

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

ОБУЧАЮЩЕЕ VR-ПРИЛОЖЕНИЕ НА ИГРОВОМ ДВИЖКЕ
UNITY

Выпускная квалификационная работа
по направлению подготовки 44.03.04 Профессиональное обучение
(по отраслям)
профилю подготовки «Информатика и вычислительная техника»
специализации «Компьютерные технологии»

Идентификационный номер ВКР: 319

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

_____ Н. С. Толстова

« ____ » _____ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ОБУЧАЮЩЕЕ VR-ПРИЛОЖЕНИЕ НА ИГРОВОМ ДВИЖКЕ
UNITY

Исполнитель:

обучающийся группы № КТ-412

А. Ю. Утемов

Руководитель:

к.пед.н., доцент

Т. В. Чернякова

Нормоконтролер:

Б. А. Редькина

АННОТАЦИЯ

Выпускная квалификационная работа состоит из обучающего приложения «Geometrica» и пояснительной записки на 55 страницах, содержащей 27 рисунков, 1 таблицу, 30 источников литературы, а также 2 приложений на 5 страницах.

Ключевые слова: ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ, ПРИЛОЖЕНИЕ, МОБИЛЬНОЕ, UNITY, ГЕОМЕТРИЧЕСКИЕ ФИГУРЫ.

Утемов, А.Ю. Обучающее VR-приложение на игровом движке Unity: выпускная квалификационная работа / А. Ю. Утемов; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2017. — 55 с.

В работе рассмотрены вопросы использования VR-приложений в обучении.

Целью работы является разработать обучающее VR-приложение на тему «Объемные геометрические фигуры - Geometrica» в среде Unity. Для достижения цели были решены следующие задачи, проанализирована литература и интернет-источники по теме создания обучающих VR-приложений в среде Unity. Проанализированы аспекты использования виртуальной реальности в обучении, требования к компьютерным программам, основанным на 3D изображениях, возможность и необходимость использования VR-приложений в процессе обучения. Разработан сценарий и общая концепция обучающего VR-приложения по теме «Объемные геометрические фигуры». Разработано обучающее VR-приложение в среде Unity.

СОДЕРЖАНИЕ

Введение.....	5
1 Технологии создания обучающих приложений виртуальной реальности.....	7
1.1 Технологии виртуальной реальности.....	7
1.1.1 Особенности и основные понятия виртуальной реальности.....	7
1.1.2 Анализ интернет-источников по предметной области	10
1.1.3 Применение VR-приложений в процессе обучения.....	13
1.2 Анализ аналогичных приложений.....	18
1.3 Педагогический анализ рассматриваемой предметной области	19
1.3.1 Педагогический адрес.....	19
1.3.2 Анализ учебно-методической документации	19
1.3.3 Характеристика и специфические особенности текущего состояния ознакомления с объемными геометрическими моделями.....	24
1.3.4 Мероприятия и рекомендации по совершенствованию педагогического процесса по преподаваемой дисциплине с использованием информационных технологий.....	25
1.4 Технологические требования к педагогическому продукту и средства его реализации.....	25
1.4.1 Функционал и требования к разрабатываемому педагогическому продукту.....	25
1.4.2 Анализ и выбор реализации приложения.....	28
Вывод.....	29
2 Проектная часть.....	31
2.1 Цель и назначение VR-приложения «Объемные геометрические фигуры – Geometrica»	31
2.2 Описание основных функций приложения.....	31
2.3 Общее описание структуры и содержания приложения	31
2.4 Основные инструменты и технологии для реализации приложения.	32

2.5	Сценарий VR-приложения «Объемные геометрические фигуры»	33
2.6	Навигация по VR-приложению «Объемные геометрические фигуры»	35
2.7	Жизненный цикл создания педагогического продукта	36
2.7.1	Раздел главное меню	36
2.7.2	Раздел меню с выбором рассматриваемых моделей	37
2.7.3	Раздел создания сцены выставки фигур	37
2.7.4	Сцена с осмотром модели и информации о фигуре	39
2.7.5	Сцена загрузки.....	40
2.7.6	Используемые скрипты	41
2.8	Методические указания по использованию педагогического продукта в учебном процессе	43
2.8.1	Установка и запуск программы	43
2.8.2	Навигация в приложении	43
2.9	Результат апробации и внедрения	44
	Вывод.....	44
	Заключение	45
	Список использованных источников	47
	Приложение 1	50
	Приложение 2	52

ВВЕДЕНИЕ

В связи со стремительным развитием современного общества в сферу жизнедеятельности человека происходит проникновение информационных технологий, что обеспечивает эффективное функционирование и успешную эволюцию общественного мира. Внедрение информационных технологий производится во всех сферах.

Не секрет, что игры и игрушки для детей являются важным инструментом познания мира, они помогают развивать ощущения и даже учиться. И в этом смысле их интерес к мобильным устройствам, играющим в нашей жизни все большую роль, вполне понятен. Сегодня на рынке появляется все больше мобильных приложений, рассчитанных на детскую аудиторию: начиная от обычных сказок, мультфильмов или игр, заканчивая весьма своеобразным интерактивным контентом, комбинирующим свойства книги, мультфильма и развивающей игры.

Немаловажными становятся специальные приложения. Одним из лучших для образования называют «VR Cardboard» виртуальная реальность с помощью картона. Разработка состоит из перечня изображений из различных областей знаний: математики и биологии, анатомия и истории, инженерного дела и архитектуры. Уникальность ее в том, виртуальная реальность применяется для обучения профессиям, где эксплуатация реальных устройств и механизмов связана с повышенным риском либо связана с большими затратами (пилот самолета, машинист поезда, диспетчер, водитель, горноспасатель и т. п.).

Объект исследования: VR-технологии создания приложения.

Предмет исследования: разработка обучающих VR-приложений в среде Unity.

Цель выпускной квалификационной работы: разработать обучающее VR-приложение на тему «Объемные геометрические фигуры - Geometrisa» в среде Unity.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать литературу и интернет-источники по теме создания обучающих VR-приложений в среде Unity.
2. Проанализировать аспекты использования виртуальной реальности в обучении, требования к компьютерным программам, основанным на 3D изображениях, возможность и необходимость использования VR-приложений в процессе обучения.
3. Разработать сценарий и общую концепцию обучающего VR-приложения по теме «Объемные геометрические фигуры».
4. Разработать обучающее VR-приложение в среде Unity.

1 ТЕХНОЛОГИИ СОЗДАНИЯ ОБУЧАЮЩИХ ПРИЛОЖЕНИЙ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

1.1 Технологии виртуальной реальности

1.1.1 Особенности и основные понятия виртуальной реальности

Виртуальная реальность, также VR – созданный техническими средствами мир (объекты и субъекты), передаваемый человеку через его ощущения: зрение, слух, обоняние, осязание и другие. Виртуальная реальность имитирует как воздействие, так и реакции на воздействие. Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности производится в реальном времени.

Объекты виртуальной реальности обычно ведут себя близко к поведению аналогичных объектов материальной реальности. Пользователь может воздействовать на эти объекты в согласии с реальными законами физики (гравитация, свойства воды, столкновение с предметами, отражение и т. п.). Однако часто в развлекательных целях пользователям виртуальных миров позволено больше, чем возможно в реальной жизни (например, летать, создавать любые предметы и т. п.).

Не следует путать виртуальную реальность с дополненной. Их коренное различие в том, что виртуальная конструирует новый искусственный мир, а дополненная реальность лишь вносит отдельные искусственные элементы в восприятие мира реального.

До эры компьютерных технологий под виртуальностью понимали объект или состояние, которые реально не существуют, но могут возникнуть при определенных условиях.

Понятие искусственной реальности было впервые введено Майроном Крюгером (англ. Myron Krueger) в конце 1960-х. В 1964 году Станислав Лем

в своей книге «Сумма Технологии» под термином «Фантомология» описывает задачи и суть ответа на вопрос «как создать действительность, которая для разумных существ, живущих в ней, ничем не отличалась бы от нормальной действительности, но подчинялась бы другим законам?». Первая система виртуальной реальности появилась в 1962 году, когда Мортон Хейлиг (англ. Morton Heilig) представил первый прототип мультисенсорного симулятора, который он называл «Сенсорамма» (Sensorama). Сенсорамма погружала зрителя в виртуальную реальность при помощи коротких фильмов, которые сопровождались запахами, ветром (при помощи фена) и шумом мегаполиса с аудиозаписи. В 1967 году Айвен Сазерленд (англ. Ivan Sutherland) описал и сконструировал первый шлем, изображение на который генерировалось при помощи компьютера. Шлем Сазерленда позволял изменять изображения соответственно движениям головы (зрительная обратная связь) [4].

В 1970-х годах компьютерная графика полностью заменила видеосъемку, до того использовавшуюся в симуляторах. Графика была крайне примитивной, однако важным было то, что тренажеры (это были симуляторы полетов) работали в режиме реального времени. Первой реализацией виртуальной реальности считается «Кинокарта Аспена» (Aspen Movie Map), созданная в Массачусетском Технологическом Институте в 1977 году. Эта компьютерная программа симулировала прогулку по городу Аспен, штат Колорадо, давая возможность выбрать между разными способами отображения местности. Летний и зимний варианты были основаны на реальных фотографиях [13].

В середине 1980-х появились системы, в которых пользователь мог манипулировать с трехмерными объектами на экране благодаря их отклику на движения руки. В 1989 году Джарон Ланьер ввел более популярный ныне термин «виртуальная реальность». В фантастической литературе поджанра киберпанк виртуальная реальность есть способ общения человека с «киберпространством» — некой средой взаимодействия людей и машин, создаваемой в компьютерных сетях [13].

В данный момент технологии виртуальной реальности широко применяются в различных областях человеческой деятельности: проектировании и дизайне, добыче полезных ископаемых, военных технологиях, строительстве, тренажерах и симуляторах, маркетинге и рекламе, индустрии развлечений и т. д. За 2015 год объем рынка технологий виртуальной реальности оценивался в 15 млрд долларов [13].

Дизайн для VR очень похож на дизайн видеоигр, поскольку в обоих случаях мы имеем дело с интерактивным 3D-опытом. Разница в том, что в VR нужно уделять особое внимание эффекту присутствия, погруженности, нелинейности повествования, не вызывающему тошноты перемещению и графической оптимизации [18].

Большинство VR-разработчиков предпочитают использовать игровые движки (если только не создают для браузеров VR-приложения), и с самого начала им приходится выбирать, на чем же работать. Самые популярные движки – Unreal Engine 4 (UE4) и Unity. Оба имеют очень широкие возможности и являются надежными инструментами. Вокруг обоих движков сложились активные сообщества с многочисленными информационными ресурсами. Оба движка позволяют управлять 3D-окружением, импортировать собственный контент (3D-модели, изображения, звук, видео), а также программировать интерактивность и процесс игры или приложения. На YouTube есть огромное количество обучающих видео, а в сети – руководств, созданных как самими авторами, так и поклонниками [29].

Среди VR-разработчиков нет общепринятого мнения, что один из этих движков лучше другого. У каждого есть свои особенности. UE4 считается более оптимизированным с точки зрения вычислений, дает более достоверную картинку, но имеет более крутую кривую обучения. Unity создавался из расчета, чтобы его возможностей хватало для создания коммерческих игр, но при этом он остается более интуитивно понятным и эффективным для начинающих разработчиков. Unreal Engine 4 можно скачать и использовать бесплатно, но авторам придется ежеквартально отчислять по 5% дохода с игры,

если он превысит \$3000. У Unity есть несколько версий разной стоимости, но можно остановиться на бесплатной Unity Personal [13].

После того, как определились с движком, надо поподробнее ознакомиться со своим выбором. Начинать нужно с азов того языка программирования, который использует ваш инструмент: C++ и Blueprints Visual Scripting (UE4), C# (Unity) или кастомный язык разметки для веб-приложений. Для Android, то скачать Android Studio и попробовать развернуть тренировочное приложение. В случае с Google Cardboard и Unity обратится к Google SDK. Так вот для начала нужно определиться с темой приложения, потом обдумать сценарий, после выбрать среду разработки. После этого как было сказано выше, узнаем какой язык использует движок. Потом создаём 3d модели и располагаем их на сцене, настраиваем окружение, и пишем скрипты для выполнения каких-либо действий или заданий. В завершении всего проверяем работоспособность приложения и при необходимости проводим оптимизацию данного приложения.

1.1.2 Анализ интернет-источников по предметной области

Анализ литературы и интернет-источников показал, что на данный момент существует достаточное количество источников, посвященных проблеме исследования, наиболее значимых можно выделить.

1. Официальное руководство пользователя программы Unity. Данное руководство создано, чтобы помочь узнать, как использовать Unity, от базовых до продвинутых приемов. Его можно читать от начала до конца, или использовать в качестве справочника. Доступен по ссылке: <https://docs.unity3d.com/ru/current/Manual/index.html>. Главная страница электронного ресурса представлена на рисунке 1.

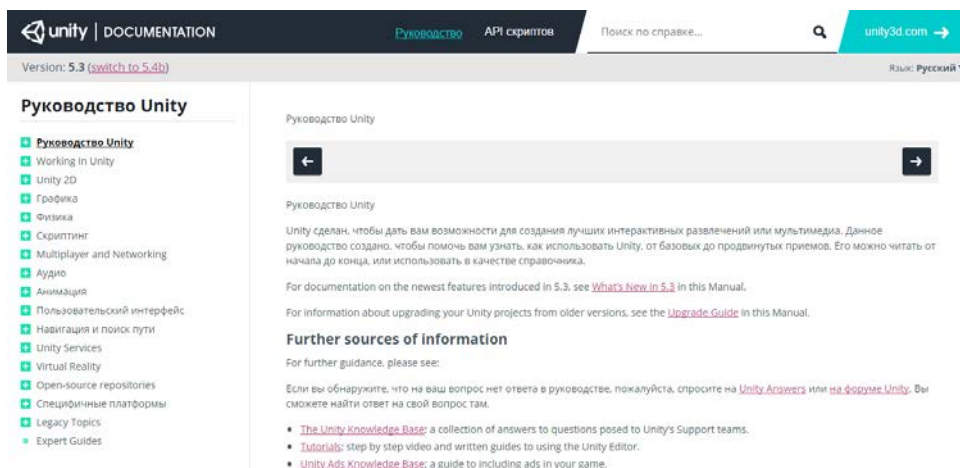


Рисунок 1 — Главная страница сайта «Руководство Unity»

2. Официальная документация Google по программированию VR-приложений. Данное руководство создано, чтобы узнать, как связать VR Cardboard с движком Unity 3d. Сайт доступен по ссылке: <https://developers.google.com/vr/android/>. Главная страница электронного ресурса представлена на рисунке 2.

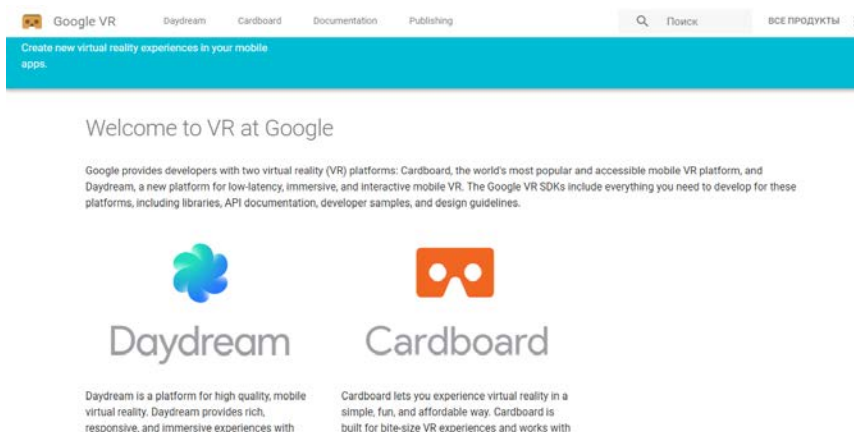


Рисунок 2 — Главная страница сайта «Google VR»

3. Официальный обучающий портал для Unity. Обучающие материалы поделены на два раздела – уроки для изучения движка и материалы, ориентированные на сервисы и производство. В каждом разделе существуют проекты – ряд пошаговых материалов, и темы, содержащие более подробные дополнительные уроки. Ссылка на сайт: <https://unity3d.com/ru/learn/tutorials>. Главная страница электронного ресурса представлена на рисунке 3.

Темы















 Interface & Essentials (18)	 2D Game Creation (45)
 Scripting (131)	 Best Practices (12)
 Graphics (77)	 Physics (27)
 Audio (12)	 Animation (17)
 User Interface (UI) (41)	 Mobile & Touch (6)
 Navigation (17)	 Tips (19)
 Ads & Analytics (9)	 Virtual Reality (8)

Рисунок 3 — Главная страница сайта «Обучающие материалы»

4. Сайт с базой курсов по разработке игр. Много бесплатных курсов по разработке видеоигр. Представлены различные курсы по разработке игр, как для начинающих, так и для опытных разработчиков. Сайт доступен по ссылке: <http://learn.indiecamp.ru/courses>. Сайт изображен на рисунке 4.

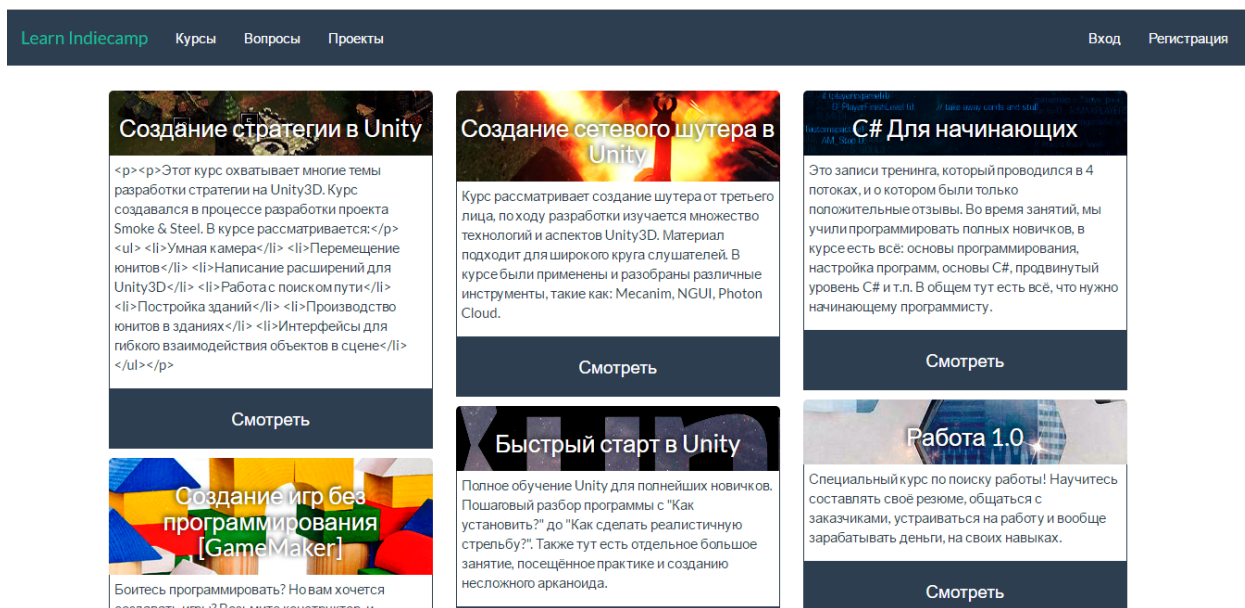


Рисунок 4 — Страница с курсами сайта «Learn Indiecamp»

5. Курсы по разработке игр в Unity. На данном сайте расположено 4 курса, которые 2 из них базовые и 2 углубленные. Где обучают всему основному для создания первой игры в среде Unity, как мобильное, так и для компьютера. Сайт расположен по ссылке: http://unity3dschool.ru/#about_school. Главная страница изображена на рисунке 5.

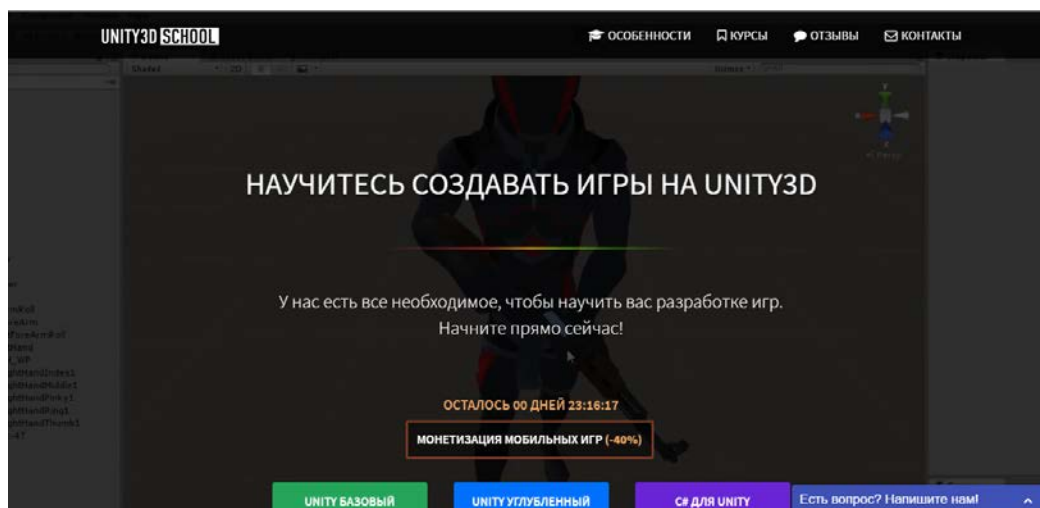


Рисунок 5 — Главная страница сайта «Unity School»

1.1.3 Применение VR-приложений в процессе обучения

В начале 60-ых годов появился первый прототип системы для создания виртуальной реальности – «Sensorama». Он позволял просматривать 3D-стереоскопические изображения, параллельно сопровождая визуальную часть стереозвуком, запахами и эффектом ветра. Sensorama заинтересовал многих и положил начало новому направлению. А в 1989 году Джарон Ланьер (Jaron Zepel Lanier) ввел термин «виртуальная реальность». Обретя столь привлекательное название, направление стало привлекать к себе еще больше внимания общественности.

Не прошло и года, как исследователем из компании Boeing, Томом Коделом был введен новый термин – «Дополненная реальность» (Augmented Reality). Это было не просто новое название существующего направления, это было новым ответвлением от виртуальной реальности, которое переросло в полноценную, самостоятельную отрасль. В отличие от виртуальной реальности, дополнения реальность не замещает человеку весь окружающий мир виртуальной альтернативой, а дополняет и обогащает ее, добавляя поверх окружающих нас предметов дополнительные слои информации как показано на рисунке 6.



Рисунок 6 — Интерфейс HoloLens

Появление мобильной революции. Одним из основных акселераторов виртуальной и дополненной реальности стали смартфоны. Мобильность и наличие датчиков положения в пространстве позволили найти практическое применения этому направлению. Стали появляться приложения, накладывающие мета-данные поверх изображения получаемой с камеры телефона. Стало возможным прокладывать маршруты к ближайшему отделению банка или к филиалу любимой сети магазинов поверх окружающего нас живого мира.

Мобильные телефоны стали основой нового витка развития и в направлении виртуальной реальности. В 2014 году на конференции Google I/O был представлен Google Cardboard, показано на рисунке 7. Легкий в изготовлении и дешевый шлем, собираемый из картона, оптических линз, магнита и застежек, в который вставляется смартфон. Благодаря доступности Google Cardboard и повсеместному использованию смартфонов, виртуальная реальность широко распространилась в качестве одного из новых, экспериментальных методов обучения. Поэтому была использована эта технология для реализации данной цели [4].



Рисунок 7 — Google Cardboard

Данной теме уделяется особое внимание в сфере E-Learning. В недавнем отчете одной из наиболее уважаемых, аналитических групп – Horizon 2016, использованию виртуальной и дополненной реальности в целях обучения посвящено несколько страниц. На данный момент, больше всего VR и AR технологии (Virtual Reality and Augmented Reality) нашли свое применение в сфере медицины и физики [9].

В чем же польза обучению от этих новых технологий? Прежде всего, VR позволяет в мгновение ока перенести студентов в любое виртуальное место в наблюдаемой вселенной и погрузить их в глубокое, увлекательное обучение. Мотивация – это еще одна положительная сторона инструментов виртуальной реальности. Что интересней? Читать сотни страниц текста с черно-белыми иллюстрациями или оказаться на Марсе и своими собственными руками собрать образцы почвы для изучения ее состава?

Направления виртуальной реальности активно развивается и пока недоступно во всем своем потенциале для большинства. Тем не менее, уже можно воспользоваться некоторыми существующими его реализациями. Например, создавать виртуальные, панорамные туры с наложением информации поверх визуального слоя и позволять ученикам самостоятельно познавать изучаемые предмет со всех сторон. В качестве хорошего примера можно посмотреть на виртуальное воплощение Боинга 787 созданное с помощью платформы Roundme. Пользователь может рассмотреть все важные детали

внутри и снаружи самолета. Это не просто набор фотографий, а интерактивная среда, в которой можно передвигаться в пространстве и переходить от модуля к модулю [30].

В Университете Мельбурна (The University of Melbourne), видимо, тоже пришли к выводу, что виртуальную реальность можно использовать в целях обучения. Сотрудниками университета было создано виртуальное пространство The Tidal River coast of Wilsons Promontory, в котором они снабдили каждую часть своего тура детальными текстовыми и визуальными пояснениями. Таким образом, студенты во время блуждания по виртуальному миру могут задержаться в определенном месте и почитать подробности о том, что они видят перед собой и как это относится к изучаемой области. О том, какие химические процессы проходят с местными каменными породами и к чему все это приводит [12].

Roundme позволяет создавать некую смесь виртуальных пространств с возможностью наложения полезной информации поверх визуального слоя. Возможности создателя виртуальной среды ограничены лишь его фантазией. Никто ведь не говорит, что виртуальные туры могут быть созданы только вокруг географических мест. Почему бы, например, на базе такой платформы не создать виртуальную операционную палату и на ее базе не обучать студентов основам работы в операционной или дать им возможность изучить всю доступную аппаратуру. А учитывая, что Roundme поддерживает гарнитуру для виртуальной реальности Google Cardboard, весь процесс обучения можно превратить в гораздо более привлекательный и глубокий процесс [13].

Создаваемые в Roundme виртуальные пространства можно экспортировать в виде Embedded-кода и использовать как готовые модули веб-страниц или как часть других сервисов и медиа-объектов. Таким образом, можно, например, импортировать виртуальную заготовку из Roundme в курсы, создаваемые в Geenio. Возможность добавления в курсы iframe-объектов, позволяет импортировать виртуальные заготовки, созданные в Roundme на лю-

бую страницу в курсе. Чтобы было понятней, зачем это нужно в курсах и как можно применять на практике, и так рассмотрим следующий пример.

Компания закупила партию нового оборудования, с которой никто не знаком и новая аппаратура не похожа на то, с чем приходилось работать сотрудникам прежде. Необходимо обучить несколько отделов в компании использованию нового оборудования. Доставка закупленной аппаратуры занимает несколько месяцев, так как все очень крупногабаритное и доставляется с другого конца земли, а компания в свою очередь не хочет допустить простаивания оборудования после его прибытия. По этой причине вместе с приобретением, компания договаривается с поставщиков отправить к ним фотографа, чтобы отснять панорамные фотографии новой аппаратуры со всех сторон для последующего создания виртуальной среды [18].

Пока оборудование находится в пути фотограф заливает все фотографии на Roundme и создает новое виртуальное пространство с переходами («порталами») от одной части аппарата к другой, не упустив никаких деталей. Затем E-Learning специалист подключается к работе. Он добавляет прямо поверх созданного пространства горячие точки (Hot Spots) с текстовыми пояснениями и иллюстрациями, которые могут быть вызваны нажатием кнопки мыши или удерживанием взгляда на нужные точки с помощью виртуальной гарнитуры, интерфейс такого приложения показан на рисунке 8.

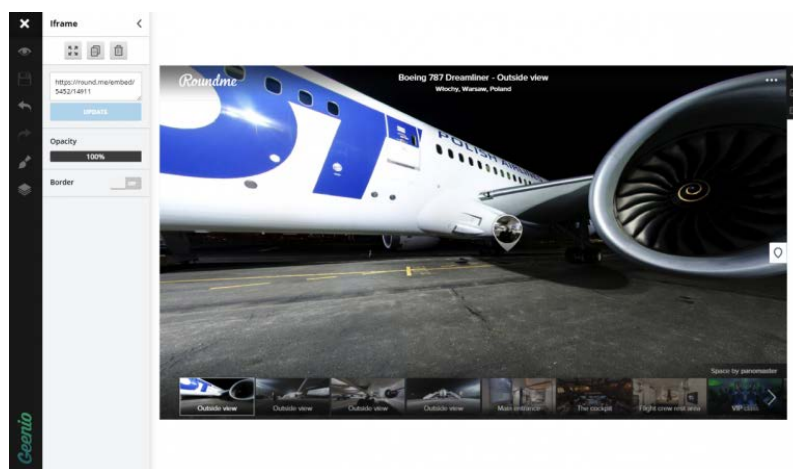


Рисунок 8 — Geenio

Затем готовая виртуальная среда экспортируется и вставляется внутрь курса созданного в Geenio, где к ней добавляются предварительные пояснения, видеоролики и сопутствующие материалы [21].

Вывод: После страницы с погружением в виртуальный мини-мир, можно попросить ученика ответить на ряд вопросов для закрепления знаний и запоминания увиденного. В итоге, мы имеем полноценный курс обучения с использованием виртуальной среды и мотивированными учениками.

1.2 Анализ аналогичных приложений

Анализ магазинов приложений «Google Play» и «App Store» для платформ Android и iOS соответственно показал, что конкурентов, на данную тему VR-приложений на данный момент нет. В обоих магазинах присутствует приложение «Chemistry VR», которое представляет из себя приложение для изучения химии. В данном приложении нужно собирать химические элементы и соединять их. Приложение «Chemistry VR» изображено на рисунке 9.

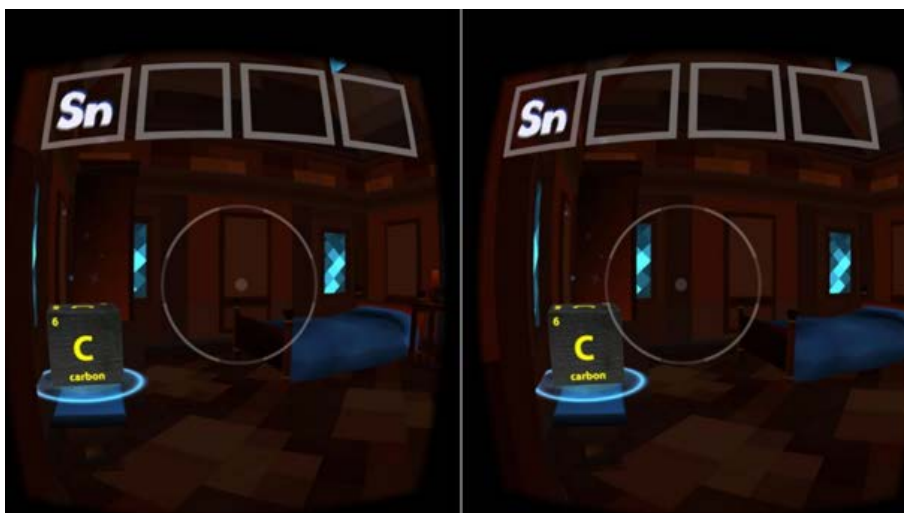


Рисунок 9 — Приложение «Chemistry VR»

Приложения совсем разные, но концепция управления одна. К отличиям можно отнести то, что в «Chemistry VR» чтобы ходить по комнате нужно нажимать на экран. И это можно отнести к минусам так как не все гарнитуры поддерживают взаимодействие с экраном телефона.

1.3 Педагогический анализ рассматриваемой предметной области

1.3.1 Педагогический адрес

VR-приложение «Объемные геометрические фигуры – Geometrica» рассчитано для учащихся школьного возраста, обучающихся в 11 классе по учебному предмету «Математика».

1.3.2 Анализ учебно-методической документации

Рассматривая типичную рабочую программу дисциплины «Математика» и государственный стандарт можно выделить следующие цели и задачи.

Цели

Изучение математики в старшей школе на профильном уровне направлено на достижение следующих целей:

- **формирование** представлений об идеях и методах математики; о математике как универсальном языке науки, средстве моделирования явлений и процессов;
- **овладение** устным и письменным математическим языком, математическими знаниями и умениями, необходимыми для изучения школьных естественно – научных дисциплин, для продолжения образования и освоения избранной специальности на современном уровне;
- **развитие** логического мышления, алгоритмической культуры, пространственного воображения, развитие математического мышления и интуиции, творческих способностей на уровне, необходимом для продолжения образования и для самостоятельной деятельности в области математики и ее приложений в будущей профессиональной деятельности;

- **воспитание** средствами математики культуры личности: знакомство с историей развития математики, эволюцией математических идей, понимание значимости математики для общественного прогресса.

Требования к уровню подготовки выпускников

В результате изучения математики на профильном уровне ученик должен знать/понимать:

- значение математической науки для решения задач, возникающих в теории и практике; широту и ограниченность применения математических методов к анализу и исследованию процессов и явлений в природе и обществе;

- значение практики и вопросов, возникающих в самой математике, для формирования и развития математической науки;

- идеи расширения числовых множеств как способа построения нового математического аппарата для решения практических задач и внутренних задач математики;

- значение идей, методов и результатов алгебры и математического анализа для построения моделей реальных процессов и ситуаций;

- универсальный характер законов логики математических рассуждений, их применимость в различных областях человеческой деятельности;

- различие требований, предъявляемых к доказательствам в математике, естественных, социально-экономических и гуманитарных науках, на практике;

- роль аксиоматики в математике; возможность построения математических теорий на аксиоматической основе; значение аксиоматики для других областей знания и для практики;

- вероятностный характер различных процессов и закономерностей окружающего мира.

Уметь:

- соотносить плоские геометрические фигуры и трехмерные объекты с их описаниями, чертежами, изображениями; различать и анализировать взаимное расположение фигур;
- изображать геометрические фигуры и тела, выполнять чертеж по условию задачи;
- решать геометрические задачи, опираясь на изученные свойства планиметрических и стереометрических фигур и отношений между ними;
- проводить доказательные рассуждения при решении задач, доказывать основные теоремы курса;
- вычислять линейные элементы и углы в пространственных конфигурациях, объемы и площади поверхностей тел и их простейших комбинаций;
- применять координатно-векторный метод для вычисления отношений, расстояний и углов;
- строить сечения многогранников и изображать сечения тел вращения;
- использовать приобретенные знания и умения в практической деятельности для:
 - исследования несложных практических ситуаций на основе изученных формул и свойств фигур;
 - вычисления длин, площадей и объемов реальных объектов при решении практических задач, используя при необходимости справочники и вычислительные устройства [3].

Таблица 1 — Примерное тематическое планирование

№	Кол-во часов в теме	Дата		Тема урока	Тип урока	Формируемые обще учебные ЗУН и способы деятельности	Форма контроля
		план	факт				
		<i>Цилиндр, конус, и шар. 17 часов</i>					
6	3			Цилиндр Понятие цилиндра.	Комбинированный	Иметь представление о цилиндре.	

Продолжение таблицы 1

						У м е т ь : различать в окружающем мире предметы-цилиндры, выполнять чертежи по условию задачи	
7		24.10		Понятие цилиндра.	Комбинированный	И м е т ь представление о цилиндре. У м е т ь : различать в окружающем мире предметы-цилиндры, выполнять чертежи по условию задачи	Фронтальный опрос
8		30.10		Цилиндр. Решение задач	Комбинированный	У м е т ь : находить площадь осевого сечения цилиндра, строить осевое сечение цилиндра З н а т ь : формулы площади боковой и полной поверхности цилиндра и уметь их выводить; используя формулы, вычислять площадь боковой и полной поверхностей	Фронтальный опрос Самостоятельная работа
		31.10		Цилиндр. Решение задач	Комбинированный		Самопроверка задач из домашней работы
9	3			Конус			
0		6.11		Конус	Комбинированный	З н а т ь : элементы конуса: вершина, ось, образующая, основание. У м е т ь : выполнять построение конуса и его сечения, находить элементы З н а т ь : элементы усеченного конуса. У м е т ь : распознавать на моделях, изображать на чертежах	Фронтальный опрос
1		7.11		Конус	Комбинированный	З н а т ь : формулы площади боковой и полной поверхности конуса У м е т ь : решать задачи на нахождение площади поверхности конуса	Математический диктант
		13.11		Усеченный конус	Комбинированный	З н а т ь : формулы площади боковой и полной поверхности конуса и усеченного конуса. У м е т ь : решать задачи на нахождение площади поверхности конуса и усеченного конуса	Фронтальный опрос

Продолжение таблицы 1

2	11			Сфера			
3		14.11		Сфера. Уравнение сферы	Комбиниро- ванный	Зн а т ь : определение сферы и шара. У м е т ь : определять взаим- ное расположение сфер и плоскости Зн а т ь : уравнение сферы. У м е т ь : составлять уравне- ние сферы по ко-ординатам точек; решать типовые задачи по теме	Самостоя- тельная работа
4		20.11		Взаимное расположе- ние сферы и плоскости	Комбиниро- ванный	Зн а т ь возможные случаи вза- имного расположения сферы и плоскости	Математиче- ский диктант
5		21.11		Касательная плоскость к сфере	Комбиниро- ванный	Зн а т ь : свойство касатель- ной к сфере, что собой пред- ставляет расстояние от цен- тра сферы до плоскости сече- ния. У м е т ь : решать задачи по теме	Проверка домашнего задания
6		27.11		Площадь сферы	Комбиниро- ванный	Зн а т ь : формулу площади сферы. У м е т ь : применять форму- лу при решении задач на нахождение площади сферы	Самостоя- тельная работа
7		28.11		Разные за- дачи на многогран- ники, ци- линдр, ко- нус и шар	Урок реше- ния задач	У м е т ь : решать типовые задачи, применять получен- ные знания в жизненных си- туациях	Опрос в ходе решения задач
8		4.12		Разные за- дачи на многогран- ники, ци- линдр, ко- нус и шар	Комбиниро- ванный		Фронталь- ный опрос

Окончание таблицы 1

9		5.12	Разные задачи на многогранники, цилиндр, конус и шар	Урок решения задач		тест
0		1.12	<i>Контрольная работа №3</i>	Урок контроля знаний	Зн а т ь : элементы цилиндра, конуса, уравнение сферы, формулы боковой и полной поверхностей	
1		2.12	<i>Зачет №2</i>	Урок контроля знаний	У м е т ь : решать типовые задачи по теме, использовать полученные знания для исследования несложных практических ситуаций	
2		8.12	<i>Обобщение по теме «Цилиндр, конус, сфера и шар»</i>	Урок систематизации знаний, умений и навыков	Систематизировать теоретический материал и совершенствовать навыки решения задач	Теоретический тест с самопроверкой

1.3.3 Характеристика и специфические особенности текущего состояния ознакомления с объемными геометрическими моделями

На данный момент в учебных заведениях геометрические модели изучают в 2 этапа:

1. На первом этапе ученикам рассказывают о детали в теории.
2. На втором этапе, преподаватель показывает рисунки в учебнике. В некоторых школах преподаватели делают 3D модели простых геометрических моделей из подручных средств.

Ознакомление с объемными фигурами достаточно банально, каких-то особых методик нет, знакомят с основными фигурами: куб, сфера, шар, рассматривают у них основные характеристики, строят и все. Никакой информации о том, что мир трехмерных фигур достаточно разнообразен и известен с времен Древней Греции нет. Поэтому, конечно, необходимо используя современные технологии показать ребятам множество геометрических фигур.

Хорошо, когда преподаватели могут дать модель ученикам посмотреть и покрутить её. Минус этого только в том, что модели эти очень простые такие как куб, параллелепипед, сфера, конус и пирамида.

1.3.4 Мероприятия и рекомендации по совершенствованию педагогического процесса по преподаваемой дисциплине с использованием информационных технологий

В учебных заведениях нет возможности показать школьникам все существующие геометрические модели, так как сложно их конструировать и их легко сломать. Поэтому в школах модели показывают в картинках, либо в учебниках, либо распечатывают на принтере изображения сложных моделей.

В разрабатываемом приложении «Объемные геометрические фигуры – Geometrica» на данный момент уже более 45 трехмерных моделей и еще будут разрабатываться. Сейчас можно не только услышать о таких моделях, но и посмотреть их в 3D. В наше время телефон есть почти у каждого, так что не составит не какого труда установить на телефон приложение и посмотреть, как выглядит та или иная геометрическая модель. И уже учащиеся будут иметь представление что это за модель и как она выглядит. В таком случае затраты минимальны, девайс для телефона создается из обычного картона. Любой ученик может собрать их дома или на прямо на уроке.

1.4 Технологические требования к педагогическому продукту и средства его реализации

1.4.1 Функционал и требования к разрабатываемому педагогическому продукту

К основным традиционным дидактическим требованиям [16] относятся:

- требование научности обучения – обеспечение достаточной глубины и корректности изложения учебного материала с учетом последних достижений науки;

- требование доступности обучения – обеспечение соответствия степени теоретической сложности и глубины изучения возрастным и индивидуальным особенностям учащихся, не допущение чрезмерной усложненности и перегруженности учебного материала;

- требование систематичности и последовательности обучения – обеспечение формирования знаний, умений и навыков, учащихся в определенной логически связанной последовательности с обеспечением преемственности;

- требование наглядности обучения – обеспечение чувственного восприятия учащимися объектов, процессов, явлений;

- требование сознательности и активности обучения – обеспечение самостоятельных и активных действий учащихся по извлечению учебной информации;

- требование прочности усвоения знаний – обеспечение закрепления знаний;

Применение информационных технологий для обучения привело к появлению новых дидактических принципов обучения и, соответственно, новых дидактических требований к средствам обучения.

К основным новым дидактическим требованиям относятся:

- требование структуризации учебного материала и структурно-функциональной связанности – обеспечение представления учебного материала с разбивкой на структурные единицы с обозначением структурно-функциональных связей между ними, отражающих внутреннюю логику изучаемого материала;

- требование интерактивности обучения – обеспечение взаимодействия учащегося с электронным учебником;

- требование адаптивности обучения – обеспечение приспособления процесса обучения к уровню знаний, умений, психологических особенностей учащегося, работающего с учебным электронным изданием.

Лаконичность интерфейса реализуется за счет определения параметров по умолчанию, пиктограмм вместо текстовых выражений, поддержки способов оперативного ввода команд. Например, обзорные сведения помечаются пиктограммой бинокля, перечень ключевых понятий – пиктограммой ключа и т. д. Эффект от применения пиктограмм основывается на том, что образная информация воспринимается человеком легче, чем вербальная. Закрепленные ассоциации позволяют пользователям проще ориентироваться в учебном материале по компактным и понятным символическим изображениям.

Дружественность пользовательского интерфейса обеспечивается благодаря его простому освоению и эффективному применению вне зависимости от степени подготовки пользователей. Дружественность интерфейса обычно осуществляется за счет создания ситуаций, хорошо знакомых пользователю; оптимального выбора цветовых решений в оформлении приложения; оптимального звукового оформления.

Обучающие мобильные приложения по возможности применения и степени интеграции в учебный процесс можно условно разделить на три группы:

- мобильные приложения-дополнения к учебным пособиям обычно являются дополнениями к учебникам и курсам, их уместно и удобно применять с основными материалами для работы в классе или вне класса;
- мобильные приложения, предназначенные для самостоятельного изучения дисциплин, например, иностранных языков, которые можно с успехом использовать как дополнительный материал в классе и вне класса для развития различных языковых навыков и речевых умений;
- мобильные приложения, предназначенные для дистанционной (мобильной) формы обучения, которые содержат основной обучающий материал

с тестами, системой проверки и ссылок, обратной связью с преподавателем и т. д.

Обычно приложения последнего вида создаются преподавателями под определенный курс в рамках какого-либо проекта и не являются общедоступными как первые два типа приложений.

1.4.2 Анализ и выбор реализации приложения

Основные элементы Unity – это объекты («GameObject») и компоненты («MonoBehaviour»). Освоив эту концепцию, вы уже можете работать с Unity. Если правильно пользоваться этой системой, она позволяет значительно улучшить организацию проекта.

В Unity включено много компонентов, обеспечивающих всем необходимым для создания игры – кроме игровой логики как таковой. Как было указано выше, компонент может быть таким маленьким, как Plane (в Unreal отсутствует), который использовали для построения сетки. Новейшие дополнения движка – компоненты «UI» и «Layout», обеспечивающие создание мощных и масштабируемых графических пользовательских интерфейсов.

Редактор можно расширять собственными сценариями. Для этого можно создать какое-либо погодное условие или физические эффекты и сохранить это в формате «Prefab», и потом использовать данную настройку в любом проекте, кроме того, в Asset Store – это встроенный магазин, доступна масса ресурсов на все случаи жизни. Хранилище Asset Store также содержит множество полезных сценариев, моделей материалов и пр. Они будут особенно кстати при прототипировании – можно просто загрузить все необходимое в виде временных ресурсов и пользоваться этим как подручными имитационными моделями [28].

Unity был одним из первых общедоступных движков, поддерживавших мобильную разработку. Поэтому он очень удобен при развертывании в мобильной среде, выглядит и действует там практически так же, как и в редак-

торе. Система постоянно совершенствуется, и развертывание протекает очень гладко. Это был существенный фактор, благодаря которому решили делать мобильный прототип.

Пожалуй, Unity может похвастаться самым широким сообществом специалистов среди всех игровых движков, поэтому если у вас возникнет вопрос скорее всего, ответ на него найдется. Пусть Unity и поддерживает множество языков для написания сценариев, документация по каждому из них очень основательная [1].

В Unity проделана огромная работа по оптимизации рендеринга для множества однотипных объектов. Чтобы добиться сопоставимой производительности в Unreal, пришлось бы задействовать Instanced Rendering, а этот механизм обычно менее гибок, чем рендеринг в Unity.

Вывод

Использование виртуальной реальности открывает много новых возможностей в обучении и образовании, которые слишком сложны, затратные по времени или дороги при традиционных подходах, если не все одновременно.

Используя 3D-графику, можно детализировано показать химические процессы вплоть до атомного уровня. Причем ничто не запрещает углубиться еще дальше и показать, как внутри самого атома происходит деление ядра перед ядерным взрывом. Виртуальная реальность способна не только дать сведения о самом явлении, но и продемонстрировать его с любой степенью детализации.

Операция на сердце, управление сверхскоростным поездом, космическим шаттлом, техника безопасности при пожаре – можно погрузить зрителя в любое из этих обстоятельств без малейших угроз для жизни.

Виртуальная реальность позволяет менять сценарии, влиять на ход эксперимента или решать математическую задачу в игровой и доступной для

понимания форме. Во время виртуального урока можно увидеть мир прошлого глазами исторического персонажа, отправиться в путешествие по человеческому организму в микрокапсуле или выбрать верный курс на корабле Магеллана.

Виртуальный мир, который окружит зрителя со всех сторон на все 360 градусов, позволит целиком сосредоточиться на материале и не отвлекаться на внешние раздражители.

Вид от первого лица и ощущение своего присутствия в нарисованном мире – одна из главных особенностей виртуальной реальности. Это позволяет проводить уроки целиком в виртуальной реальности.

Как видно, технологии, казавшиеся фантастикой еще десяток лет назад, сегодня реальны и уже применимы в образовании: интеграция между предметами и взаимодействие образовательных заведений по сети; телемосты и различные видеоконференции; трехмерные электронные обучающие ресурсы, материалы для презентаций; виртуальные музеи, планетарии, залы для лекций, лаборатории и практикумы. Следует готовиться к тому, что в ближайшее время VR-технологии станут повседневностью.

На основании анализа требований, предъявляемых к VR-приложениям были выделены критерии к содержательному наполнению и интерфейсу VR-приложения, такие как ландшафтная ориентация экрана, запуск приложения на полный экран и простое взаимодействие с объектами с помощью взгляда, без необходимости доставать телефон из гарнитуры cardboard, которые были учтены при разработке.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Цель и назначение VR-приложения «Объемные геометрические фигуры – Geometrica»

Целью разработки данного приложения было познакомить учеников школы с разнообразными трехмерными геометрическими моделями, а также их основными характеристиками в рамках учебной дисциплины «Математика».

2.2 Описание основных функций приложения

VR-приложения «Объемные геометрические фигуры – Geometrica», является мобильным справочным, образовательным приложением. Приложение включает в себя следующие программные функции:

- геометрические модели разделены на разделы;
- каждую модель можно рассмотреть в режиме 3d;
- для каждой модели есть справочная информация, что это за модель и основные формулы по вычислению площади или объема.

2.3 Общее описание структуры и содержания приложения

Структура обучающего мобильного приложения состоит из одного основного блока.

Который поделен на 4 раздела, такие как главное меню, выбор категории геометрических моделей, сами сцены в которых представлена выставка данных моделей и осмотр в 3d геометрических фигур с небольшой справкой о фигуре.

2.4 Основные инструменты и технологии для реализации приложения

Приложение реализовано для мобильной платформы Android, разработанной Google. Android – это операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. В будущем планируется поддержка автомобилей и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android inc., которую затем купила Google [18].

Логика приложения, обработка событий на экране и взаимодействие с акселерометром разрабатывались на языке Java – объектно-ориентированный язык программирования, разработанный компанией Sun TGCsystems (в последующем приобретенной компанией Oracle). Дата официального выпуска – 23 мая 1995 года [30].

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые повышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Приложение «Объемные геометрические фигуры – Geometrica» разрабатывалось с помощью игрового движка Unity.

Unity — это инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и OS X. Созданные с помощью Unity приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One и MotionParallax3D дисплеях (устройства для воспроизведения виртуальных голограмм), например, Nettlebox. Есть возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity (Unity Web Player), а также с помощью реализации технологии WebGL. Ранее была экспериментальная поддержка реализации проектов в рамках модуля Adobe Flash Player, но позже команда разработчиков Unity приняла сложное решение по отказу от этого.

Геометрические фигуры моделировались с помощью программного пакета Autodesk 3ds Max.

Autodesk 3ds Max – полнофункциональная профессиональная программная система для создания и редактирования трёхмерной графики и анимации, разработанная компанией Autodesk. Содержит самые современные средства для художников и специалистов в области мультимедиа. Работает в операционных системах Windows и Windows NT (как в 32 битных, так и в 64 битных). В апреле 2017 года выпущена версия 20.0 этого продукта под названием Autodesk 3ds Max 2018.

2.5 Сценарий VR-приложения «Объемные геометрические фигуры»

Разработка любого обучающего ресурса начинается с продумывания его концепции, правил работы с ним, а также цели его создания и ожидаемого результата.

VR-приложения «Объемные геометрические фигуры – Geometrica», предназначено для детей школьного возраста с целью представить, как вы-

глядит та или иная геометрическая фигура, иметь возможность рассмотреть ее со всех сторон и понять ее свойства.

В связи с чем, первое, что было сделано при разработке VR-приложения «Объемные геометрические фигуры – Geometrica» это проектирование сценария, того каким образом обучающийся будет работать с данным обучающим средством.

Общая концепция такова, что в приложении можно подойти к любой из представленных фигур и прочесть, что это за фигура и при желании приблизить ее для рассмотрения, и посмотреть основную информацию о ней.

На рисунке 10 представлена блок схема обучающего приложения. При запуске приложения учащийся видит меню приложения расположенное в комнате. Далее пользователь выбирает, игровое поле с моделями, а на поле выбирает нужную модель для осмотра вблизи, под разными углами.

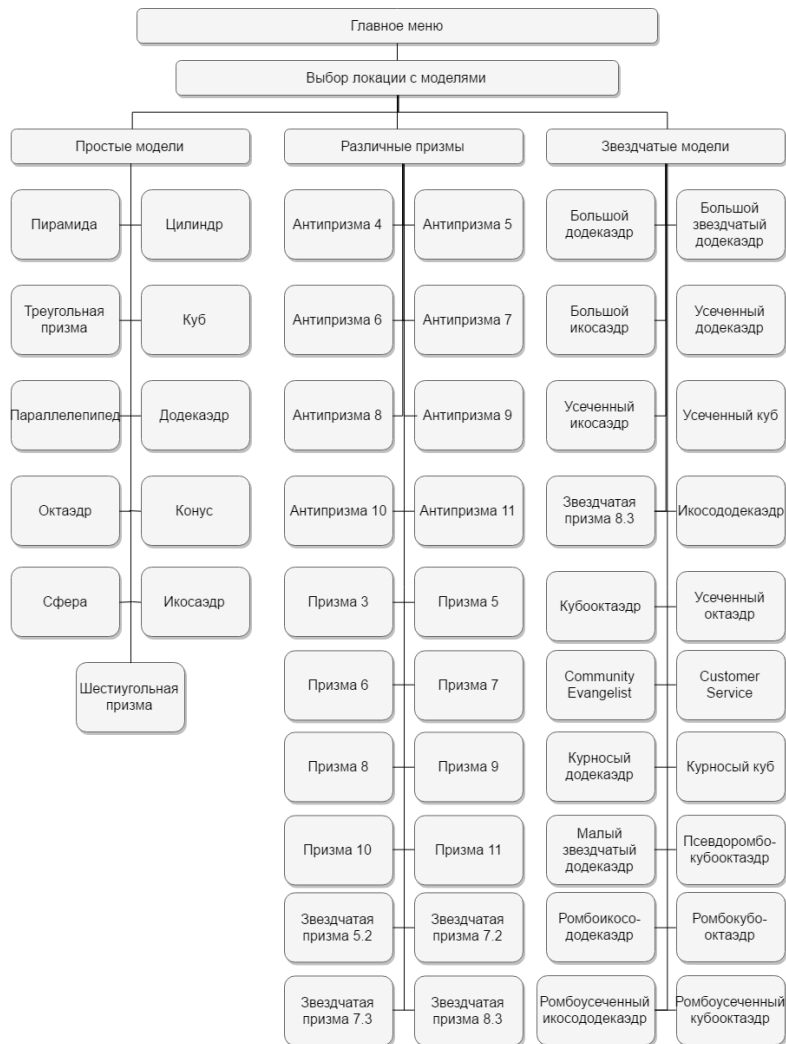


Рисунок 10 — Блок схема приложения

В данном приложении интуитивно понятный интерфейс, для того чтобы выбрать какую-либо кнопку, стоит просто посмотреть на нее 3 секунды, либо нажать кнопку на соответствующем пульте, если такой имеется.

2.6 Навигация по VR-приложению «Объемные геометрические фигуры»

Существует множество новых моделей навигации для мобильных приложений, но если использовать одно из распространенных, то нужно быть бдительным при выборе наиболее подходящей для данного приложения.

Если новый пользователь запутается или будет разочарован при попытке познакомиться с приложением, то интерес к нему достаточно быстро сойдет на нет. Если приложение обеспечивает комплексную функциональность, то можно включить «Помощь». Если добавить большой текст в качестве подсказки, это может означать, что пользовательский интерфейс требует доработки.

Итак, на основании вышесказанного, можно выделить необходимые требования для VR-приложения:

1. Приложение «не выкидывает» в 2D-интерфейс.

Google: «Можете ли вы использовать приложение без необходимости в извлечении смартфона из гарнитуры?» Также компания предостерегает от запросов на доступ к камере и контактам, потому что это вызовет 2D-диалог.

2. Приложение остается в ландшафтном режиме.

Google: «Остается ли приложение в ландшафтном режиме, когда телефон вращается?» Необходимо, чтобы телефон был всегда в ландшафтном режиме, даже если он находится в книжной форме.

3. Используется полноэкранный режим.

Google: «Не показывайте системные панели».

4. Пользователи могут сфокусироваться на отдельных объектах.

Google: «Избегайте размещения объектов так близко к пользователям, что они не могут сосредоточиться на них».

5. Приложение поддерживает отслеживание положения головы.

Google: «Приложение продолжает обновлять дисплей на основе движений даже во время загрузки сцены».

6. Сохраняется неизменная линия горизонта

Google: «Если приложение имеет видимую линию горизонта, она должна оставаться неизменной».

7. Именно пользователь инициирует движение.

Google: «Перемещение камеры должно быть инициировано пользователем».

В приложении «Объемные геометрические фигуры» учтены все необходимые требования для достижения комфортного пользования данным приложением. Для того чтобы приложение не вылетало в 2d режим, убраны какие-либо сообщения. Пользователю не понадобится извлекать телефон из гарнитуры, при загрузке сцены отслеживается положение головы, так же оно запускается в полноэкранный режим. И так как данное VR-приложение для гарнитуры, оно всегда остается в ландшафтном режиме.

2.7 Жизненный цикл создания педагогического продукта

2.7.1 Раздел главное меню

В среде Unity каждый уровень называется сценой. Так вот в сцене «Menu» присутствует источник света, комната состоит из куба, установлена камера, также используется само поле для картинки и кнопок, данная сцена с интерфейсом Unity изображена на рисунке 11.

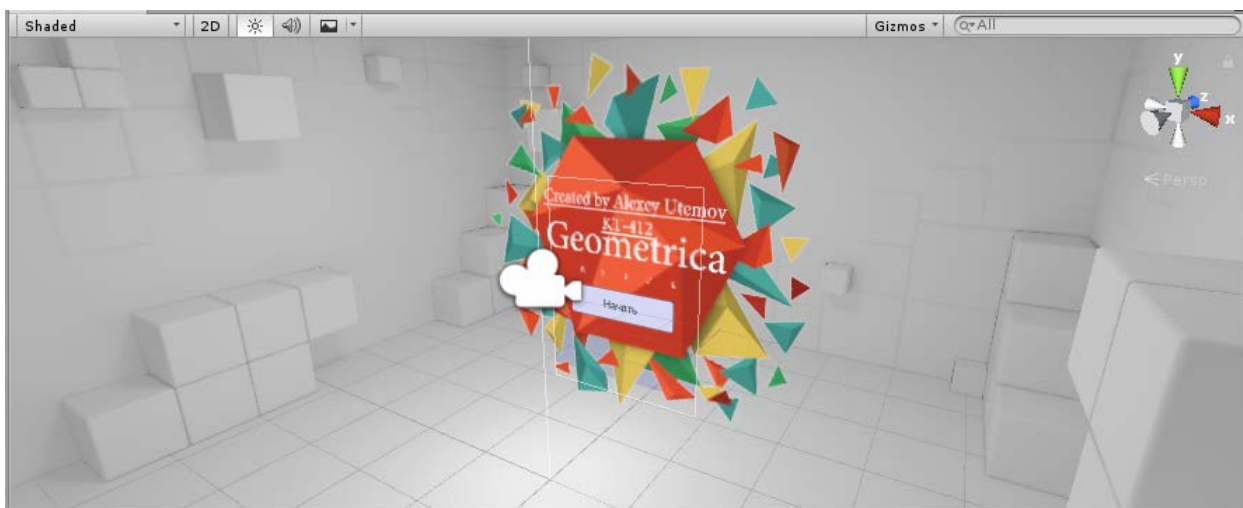


Рисунок 11 — Сцена «Menu»

2.7.2 Раздел меню с выбором рассматриваемых моделей

В сцене «Main menu» с выбором выставок геометрических фигур используются все те же объекты, как и в сцене «Main Menu». Но вместо кнопок для выбора используется изображение, данная сцена изображена на рисунке 12.

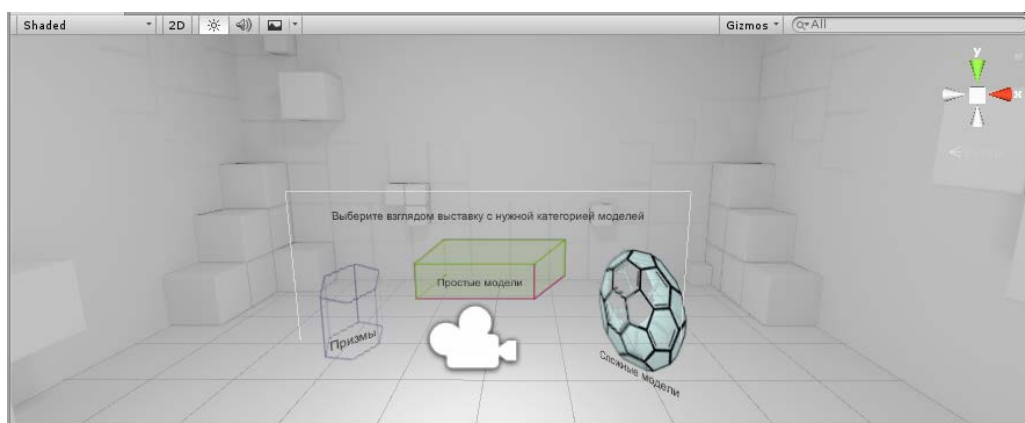


Рисунок 12 — Сцена «Menu»

2.7.3 Раздел создания сцены выставки фигур

В данном приложении существует 3 сцены с геометрическими фигурами, в данной сцене объекты поделены на группы:

- окружение;

- свет;
- игрок;
- геометрические фигуры.

Объекты окружения созданы из примитивов и стандартных объектов, трава, здания, деревья, дорога и тротуар созданы с помощью куба и плоскости. Объекты окружения показаны на рисунке 13.

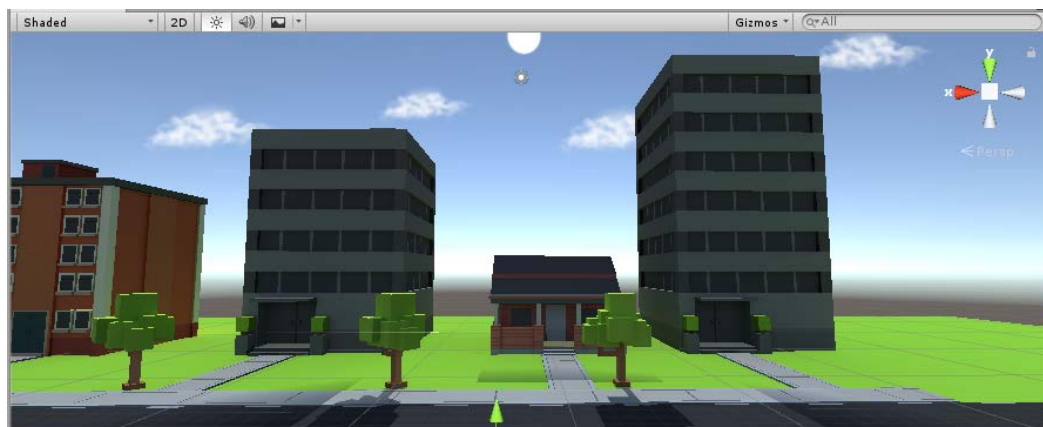


Рисунок 13 — Окружение

В группе «Свет» находятся источники света, глобальный (солнце) и в зависимости от сцены 2-3 источника света (лампы).

В группе «Игрок» находятся такие объекты как камера, объект капсула, с коллизией, чтобы игрок мог ходить по земле, не проваливаясь сквозь объекты. Также к объекту привязан музыкальный файл.

И так объекты в группе «Геометрические фигуры», состоят из подставок, табличек (кнопок) и сами фигуры. Подставки сделаны из куба где слегка закруглены углы и грани. Таблички созданы из текстуры золотой таблички и применены к стандартной кнопке.

И, так сами фигуры смоделированы с помощью программы Autodesk 3DS Max, интерфейс программы изображен на рисунке 14.

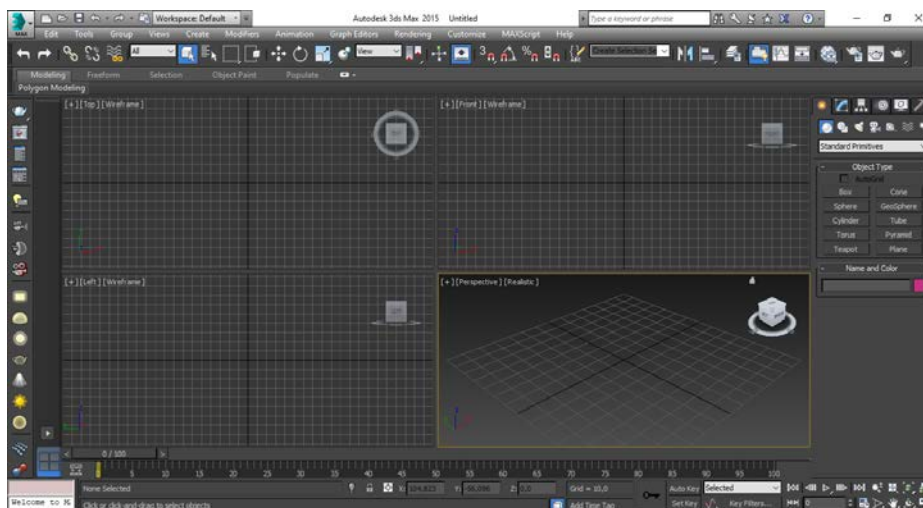


Рисунок 14 — Интерфейс программы «Autodesk 3D max»

Грани каждой модели покрашены в разный цвет, чтобы отчетливо было видно грани и ребра данной геометрической фигуры, пример модели изображен на рисунке 15.

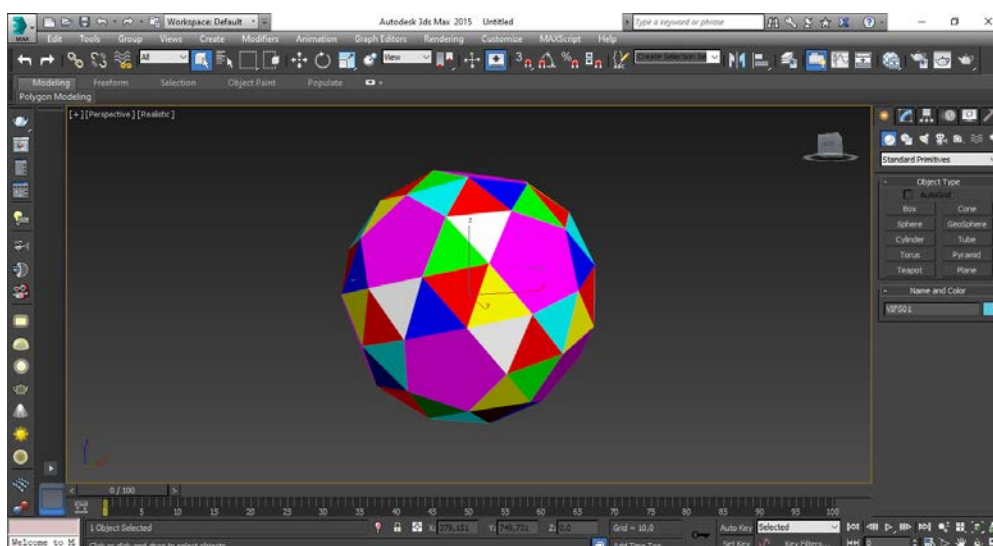


Рисунок 15 — Геометрическая модель «Курносый додекаэдр»

2.7.4 Сцена с осмотром модели и информации о фигуре

На данный момент количество моделей превышает 45 штук. Как было описано выше, окружение было взято с главного меню, добавлена информационная доска, перед камерой расположена определенная модель, которую требуется рассмотреть, одна из сцен изображена на рисунке 16.

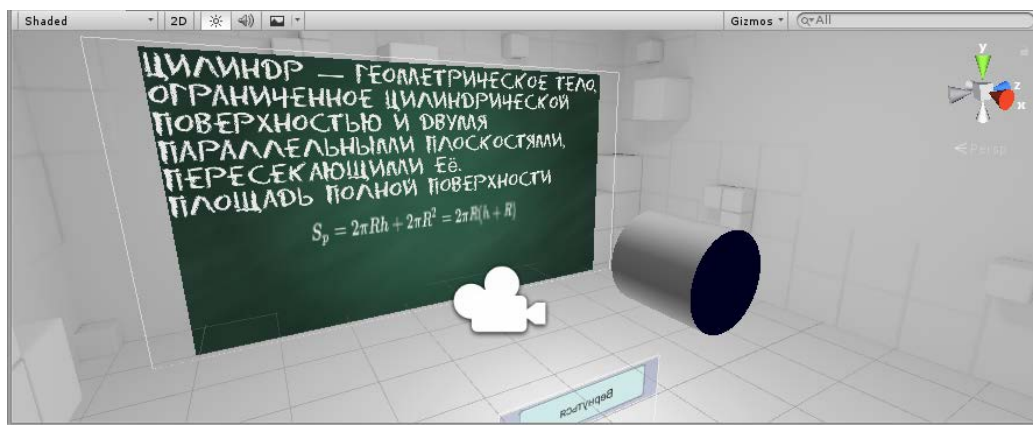


Рисунок 16 — Сцена осмотра модели с информацией цилиндра

Также на следующем рисунке 17 изображен додекаэдр.

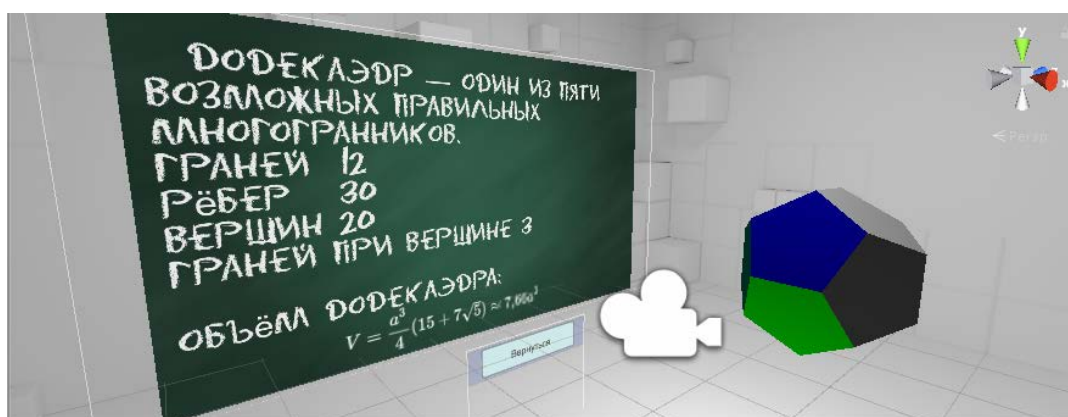


Рисунок 17 — Сцена с моделью «додекаэдр»

2.7.5 Сцена загрузки

Сцена «LoadScene» это промежуточная «комната», которая появляется в момент загрузки всех элементов и моделей в игровом пространстве. В данной сцене нет кнопок. В данной «комнате» используется сфера с текстурой музея в 360 градусов, для эффекта погружения. Данная сцена изображена на рисунке 18.



Рисунок 18 — Сцена «LoadScene»

2.7.6 Используемые скрипты

В приложении также используются скрипты, написанные на языке C#. Основной скрипт связки гироскопа в телефоне и камеры в приложении бесплатно распространяет Google, так как технология и гарнитуры Cardboard этой же компании.

Также в приложении используются такие скрипты как:

1. Передвижение персонажа, скрипт изображен на рисунке 19.

```

4     [RequireComponent (typeof (CharacterController))]
5     [System.Serializable]
6     public class Navigation : MonoBehaviour {
7
8
9         public static float pitch;
10        public float forwardLowerBound = 20;
11        public float forwardUpperBound = 30;
12        public float forwardLimit = 50;
13        public float backwardLowerBound = 20;
14        public float backwardUpperBound = 30;
15        public float backwardLimit = 50;
16
17        public static float ForwardLowerBound;
18        public static float ForwardUpperBound;
19        public static float BackwardLowerBound ;
20        public static float BackwardUpperBound;
21        public static float ForwardLimit;
22        public static float BackwardLimit;
23
24
25        private CharacterController controller;
26        private bool moving;
27        private GameObject head;
28
29
30        public void Start () {
31            ForwardLowerBound = forwardLowerBound;
32            ForwardUpperBound = forwardUpperBound;
33            BackwardLowerBound = 360 - backwardLowerBound;
34            BackwardUpperBound = 360 - backwardUpperBound;

```

Рисунок 19 — Скрипт «Движение наклоном головы»

2. Переход между сценами, скрипт изображен на рисунке 20.

```
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3  using System.Collections;
4
5
6  public class Prost : MonoBehaviour {
7
8      public void PalyGame() {
9          SceneManager.LoadScene("Prost");
10     }
11 }
12
```

Рисунок 20 — Скрипт «Загрузка сцен»

3. Вращение геометрических фигур, скрипт изображен на рисунке 21.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Rotate : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9
10     }
11
12     // Update is called once per frame
13     void Update () {
14         //transform.RotateAround(Vector3.zero, Vector3.up, 1 * Time.deltaTime);
15         transform.Rotate(Vector3.left * 20 * Time.deltaTime);
16     }
17 }
18
```

Рисунок 21 — Скрипт «Вращение объектов»

4. Действие взглядом, скрипт изображен на рисунке 22.

```
7  public class LookButton : MonoBehaviour {
8      public float time = 2f;
9      private float timer;
10     private bool gaze;
11     void Start () {
12     }
13     void Update () {
14         if (gaze)
15         {
16             timer += Time.deltaTime;
17             if (timer >= time)
18             {
19                 ExecuteEvents.Execute(gameObject, new PointerEventData(EventSystem.current), ExecuteEvents.pointerDownHandler);
20                 timer = 0f;
21             }
22         }
23     }
24     public void PointerEnter()
25     {
26         gaze = true;
27     }
28     public void PointerExit()
29     {
30         gaze = false;
31         timer = 0f;
32     }
33     public void PointerDown()
34     { }
35 }
36
```

Рисунок 22 — Скрипт «Выбор взглядом»

2.8 Методические указания по использованию педагогического продукта в учебном процессе

2.8.1 Установка и запуск программы

Данное приложение имеет стандартное расширение для Android «APK», для установки понадобится телефон на операционной системе Android версии не ниже 4.4. Скачиваем файл на телефон и запускаем, после чего ярлык игры появится в меню приложений. После, можно запускать приложение «Geometrica», ярлык приложения показан на рисунке 23.

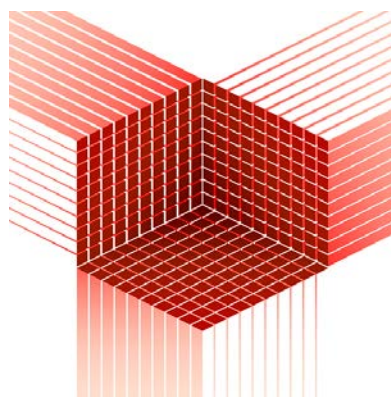


Рисунок 23 — Иконка приложения «Объемные геометрические фигуры – Geometrica»

2.8.2 Навигация в приложении

В данном приложении «Объемные геометрические фигуры – Geometrica» очень простое управление, поворотом головы крутится камера. При наклоне головы в 45 градусов вперед, персонаж идет вперед, также если наклонить назад, персонаж соответственно пойдет назад. На экране существует белая точка, при наведении точки на объект, с которым можно взаимодействовать, точка увеличивается и превращается в круг. В этом состоянии начинается отчет времени, после истечения 3 секунд эмитируется нажатие кнопки.

2.9 Результат апробации и внедрения

Данный продукт был испробован 10 людьми, которые принимали независимое тестирование приложения. Восемь человек сказали, что данное приложение интересно для справки и данный интерфейс быстро осваивается.

Вывод

Было разработано обучающее VR-приложение для учеников школьного возраста, обучающихся в 11 классе. С целью представить, как выглядит та или иная геометрическая фигура, иметь возможность рассмотреть ее со всех сторон и понять ее свойства. В данном приложении используются программные функции такие как, геометрические модели разделены на разделы, каждую модель можно рассмотреть в режиме 3d, для каждой модели есть справочная информация, что это за модель и основные формулы по вычислению площади или объема. Для реализации проекта, использовались программные пакеты как Autodesk 3d max, Blender, Unity, Microsoft Visual Studio.

ЗАКЛЮЧЕНИЕ

Использование виртуальной реальности открывает много новых возможностей в обучении и образовании, которые слишком сложны, затратные по времени или дороги при традиционных подходах, если не все одновременно.

Используя 3D-графику, можно детализировано показать химические процессы вплоть до атомного уровня. Причем ничто не запрещает углубиться еще дальше и показать, как внутри самого атома происходит деление ядра перед ядерным взрывом. Виртуальная реальность способна не только дать сведения о самом явлении, но и продемонстрировать его с любой степенью детализации.

Операция на сердце, управление сверхскоростным поездом, космическим шаттлом, техника безопасности при пожаре – можно погрузить зрителя в любое из этих обстоятельств без малейших угроз для жизни.

Виртуальная реальность позволяет менять сценарии, влиять на ход эксперимента или решать математическую задачу в игровой и доступной для понимания форме. Во время виртуального урока можно увидеть мир прошлого глазами исторического персонажа, отправиться в путешествие по человеческому организму в микрокапсуле или выбрать верный курс на корабле Магеллана.

Виртуальный мир, который окружит зрителя со всех сторон на все 360 градусов, позволит целиком сосредоточиться на материале и не отвлекаться на внешние раздражители.

Вид от первого лица и ощущение своего присутствия в нарисованном мире – одна из главных особенностей виртуальной реальности. Это позволяет проводить уроки целиком в виртуальной реальности.

Как видно, технологии, казавшиеся фантастикой еще десяток лет назад, сегодня реальны и уже применимы в образовании: интеграция между пред-

метами и взаимодействие образовательных заведений по сети; телемосты и различные видеоконференции; трехмерные электронные обучающие ресурсы, материалы для презентаций; виртуальные музеи, планетарии, залы для лекций, лаборатории и практикумы. Следует готовиться к тому, что в ближайшее время VR-технологии станут повседневностью.

В рамках дипломной работы были рассмотрены вопросы разработки сценария, интерфейса и непосредственно самого обучающего VR-приложения «Объемные геометрические фигуры – Geometrica» с помощью игровой платформы Unity.

На первом этапе работы было рассмотрено понятие VR-технология и практическое применение виртуальной реальности в образовании, далее подробно разобран выбор игровой платформы Unity для реализации VR-приложения.

На следующем этапе был представлен сценарий VR-приложения «Объемные геометрические фигуры – Geometrica», который приведен в виде блок-схемы, демонстрирующий процесс работы и основные возможности приложения.

На основании анализа требований, предъявляемых к VR-приложениям были выделены критерии к содержательному наполнению и интерфейсу VR-приложения, такие как ландшафтная ориентация экрана, запуск приложения на полный экран и простое взаимодействие с объектами с помощью взгляда, без необходимости доставать телефон из гарнитуры Cardboard, которые были учтены при разработке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гончаров А. А. Самоучитель 3ds Max 2016: основы моделирования [Текст] / А. А. Гончаров. – Санкт Петербург: БХВ-Петербург, 2016. – 528 с.
2. Аудио [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/Audio.html> (дата обращения: 28.05.2017).
3. Биркгофф Г.С. Математика и психология [Текст] / Г.С Биркгофф. – Москва: ЛКИ, 2008. – 529 с.
4. Все о создании игр [Электронный ресурс]. – Режим доступа: <http://gamesmaker.ru/books/osnovy-razrabotki-igr-na-unity-3d/> (дата обращения: 28.05.2017).
5. Гессен С.И. Основы педагогики. Введение в прикладную философию [Текст] / С.И. Гессен. – Москва: Школа-Пресс, 1995. – 448 с.
6. Интерактивный учебник по Visual Studio [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/bb514232\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/bb514232(v=vs.90).aspx) (дата обращения: 28.05.2017).
7. Камеры в Unity [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/CamerasOverview.html> (дата обращения: 28.05.2017).
8. Ландшафтный движок [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/script-Terrain.html> (дата обращения: 28.05.2017).
9. Новости VR индустрии [Электронный ресурс]. – Режим доступа: https://vrgeek.ru/2016/07/21/2467_obrazovanie-v-vr/ (дата обращения: 25.03.2017).
10. Оптимизация графики [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/OptimizingGraphicsPerformance.html> (дата обращения: 28.05.2017).

11. Официальный мануал Unity [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/index.html> (дата обращения: 28.05.2017).
12. Официальный мануал от Google [Электронный ресурс]. – Режим доступа: <https://developers.google.com/vr/unity/> (дата обращения: 28.05.2017).
13. Статья о VR-технологиях [Электронный ресурс]. – Режим доступа: [mylsa](http://mylsa.ru) (дата обращения: 25.03.2017).
14. Уроки 3ds max [Электронный ресурс]. – Режим доступа: <http://junior3d.ru/index.html> (дата обращения: 28.05.2017).
15. Уроки и документация для Unity3d [Электронный ресурс]. – Режим доступа: <http://gamesmaker.ru/3d-game-engines/unity3d/> (дата обращения: 28.05.2017).
16. Уроки по моделированию в 3D MAX [Электронный ресурс]. – Режим доступа: <http://www.3dmax-tutorials.ru/> (дата обращения: 28.05.2017).
17. Физика [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/PhysicsSection.html> (дата обращения: 28.05.2017).
18. Школа-студия визуализации [Электронный ресурс]. – Режим доступа: <http://visschool.ru/> (дата обращения: 28.05.2017).
19. 3D-моделирование [Электронный ресурс]. – Режим доступа: <http://younglinux.info/blender.php> (дата обращения: 28.05.2017).
20. 4Brain [Электронный ресурс]. – Режим доступа: <https://4brain.ru/pedagogika/principy-didaktiki.php> (дата обращения: 15.04.2017).
21. Google VR SDK for Unity [Электронный ресурс]. – Режим доступа: <https://developers.google.com/vr/unity/guide> (дата обращения: 28.05.2017).
22. Interactive patterns [Электронный ресурс]. – Режим доступа: <https://www.google.com/design/spec-vr/interactive-patterns/controls.html#> (дата обращения: 28.05.2017).

23. Lighting Overview [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/Lighting.html> (дата обращения: 28.05.2017).

24. Materials, Shaders & Textures [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/Shaders.html> (дата обращения: 28.05.2017).

25. Render [Электронный ресурс]. – Режим доступа: <http://render.ru/> (дата обращения: 28.05.2017).

26. TProger [Электронный ресурс]. – Режим доступа: <https://tproger.ru/tag/unity/> (дата обращения: 28.05.2017).

27. Unity SDK Plugin Contents [Электронный ресурс]. – Режим доступа: <https://developers.google.com/vr/unity/plugin> (дата обращения: 28.05.2017).

28. Unity3d по русски [Электронный ресурс]. – Режим доступа: <http://unity3d.ru/distribution/viewforum.php?f=11> (дата обращения: 28.05.2017).

29. VRmania [Электронный ресурс]. – Режим доступа: <http://vrmania.ru/vr/virtualnaya-realnost-v-obrazovanii.html> (дата обращения: 28.05.2017).

30. Working In Unity [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/current/Manual/UnityOverview.html> (дата обращения: 28.05.2017).

ПРИЛОЖЕНИЕ 1

**Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования**

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
направление 44.03.04 Профессиональное обучение (по отраслям)
профиль «Информатика и вычислительная техника»
профилизация «Компьютерные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Н. С. Толстова

« ____ » _____ 2016 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студента 4 курса, группы КТ-412 Утемов Алексей Юрьевич

1. Тема Обучающее VR-приложение в среде Unity
утверждена распоряжением по институту от 07.02.2017 г. № 073.
2. Руководитель Чернякова Татьяна Викторовна, к.пед.н., доцент
3. Место преддипломной практики ФГАОУ ВО РГППУ кафедра ИС
4. Исходные данные к ВКР Рабочая программа по математике 10-11 классы (Мордкович, Атанасян), официальное руководство пользователя программы Unity, официальная документация Google по программированию VR-приложений, сайт с базой курсов по разработке игр.
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
Проанализировать литературу и интернет-источники.
Проанализировать аспекты использования виртуальной реальности в обучении.
Разработать сценарий и общую концепцию обучающего VR-приложения.
Разработать обучающее VR-приложение в среде Unity.
6. Перечень демонстрационных материалов
Презентация, выполненная средствами Microsoft PowerPoint
Обучающее VR-приложение

7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
1	Сбор информации по выпускной работе и сдача зачета по преддипломной практике	25.05.2017	15	
2	Выполнение работ по разрабатываемым вопросам их изложение в выпускной работе:			
	Анализ литературы и интернет-источники.	18.04.2017	5	
	Анализ аспектов использования виртуальной реальности в обучении.	25.04.2017	5	
	Разработка сценария и общей концепции обучающего VR-приложения.	03.05.2017	20	
	Разработка обучающего VR-приложения в среде Unity.	02.06.2017	35	
3	Оформление текстовой части ВКР	05.06.2017	5	
4	Выполнение демонстрационных материалов к ВКР	10.06.2017	5	
5	Нормоконтроль	12.06.2017	5	
6	Подготовка доклада к защите в ГЭК	19.06.2017	5	

8. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
		подпись	дата	подпись	дата

Руководитель _____
подпись дата

Задание получил _____
подпись студента дата

9. Выпускная квалификационная работа и все материалы проанализированы. Считаю возможным допустить Утемова Алексея Юрьевича к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель _____
подпись дата

10. Допустить Утемова Алексея Юрьевича к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от 14.06.2017 №12)

Заведующий кафедрой _____
подпись дата

ПРИЛОЖЕНИЕ 2

Скрипт взят с сайта Google, отслеживание положения телефона в пространстве, показан ниже на рисунках.

```
using UnityEngine;
using System;
using System.Collections.Generic;

using Gvr.Internal;

/// The GvrViewer object communicates with the head-mounted display.
/// Is is responsible for:
/// - Querying the device for viewing parameters
/// - Retrieving the latest head tracking data
/// - Providing the rendered scene to the device for distortion correction (optional)
///
/// There should only be one of these in a scene. An instance will be generated automatically
/// by this script at runtime, or you can add one via the Editor if you wish to customize
/// its starting properties.
[AddComponentMenu("GoogleVR/GvrViewer")]
public class GvrViewer : MonoBehaviour {
    public const string GVR_SDK_VERSION = "1.30";

    /// The singleton instance of the GvrViewer class.
    public static GvrViewer Instance {
        get {
            #if UNITY_EDITOR
                if (instance == null && !Application.isPlaying) {
                    instance = UnityEngine.Object.FindObjectOfType<GvrViewer>();
                }
            #endif
            if (instance == null) {
                Debug.LogError("No GvrViewer instance found. Ensure one exists in the scene, or call "
                    + "GvrViewer.Create() at startup to generate one.\n"
                    + "If one does exist but hasn't called Awake() yet, "
                    + "then this error is due to order-of-initialization.\n"
                    + "In that case, consider moving "
                    + "your first reference to GvrViewer.Instance to a later point in time.\n"
                    + "If exiting the scene, this indicates that the GvrViewer object has already "
                    + "been destroyed.");
            }
            return instance;
        }
    }
    private static GvrViewer instance = null;

    /// Generate a GvrViewer instance. Takes no action if one already exists.
    public static void Create() {
        if (instance == null && UnityEngine.Object.FindObjectOfType<GvrViewer>() == null) {
            Debug.Log("Creating GvrViewer object");
            var go = new GameObject("GvrViewer", typeof(GvrViewer));
            go.transform.localPosition = Vector3.zero;
            // sdk will be set by Awake().
        }
    }

    /// The StereoController instance attached to the main camera, or null if there is none.
    /// @note Cached for performance.
    public static StereoController Controller {
        get {
            #if !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
                Camera camera = Camera.main;
                // Cache for performance, if possible.
                if (camera != currentMainCamera || currentController == null) {
                    currentMainCamera = camera;
                    currentController = camera.GetComponent<StereoController>();
                }
            #endif
            return currentController;
        }
    }
}
#endif
```

Рисунок 24 — Начало скрипта

```

        return null;
    #endif // !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
    }
}
private static StereoController currentController;
private static Camera currentMainCamera;

/// Determine whether the scene renders in stereo or mono.
/// Supported only for versions of Unity *without* the GVR integration.
/// VRModeEnabled will be a no-op for versions of Unity with the GVR integration.
/// _True_ means to render in stereo, and _false_ means to render in mono.
public bool VRModeEnabled {
    get {
    #if !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
        return vrModeEnabled;
    #else
        return UnityEngine.VR.VRSettings.enabled;
    #endif // !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
    }
    set {
    #if !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
        if (value != vrModeEnabled && device != null) {
            device.SetVRModeEnabled(value);
        }
        vrModeEnabled = value;
    #endif // !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
    }
}

/// Ignore private field is assigned but its value is never used compile warning.
#pragma warning disable 414
[SerializeField]
private bool vrModeEnabled = true;
#pragma warning restore 414

/// Methods for performing lens distortion correction.
public enum DistortionCorrectionMethod {
    None, ///< No distortion correction
    Native, ///< Use the native C++ plugin
    Unity, ///< Perform distortion correction in Unity (recommended)
}

/// Determines the distortion correction method used by the SDK to render the
/// #StereoScreen texture on the phone. If _Native_ is selected but not supported
/// by the device, the _Unity_ method will be used instead.
public DistortionCorrectionMethod DistortionCorrection {
    get {
        return distortionCorrection;
    }
    set {
        if (device != null && device.RequiresNativeDistortionCorrection()) {
            value = DistortionCorrectionMethod.Native;
        }
        if (value != distortionCorrection && device != null) {
            device.SetDistortionCorrectionEnabled(value == DistortionCorrectionMethod.Native
                && NativeDistortionCorrectionSupported);
            device.UpdateScreenData();
        }
        distortionCorrection = value;
    }
}
[SerializeField]
#if UNITY_HAS_GOOGLEVR && UNITY_ANDROID && !UNITY_EDITOR
private DistortionCorrectionMethod distortionCorrection = DistortionCorrectionMethod.Native;
#else
private DistortionCorrectionMethod distortionCorrection = DistortionCorrectionMethod.Unity;
#endif // UNITY_HAS_GOOGLEVR && UNITY_ANDROID && !UNITY_EDITOR

/// The native SDK will apply a neck offset to the head tracking, resulting in
/// a more realistic model of a person's head position. This control determines
/// the scale factor of the offset. To turn off the neck model, set it to 0, and
/// to turn it all on, set to 1. Intermediate values can be used to animate from
/// on to off or vice versa.
public float NeckModelScale {
    get {
        return neckModelScale;
    }
    set {
        value = Mathf.Clamp01(value);
        if (!Mathf.Approximately(value, neckModelScale) && device != null) {
            device.SetNeckModelScale(value);
        }
        neckModelScale = value;
    }
}
[SerializeField]
private float neckModelScale = 0.0f;

#if UNITY_EDITOR
/// Restores level head tilt in when playing in the Unity Editor after you
/// release the Ctrl key.
public bool autoUntiltHead = true;

/// @cond
/// Use unity remote as the input source.
public bool UseUnityRemoteInput = false;
/// @endcond

/// The screen size to emulate when testing in the Unity Editor.
public GvrProfile.ScreenSizes ScreenSize {

```

Рисунок 25 — Продолжение скрипта

```

public GvrProfile.ScreenSizes ScreenSize {
    get {
        return screenSize;
    }
    set {
        if (value != screenSize) {
            screenSize = value;
            if (device != null) {
                device.UpdateScreenData();
            }
        }
    }
}
[SerializeField]
private GvrProfile.ScreenSizes screenSize = GvrProfile.ScreenSizes.Nexus5;

/// The viewer type to emulate when testing in the Unity Editor.
public GvrProfile.ViewerTypes ViewerType {
    get {
        return viewerType;
    }
    set {
        if (value != viewerType) {
            viewerType = value;
            if (device != null) {
                device.UpdateScreenData();
            }
        }
    }
}
[SerializeField]
private GvrProfile.ViewerTypes viewerType = GvrProfile.ViewerTypes.CardboardMay2015;
#endif

/// The VR device that will be providing input data.
private static BaseVRDevice device;

/// Whether native distortion correction functionality is supported by the VR device.
public bool NativeDistortionCorrectionSupported { get; private set; }

/// Whether the VR device supports showing a native UI layer, for example for settings.
public bool NativeUILayerSupported { get; private set; }

/// Scales the resolution of the #StereoScreen. Set to less than 1.0 to increase
/// rendering speed while decreasing sharpness, or greater than 1.0 to do the
/// opposite.
public float StereoScreenScale {
    get {
        return stereoScreenScale;
    }
    set {
        value = Mathf.Clamp(value, 0.1f, 10.0f); // Sanity.
        if (stereoScreenScale != value) {
            stereoScreenScale = value;
            StereoScreen = null;
        }
    }
}
[SerializeField]
private float stereoScreenScale = 1;

/// The texture that Unity renders the scene to. After the frame has been rendered,
/// this texture is drawn to the screen with a lens distortion correction effect.
/// The texture size is based on the size of the screen, the lens distortion
/// parameters, and the #StereoScreenScale factor.
public RenderTexture StereoScreen {
    get {
        // Don't need it except for distortion correction.
        if (distortionCorrection == DistortionCorrectionMethod.None || !VRModeEnabled) {
            return null;
        }
        if (stereoScreen == null) {
            // Create on demand.
            StereoScreen = device.CreateStereoScreen(); // Note: uses set{}
        }
        return stereoScreen;
    }
    set {
        if (value == stereoScreen) {
            return;
        }
        if (stereoScreen != null) {
            stereoScreen.Release();
        }
        stereoScreen = value;
        if (OnStereoScreenChanged != null) {
            OnStereoScreenChanged(stereoScreen);
        }
    }
}
}

```

Рисунок 26 — Продолжение скрипта

```

public void UpdateState() {
    if (updatedToFrame != Time.frameCount) {
        updatedToFrame = Time.frameCount;
        device.UpdateState();

        if (device.profileChanged) {
            if (distortionCorrection != DistortionCorrectionMethod.Native &&
                device.RequiresNativeDistortionCorrection()) {
                DistortionCorrection = DistortionCorrectionMethod.Native;
            }
            if (stereoScreen != null &&
                device.ShouldRecreateStereoScreen(stereoScreen.width, stereoScreen.height)) {
                StereoScreen = null;
            }
        }

        DispatchEvents();
    }
}

private void DispatchEvents() {
    // Update flags first by copying from device and other inputs.
    Triggered = Input.GetMouseButtonDown(0);
#if UNITY_HAS_GOOGLEVR && (UNITY_ANDROID || UNITY_EDITOR)
    Triggered |= GvrController.ClickButtonDown;
#endif // UNITY_HAS_GOOGLEVR && (UNITY_ANDROID || UNITY_EDITOR)

    Tilted = device.tilted;
    ProfileChanged = device.profileChanged;
    BackButtonPressed = device.backButtonPressed || Input.GetKeyDown(KeyCode.Escape);
    // Reset device flags.
    device.tilted = false;
    device.profileChanged = false;
    device.backButtonPressed = false;
}

/// Presents the #StereoScreen to the device for distortion correction and display.
/// @note This function is only used if #DistortionCorrection is set to _Native_,
/// and it only has an effect if the device supports it.
public void PostRender(RenderTexture stereoScreen) {
    if (NativeDistortionCorrectionSupported && stereoScreen != null && stereoScreen.IsCreated()) {
        device.PostRender(stereoScreen);
    }
}

/// Resets the tracker so that the user's current direction becomes forward.
public void Recenter() {
    device.Recenter();
}

/// Launch the device pairing and setup dialog.
public void ShowSettingsDialog() {
    device.ShowSettingsDialog();
}

#if !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
/// Add a StereoController to any camera that does not have a Render Texture (meaning it is
/// rendering to the screen).
public static void AddStereoControllerToCameras() {
    for (int i = 0; i < Camera.allCameras.Length; i++) {
        Camera camera = Camera.allCameras[i];
        if (camera.targetTexture == null &&
            camera.cullingMask != 0 &&
            camera.GetComponent<StereoController>() == null &&
            camera.GetComponent<GvrEye>() == null &&
            camera.GetComponent<GvrPreRender>() == null &&
            camera.GetComponent<GvrPostRender>() == null) {
            camera.gameObject.AddComponent<StereoController>();
        }
    }
}
#endif // !UNITY_HAS_GOOGLEVR || UNITY_EDITOR

void OnEnable() {
#if UNITY_EDITOR
    // This can happen if you edit code while the editor is in Play mode.
    if (device == null) {
        InitDevice();
    }
#endif
    device.OnPause(false);
}

void OnDisable() {
    device.OnPause(true);
}

void OnApplicationPause(bool pause) {
    device.OnPause(pause);
}

void OnApplicationFocus(bool focus) {
    device.OnFocus(focus);
}

void OnApplicationQuit() {
    device.OnApplicationQuit();
}

void OnDestroy() {
#if !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
    VRModeEnabled = false;
#endif // !UNITY_HAS_GOOGLEVR || UNITY_EDITOR
    if (device != null) {
        device.Destroy();
    }
    if (instance == this) {
        instance = null;
    }
}
}

```

Рисунок 27 — Завершение скрипта