

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

ТЕХНОЛОГИЯ ТРЕХМЕРНОЙ ВИЗУАЛИЗАЦИИ ИНТЕРЬЕРА В ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСИТ-1501
Института математики, физики, информатики
и технологий
Портнов Ю. К.

Руководитель: к.п.н., доцент кафедры ИКТО
Арбузов С. С.

Работа допущена к защите
«__» _____ 2019 г.
Зав. кафедрой _____

Екатеринбург – 2019

Реферат

Портнов Ю.К. ТЕХНОЛОГИЯ ТРЕХМЕРНОЙ ВИЗУАЛИЗАЦИИ ИНТЕРЬЕРА В ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, выпускная квалификационная работа: 63 стр., рис. 41, библи. 35 назв.

Ключевые слова: ТРЕХМЕРНАЯ ВИЗУАЛИЗАЦИЯ ИНТЕРЬЕРА, ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

Предмет разработки – технология трехмерной визуализации интерьера в виртуальной реальности.

Цель работы – разработать и описать технологию визуализации интерьера в виртуальной реальности.

В работе описаны результаты проектирования и программной реализации технологии, реализующей трехмерную визуализацию интерьера в виртуальной реальности с возможностью взаимодействия с ней в нескольких режимах.

Технология реализована с помощью игрового движка Unity, сценарии написаны с использованием языка C#. Для наполнения трехмерной сцены использовался магазин «Unity Asset Store» и 3D-редактор Blender.

В качестве сопроводительной документации создано руководство пользователя, доступ к скомпилированному установочному файлу осуществляется путем предоставления ссылки с доступом к облачному хранилищу Google Диск.

Оглавление

ВВЕДЕНИЕ.....	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТРЕХМЕРНОЙ ВИЗУАЛИЗАЦИИ.....	5
1.1 ТЕХНОЛОГИЯ ТРЕХМЕРНОЙ ВИЗУАЛИЗАЦИИ ИНТЕРЬЕРА	5
1.2 МЕТОДЫ И СРЕДСТВА РЕАЛИЗАЦИИ ТЕХНОЛОГИИ ТРЕХМЕРНОЙ ВИЗУАЛИЗАЦИИ ИНТЕРЬЕРА	17
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ТЕХНОЛОГИИ	25
ГЛАВА 2. РАЗРАБОТКА ТЕХНОЛОГИИ	30
2.1 МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ РАЗРАБАТЫВАЕМОЙ ТЕХНОЛОГИИ.....	30
2.2 ОПИСАНИЕ ПРОЦЕССА РАЗРАБОТКИ ТЕХНОЛОГИИ.....	38
2.3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.....	55
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	60

Введение

Рост вычислительных мощностей в последние 30 лет привел к повсеместному развитию компьютерной графики. В таких сферах, как кино, реклама и игры она является одним из основных и неотъемлемых элементов. Сегодня графика достигла больших высот и плотно вошла в нашу жизнь. Одной из сфер, в которой компьютерная графика представляет собой неотъемлемый элемент, является архитектурная визуализация, в том числе и визуализация интерьерных пространств.

Большую часть рынка архитектурной визуализации занимает создание статичных изображений того или иного объекта с разных ракурсов. Однако, развитие вычислительных мощностей привело к тому, что построение изображений в реальном времени, с использованием технологии растеризации, практически не уступает в фотореалистичности технологии трассировки лучей. Это дает возможность реализовывать туры по виртуальным пространствам, созданным при помощи игровых движков, подавляющее большинство из которых для рендера используют технологию растеризации. Такой подход позволяет клиенту наиболее точно оценить предлагаемый интерьерный объект.

Предмет разработки: технология трехмерной визуализации интерьера в виртуальной реальности.

Цель работы: разработать и описать технологию визуализации интерьера в виртуальной реальности.

Задачи:

1. Произвести анализ состояния проблемы интерьерной визуализации и существующих подходов для ее решения.
2. Произвести анализ и обосновать выбор инструментов для разработки технологии трехмерной визуализации интерьера в виртуальной реальности.
3. В соответствии с техническим заданием разработать и описать технологию.
4. Подготовить руководство пользователя по разработанной технологии.

Глава 1. Теоретические основы трехмерной визуализации

1.1 Технология трехмерной визуализации интерьера

В 1990-х годах область архитектурного проектирования была преобразована в результате широкого распространения компьютеров и программ CAD. Эта революция повлияла на весь процесс проектирования от начала до конца, включая методы презентации. Традиционные акварельные картины были заменены компьютерными изображениями, которые могли показать дизайн с разных сторон. Виртуальная камера может даже пролететь сквозь дизайн и создать видео-тур по концепции, которая еще не создана.

Этот новый рабочий процесс в конечном итоге сэкономил время и деньги архитекторов, инженеров и дизайнеров и дал им новые способы представить свои работы более широкой аудитории [7], [20, с. 3].

Ожидания клиентов продолжают расти с каждой новой инновацией. Благодаря тому, что фотореалистичная визуализация стала обычным явлением в телешоу и художественных фильмах, клиенты ожидают того же уровня качества в своих архитектурных презентациях. Достижения в технологии рендеринга сделали возможным создание фотореалистичных изображений с использованием симуляции физики освещения реального мира, но с большими финансовыми затратами.

Трассировка лучей всегда была дорогой и с точки зрения времени и ресурсов, для отрисовки каждого кадра требовались минуты или часы. Но активное развитие информационных технологий привело к тому, что метод растеризации, реализующий рендеринг в реальном времени, перестал уступать в фотореалистичности методу трассировки лучей [15, с. 387]. На схеме ниже (см. Рис. 1.1) можно увидеть различия этих двух технологий.

Все началось с компьютерных игр. Рендеринг в реальном времени давно используется в компьютерных играх. Понятно, что игры предполагают высокий уровень интерактивности. Ландшафт, персонажи и все иные элементы игр

должны постоянно меняться в зависимости от движений пользователя, и разработчик просто не может предсказать, как пользователь будет двигаться, чтобы отрисовать все кадры заранее. Но рендеринг в реальном времени помогает разработчикам игр создавать интерактивные миры. Когда пользователь играет в трехмерную игру, он должен получить впечатление, что его персонаж перемещается в окружающей среде или выполняет какое-то быстрое действие плавно [34]. Для этого необходимо, чтобы частота кадров была не менее 30 кадров в секунду. При выполнении данного условия, игра рендерит изображения так быстро, что движения кажутся плавными, как в случае с анимацией.

Экранизация в компьютерной графике

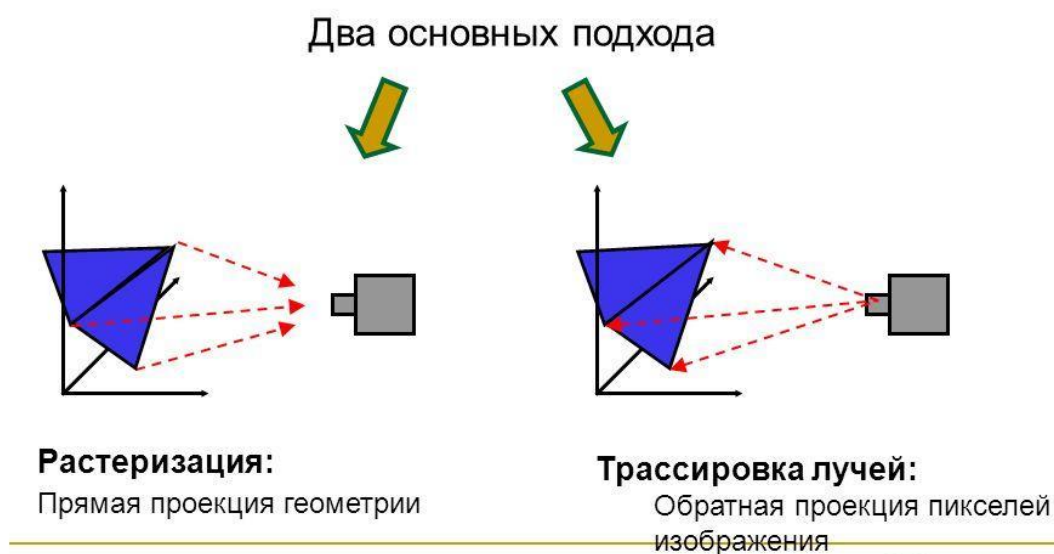


Рис. 1.1. Схема методов экранизации в компьютерной графике¹

Метод рендеринга в реальном времени постепенно распространяется от игр к другим секторам. Одним из лучших примеров в этом отношении является проект «ArchVizPRO Interior Vol.1» (см. Рис. 1.2), виртуальный демонстрационный тур, созданный итальянским визуализатором архитектуры Рутгеро Кор-

¹ http://images.myshared.ru/6/676302/slide_5.jpg

ридори с использованием игрового движка Unity в реальном времени. Особое внимание было уделено созданию освещения и материалов [17]. Проект стал широко известен не только в кругах визуализаторов, но и привлек внимание потенциальных клиентов.



Рис. 1.2. Скриншот из проекта «ArchVizPRO Interior Vol.1» Руггеро Корридо²

С точки зрения клиента, статические изображения – не лучший способ проиллюстрировать всю глубину видения архитектора или дизайнера. Они не могут предложить гибкость, подразумевающую возможность увидеть дизайн под другим углом, а не только с фиксированного ракурса, выбранного визуализатором. Клиент не может увидеть весь проект до того, как он будет построен, поэтому фактические объекты могут отличаться от ожиданий клиента. Анимация также не является лучшим методом для презентации проектов, тут на помощь приходят интерактивные технологии, которые реализуются с помощью использования все тех же игровых движков. Рендеринг в реальном времени и интерактивные технологии позволяют клиенту осмотреть виртуальное про-

² <https://bit.ly/2Kdy9vd>

странство в любом ракурсе по своему выбору [33]. Лучший пользовательский опыт среди интерактивных предоставляет технология «Virtual Reality», погружающая пользователя в виртуальное пространство [11, с. 915] (см. Рис. 1.3).



Рис. 1.3. Фотография «Asian Girl with VR headset»³

Рендеринг в режиме реального времени обеспечивает дополнительные преимущества в том числе и исполнителям. Данный режим визуализации помогает архитекторам и дизайнерам четко представлять свои проекты в режиме реального времени, реализует возможность постепенно добавлять в виртуальное пространство трехмерные модели объектов интерьера и физические эффекты, такие как солнечный свет, тени и отражения, а результаты генерировать в реальном времени [8]. Это позволяет вносить правки без необходимости продолжительное время ждать нового результата, что обеспечивает быструю скорость выполнения и приводит к большей свободе творчества. Просто взяв каждую часть рабочего процесса, которая использует традиционный рендеринг, и заменив ее рендерингом в реальном времени, исполнитель экономит огромное ко-

³ <https://bit.ly/2M80oxP>

личество времени [23]. Эти факты привлекают все больше дизайнеров и архитекторов.

В недавнем опросе архитектурных визуализаторов [1] большинство тех, кто экспериментировал с рендерингом в реальном времени, заявили, что они намерены использовать его для каждой части своего рабочего процесса. Опрос также показал, что 73 процента респондентов считают, что рендеринг в реальном времени важен для их рабочих процессов, указывая на то, что основной сдвиг в сторону рендеринга в реальном времени уже идет полным ходом. Unity был назван одним из самых популярных движков реального времени среди респондентов, которые экспериментируют с новыми движками рендеринга (см. Рис. 1.4).

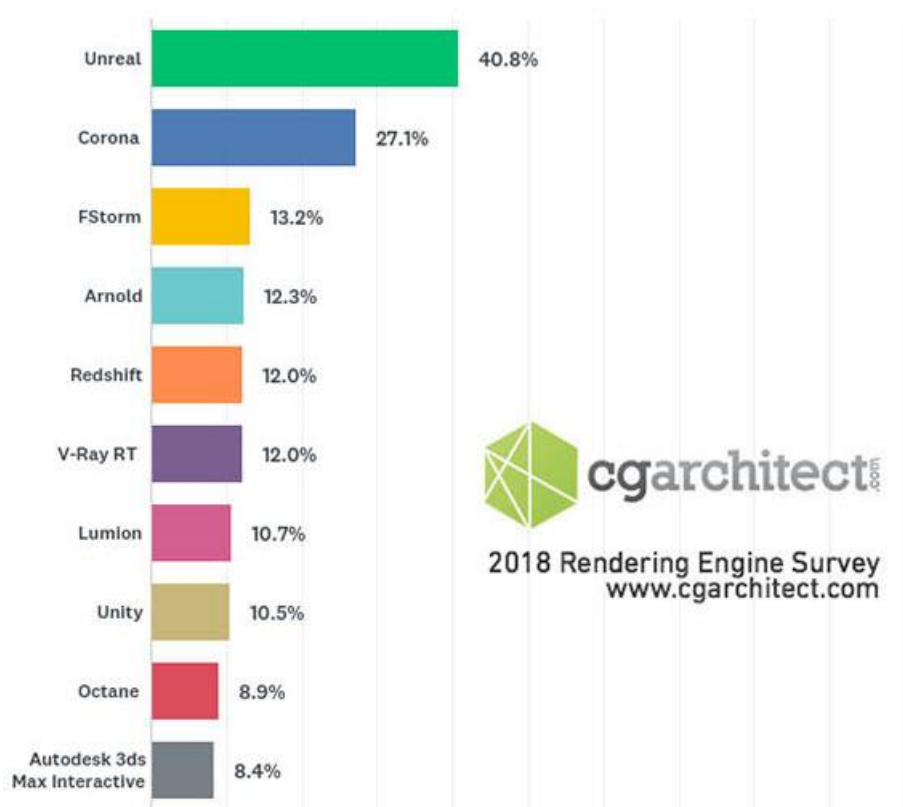


Рис. 1.4. Диаграмма распределения предпочтений респондентов различным рендер-движкам⁴

⁴ http://www.cgarchitect.com/content/posts/features/2018_02/Q8-2018.jpg

Проблема трехмерной визуализации интерьеров в виртуальной реальности актуальна в том числе и из-за интереса к данной технологии со стороны гигантов сферы архитектурного дизайна и визуализации. Для компаний, стремящихся получить конкурентное преимущество, рендеринг в реальном времени может сыграть решающую роль. Такие компании, как Neoscape, НОК, HNTB и Gensler, используют технологии реального времени, так как они изменяют процессы создания того или иного продукта в выгодную сторону от начала до конца. Использование таких технологий улучшения процесса проектирования для поддержки брендов их клиентов стало неотъемлемой и эффективной частью их рабочего процесса.

Каждая сложность прошлого довольно сильно упростилась благодаря быстрому росту технологий. Каждая визуальная презентация сегодня движется к трехмерной интерактивности. Данный процесс требует огромных затрат, но тем не менее, с учетом того, какие усилия уже предпринимаются, рендеринг в реальном времени может стать обычным требованием в архитектурной визуализации в ближайшем будущем.

Далее приведен обзор и дан критический анализ методов визуализации архитектурного внутреннего пространства отдельного помещения. Данные методы можно разделить на аналоговые и цифровые.

Аналоговые включают в себя:

- создание чертежа плана квартиры от руки;
- создание рисунка от руки.

Цифровые включают в себя:

- создание чертежа плана квартиры с использованием чертежных программ;
- создание статичного изображения с использованием 2D-редакторов;
- создание статичного изображения с использованием 3D-редакторов;
- создание анимации с использованием 3D-редакторов;
- создание интерактивного тура с использованием 3D-редакторов.

Составление чертежей является важным аспектом любой функции, связанной с созданием архитектурных объектов. Проектировщики и дизайнеры готовят технические чертежи и планы, которые содержат всю необходимую для работников сферы строительства информацию. Также, чертежи в небольшой мере являются методом визуализации, который демонстрирует схематичное изображение будущего объекта в общем плане, обычно с видом сверху.

Ручная разработка была методом, который использовался профессионалами некоторое время назад. Архитекторы и дизайнеры сидели за чертежным столом, использовали бумагу, карандаш и ластик, и тщательно рисовали каждую деталь вручную (см. Рис. 1.5). Появление программных систем, таких как CAD (система автоматизированного проектирования), сделало черчение более простым и менее трудоемким.

Photo # NH 53194 Scene in a Bureau of Construction & Repair drafting office, 1918



Рис. 1.5. Фотография «Scene in a Bureau of Construction & Repair drafting office, 1918»⁵

⁵ https://cs8.pikabu.ru/post_img/2017/12/16/10/1513442442144457538.jpg

Составление чертежей в САД имеет несколько преимуществ по сравнению с составлением чертежей от руки, в первую очередь это происходит быстрее. Черчение в САД также обеспечивает гибкость проектирования и удобство рабочей среды по сравнению с составлением вручную [20, с. 3]. Однако, некоторые эксперты считают, что черчение от руки крайне важно для понимания и сохранения основных концепций создания чертежей. В конечном итоге, учащиеся, которые знакомятся с черчением от руки, становятся более квалифицированными специалистами, используя программное обеспечение САД более эффективно.

Наибольшим преимуществом САД перед ручным составлением является сокращение времени проектирования, что приводит к снижению затрат на проектирование. В чертежах важна точность, так как любая ошибка может привести к катастрофическим последствиям. Разработчикам требуется больше времени для получения желаемых результатов при составлении вручную, поскольку каждая линия должна быть точно измерена и нарисована. Программное обеспечение САД обеспечивает полную точность измерений с помощью нескольких простых команд, тем самым экономя время. САД позволяет хранить ранее созданные шаблоны чертежей, поэтому процесс проектирования не нужно начинать с нуля для каждого нового проекта. Существующие шаблоны могут быть настроены в соответствии с требованиями нового проекта, сокращая время проектирования [14, с. 215].

Изменения дизайна часто требуются из-за предпочтений клиента. Программное обеспечение САД предоставляет разработчикам огромную гибкость для внесения необходимых изменений или модификаций. Доработка может быть сделана просто и легко, не перерисовывая планы [21, с. 3]. С ручным составлением, чтобы сделать какие-либо изменения, архитекторы и дизайнеры не имеют никакой альтернативы, кроме как вернуться к чертежной доске и начать все заново.

В отличие от черчения с помощью CAD (см. Рис. 1.6), в аналоговом способе используются такие инструменты, как чертежная доска, линейки, специальные карандаши, ластик, булавки и так далее. Эти инструменты занимают место, и рабочая среда кажется загроможденной (см. Рис. 1.5).

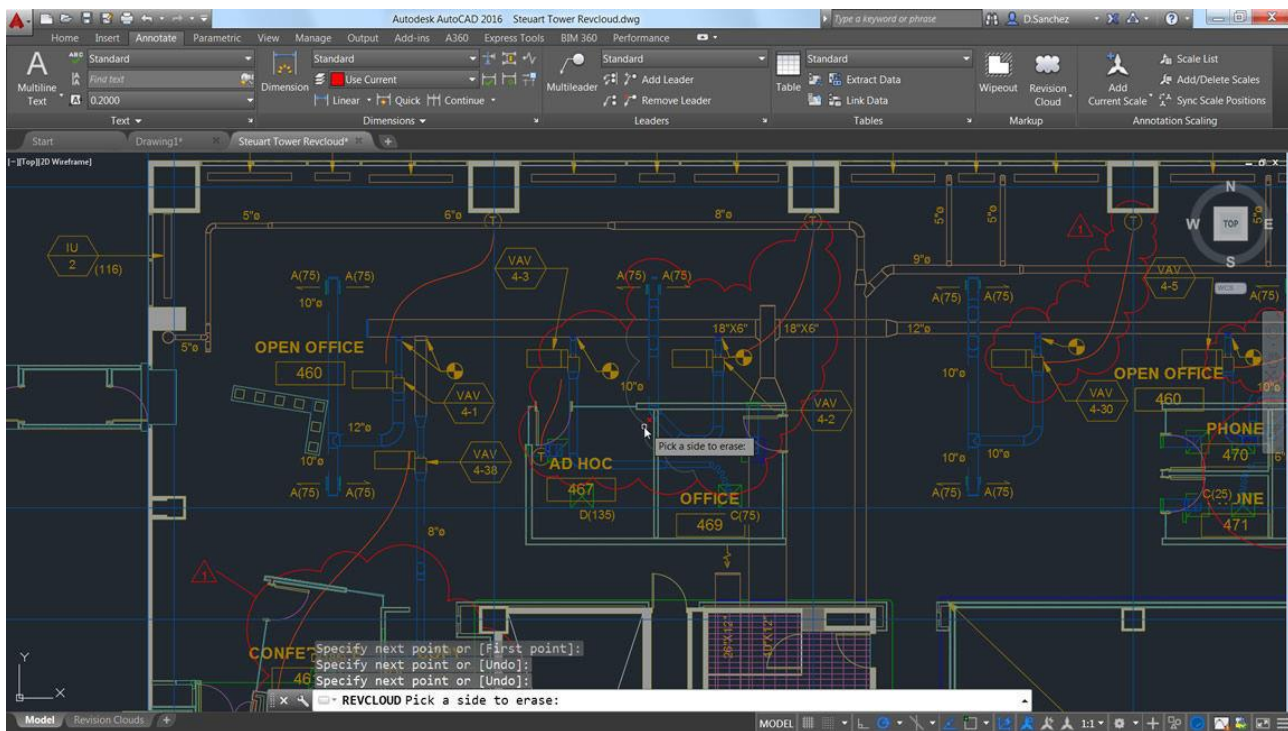


Рис. 1.6. Скриншот окна редактора программы AutoCAD.⁶

Никакой чертеж в качестве самостоятельного не является достаточным методом интерьерной визуализации, так как может предоставить клиенту только схематическое представление будущего объекта. Поэтому, чертежи планов квартир являются лишь основой для остальных методов визуализации.

Метод создания статичных изображений с использованием 2D-редакторов имеет аналогичные преимущества в отношении ручного метода, что и CAD-системы в отношении черчения от руки. Используя данные методы и приложив большие усилия, можно получить фотореалистичное изображение. Однако рисование все равно будет уступать методу визуализации с использованием 3D-редакторов [35].

⁶ http://img.directindustry.com/images_di/photo-g/14521-8997071.jpg

Появление трехмерного моделирования сильно повлияло на представление архитектурных услуг и, несомненно, является самым преобразующим событием, которое когда-либо происходило со сферой визуализации. Далее рассмотрены преимущества пришедшие вместе с трехмерной графикой.

При использовании трехмерной графики перспектива, свет и материалы строятся в автоматическом режиме, необходимо лишь создать и настроить сцену, после чего, изменяя параметры камеры можно получать изображения с разных ракурсов [28, 18 с. 8]. Это позволяет сохранить большое количество времени, а также дает возможность создавать анимацию.

Изображение, созданное с использованием 3D-редакторов, несет в себе больше информации и имеет более презентабельный вид. Для того, чтобы понять, насколько сильно рендеринг с помощью 2D-редакторов уступает рендерингу с использованием 3D-редакторов достаточно сравнить два изображения ниже (см. Рис. 1.7, Рис. 1.8).



Рис. 1.7. Рендер «Interior Design Kit»⁷

⁷ <https://bit.ly/2W0bkx2>



Рис. 1.8. Рендер «3D Furniture Rendering for Sofa in a Roomset»⁸

При использовании трехмерной графики легче увидеть влияние на общий дизайн незначительных или серьезных изменений. Это помогает завершить проектирование при меньших затратах и при меньшем количестве правок. В докладе, представленном Шерил Штауб и Атул Ханзоде [27], установлено, что эти технологии оказали значительное влияние на выполнение проекта, в том числе:

- меньше переделок в процессе исполнения проекта;
- увеличение производительности исполнителей;
- меньше запросов на дополнительную информацию;
- меньше запросов на внесение правок;
- меньший объем финансовых затрат;
- сокращение времени от начала строительства до сдачи объекта.

Помимо фотореалистичных изображений и анимации, 3D-редакторы позволяют создавать интерактивные туры по виртуальному трехмерному про-

⁸ http://www.cgarchitect.com/content/portfolioitems/2019/02/151433/3d-furniture-rendering-sofa-in-a-living-room_medium.jpg

странству. Данный метод визуализации является наиболее предпочтительным у клиентов, так как имеет ряд преимуществ над всеми вышеописанными.

Для реализации интерактивного тура, необходимо использовать технологии рендера в реальном времени, которые в свою очередь привносят все ранее описанные преимущества: сохранение времени на рендер и внесение правок, в следствии чего уменьшаются финансовые затраты, а также появляется возможность для пользователя в полной мере оценить предлагаемый ему объект, манипулируя виртуальной камерой в реальном времени.

В данной работе для визуализации интерьера планируется применить именно такой метод. Реализуется это путем создания трехмерной сцены в виртуальной реальности и реализации приложения под операционную систему Android для взаимодействия со сценой в различных режимах. Реализация стерео и моно отображения, а также режим трехмерной планировки обеспечит весь спектр преимуществ различных методов визуализации. Оптимизация приложения под мобильные девайсы обеспечит доступность в использовании.

Иные приложения, использующие интерактивный метод визуализации интерьера, реализуют лишь один способ отображения, что не в полной мере позволяет пользователю оценить предлагаемый ему объект. Также, нехватка того или иного режима может повредить доступности в использовании из-за отсутствия у пользователя необходимых аппаратных средств или недостатка мощностей девайса. Например, серия проектов по визуализации трехмерных сцен интерьеров «ArchVizPRO Interior» Руггеро Корридори имеет скомпилированные установочные файлы под мобильные девайсы, однако в них реализован лишь один режим взаимодействия в виде отображения на весь экран с управлением джойстиком [17].

Таким образом, наиболее оптимальным подходом к визуализации интерьера является реализация приложения под мобильные устройства, обеспечивающее интеракцию с виртуальной реальностью, а также разные режимы отображения и взаимодействия с ней.

1.2 Методы и средства реализации технологии трехмерной визуализации интерьера

В работе «Философия науки и техники» Н.А. Некрасова и С.И. Некрасов определяют технологию, как «совокупность методов и инструментов для достижения желаемого результата...» [4, с. 301]. В качестве методов и инструментов подразумевается приложение (программный продукт), а желаемый результат – изображение интерьера в виртуальной реальности. В целях наиболее оптимальной реализации технологии, необходимо выбрать подходящий инструмент для разработки. Поэтому был проведён сопоставительный анализ между двумя наиболее популярными игровыми движками – Unity и Unreal Engine, чтобы выяснить, какой из них наиболее подходит для реализации такого рода технологии. Также обоснован выбор вспомогательного инструмента для создания элементов трехмерной графики Blender.

Ниже представлены положительные и отрицательные моменты при использовании редактора Unity (см. Рис. 1.9).

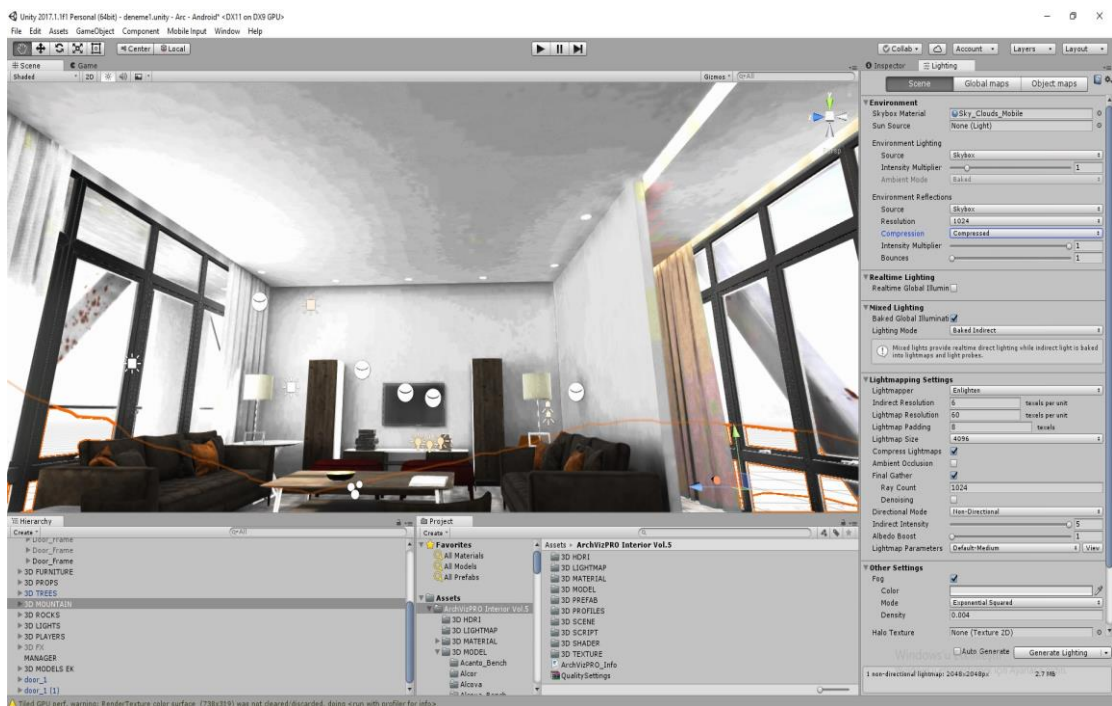


Рис. 1.9. Скриншот редактора Unity⁹

⁹ https://forum.unity.com/attachments/screen_01-jpg.247649/

Ключевыми сущностями Unity являются объекты «GameObject» и компоненты «MonoBehaviour» [26, С. 63-65]. При понимании этой концепции, можно без больших затруднений работать в Unity. Эта система, при правильном использовании, может значительно улучшить организацию вашего проекта.

В Unity есть много компонентов, которые предоставляют все необходимое для создания игры – кроме самой игровой логики. В последних обновлениях появились такие дополнения к движку, как компоненты «UI» и «Layout», которые позволяют создавать мощные и масштабируемые пользовательские интерфейсы.

Редактор может быть расширен с помощью собственных сценариев, а помимо этого в магазине «Unity Asset Store» есть множество ресурсов для любых целей. Данный магазин содержит множество полезных скриптов, моделей, материалов и многое другое. Это особенно полезно, когда дело доходит до прототипирования, поскольку появляется возможность просто загрузить временные ресурсы и использовать их для быстрого создания макетов.

Unity был одним из первых публичных игровых движков с мобильной поддержкой. Таким образом, мобильное развертывание работает стабильно, и выдаваемый визуальный результат похож на тот, который видно внутри редактора. Система периодически совершенствуется и развертывание на мобильных устройствах проходит без проблем [22, 29, с. 534].

У Unity, пожалуй, самое большое сообщество среди всех игровых движков, поэтому вероятно, что вы получите ответ, если у вас есть вопрос. Хотя Unity поддерживает несколько языков для написания сценариев, документация для каждого из них охватывает все нюансы. Более того, даже если найти ответ на другом языке, вы все равно можно будет понять логику и адаптировать его под свои нужды.

Unity проделала большую работу по оптимизации рендеринга для нескольких объектов одного типа [19, с. 196]. Чтобы получить аналогичную про-

изводительность в Unreal Engine, необходимо использовать «Instanced Rendering», что является менее гибким способом.

Из отрицательных моментов при использовании Unity можно отметить, что движок поставляется с закрытым исходным кодом. Это может быть проблематично, если необходимая функция не работает, единственным выходом из данной ситуации является ожидание обновления.

Новая система пользовательского интерфейса справляется со своими функциями. Однако, вместо того чтобы редактировать элементы используя дополнительный редактор, все манипуляции необходимо производить прямо внутри сцены, но элементы слишком большие по сравнению с остальной сценой [13, с. 447]. При открытии сцены, для редактирования пользовательского интерфейса, нужно сначала уменьшить масштаб.

Далее приведены положительные и отрицательные моменты касательно использования для создания мобильного приложения редактора Unreal Engine.

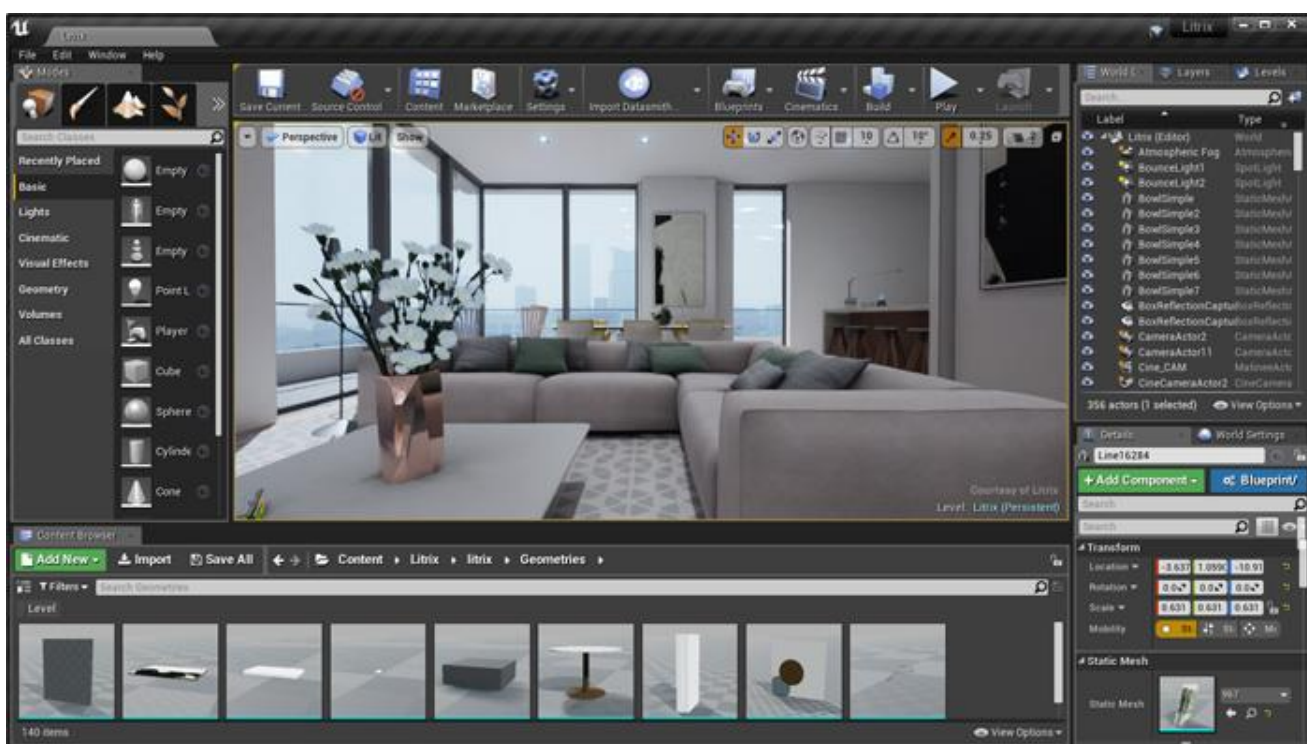


Рис. 1.10. Скриншот редактора Unreal Engine¹⁰

¹⁰ https://www.aecmag.com/images/stories/2018/03_2018/Epic_Unreal/Interior_Litrix_1.jpg

Unreal Engine имеет совершенно бесплатную пробную версию, которая не имеет никаких ограничений и является полнофункциональной. У Unity есть бесплатная версия, но для доступа к полной версии необходимо потратить большие финансовые средства.

Unreal Engine имеет мощный редактор с большим количеством специализированных редакторов внутри для различных целей [16, с 6-11]. При наличии опыта взаимодействия с этими подчиненными редакторами, они в полной мере поддерживают рабочий процесс и часто предоставляют информацию, которую Unity не показывает. Некоторые редакторы достаточно функциональны, чтобы заменить полноценные программы. То, как все эти подсистемы взаимодействуют является одним из основных преимуществ Unreal Engine.

Движок поставляется с полным исходным кодом [32]. Это позволяет исследовать код и понимать, как работают определенные части движка. Кроме того, присутствует возможность исправлять ошибки в движке или напрямую добавлять функциональность. Сообщество и компания Epic Games постоянно улучшают и обновляют кодовую базу.

В Unreal Engine реализовано большое количество используемых вариантов отображения рендеринга внутри редактора (например, отображение только освещения или полигональной сетки) [25]. Есть множество современных шейдеров, поставляемых с движком. Рендер Unreal Engine является одним из лучших на рынке.

«Blueprints» (чертежи) – это нодовая система, которая является быстрым способом создать что-то простое или реализовать базовую игровую логику без написания кода (см. Рис. 1.11). Данная система интегрирована с языком C++ [24, с. 3]. Такой инструмент полезен как для начинающих, так и для опытных разработчиков, а также позволяет им взаимодействовать лучше.

Unreal Engine в том числе имеет и отрицательные стороны. С ним сложно разогнаться. Даже будучи экспертом в C++, придется потратить много времени на изучение различных макросов и функций данного движка. Это может быть

трудно для людей, которые изучают язык C++ одновременно с изучением инструментов движка.

Нодовая система может быстро стать запутанной (см. Рис. 1.11). Логика, состоящая из десятков узлов, часто может быть упрощена до одной или двух строк эквивалентного кода в C++. Обычно это не проблема, поскольку можно использовать C++, но для некоторых вещей, таких как «UMG» (Система пользовательского интерфейса), требуются чертежи, которые могут быть сильно загромождены [24, с. 47].

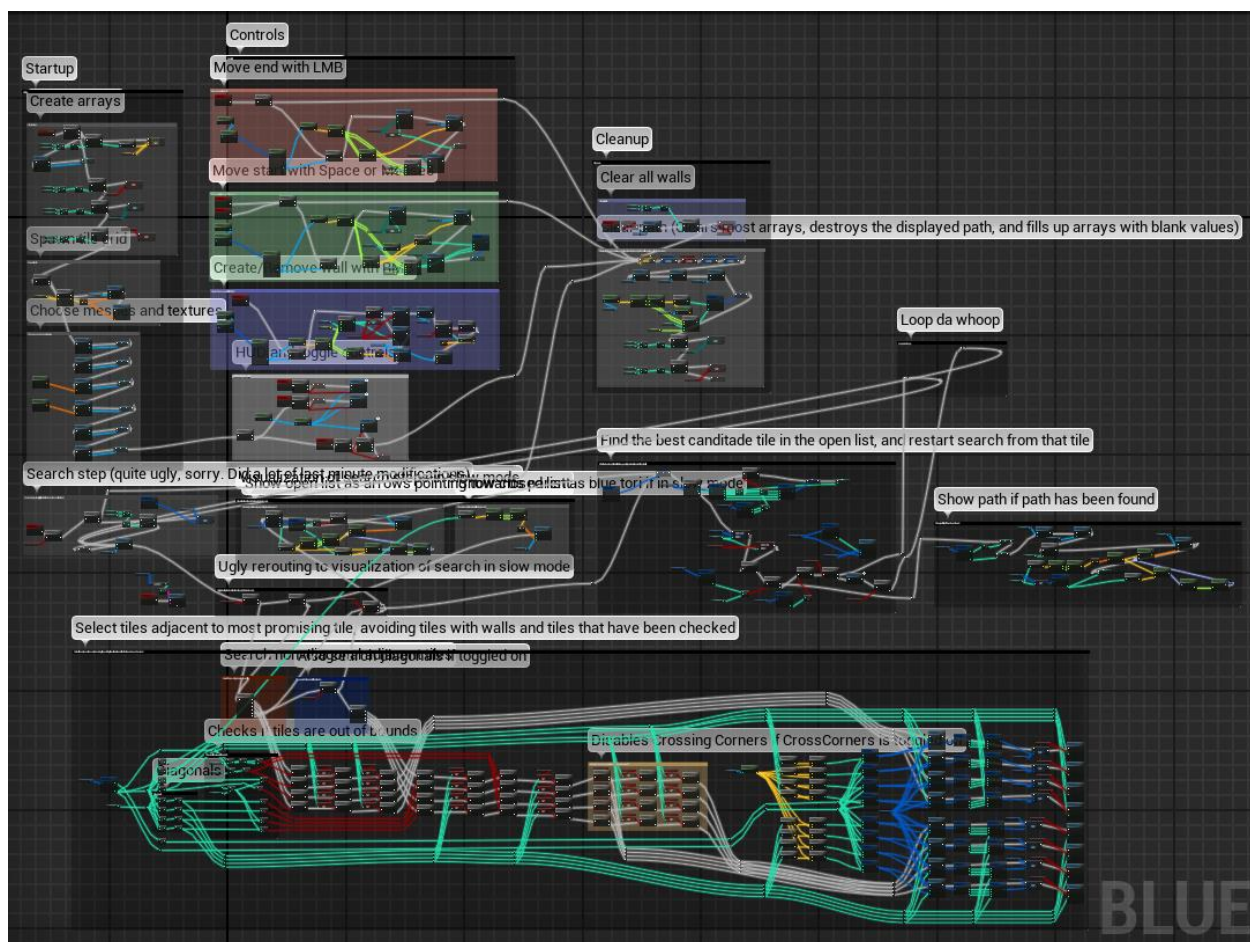


Рис. 1.11. Скриншот игровой логики, созданной с помощью «Blueprints»¹¹

Хотя Unreal Engine имеет большое сообщество, оно редко дает ответы на вопросы. Кроме того, почти вся предоставляемая поддержка предназначена лишь для нодовой системы. Вокруг Unreal Engine пользователи пытаются со-

¹¹ <http://i.imgur.com/muveguB.png>

здать активное сообщество, однако сообщество Unity уже является достаточно большим и активным.

Документация по языку C++ испытывает серьезную нехватку в информации. Справочные онлайн материалы по классам C++ бесполезны. Компания Epic Games это признает и расставляет приоритеты для их улучшения. Кроме того, постоянные обновления делают неактуальными множество функций. Необходимо быть осторожным при просмотре обучающих видео, так как они могут использовать более старую версию движка и функции, которые больше не используются.

Для графического интерфейса используется интегрированная система пользовательского интерфейса «UMG». Она основана на «Blueprints» и может быть полезной, однако эта система не всегда работает корректно и не имеет некоторых элементов управления, например, кнопки переключения. Кроме того, можно столкнуться с неожиданными сбоями редактора при разработке пользовательского интерфейса, которые могут привести к потере часов работы. Epic Games пока что лишь развивают эту систему.

Мобильная разработка с Unreal идет медленно. Развертывание на устройстве занимает много времени. На Android возникают некоторые визуальные проблемы, такие как размытые текстуры или неосвещенные деревья. Но для iOS проблемы были намного больше. По вине компании Apple, на устройствах с операционной системой iOS Unreal Engine поддерживает возможность сборки только в том случае, если проект создан полностью с использованием лишь системы «Blueprint». Таким образом, Unity с большим опережением выигрывает Unreal Engine в разработке под мобильные операционные системы.

В заключении можно отметить много положительных вещей, которые присутствуют в обоих движках. Было выявлено множество областей, в которых эти движки можно улучшить, чтобы обеспечить лучший опыт взаимодействия для разработчиков. Ни один движок не является заметно лучше, чем другой, и с постоянно меняющимися функциями и добавлением поддержки данный анализ

вполне может устареть через несколько месяцев. Эти два движка выжимают мощности устройства до предела и показывают сопоставимые показатели производительности. Эта ситуация придает особое значение удобству пользовательского взаимодействия и интуитивно понятной разработке. Учитывая все аспекты, которые были выявлены в ходе анализа этих движков, было принято решение для разработки технологии использовать движок Unity.

Несмотря на то, что Unreal Engine был изначальным выбором, после сопоставительного анализа Unity оказался лучшим кандидатом для разработки технологии.

Основными факторами, которые способствовали этому решению, были [30, 31]:

1. Unity визуально смотрится более согласованно на каждой платформе и обеспечивает беспроблемное развертывание одним нажатием на любых устройствах.
2. Unity-проекты занимают меньше места на каждом устройстве, что обеспечивает более удобный опыт взаимодействия для конечного пользователя. Меньший размер установочного файла важен в магазине Google Play, где APK разделяются, если они больше 50 МБ.
3. Unity намного легче учить и понимать. С данным движком неопытные разработчики могут создавать продукты быстрее, а также рассчитывать на поддержку большого сообщества.
4. В Unity меньше требуется времени на итерации (более быстрое развертывание, более быстрая компиляция исходного кода, шейдеры компилируются практически мгновенно).

В качестве вспомогательного инструмента, который будет использоваться в проекте для создания некоторых элементов виртуальной трехмерной сцены интерьера, был выбран 3D-редактор Blender. Ниже представлены основные преимущества данного редактора над остальными на рынке трехмерной графики.

Blender – это бесплатный 3D-пакет с открытым исходным кодом. Он поддерживает весь процесс создания трехмерной графики – моделирование, риггинг, анимацию, симуляцию, рендеринг, композитинг и отслеживание движения, даже редактирование видео и создание игр [10]. Одновременно с наличием такого количества инструментов, Blender является одним из самых легковесных редакторов на рынке, а также отличается большой производительностью даже на слабых компьютерах.

Новая версия Blender 2.80 привнесла в качестве изменений, пожалуй, лучший пользовательский интерфейс (см. Рис. 1.12) и опыт взаимодействия среди всех остальных трехмерных редакторов.

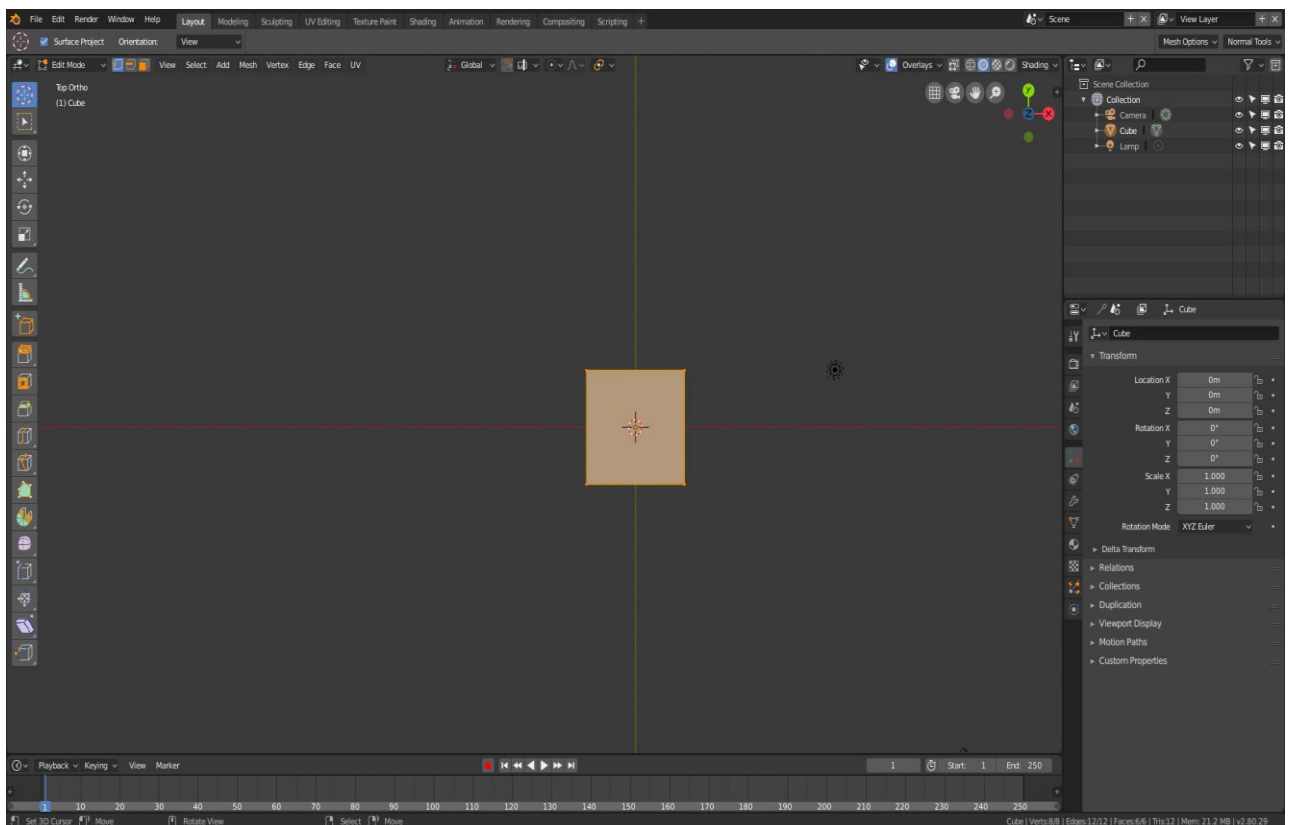


Рис. 1.12. Скриншот редактора Blender 2.80¹²

Опытные пользователи используют API Blender для написания сценариев Python, чтобы настроить приложение или создать специализированные инструменты; часто эти изменения включаются в будущие выпуски Blender.

¹² <https://bit.ly/2Wi4zeW>

Данный редактор подходит для частных лиц и небольших студий, которые используют преимущества унифицированного и гибкого процесса разработки.

Blender является кроссплатформенным и одинаково работает на компьютерах Linux, Windows и Macintosh. Его интерфейс использует OpenGL для обеспечения согласованности. Для подтверждения конкретной совместимости в списке поддерживаемых платформ указаны те, которые регулярно тестируются группой разработчиков.

Как проект, управляемый сообществом в рамках GNU General Public License (GPL), общественность уполномочена вносить различные изменения в базу кода, что приводит к появлению новых функций, исправлению ошибок и повышению удобства использования.

1.3 Техническое задание на разработку технологии

Составлено на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» [2].

1. Общие сведения.

1.1. Название организации заказчика.

Заказчик: «Уральский государственный педагогический университет».

1.2. Название технологии.

Название: «VR Interior».

1.3. Назначение технологии.

Назначение технологии состоит в визуализации интерьерных пространств, используя доступные аппаратные средства.

1.4. Категория пользователей.

Заказчики на покупку или изменение интерьерных объектов.

1.5. Плановые сроки выполнения.

Дата начала 01.09.2018, дата окончания 20.05.2019.

2. Характеристика области применения.

2.1. Процессы и структуры, в которых предполагается использование технологии.

Использование технологии предполагается в процессе ремонта или покупки квартиры.

2.2. Характеристика персонала.

В работе с технологией будет участвовать один человек с базовыми знаниями о работе с операционной системой Android.

3. Требования к технологии.

3.1. Требования к технологии в целом.

Технология должна содержать в себе описание создания трехмерной сцены интерьера и программной части для взаимодействия с сценой в разных режимах.

Трехмерная сцена должна включать в себя трехмерные объекты стен, мебели и декора, текстуры и материалы для этих объектов, освещение и виртуальную камеру.

Программная составляющая должна в себя включать интерфейс для взаимодействия с трехмерной сценой, который, в свою очередь, должен состоять из пользовательского интерфейса для выбора режима взаимодействия и для настройки параметров графики. Под режимами взаимодействия подразумеваются три контроллера, реализующие три разных способа управления параметрами виртуальной камеры.

Режим «VR» подразумевает использование очков виртуальной реальности. Вращение виртуальной камеры должно происходить за счет вращения джойстика пользователя, используя гироскоп и акселерометр. Передвижение должно происходить по направлению камеры при касании сенсорного экрана с помощью физической кнопки на очках виртуальной реальности.

В режим «3D» вращение виртуальной камеры должно быть реализовано по аналогии с режимом «VR», а передвижение должно происходить с помощью виртуального джойстика.

В режиме «3D, вид сверху» должны быть реализованы функции вращения камеры вокруг интерьера, отдаления и приближения, а также перемещение в плоскости перпендикулярной направлению камеры. Управление должно происходить за счет скольжения пальцев по сенсорному экрану пользователя.

3.2. Аппаратные требования.

3.2.1. Аппаратные требования для разработки

Для разработки необходим персональный компьютер.

3.2.2. Аппаратные требования для эксплуатации.

Для эксплуатации технологи необходимо мобильное устройство на операционной системе Android, а также очки виртуальной реальности с физической кнопкой для касания экрана мобильного устройства.

3.3. Программное обеспечение.

3.3.1. Программное обеспечение для разработки.

3.3.1.1. Системное программное обеспечение.

ОС Windows 10.

3.3.1.2. Прикладное программное обеспечение.

Unity 2017.4.26f1 Personal, Microsoft Visual Studio 2017, Blender 2.80.

3.3.2. Программное обеспечение для эксплуатации.

3.3.2.1. Системное программное обеспечение.

ОС Android не ниже версии 4.4 Kit Kat.

3.3.2.2. Прикладное программное обеспечение.

Не требуется.

3.4. Форматы входных/выходных данных, порядок их ввода в систему.

В качестве входных данных выступает информация с таких датчиков, как гироскоп, акселерометр и сенсорный экран. В качестве выходных данных – выводимое на экран устройства изображение.

3.5. Порядок взаимодействия с другими системами.

Не предусмотрен.

3.6. Меры защиты информации.

Не предусмотрены.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

Пользовательский интерфейс должен состоять из главного меню и меню настроек. Главное меню должно включать в себя кнопки для выбора режима взаимодействия, кнопку для перехода к меню настроек и кнопку для выхода из приложения. Меню настроек должно включать в себя элементы для изменения параметров графики, а также кнопки для возвращения в главное меню и выхода.

Непосредственно при взаимодействии с трехмерной сценой, интерфейс режима «VR» должен состоять из двух изображений для каждого глаза, режим «3D» предусматривает наличие графических элементов виртуального джойстика, режим «3D, вид сверху» не должен иметь каких-либо графических элементов пользовательского интерфейса.

4.2. Размещение информации на экране, макет дизайна экрана.

Макеты главного меню, меню настроек представлены на рисунках ниже (см. Рис. 1.13, Рис. 1.14). Расположение информации непосредственно при взаимодействии с трехмерной сценой должно определяться режимом. Для режима «3D» – это вывод изображения на весь экран с виртуальным джойстиком в левом нижнем углу, для режима «VR» – это стерео вывод изображений для левого и правого глаза, для режима «3D, вид сверху» – вывод изображения на весь экран без видимых элементов пользовательского интерфейса.

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

В качестве сопроводительной документации, должно быть разработано руководство пользователя.

6. Порядок сдачи приема технологии.

Технология считается принятой при его положительной оценке экспертами-преподавателями вуза.

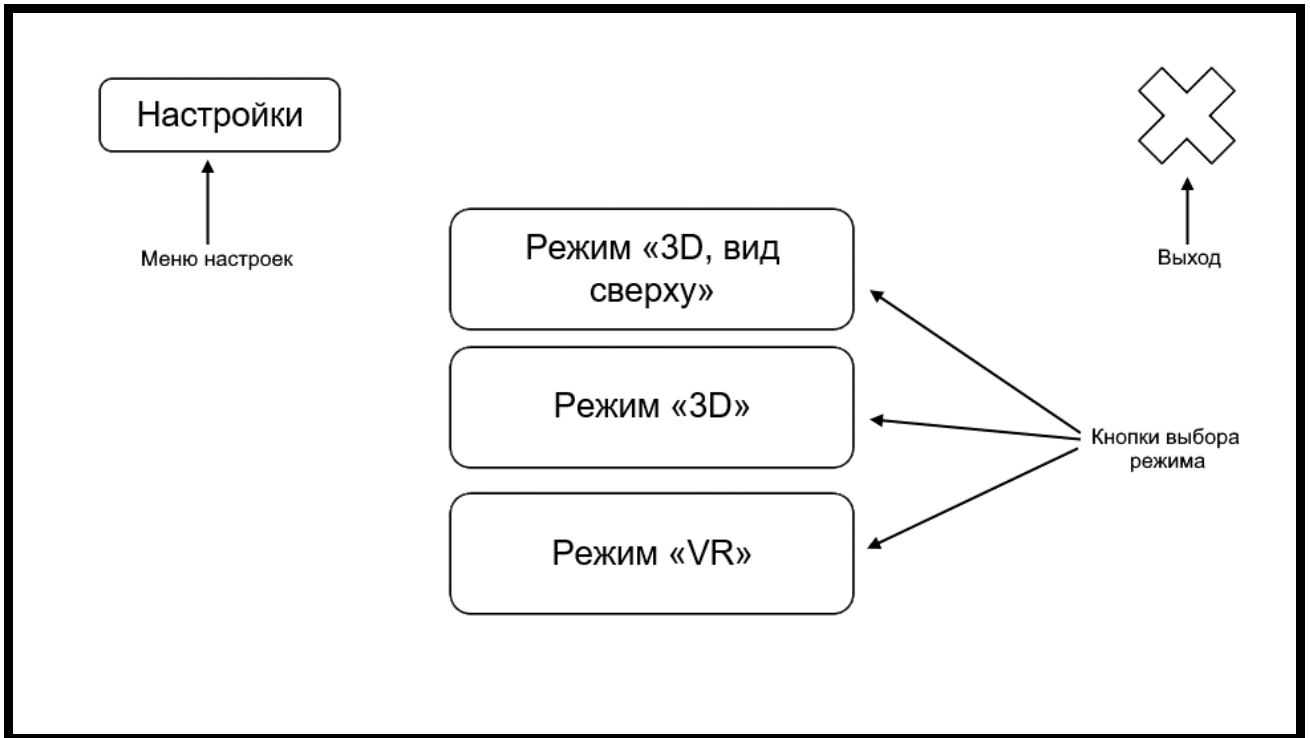


Рис. 1.13. Расположение элементов главного меню

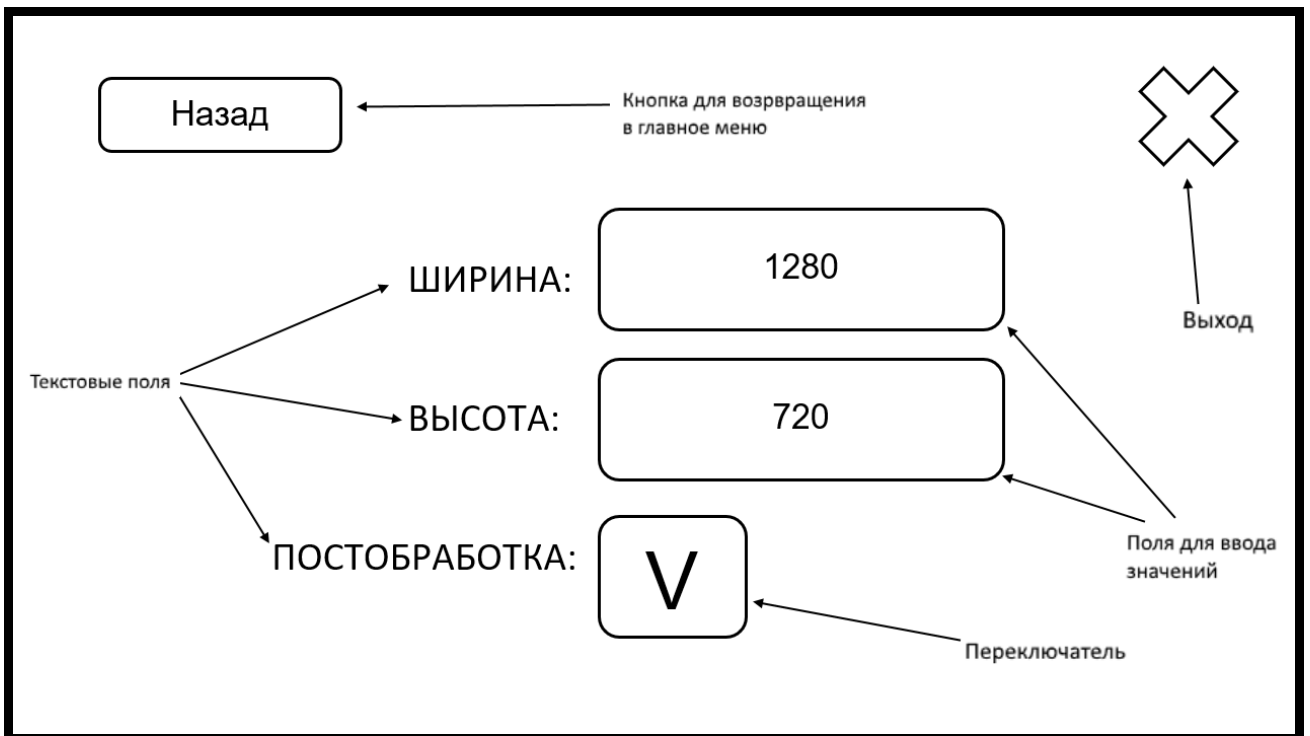


Рис. 1.14. Расположение элементов меню настроек

Глава 2. Разработка технологии

2.1 Модельные представления разрабатываемой технологии

На изображении ниже продемонстрирована последовательность технологического процесса.



Рис. 2.1. Последовательность технологического процесса

Данная последовательность демонстрирует чередование работ, выполнение которых необходимо для визуализации трехмерной сцены в виртуальной реальности.

Продолжить разработку технологии необходимо с составления неформализованного описания ее основных элементов, их характеристик и взаимодействий между ними.

Итак, исходя из технического задания, разрабатываемая технология должна состоять из двух основных частей. Первая часть – это трехмерная сцена, содержащая в себе набор архитектурных объектов и представляющая собой архитектурное внутреннее пространство отдельного помещения. Вторая часть — это программное обеспечение, реализующее взаимодействие пользователя и трехмерной сцены посредством интерфейса, использующего технологии виртуальной реальности.

Пользователю необходимы несколько способов взаимодействия с трехмерной сценой, чтобы исключить вариант, при котором он не сможет ее осмотреть из-за отсутствия необходимого приспособления (например, VR – очков). Также, у программного обеспечения, как и у трехмерной сцены необходимо максимизировать выгодные характеристики, минимизировать расходы аппаратных ресурсов и скомпилировать приложение под операционную систему Android.

На основе вышесказанного и нормативных документов [6] составлена концептуальная функциональная модель работы приложения разрабатываемой технологии (см. Рис. 2.2 и Рис. 2.3), а также разработана концептуальная структурная модель технологии (см. Рис. 2.4).

Функциональная модель показывает логические отношения между работами приложения, а структурная модель наглядно демонстрирует из каких элементов должна состоять готовая технология, определяет типы связей между этими элементами, а также содержит информацию о том, каким образом пользователь управляет трехмерной сценой.

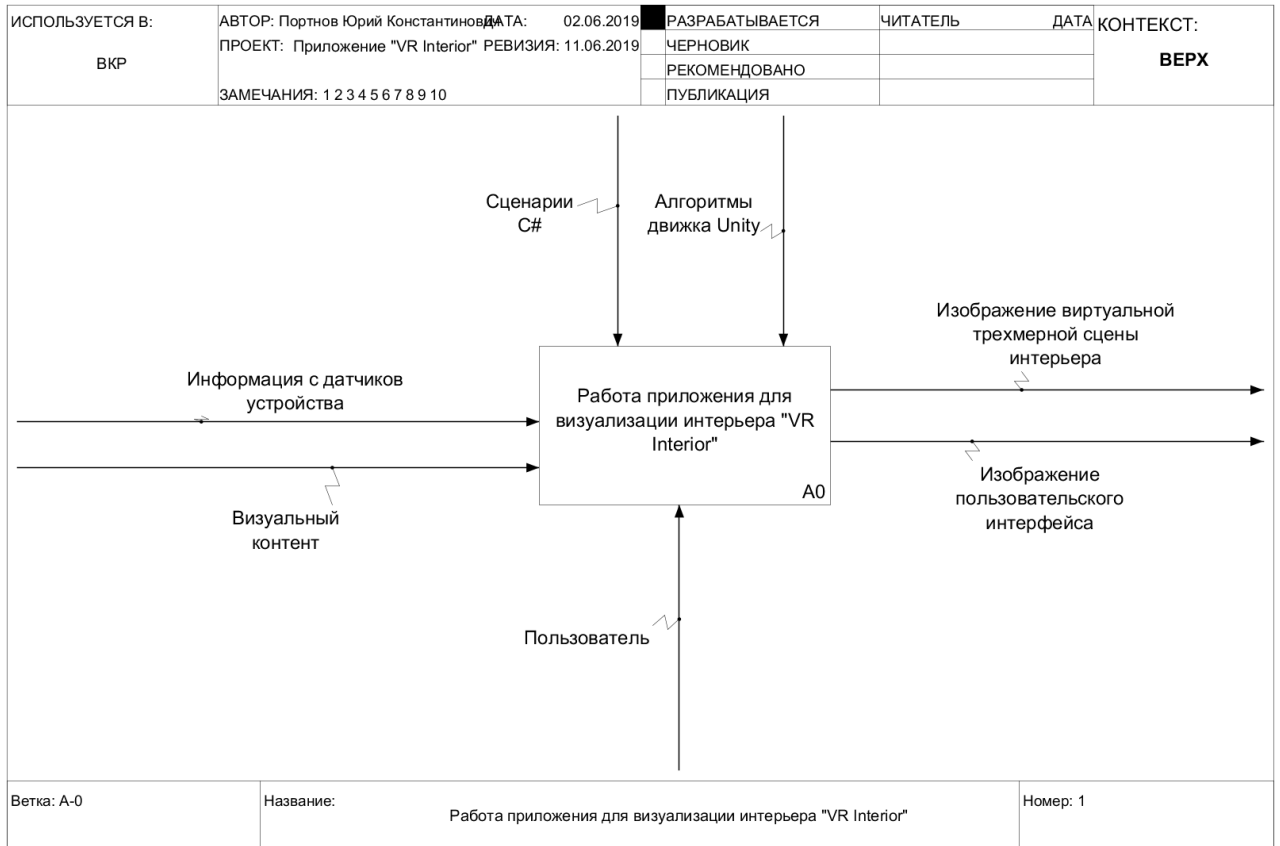


Рис. 2.2. Контекстная диаграмма работы приложения

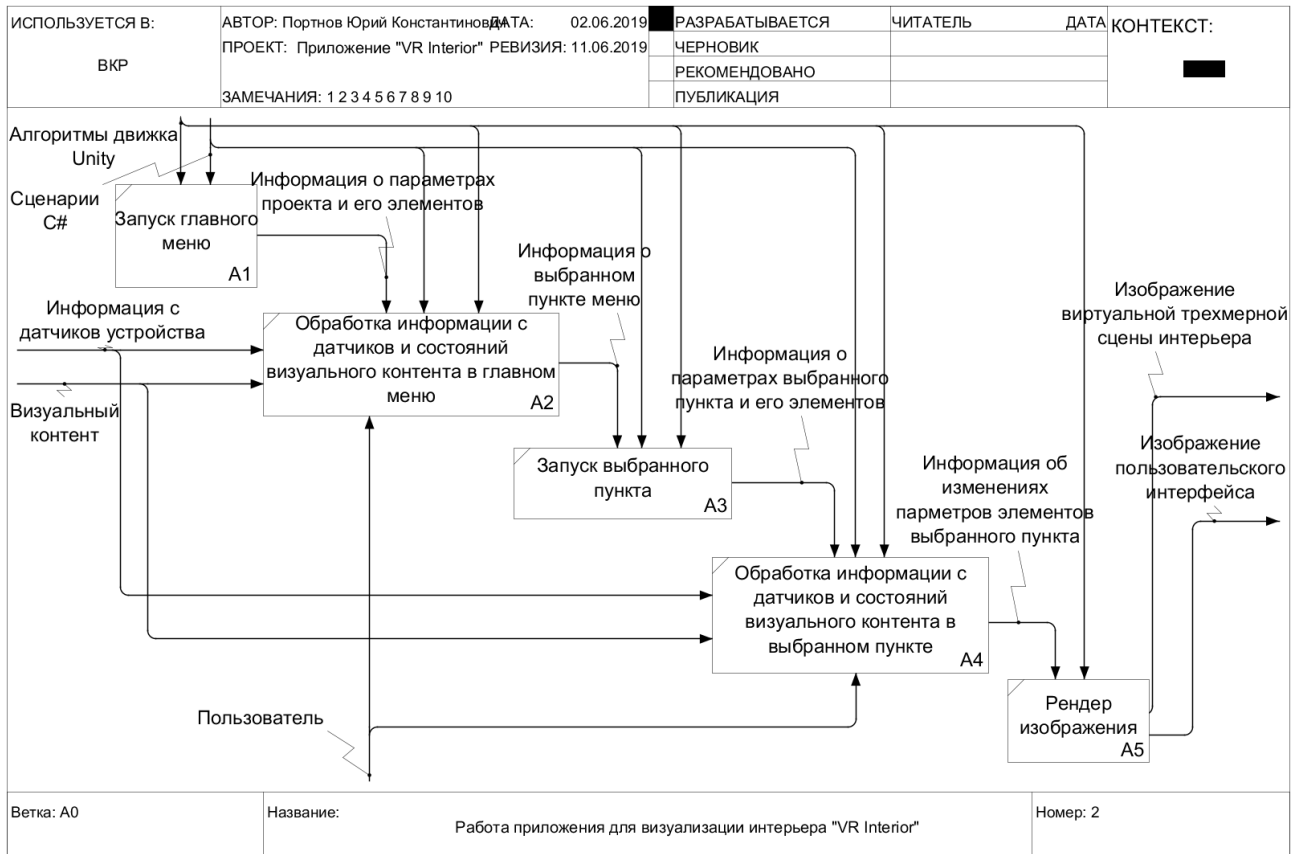


Рис. 2.3. Диаграмма декомпозиции работы приложения

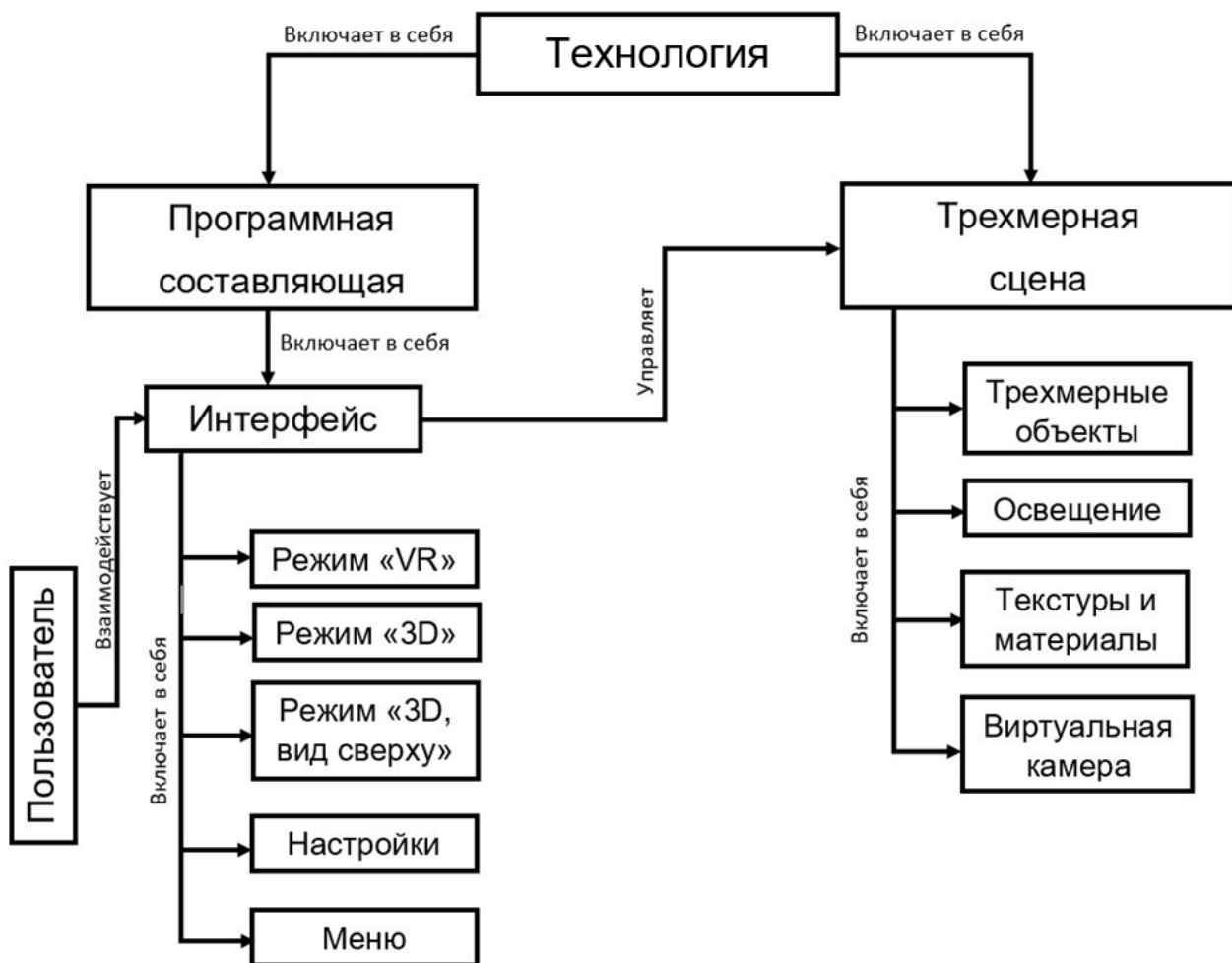


Рис. 2.4. Концептуальная структурная модель

Итак, модели показывают, что программная составляющая состоит из интерфейса, который в свою очередь содержит ряд элементов, реализующих возможность для пользователя различными способами манипулировать трехмерной сценой. Ниже представлено более подробное описание этих элементов.

Начать необходимо с описания меню, так как именно этот элемент предоставляет пользователю выбор между другими составляющими интерфейса. Схематичное представление меню показано в техническом задании (см. Рис. 1.13).

Пункт настройки позволяет пользователю устанавливать параметры, которые необходимы для оптимизации работы приложения на имеющемся устройстве. Главными параметрами, сильнее всего влияющими на аппаратные

ресурсы, являются разрешение экрана и эффекты постобработки. В техническом задании можно увидеть схематичный рисунок, на котором показано расположение элементов меню настроек (см. Рис. 1.14).

Программная часть также предусматривает несколько режимов взаимодействия пользователя и трехмерной сцены. Это необходимо для исключения невозможности использования технологии, а также для предоставления пользователю различных вариаций опыта взаимодействия.

Режим «3D, вид сверху» подразумевает под собой возможность вращения виртуальной камеры вокруг центра трехмерной сцены. Такой способ позволяет пользователю рассматривать интерьер целиком. По своей сути, данный режим является способом представления трехмерной планировки. Пример представлен на рисунке ниже (см. Рис. 2.5).



Рис. 2.5. Пример трехмерной планировки квартиры¹³

¹³ https://img.avaho.ru/upload/objects_flatplan_photo/915627_c884caa85f97972a7cfbf9c2cf4649b9.jpg

Режим «3D» необходим пользователю в том случае, если у него отсутствуют аппаратные приспособления для прогулки по сцене с использованием технологии виртуальной реальности. Данный режим подразумевает вид от первого лица, который необходим для реализации легкого эффекта присутствия. На примере ниже (см. Рис. 2.6) можно увидеть скриншот, сделанный в подобном приложении, в данном режиме интерфейс представляет из себя 2 джойстика, которые отвечают за перемещение и вращение виртуальной камеры. В данной работе для перемещения камеры будет использоваться джойстик, а для вращения, вместо джойстика будут использоваться гироскоп и акселерометр, что усилит у пользователя эффект присутствия.



Рис. 2.6. Скриншот из приложения «New Scandinavian»¹⁴

Основным режимом в данной работе является «VR». Данный способ взаимодействия пользователя и трехмерной сцены характеризуется высоким уровнем эффекта присутствия. Обеспечивается это за счет шлема виртуальной реальности, который изолирует взгляд пользователя от внешнего мира, а также

¹⁴ <https://i.ytimg.com/vi/vgIYDFEZ4dw/maxresdefault.jpg>

двух изображений, которые предназначены для левого и правого глаза. В режиме «VR» вращение виртуальной камеры реализуется с помощью гироскопа и акселерометра, а перемещение за счет касания сенсорного экрана девайса. Ниже представлен скриншот из приложения «SF Shipyard Engel Home Tour», который показывает, как выглядит интерьер в виртуальной реальности на экране телефона.



Рис. 2.7. Скриншот из приложения «SF Shipyard Engel Home Tour»¹⁵

Проект включает в себя трехмерную сцену интерьера, с которой пользователь взаимодействует с помощью интерфейса, описанного выше.

Необходимым элементом любой трехмерной сцены является виртуальная камера, от нее зависит точка и угол построения проекции, которая транслируется на экран пользователя. В данном проекте камера является основным элементом сцены, так как основной задачей интерфейса является перемещение пользователя по сцене. Соответственно, главными параметрами для манипуляций являются угол поворота и позиция виртуальной камеры в трехмерной сцене.

¹⁵ <https://www.transparenthouse.com/wp-content/uploads/2015/08/unnamed-2.jpg>

Режимы интерфейса, описанные выше, могут использоваться в различных сферах (например, компьютерные игры). Наиболее практично данные способы используются в сфере архитектурной визуализации, поэтому в качестве трехмерной сцены в данном проекте вступает интерьер квартиры.

Для корректного составления сцены необходим дизайн-проект. Для данной работы был выбран проект «ЖК Мечта студия» [3] от компании «TrioDesign». Планировка представлена на рисунке ниже (см. Рис. 2.8).



Рис. 2.8. Планировка квартиры¹⁶

¹⁶ https://st.hzcdn.com/simgs/3f21c92a0b955863_9-4149/---.jpg

На основе этой планировки можно составлять трехмерные объёмы помещений в соответствии с указанными размерами.

Также данный проект уже реализован на реальном примере и имеет фотографии, которые содержат информацию о дизайнерских решениях. На фотографии ниже изображен общий план квартиры (см. Рис. 2.9). На его основе можно наполнять сцену трехмерными моделями мебели, настраивать материалы и освещение.



Рис. 2.9. Общий план интерьера¹⁷

2.2 Описание процесса разработки технологии

Разработка данной технологии включает в себя несколько основных этапов:

- 1) создание unity-проекта и его настройка под операционную систему Android;

¹⁷ https://st.hzcdn.com/simgs/38218dbc094405f0_4-0800/sovremennaya-klassika-gostinaya.jpg

- 2) создание главной сцены и разработка основных элементов управления;
- 3) создание сцены главного меню и меню настроек;
- 4) наполнение сцены трехмерными моделями, текстурами и материалами;
- 5) настройка освещения;
- 6) настройка эффектов постобработки;
- 7) оптимизация и компиляция.

Создание Unity-проекта начинается с Unity-Hub – центра управления Unity-проектами. При создании указывается название проекта, расположение, в которое он компилируется, а также шаблон, отвечающий за измерение проекта. На скриншоте ниже (см. Рис. 2.10) изображено окно создания Unity-проекта с данными разрабатываемого проекта.

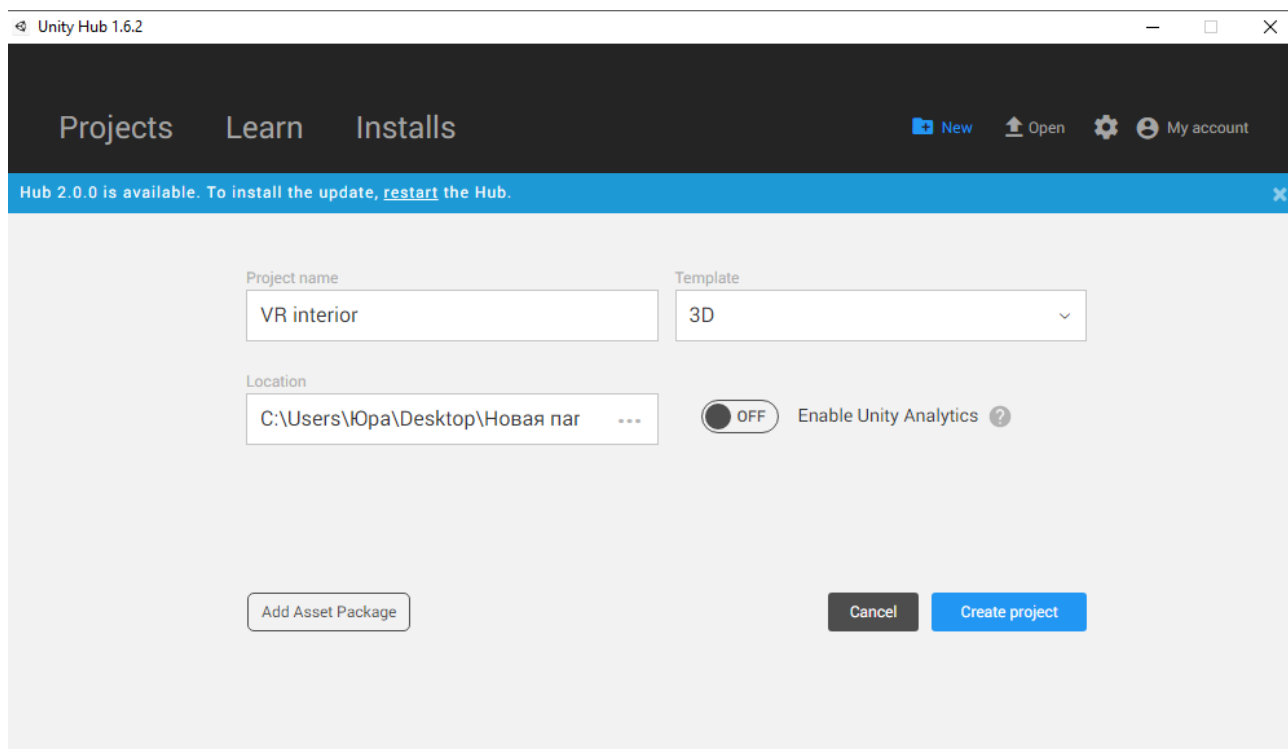


Рис. 2.10. Создание Unity-проекта

Далее произведены настройки созданного проекта, направленные на возможность компиляции под операционную систему Android. В настройках проекта изменена платформа, подключена возможность работы с виртуальной ре-

альностью, заполнены идентификационные данные, выбраны необходимый для работы с виртуальной реальностью минимальный уровень Android API, а также изменена система для построения арк-файла с Gradle на Internal в связи с ее большей стабильностью. На изображении ниже (см. Рис. 2.11) продемонстрированы все значения, установленные в ходе настройки проекта.

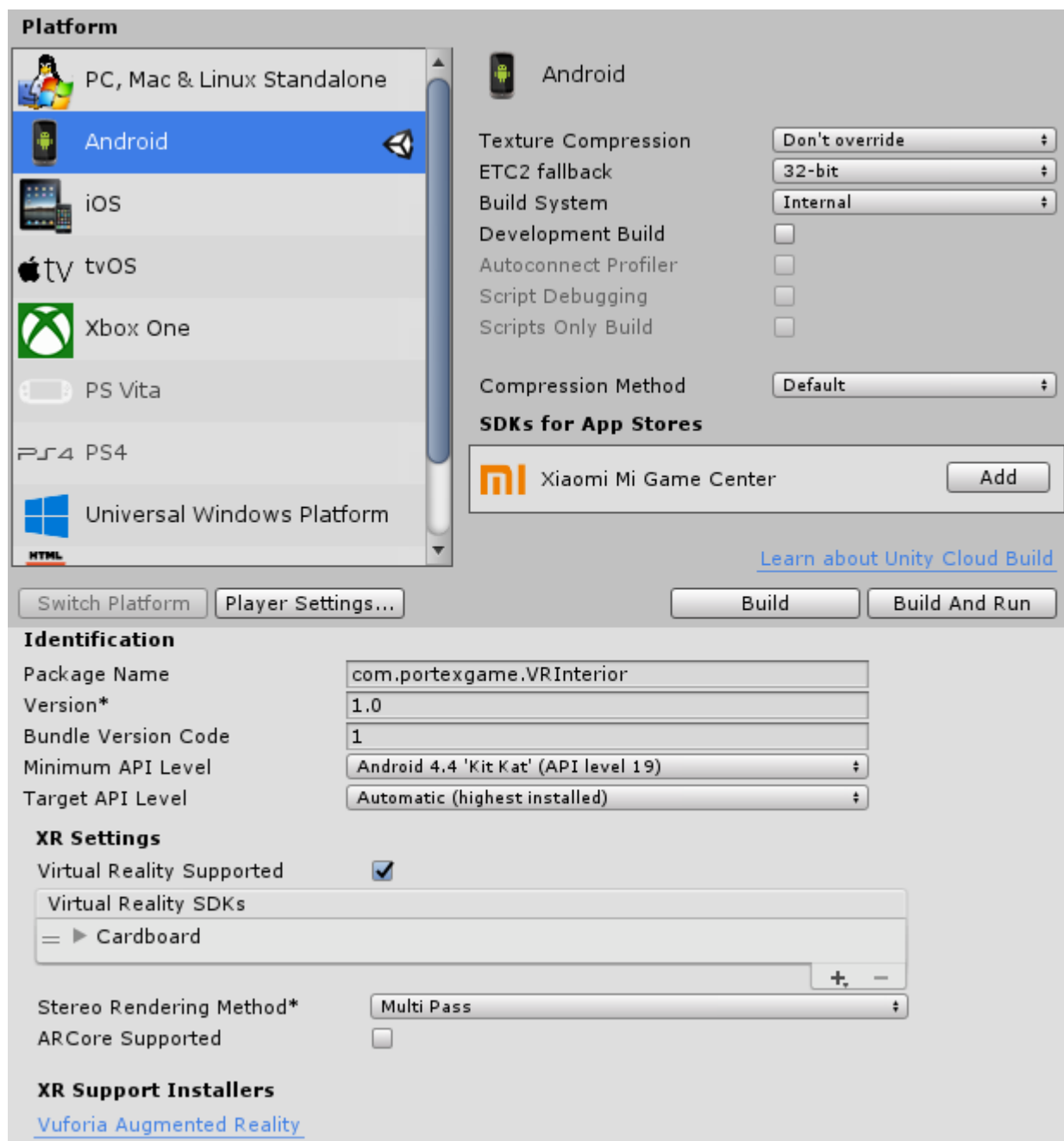


Рис. 2.11. Скриншоты настроек Unity-проекта

Следующим после настройки шагом стало создание главной сцены, которая должна содержать в себе все элементы трехмерной графики, в том числе и

объект «Player» класса «GameObject», отвечающий за все манипуляции с виртуальной камерой. Данному объекту добавлены в качестве компонентов все классы, реализующие различные режимы управления камерой.

Перед написанием всех классов-контроллеров в проект импортирован набор средств разработки «Google VR», а в сцену из данного пакета добавлен объект с присвоенным ему классом «GvrEditorEmulator», который эмулирует вращение девайса, но используя при этом клавиатуру и мышь. Это позволяет тестировать режимы «VR» и «3D» прямо в редакторе, без необходимости каждый раз компилировать проект.

При создании сцены по умолчанию в ней находится камера, но для взаимодействия с ее позицией был создан объект «Player» класса «GameObject». К данному объекту добавлен компонент «Box Collider», имитирующий размеры человека и призванный обнаруживать пересечения с коллайдерами других объектов в сцене. Также объекту добавлен компонент «Rigidbody», который получает информацию от коллайдера и управляет положением объекта через имитацию физики. Камера сделана дочерним объектом по отношению к «Player». Таким образом, изменяя позицию родительского объекта, изменяется позиция камеры, но при этом вращения камеры не отражаются на параметрах вышестоящего объекта. На скриншоте ниже (см. Рис. 2.12) можно увидеть параметры компонентов объекта «Player».

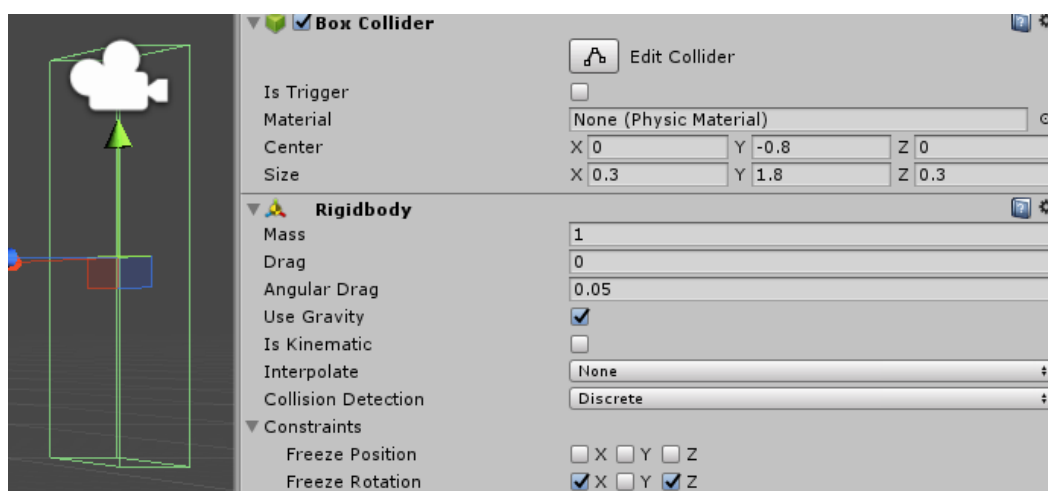


Рис. 2.12. Скриншот с параметрами компонентов объекта «Player»

После выполнения всех шагов, описанных выше, можно приступать к написанию контроллеров, отвечающих за различные режимы. Первым создан класс «VRController», отвечающий за управление главным объектом в режиме «VR». В методе «Update ()» данного класса расположен алгоритм, включающий в себя получение вектора оси Z. Эта ось указывает в ту сторону, в которую направлена виртуальная камера. Ось Y полученного вектора приравняется к нулю, чтобы объект не перемещался вверх или вниз. Далее реализован обработчик нажатия на экран девайса, при истинности которого происходит сложение текущей позиции объекта с умножением полученного вектора на заранее объявленную переменную скорости и время в секундах, которое потребовалось для отрисовки последнего кадра. На скриншоте ниже (см. Рис. 2.13) показано то, как выглядит интерфейс данного режима на мобильном устройстве.



Рис. 2.13. Скриншот интерфейса режима «VR»

Далее был реализован контроллер, отвечающий за управление в режиме «3D». Для этого из официального магазина «Unity Asset Store» импортирован пакет «Simple Touch Controller», содержащий инструменты, управляющие вводом с использованием виртуальных джойстиков. В сцену добавлен объект

«Canvas», который представляет собой абстрактное пространство, в котором производится настройка и отрисовка пользовательского интерфейса. К этому объекту, из импортированного пакета в качестве дочернего был добавлен объект «SimpleTouch Joystick», который содержит графические элементы интерфейса, а также класс «SimpleTouchController», содержащий метод «GetTouchPosition ()», возвращающий значения, вводимые пользователем при использовании виртуального джойстика. Следующим шагом создан класс «JoystickController», содержащие две глобальные переменные. Первая типа «Float» для изменения скорости непосредственно в редакторе, а вторая переменная типа «SimpleTouchController», которой присваивалась ссылка на объект из сцены. В методе «Update ()» алгоритм перемещения реализован по аналогии с передвижением в режиме «VR», с добавлением в него вектора оси X, отвечающего за передвижение в стороны. Также данный режим подразумевает отключение настроек «XRSettings», которые помимо всего прочего контролируют управление с помощью гироскопа и акселерометра. Чтобы решить данную проблему, за поворот камеры с использованием этих датчиков отвечает глобальный статичный метод «InputTracking.GetLocalRotation (XRNode.CenterEye)». На скриншоте ниже изображен интерфейс режима «3D» на мобильном устройстве.



Рис. 2.14. Скриншот интерфейса режима «3D»

Следующим шагом стала реализация контроллера для режима «3D, вид сверху». Для этого создан класс «OrbitViewController». В данном классе содержится глобальная переменная класса «Transform», отвечающая за цель от значений позиции которой зависят дальнейшие изменения параметров объекта «Player». Также глобальными являются переменные, отвечающие за смещение позиции относительно цели, за минимальную и максимальную дистанцию управляемого объекта от цели, за скорость вращения по осям X и Y, за минимальный и максимальный угол вращения по оси X, а также за скорость приближения и движения в плоскости перпендикулярной направлению камеры.

Из класса «MonoBehaviour» переопределены методы «Start ()» и метод «Update ()». В первом методе изменяемые переменные принимают значения параметров объекта «Player», высчитывается дистанция от этого объекта до цели, а также задаются значения поворота. Во втором методе реализованы три обработчика ввода, содержащие алгоритмы, отвечающие за приближение к цели, вращение вокруг нее и движение в плоскости перпендикулярной направлению камеры.

Алгоритм приближения и отдаления строится на основе высчитывания разности длин векторов между двумя пальцами на экране в предыдущее обновление и текущее.

Алгоритм вращения строится на основе расстояния, на которое сместился палец за предыдущее обновление. Также данный алгоритм предусматривает ограничение вращения по оси X с помощью метода «ClampAngle (float angle, float min, float max)». Данный метод реализован с использованием глобальной статической функции «Clamp(float value, float min, float max)» класса «Mathf», которая ограничивает значение между минимальным и максимальным.

Движение в плоскости перпендикулярной направлению камеры выполняется на основе высчитывания разницы между положением при первом касании экрана двумя пальцами и текущим положением пальцев.

Данный режим не имеет видимых элементов пользовательского интерфейса, но отличается от других двух видом на сцену сверху. На скриншоте ниже (см. Рис. 2.15) изображен интерфейс режима «3D, вид сверху»



Рис. 2.15. Скриншот интерфейса режима «3D, вид сверху»

После реализации всех классов-контроллеров, необходимо добавить их в качестве компонентов к объекту «Player». Следом, непосредственно в редакторе, стали доступны глобальные переменные. Класс «VRController» имеет одну глобальную переменную, отвечающую за скорость передвижения объекта, данной переменной присвоено значение 1.3, являющееся наиболее оптимальным для перемещения по небольшой интерьерной сцене. Класс «JoystickController» помимо переменной, отвечающей за скорость, имеет глобальную переменную класса «SimpleTouchController», ей присвоена ссылка на объект, который заранее был добавлен в сцену. Класс «OrbitViewController» содержит одиннадцать глобальных переменных, отвечающих за различные параметры. В ходе тестирования данного контроллера, этим переменным заданы наиболее оптимальные для управления объектом значения. На скриншоте ниже (см. Рис. 2.16) проде-

монстрированы все значения параметров компонентов-контроллеров объекта «Player».

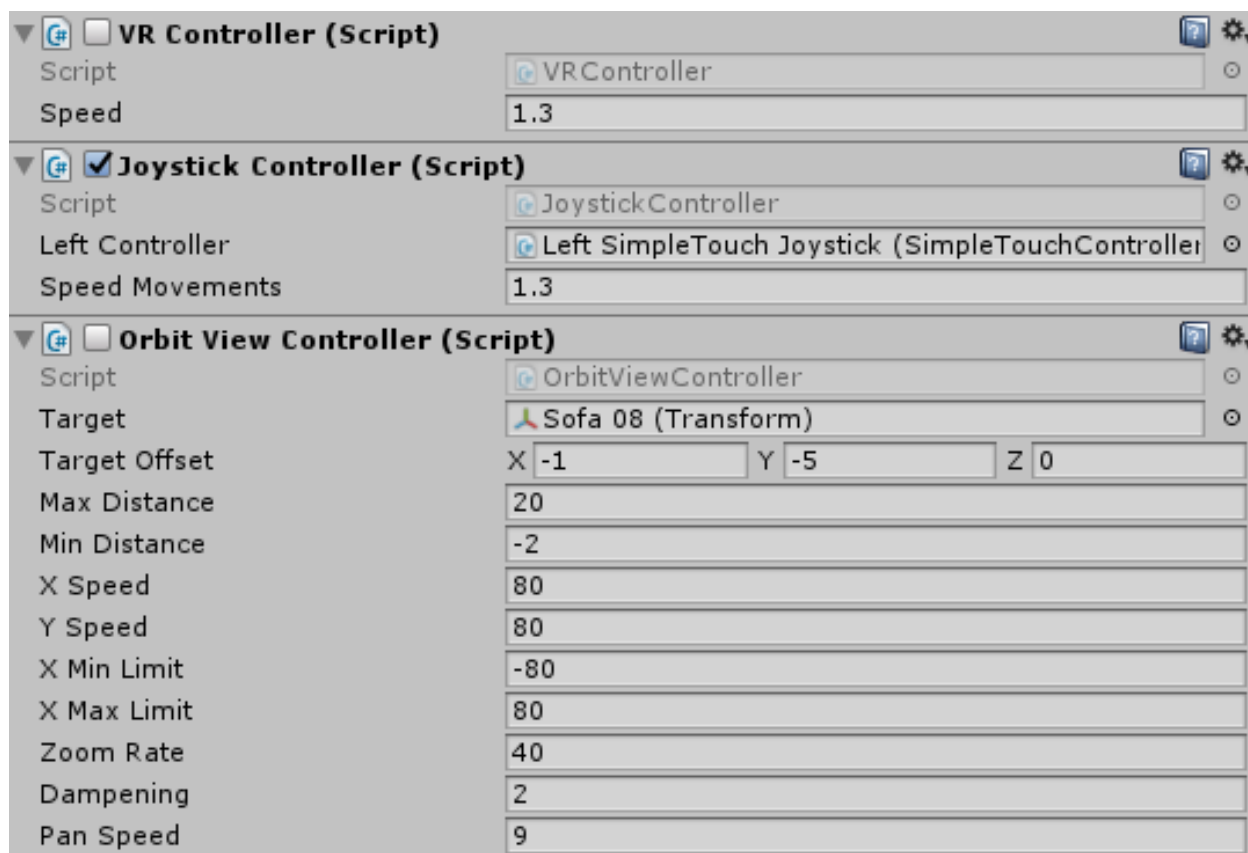


Рис. 2.16. Скриншот значений параметров контроллеров объекта «Player»

После реализации и настройки всех классов-контроллеров была создана сцена «MenuScene», содержащая в себе элементы графического интерфейса, отвечающие за выбор режима взаимодействия с главной сценой. Также данная сцена содержит меню настроек, элементы которого предназначены для корректирования параметров графики.

Прежде всего для настройки и отрисовки пользовательского интерфейса в сцену был добавлен объект «Canvas». В качестве дочерних были добавлены объекты класса «GameObject», а именно «MainMenu», содержащий в качестве дочерних элементы пользовательского интерфейса основного меню, и объект «SettingsMenu», содержащий элементы меню настроек. Также «Canvas» содержит объект «background» с компонентом, содержащим класс «Image». Этот объект отвечает за отрисовку заднего фона.

Объекту «MainMenu» были добавлены пять объектов-кнопок и расположены в соответствии со схемой «Расположение элементов главного меню» (см. Рис. 1.13). Объекты «VR_button», «3D_button» и «ORB_button» отвечают за выбор режима взаимодействия с главной сценой. Объект «SET_button» предназначен для перехода в меню настроек, а «Quit_button» для выхода из приложения. Для реализации функционала этих кнопок создан класс «MainMenuScript» и добавлен в качестве компонента к объекту «MainMenu». После этого на обработчики нажатий кнопок назначены соответствующие методы из этого класса. Алгоритм методов, предназначенных для выбора режима, заключается в записи в хранилище предпочтений значения по ключу «SelectedMode», которое в дальнейшем будет использовано при загрузке главной сцены. Действия при нажатии кнопки «SET_button» назначены непосредственно в редакторе и заключаются в деактивации объекта «MainMenu» и активации объекта «SettingsMenu». На обработчик кнопки «Quit_button» назначен метод «QuitApp ()», в котором вызывается метод «Quit ()» класса «Application», отвечающий за прекращение работы приложения. На скриншоте ниже (см. Рис. 2.17) изображен пользовательский интерфейс главного меню приложения.



Рис. 2.17. Скриншот интерфейса главного меню приложения.

В качестве дочерних к объекту «SettingsMenu» были добавлены такие элементы пользовательского интерфейса, как «BACK_button» и «Quit_button» – кнопки, отвечающие за возвращение в меню настроек и выход из приложения соответственно. Добавлены объекты «Width_input» и «Height_input» – поля ввода для изменения ширины и высоты. Добавлен «Toggle» – переключатель для эффектов постобработки. Также добавлены текстовые поля «Width_txt», «Height_txt» и «PostProcess_txt» для подписи элементов. Все объекты меню настроек расположены в соответствии со схемой «Расположение элементов меню настроек» (см. Рис. 1.14). Функционал элементов меню настроек реализован в классе «SettingsMenuScript», который добавлен в качестве компонента к объекту «SettingsMenu». На обработчик кнопки «BACK_button» из класса добавлен метод «SetResolution ()», который считывает значение переключателя постобработки и записывает в хранилище предпочтений значение по ключу «PPE», а также с помощью метода «SetResolution(int width, int height, bool fullscreen)» класса «Screen» устанавливает значения из соответствующих полей ввода. Переход к главному меню реализован по аналогии с кнопкой «SET_button», но также класс «SettingsMenuScript» переопределяет метод «Update ()», в котором реализован обработчик аппаратной кнопки Back для возвращения в главное меню. Кнопка «Quit_button» аналогична одноименной кнопке из главного меню. На скриншоте ниже (см. Рис. 2.18) изображен пользовательский интерфейс меню настроек приложения.

Для применения настроек заданных в сцене «MenuScene», в сцене «MainScene» был создан объект «SceneLoader», которому в качестве компонента назначен класс «SceneLoaderScript», содержащий методы, отвечающие за включение и выключение различных элементов сцены в зависимости от выбранного режима в главном меню и заданных параметров в меню настроек. Все эти данные заранее записываются в хранилище предпочтений пользователя с помощью метода «SetString(string key, string value)» класса «PlayerPrefs», а метод «GetString(string key, string value)» возвращает записанные значения.

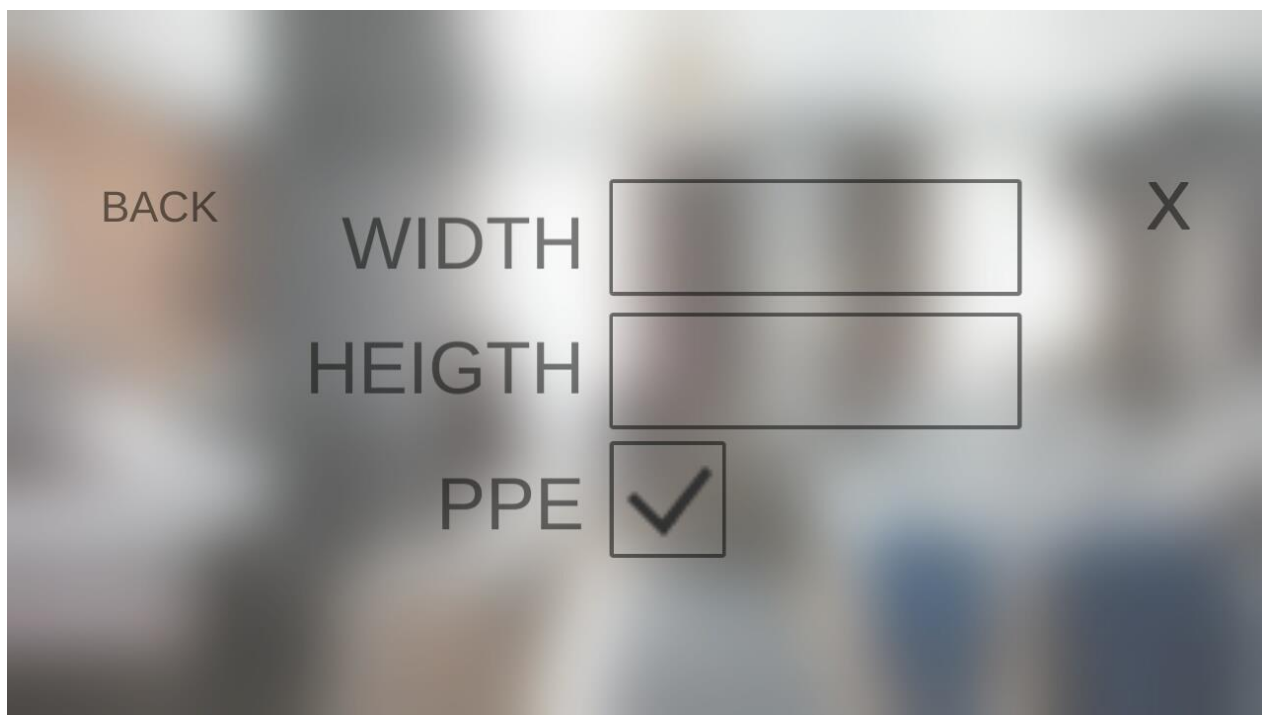


Рис. 2.18. Скриншот интерфейса меню настроек приложения

Класс «SceneLoaderScrip» содержит метод `OnBeforeSceneLoadRuntimeMethod()`, который вызывается при запуске приложения и отвечает за выключение настроек «XRSettings». В переопределенном методе «Awake ()» вызывается метод «startSelectedMode(string SelectedMode, string isPPEon)», в который в качестве параметров передаются значения из хранилища предпочтений пользователя, далее в методе с помощью условных операторов включаются или выключаются эффекты постобработки ип вызывается один из методов: «enableVRMode()», «enable3DMode()» или «enableOrbMode ()», в которых и происходит включение или отключение различных элементов сцены. Также этот класс переопределяет метод «OnSceneLoaded(Scene scene, LoadSceneMode mode)», в котором выполняется отключение настроек «XRSettings» при загрузке сцены «MenuScene». В переопределенном методе «Update()» реализован обработчик аппаратной кнопки Back, по нажатию на которую загружается сцена меню. Методы `LoadDevice(string newDevice, bool enable)` и `EnableVR()` отвечают за включение настроек «XRSettings», которые необходимы для режима «VR».

Следующим шагом, после реализации программной составляющей технологии, стало создание трехмерных элементов сцены.

Сперва в редакторе трехмерной графики Blender были смоделированы стены, с сохранением размеров и пропорций планировки (см. Рис. 2.8) из проекта «ЖК Мечта студия» [3] от компании «TrioDesign». Созданы дверные и оконные проемы, а также плинтуса и карнизы по периметру комнаты. Определенным полигонам назначены слоты для материалов. Данные модели были экспортированы в формате FBX и импортированы в unity-проект. На скриншоте ниже (см. Рис. 2.19) изображено созданное помещение в интерфейсе редактора Blender.

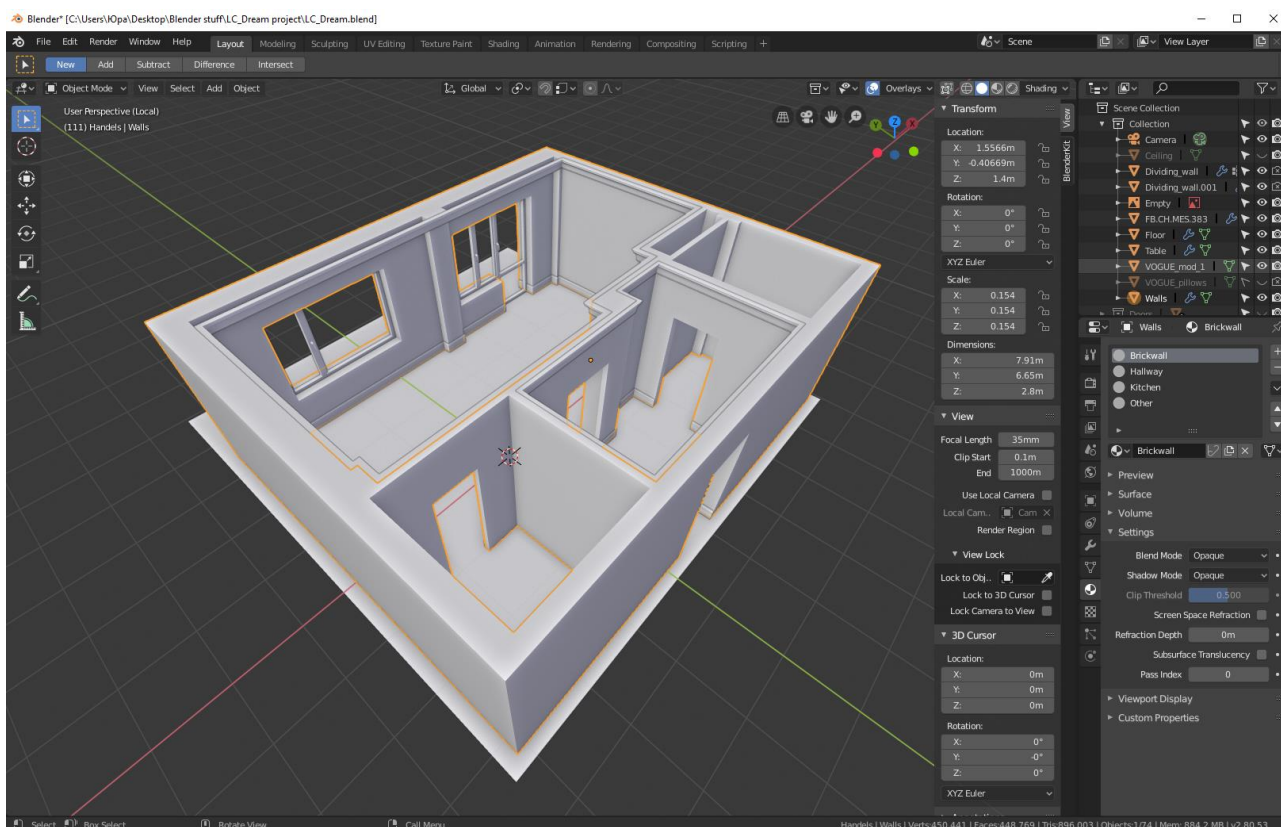


Рис. 2.19. Трехмерная модель стен в редакторе Blender

После, из официального магазина «Unity Asset Store» импортированы пакеты «ArchVizPRO Interior Vol.1», «ArchVizPRO Interior Vol.2» и «House Furniture Pack», содержащие трехмерные объекты мебели и декора, а также текстуры и материалы для них. Из пакетов в сцену добавлялись объекты ориентируясь на планировку (см. Рис. 2.8) и общий план интерьера (см. Рис. 2.9). На

скриншоте ниже (см. Рис. 2.20) изображены объекты, добавленные в сцену из импортированных пакетов.

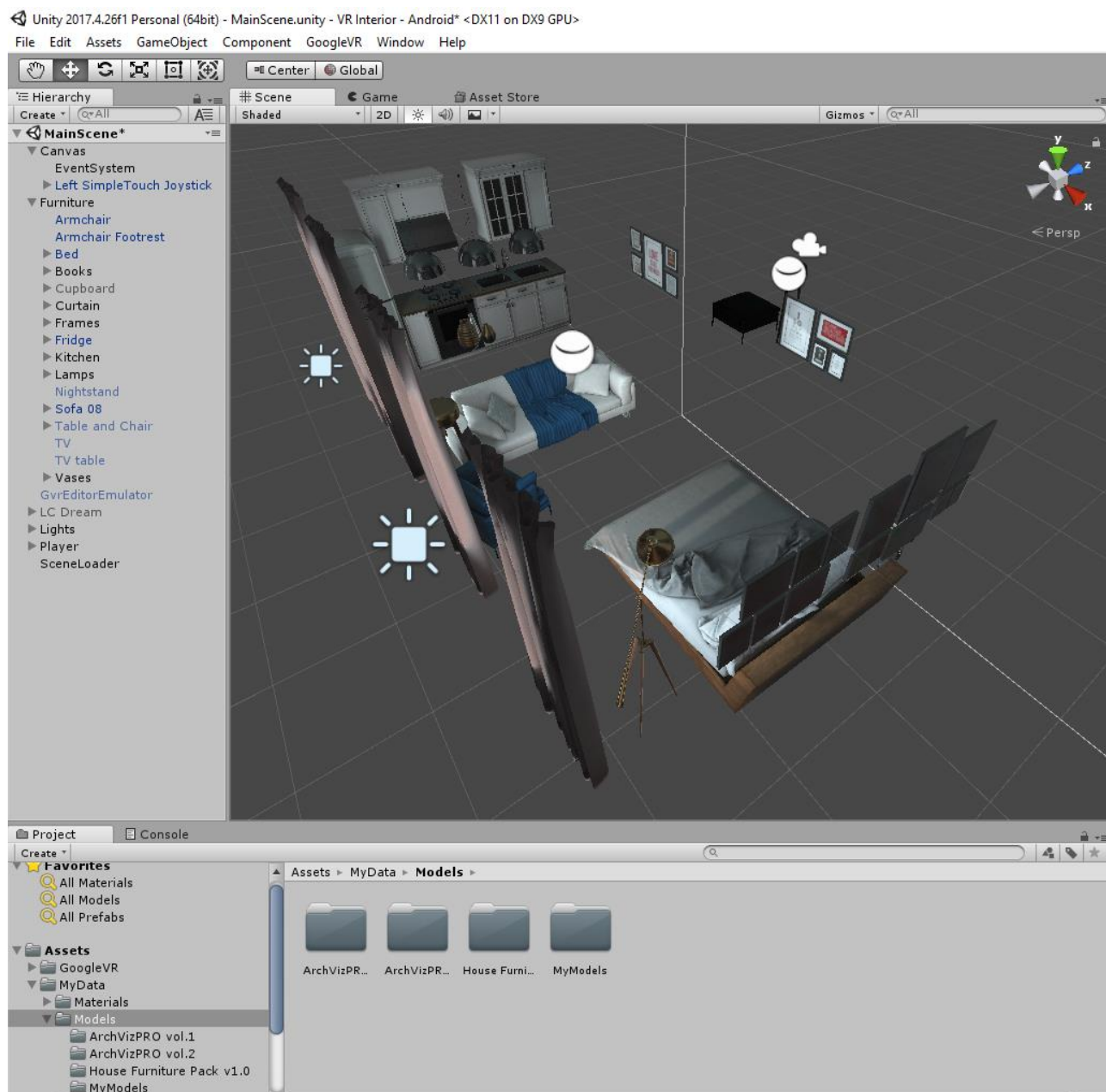


Рис. 2.20. Скриншот объектов, импортированных из пакетов

Далее в редакторе Blender были созданы геометрия недостающих трехмерных объектов интерьера, построена UV-развертка для корректного наложения текстур и назначены слоты для материалов. После чего объекты экспортированы в формате FBX и импортированы в unity-проект. Затем, ориентируясь на планировку (см. Рис. 2.8), объекты добавлены в сцену. Материалы создавались непосредственно в unity-редакторе. Для этого, с сайта «Poliigon» [12] ска-

чаны текстурные карты, импортированы в проект и добавлены в соответствующие ячейки материала. После этого материалы присвоены в качестве компонентов к трехмерным объектам. На скриншоте ниже (см. Рис. 2.21) изображены импортированные в сцену объекты в интерфейсе редактора Blender.

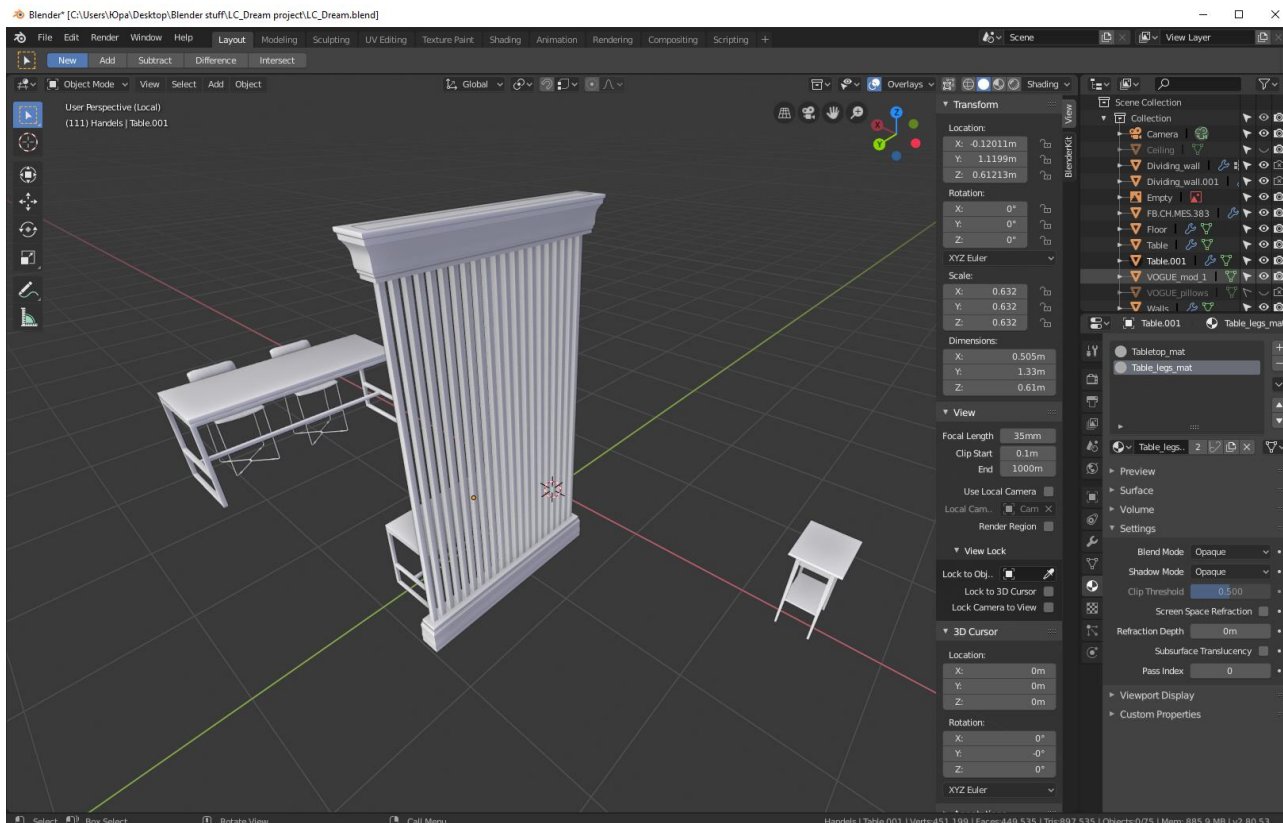


Рис. 2.21. Трехмерные объекты мебели в редакторе Blender

После размещения всех трехмерных объектов мебели и назначения им материалов можно настраивать освещение. В сцене были использованы два типа источников света: «Directional light» – направленный свет, имитирующий свет от солнца и «Area light» – имитирующий рассеянный свет от неба. Также в сцену добавлены объекты «Reflection probe», которые фиксируют сферическую картину своего окружения во всех направлениях, затем захваченное изображение сохраняется в виде кубической карты, которая может использоваться объектами с отражающими материалами. После настройки источников освещения можно приступить к процессу запекания света в текстуру, которая наложится на объекты. На скриншоте ниже (см. Рис. 2.22) можно увидеть настройки, отвечающие за отрисовку карт света.

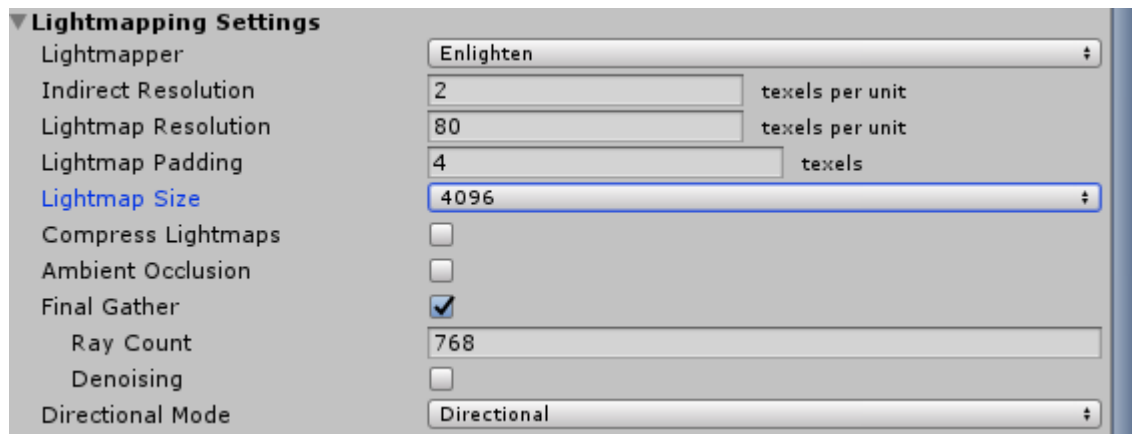


Рис. 2.22. Скриншот настроек запекания карт света

Следующим шагом стала реализация эффектов постобработки. Для этого из официального магазина «Unity Asset Store» импортирован пакет «Post Processing Stack», содержащий необходимые инструменты. После этого в редакторе стала доступна возможность создания профиля постобработки. Из пакета к объекту «MainCamera» в качестве компонента добавлен скрипт с классом «PostProcessingBehaviour». А профиль постобработки передан классу в качестве значения глобальной переменной «Profile». В профиле активированы и настроены такие эффекты, как сглаживание, затенение и цветокоррекция. На скриншотах ниже (см. Рис. 2.23) изображена сцена с до и после эффектов постобработки.

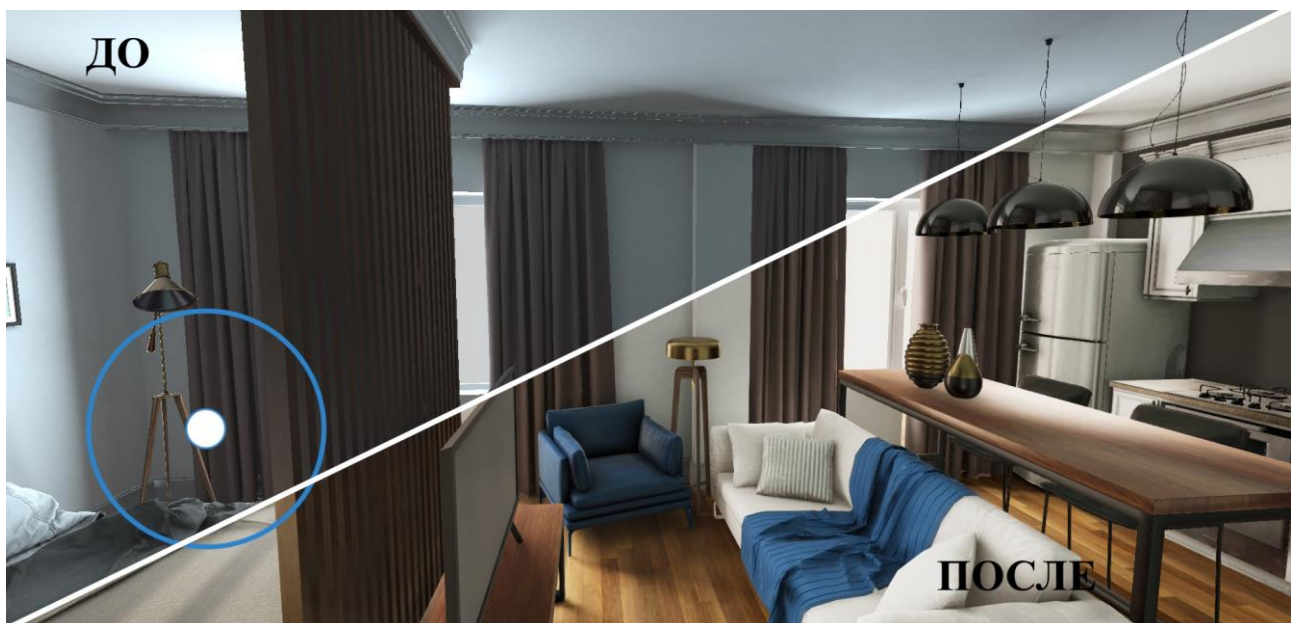


Рис. 2.23. Скриншоты до и после постобработки

Перед компиляцией проекта в арк-файл необходимо произвести его оптимизацию под мобильные девайсы. Некоторые из пунктов произведены в процессе реализации проекта, например, запекание всех источников света, понижение настроек разрешения объектов «Reflection probe», импортированные трехмерные объекты мебели являются низкополигональными, а также реализована возможность понижать разрешение экрана и отключать эффекты постобработки. Непосредственно перед компиляцией произведены настройки качества, значения параметров которых можно увидеть на скриншоте ниже (см. Рис. 2.24)

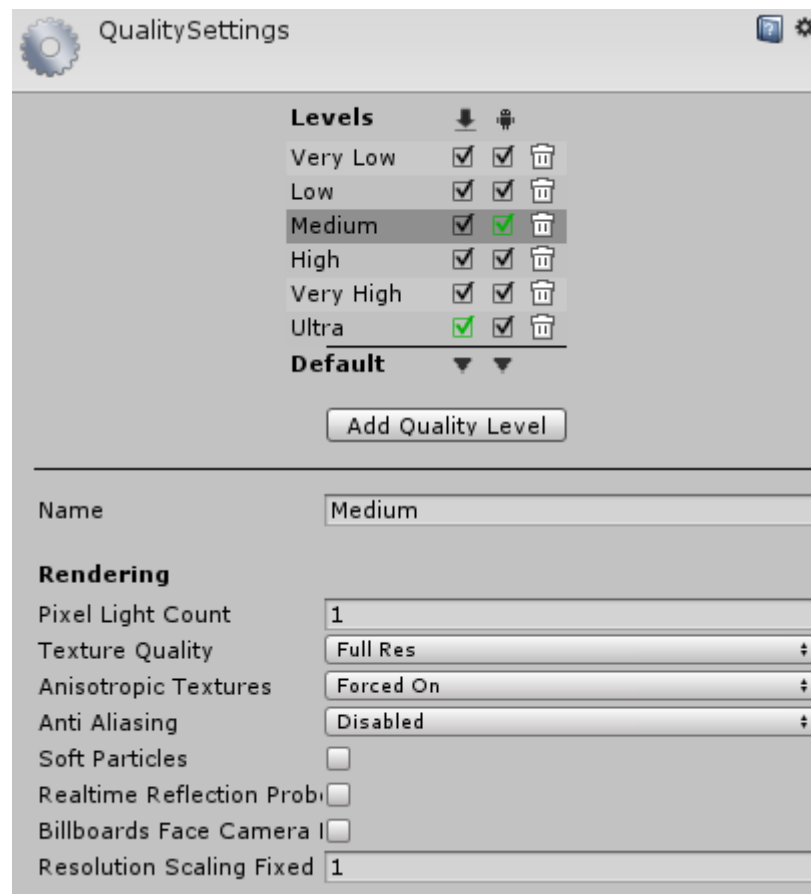


Рис. 2.24. Скриншот настроек качества

После того, как были внесены корректировки во все настройки, отвечающие за максимизацию выгодных характеристик и минимизацию расходов аппаратных ресурсов, был скомпилирован арк-файл, который можно использовать для установки приложения на девайсах с операционной системой Android.

2.3 Руководство пользователя

В соответствии с руководящими стандартами для создания руководства пользователя [1], были выделены следующие основные разделы руководства пользователя:

- назначение приложения;
- условия применения приложения;
- подготовка приложения к работе;
- описание операций;
- аварийные ситуации.

Назначение приложения состоит в визуализации интерьерных пространств, используя доступные аппаратные средства. Такой подход помогает пользователю более точно оценить предлагаемый ему интерьерный объект.

Условием применения является наличие мобильного девайса на операционной системе Android версией не ниже Android 4.4 Kit Kat. Пользователь должен обладать навыками работы с операционной системы Android.

Для начала работы требуется скачать на девайс и установить арк-файл, после чего приложение готово к работе.



Рис. 2.25. Скриншот главного меню и функции его элементов

При запуске приложения пользователь попадает в главное меню, отвечающее за выбор режима, переход в меню настроек или выход из приложения. На скриншоте выше (см. Рис. 2.25) указаны кнопки главного меню и подписаны соответствующие им функции.

Кроме главного меню, в приложении есть меню настроек, отвечающее за изменение разрешения экрана и включение или отключение эффектов постобработки. На скриншоте ниже (см. Рис. 2.26) изображено меню настроек с подписанными функциями его элементов. Также, можно вернуться в главное меню, используя аппаратную кнопку Back.

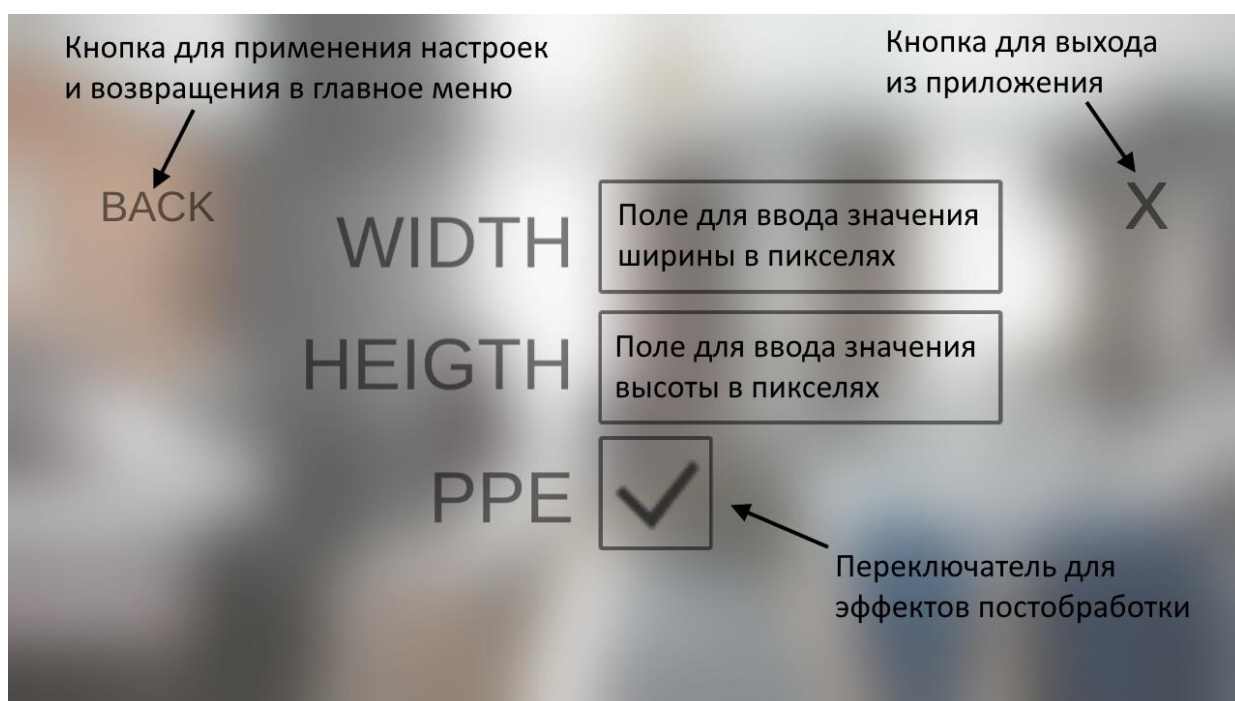


Рис. 2.26. Скриншот меню настроек и функции его элементов

При выборе в главном меню режима проходит некоторое время, прежде чем загрузится главная сцена. Для возвращения в главное меню из любого режима необходимо нажать аппаратную кнопку Back.

Для режима «VR» требуются очки виртуальной реальности с физической кнопкой, отвечающей за касание экрана девайса. В эти очки помещается устройство пользователя, вращение виртуальной камеры синхронизировано с вращением девайса, передвижение происходит вперед относительно направления камеры, если зажата кнопка для касания экрана устройства.

Режим «3D» не требует дополнительного оборудования. Вращение виртуальной камеры синхронизировано с вращением девайса, передвижение происходит за счет виртуального джойстика (см. Рис. 2.27), который реализует возможность движения в направлениях вперед – назад и влево-вправо. Для этого необходимо зажать джойстик и перемещать его в соответствующую сторону, учитывая при этом направление камеры.

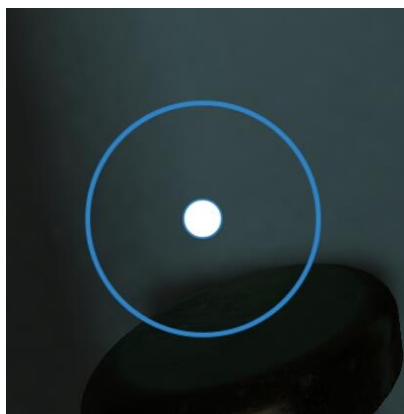


Рис. 2.27. Скриншот виртуального джойстика

Перемещение и вращение виртуальной камеры в режиме «3D, вид сверху» происходит с помощью сенсорного экрана. Для вращения необходимо скользить по экрану одним пальцем. Для передвижения в плоскости перпендикулярной направлению камеры необходимо скользить по экрану параллельно двумя пальцами. Для приближения и отдаления необходимо разводить и сводить два пальца на экране, соответственно.

При отказе или сбое в работе приложения необходимо его перезапустить. В случае, если перезапуск не решил проблему, приложение необходимо удалить и установить заново.

Заключение

В ходе проделанной работы, с помощью игрового движка Unity и 3D редактора Blender было реализовано приложение под операционную систему Android, которое визуализирует трехмерную сцену интерьера в виртуальной реальности, а также обеспечивает взаимодействие с ней в различных режимах.

В процессе выполнения работы в рамках ранее сформулированных задач было выполнено следующее:

1. Произведен анализ состояния проблемы интерьерной визуализации и существующих подходов для ее решения.
2. Произведен анализ и обоснован выбор инструментов для разработки технологии трехмерной визуализации интерьера в виртуальной реальности.
3. В соответствии с техническим заданием проведена разработка технологии.
4. Подготовлено руководство пользователя разработанного программного технологии.

На основании анализа состояния проблемы интерьерной визуализации было выявлено, что распространенные подходы являются неоптимальными, а в связи с преимуществами рендеринга в реальном времени и интерактивных технологий, спрос на них растет в геометрической прогрессии. Поэтому, для реализации были выбраны именно эти подходы, с созданием различных режимов отображения и взаимодействия с трехмерной сценой.

Анализ инструментов включал в себя сравнение игровых движков Unreal Engine и Unity, выделены преимущества, обосновывающие выбор в пользу Unity. Также, анализ редактора трехмерной графики Blender, выявил его преимущества по сравнению с другими решениями на рынке, что обосновало его выбор.

После, была произведена разработка технологии в соответствии с техническим заданием, а также разработано руководство пользователя.

Таким образом, следует считать, что результаты разработки соответствуют всем требованиям технического задания, поставленная цель достигнута. Работа носит законченный характер.

Список информационных источников

1. ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению. М.: Стандартиформ, 2010. 2 с.
2. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. М.: Стандартиформ, 2009. 11 с.
3. ЖК Мечта студия // Houzz URL: <https://www.houzz.ru/projects/2553480/> (дата обращения: 03.06.2019).
4. Некрасова Н.А., Некрасов С.И. Философия науки и техники: Тематический словарь справочник. Учебное пособие. М.: МИИТ, 2009. 424 с.
5. Орёл или решка: сравнение Unity и Unreal Engine // DTF URL: <https://dtf.ru/gamedev/7227-orel-ili-reshka-sravnenie-unity-i-unreal-engine> (дата обращения: 03.06.2019).
6. Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. М.: ИПК Издательство стандартов, 2001. 49 с.
7. Системы автоматизированного проектирования // АСПЕКТ СПБ URL: <http://www.aspectspb.ru/solutions/is/cad.html> (дата обращения: 03.06.2019).
8. Что такое рендеринг в реальном времени и зачем он нужен // Aion Pro URL: <https://www.aion.pro/post/что-такое-rendering-v-realnom-vremeni-i-zachem-on-nuzhen> (дата обращения: 03.06.2019).
9. 2018 Architectural Visualization Rendering Engine Survey - RESULTS // CGarchitect URL: http://www.cgarchitect.com/2018/02/2018-architectural-visualization-rendering-engine-survey?utm_medium=website&utm_source=archdaily.com (дата обращения: 03.06.2019).
10. About // blender.org – Home of the Blender project URL: <https://www.blender.org/about/> (дата обращения: 03.06.2019).

11. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering, Fourth Edition. 4 изд. A K Peters/CRC Press, 2018. 1198 с.
12. All Textures // POLIIGON URL: <https://www.poliigon.com/> (дата обращения: 03.06.2019).
13. Blackman S. Unity for Absolute Beginners. 1 изд. Apress, 2014. 612 с.
14. CADFolks. AutoCAD 2019 For Beginners. 6 изд. CreateSpace Independent Publishing Platform, 2018. 318 с.
15. Computer Graphics: Principles and Practice (3rd Edition) / Hughes, Dam, McGuire, Sklar, Foley, Feiner, Akeley, 3 изд. Addison-Wesley Professional, 2007. 1264 с.
16. Doran J. P. Unreal Engine Game Development Cookbook. Packt Publishing, 2015. 326 с.
17. Exploring the VR Boundaries of ArchViz in Unity // 80 LEVEL URL: <https://80.lv/articles/exploring-the-vr-boundaries-of-archviz-in-unity-004adk-001agt/> (дата обращения: 03.06.2019).
18. Graphics and Visualization: Principles & Algorithms / Theoharis, Papaioannou, Platis, Patrikalakis, A K Peters/CRC Press, 2007. 752 с.
19. Linowes J. Unity Virtual Reality Projects: Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D. Packt Publishing, 2015. 286 с.
20. Muccio D. AutoCAD 2018 for the Interior Designer: AutoCAD for Mac and PC. ASDC Publications, 2017. 450 с.
21. Omura G., Bento B. C. Mastering AutoCAD 2018 and AutoCAD LT 2018. 1 изд. Sybex, 2017. 1088 с.
22. Products – Unity // Unity URL: <https://unity3d.com/unity> (дата обращения: 03.06.2019).
23. Real-Time Rendering: Ushering in a New Era of Design Visualization // b+a architecture URL: <http://baarchitecture.com/production/realtime-rendering->

- [ushering-in-a-new-era-of-design-visualization-76f3d](#) (дата обращения: 03.06.2019).
- 24.Sewell B. Blueprints Visual Scripting for Unreal Engine. Packt Publishing, 2015. 188 с.
- 25.Shannon T. Unreal Engine 4 for Design Visualization: Developing Stunning Interactive Visualizations, Animations, and Renderings (Game Design). 1 изд. Addison-Wesley Professional, 2017. 384 с.
- 26.Sinicki A. Learn Unity for Android Game Development: A Guide to Game Design, Development, and Marketing. 1 изд. Apress, 2017. 248 с.
- 27.Staub-French S., Khanzode A. 3D and 4D modeling for design and construction coordination: Issues and lessons learned // Electronic Journal of Information Technology in Construction. 2007. №12. с. 381-407.
- 28.THE IMPORTANCE OF 3D VISUALIZATION // VRender URL: <https://vrender.com/the-importance-of-3d-visualization/> (дата обращения: 03.06.2019).
- 29.Thorn Mastering A. Unity 2017 Game Development with C#: Create professional games with solid gameplay features and professional-grade workflow. 2 изд. Packt Publishing, 2017. 568 с
- 30.Unity VS Unreal – Which Engine Should You Choose? // Sunday Sundae URL: <https://sundaysundae.co/unity-vs-unreal/> (дата обращения: 03.06.2019).
- 31.Unreal Engine 4 vs. Unity: Which Game Engine Is Best for You? // Pluralsight URL: <https://www.pluralsight.com/blog/film-games/unreal-engine-4-vs-unity-game-engine-best> (дата обращения: 03.06.2019).
- 32.Unreal Studio | Features // What is Unreal Engine 4 URL: <https://www.unrealengine.com/en-US/features> (дата обращения: 03.06.2019).
- 33.We Tried Real-Time Rendering. Here's What We Found Out. // ENGINEERING.COM URL: <https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/>

[17556/We-Tried-Real-Time-Rendering-Heres-What-We-Found-Out.aspx](https://www.easyrender.com/3d-rendering/what-is-real-time-rendering-and-why-it-matters) (дата обращения: 03.06.2019).

34. What is Real Time Rendering and Why It Matters // Architectural Rendering Services | Easy Render URL: <https://www.easyrender.com/3d-rendering/what-is-real-time-rendering-and-why-it-matters> (дата обращения: 03.06.2019).

35. Why 3D Modeling is so Important // LinkedIn URL: <https://www.linkedin.com/pulse/why-3d-modeling-so-important-thomas-a-anderson-p-e-> (дата обращения: 03.06.2019).