать с 3d-графикой. В таком случае, обучение основам у такого разработчика займет существенно меньше времени, за счет чего он сможет

сосредоточится на тонкостях работы с движком. Зная особенности инструмента, уже можно пытаться добавлять в проект продвинутой механики, для чего и может потребоваться продвинутый функционал движка. В таком случае важными станут функциональные параметры движка, а также активность форумов, где разработчик будет искать решения продвинутых проблем и изучать дальнейшие тонкости.

Подразумевается, что если разработчик хорошо знаком с одним из движков, то именно его следует выбрать для проекта. Также, если разработчик хорошо знаком с обоими движками, то подразумевается, что такой разработчик сам может определить наилучшим образом подходящий инструмент для своей конкретной задачи.

После определения категорий пользователей, необходимо, непосредственно, вычислить значения параметров для сравнения. В предыдущем разделе были перечислены выделенные для сравнения параметры, приведены и обусловлены их способы вычисления. Все параметры можно разделить на 2 категории: параметры, связанные с функциональными возможностями движка, и параметры, связанные с документацией и обучающим материалом. Далее будут приведены описания произведенных вычислений и итоговая таблица с результатами для всех параметров.

Функциональные возможности

Охват форматов 3d-моделей

Расчет показателей охвата форматов 3d-моделей (K1) для Unity и Unreal Engine был осуществлен в соответствии с методикой, определенной в предыдущем разделе. Результаты расчета включены в таблицу 5.

Возможности работы с виртуальным окружением

Соотношение функций и движков представлено в таблице 3 («+» означает поддержку функции, «-» отсутствие поддержки, «*» поддержку только с плагином). Проверка поддержки функций осуществлялась путем изучения документации движков [7][8], поиска плагинов на Github и непосредственной проверкой самого движка вручную.

Таблица 3

Функция	Unity	Unreal Engine
Отслеживание положения головы пользователя	+	+
Отслеживание положения глаз пользователя	*	+
Отслеживание положения рук пользователя (с использованием дополнительных контроллеров)	*	*
Использование камеры устройства для корректировки динамического освещения	-	*
Поддержка фотограмметрии (трехмерного фотосканирования для реалистичного воссоздания виртуального окружения)	-	*
Поддержка Google VR сервисов	*	+
Поддержка HDR	*	+

Расчет показателей возможностей работы с виртуальным окружением (K2) для Unity и Unreal Engine был осуществлен в соответствии с методикой, определенной в предыдущем разделе. Результаты расчета включены в таблицу 5.

Возможности работы со светом

Соотношение опций и движков представлено в таблице 4 («+» означает поддержку функции, «-» отсутствие поддержки, «*» поддержку только с плагином). Проверка поддержки функций осуществлялась путем изучения документации движков [7][8], поиска плагинов на Github и непосредственной проверкой самого движка вручную.

Таблица 4

Опция	Unity	Unreal Engine
Источник света типа Directional light (имитирует солнечный свет)	+	+
Источник света типа Point Light (имитирует точечный источник света)	+	+
Источник света типа Area Light (имитирует источник света, имеющий площадь)	+	*
Источник света типа Ambient Light (Заполняющий свет, не имеющий источника)	+	*
Источник света типа Spot Light (имитирует источник, испускающий свет из одной точки, но направленно, подобно прожектору)	*	+
Источник света типа Sky (имитирует источник, применяющий фон сцены как общее освещение)	-	+
Динамическое затенение в углах	*	+
Обработка рассеянного света	*	+

Расчет показателей возможностей работы со светом (К3) для Unity и Unreal Engine был осуществлен в соответствии с методикой, определенной в предыдущем разделе. Результаты расчета включены в таблицу 5.

Таблица 5

Обозначение	Параметр	Unity	Unreal Engine
К ₁	Охват форматов 3d-моделей	4,74	3,72
К ₂	Возможности работы с виртуальным окружением	0,43	0,78
К ₃	Возможности работы со светом	0,69	0,87

Обучающий материал и готовые решения

Параметры К4 – К8 связаны с материалами, которые должны помочь разработчику разобраться с использованием возможностей инструментария и создать проект. Данные параметры были рассчитаны в соответствии с методикой, определенной в предыдущем разделе. Стоит отметить, что точные значения параметров актуальны на момент написания статьи, но могут измениться со временем. Результаты расчета включены в таблицу 6.

Таблица 6

Обозначение	Параметр	Unity	Unreal Engine
К ₄	Количество обучающих курсов	6270	2890
К ₅	Количество тем на форуме	7140	5329
К ₆	Количество бесплатных готовых решений	1070	116
К ₇	Количество платных готовых решений	5474	6374
К ₈	Количество репозиторий на github	4508	425

В табл. 7 приведены значения рассматриваемых параметров для Unity3d и Unreal Engine.

Таблица 7

Обозначение	Параметр	Unity	Unreal Engine
К ₁	Охват форматов 3d-моделей	4,74	3,72
К ₂	Возможности работы с виртуальным окружением	0,43	0,78
К ₃	Возможности работы со светом	0,69	0,87
К ₄	Количество обучающих курсов	6270	2890

K ₅	Количество тем на форуме	7140	5329
K ₆	Количество бесплатных готовых решений	1070	116
K ₇	Количество платных готовых решений	5474	6374
K ₈	Количество репозиторий на github	4508	425

Как можно заметить, для разработчиков, начинающих свой путь в VR и ожидающих от инструментария доступность обучения и большее количество готовых бесплатных решений, Unity выигрывает по наиболее важным показателям. Для случая с более опытным разработчиком ситуация не такая очевидная, но если в приоритет ставится возможности работы с виртуальным окружением и светом, то в данном случае, возможно, выигрывает Unreal Engine.

Создание и оценка тестовых приложений с использованием выбранных инструментов

Определение задачи

Для того, чтобы проверить в действии оба представленных выше инструментария, проведен эксперимент в виде загрузки обоих инструментариев разработчиков и создания VR-приложения для Android.

Тогда задача будет сформирована следующим образом: создать работающее VR-приложение для ОС Android с помощью Unreal Engine и Unity. Приложение должно содержать 3d-объект, должно уметь отслеживать взгляд пользователя и использовать эти данные для взаимодействия с 3d-объектом.

В качестве устройства будем использовать обычный смартфон на Android, в частности Android 10, процессор Qualcomm Snapdragon 720G (2 ядра 2,3ГГц, 6 ядер 1,8ГГц), 6 Гб ОЗУ и видеоадаптер Adreno 618. В качестве VR-очков для проверки взаимодействия VR-приложения с линзами и реалистичности VR-окружения будет использоваться шлем VRBox.

Настройка инструментариев и создание проектов

Для создания экспериментальных проектов с целью тестирования инструментариев, необходимо их загрузить, установить и настроить для работы. Официальные руководства для начинающих включают инструкции по загрузке и настройке инструментариев, в соответствии с которыми была проведена дальнейшая работа по их установке.

Основной набор инструментов Unreal Engine был загружен с использованием Epic Games Launcher – цифрового магазина игр от Epic Games, предоставляющих данный инструментарий бесплатно для некоммерческого использования. Для работы с мобильными платформами, требуются дополнительные модули, такие как Android SDK, Java и JDK [6]. Согласно руководству [7], пакеты установки модулей должны прилагаться к основным файлам, загруженным ранее, однако, возможно, из-за изменений в структуре распространяемых файлов в новых версиях движка, их там не оказалось, поэтому они были вручную загружены с официальных сайтов Java, Google и Nvidia Codeworkers. Настройки проекта также были выставлены в соответствии с официальной инструкцией.

Несмотря на отсутствие проблем с установкой и начальной настройкой инструментария, позже, на этапе компиляции возникли проблемы, описание которых отсутствует в документации и обучающем материале, в связи с чем возникла необходимость в самостоятельном поиске решения на официальном форуме, где была найдена тема с обсуждением данного вопроса. Суть проблемы оказалась в конфликте между версиями дополнительных модулей, так как официальная утилита от Nvidia Codeworkers загружала устаревшие версии некоторых из них. Также проблемы возникли с плагином от Google для работы с VR на смартфонах, так как версия этого плагина, прилагающаяся к актуальной версии Unreal Engine, была несовместима с тестовым смартфоном. Проблема была решена загрузкой старой версии Unreal Engine,

которая поставлялась с более старой версией плагина, а также заменой новой версии плагина на загруженную старую.

Для установки инструментария Unity была использована официальная утилита UnityHub, которая автоматически установила Unity, а также все необходимые для работы, в том числе и с мобильными устройствами, модули, а также среду разработки MS Visual Studio. Загруженные модули были автоматически импортированы в проект. Специальный плагин для работы с VR был загружен с Github и без проблем импортирован в проект.

Проекты были созданы в соответствии с официальными руководствами для начинающих и включают в себя такие базовые вещи, как настройка виртуального окружения, настройка запуска на мобильном устройстве, отслеживание взгляда и взаимодействие с виртуальным окружением. Далее приведены скриншоты готовых продуктов, запущенные на тестовом устройстве.

Скриншот объекта в проекте Unity

Скриншот объекта в проекте Unreal Engine

Тестирование

Оба проекта были установлены на тестовое оборудование и испытаны. Тесты предполагают запуск приложения и проверку определенной характеристики. Далее описаны методики испытания отдельных характеристик оценивание проектов:

1. *Частота кадров* измеряется при помощи специальной утилиты для Android-устройств «Gameboost». Минимальная частота кадров – худший зафиксированный в процессе работы показатель, средняя – средний показатель на протяжении 30 секунд работы.

2. *Загрузка ЦП* измеряется при помощи специальной утилиты для Android-устройств «CPU Meter». Максимальная загрузка – пиковое значение загрузки, средняя – средний показатель на протяжении 30 секунд работы.
3. *Загрузка ОЗУ* измеряется при помощи стандартного диспетчера приложений Android. Диспетчер показывает среднее значение с момента запуска.
4. *Занимаемая память* измеряется при помощи стандартного диспетчера приложений Android. Сумма занимаемой всеми файлами установленного приложения памяти устройства.
5. *Использование батареи* измеряется при помощи стандартного диспетчера энергосбережения Android. Диспетчер показывает процент использования батареи устройства за последние 10 минут.
6. *Время запуска приложения* измеряется от момента нажатия на ярлык приложения до момента загрузки виртуального окружения. Для точности измерения использована запись экрана.
7. *Отклонение угла наклона камер* для левого и правого глаза измеряется путем установки тестируемого устройства под углом 45° к горизонтальной поверхности, а затем измерением угла наклона между одинаковыми поверхностями для левого и правого глаза. Чем больше показатель, тем сильнее затрудняется восприятие.
8. *Погрешность отслеживания взгляда* измеряется как расстояние в пикселях от центра отслеживания взгляда до края объекта в момент, когда система регистрирует взаимодействие с объектом. Чем больше показатель, тем менее точным является отслеживание взгляда.
9. *Задержка отслеживания взгляда* тестируется путем определения задержки между физическим поворотом устройства и реакцией приложения и системы отслеживания взгляда.

Результаты тестов представлены в таблице 8.

Таблица 8

	Unity	Unreal Engine
Частота кадров (средняя), к/с	59	55
Частота кадров (минимальная), к/с	57	52
Загрузка ЦП (средняя), %	23	36
Загрузка ЦП (Максимальная), %	100	92
Загрузка ОЗУ, Мб	173	400
Занимаемая память, Мб	57	305
Использование батареи, %	1,3	1,9
Время запуска приложения, с	6	4,3
Отклонение угла наклона камер, °	1	19 (1)*
Погрешность отслеживания взгляда, пикс.	26	150 (6)*
Задержка отслеживания взгляда, мс	33	33

Символом «*» отмечены результаты тестов Unreal Engine, которые помимо мобильного устройства проводились в VR-режиме на персональном компьютере (ПК), так как из-за смещения центра стереоскопической синхронизации в сторону правого глаза, которое возникает на тестовом устройстве, скорее всего, из-за проблем с совместимостью тестового устройства и движка, достоверно проверить эти параметры с помощью только тестового устройства оказалось невозможным (результаты ПК записаны в скобках).

Помимо параметров производительности, была проведена оценка графических возможностей движка в его базовой комплектации. Общее качество графики трудно

оценить с помощью числовых показателей. Далее приведены типовые настройки графики, повсеместно встречающиеся в играх и сравнение их качества в испытываемых движках.

- Качество обработки теней

Сравнение рендера теней у Unity (слева) и Unreal Engine (справа)

Невооруженным глазом видно, что качество теней у Unreal Engine заметно лучше. Тень соответствует контуру объекта, контур тени плавный и явно выражен. На статичном изображении этого не заметно, но у Unity тень немного дергается при повороте камеры, что является очень неприятным эффектом для глаза. Тень в Unity выглядит однотонной и темной, что объясняется тем, что стандартный шейдер не учитывает рассеяние света, в отличие от Unreal Engine. Этот результат доказывает, что возможности работы со светом в Unreal Engine действительно превосходят базовые возможности Unity.

- Сглаживание краев объектов

Сравнение обработки краев объектов у Unity (слева) и Unreal Engine (справа)

Здесь также заметно, что базовый рендер Unreal Engine справляется с задачей лучше – ребро куба выглядит практически гладко, в то время как у Unity можно увидеть эффект зубчатости, который также негативно сказывается на визуальном восприятии.

Результаты тестов

Результаты тестов показывают, что приложение, разработанное с использованием движка Unity, требует меньше ресурсов для работы, занимает меньше места и выдает большую производительность. Также Unity показывает хорошие результаты в создании реалистичного отображения виртуального окружения и стереоскопического эффекта, однако базовый уровень качества графики ниже, чем у Unreal Engine, поэтому улучшение графики потребует доработок проекта, которые в свою очередь скорее всего потребуют подключить к движку дополнительные модули.

Приложение было создано быстрее и с меньшим количеством проблем благодаря наличию большего количества обучающего материала, такого, как темы на форуме и разделы официальной документации.

Unreal Engine более требователен к ресурсам устройства, но взамен предлагает более высокое качество графики, особенно это заметно в работе со светом на примере отрисовки тени. Также, приложение, созданное с помощью Unreal Engine, запускается быстрее. Проблемы со стереоскопическим эффектом, выявленные при тестировании, могут быть объяснены несовместимостью тестового устройства и движка, так как в VR-симуляции на ПК таких проблем не возникает.

Заключение

В данной статье были рассмотрены инструментарии для VR-разработки и произведен сравнительный анализ движков Unity и Unreal Engine для разработки VR-приложений для мобильных платформ.

Результаты теоретических и практических исследований показывают, что движок Unity хорошо подходит разработчикам, имеющим небольшой опыт в создании 3d-приложений, которые смогут с его помощью быстрее и проще создать проект, благодаря более простому процессу разработки и большому количеству готовых решений. В то же время, продвинутые возможности работы со светом и виртуальным окружением Unreal Engine оценят более опытные разработчики.

Приведенные в статье результаты исследований, описания и результаты тестов должны помочь читателю сформировать мнение относительно приоритетности использования рассмотренных инструментов в их конкретных реализациях.

1. Разработка приложений виртуальной реальности // Umbrella IT URL: <https://umbrellait.com/ru/blog/grown-out-of-playground-vr-is-expanding-boundaries/> (дата обращения: 25.12.2020).
2. Best virtual reality SDKs to build VR apps // ThinkMobiles URL: <https://thinkmobiles.com/blog/best-vr-sdk/> (дата обращения: 25.12.2020).
3. Орёл или решка: сравнение Unity и Unreal Engine // DTF URL: <https://dtf.ru/gamedev/7227-orel-ili-reska-sravnenie-unity-i-unreal-engine> (дата обращения: 25.12.2020).
4. Unity vs Unreal Engine for XR Development: Which One Is Better? // Circuit Stream URL: <https://circuitstream.com/blog/unity-vs-unreal/> (дата обращения: 25.12.2020).
5. Джонатан Линовес. Виртуальная реальность в Unity. / Пер. с англ. Рагимов Р. Н. – М.: ДМК Пресс, 2016. – 316 с.
6. Макеффри М. Unreal Engine VR для разработчиков. / Пер. с англ. Н. И. Веселко – М.: Эксмо, 2019. – 256 с.
7. Google VR Quick Start // Unreal Engine URL: <https://docs.unrealengine.com/en-US/ShippingAndReleasing/XRDevelopment/VR/GoogleVR/QuickStart/index.html> (дата обращения: 25.12.2020).
8. Quickstart for Google VR SDK for Unity with Android // Google VR URL: <https://developers.google.com/vr/develop/unity/get-started-android?hl=ru> (дата обращения: 25.12.2020).

Бадкина Т.Е., Полетаев А.Л.

Разработка мультязычности на фреймворке laravel

ФГБОУ ВО «МГУ им. Н. П. Огарёва»

(Россия, Саранск)

doi: 10.18411/lj-04-2021-02

Аннотация

В статье приведено краткое описание особенностей разработки мультязычности на laravel, реализующие переключение языков в CRM-системе.

Ключевые слова: мультязычность, Laravel, php, MySQL, SQL-запросы.