

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»
Высшая школа информационных технологий и автоматизированных систем
(наименование высшей школы / филиала / института)

Орлов Данил Денисович
(ФИО обучающегося)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

09.03.03 «Прикладная информатика»

(код и наименование направления подготовки / специальности)

Разработка Web и мультимедийных приложений

(наименование направленности образовательной программы (профиля / специальности / название магистерской программы))

Разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта»

(тема ВКР)

Утверждена приказом от 7 декабря 2022 года № 3023-А

Руководитель ВКР

Деменкова Е.А., к.т.н.,
доцент кафедры ИСиИБ

Консультант

Пархимович М.Н., старший
преподаватель кафедры
ИСиИБ

Нормоконтроль

Деменкова Е.А., к.т.н.,
доцент кафедры ИСиИБ

Руководитель ОПОП

Кочкин С.А., к.ф.-м.н.,
доцент кафедры ВиПМ

(дата)

(подпись)

(ФИО, должность / степень / звание)

Постановление ГЭК от «__» _____ 20__ г.

Признать, что обучающийся Д.Д. Орлов

выполнил и защитил ВКР с отметкой

(отметка прописью)

Председатель ГЭК

(подпись)

А.С. Проскуряков
(инициалы, фамилия)

Секретарь ГЭК

(подпись)

А.О. Яковленкова
(инициалы, фамилия)

Архангельск 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»

Высшая школа информационных технологий и автоматизированных систем
(наименование высшей школы / филиала / института)

ЗАДАНИЕ ДЛЯ ПОДГОТОВКИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

09.03.03 «Прикладная информатика»
(код и наименование направления подготовки / специальности)

Тема ВКР: Разработка модулей виртуальной и дополненной реальности для
печатных изданий на примере романа «Ключ от лабиринта»

Утверждена протоколом заседания кафедры от «07» декабря 2022 г. № 7

Обучающемуся (-ейся):

Орлову Данилу Денисовичу
(Ф.И.О.)

Курс: 4

Группа: 351915

Срок сдачи выпускником законченной работы: «16» июня 2023 г.

Исходные данные к работе Требования заказчика на разработку, официальная
документация A-frame, ARCore, ARkit, AR.js, MindAR,

Основные разделы работы с указанием вопросов, подлежащих рассмотрению
Анализ текущего уровня развития технологий VR и AR, основные тенденции и сферы
применения. Выбор инструментов и технологий для реализации модулей VR/AR в
печатных изданиях. Разработка конструктора для генерации стартовых страниц модулей
виртуальной и дополненной реальности и тестирования разрабатываемых моделей.
Разработка VR сцен для романа «Ключ от лабиринта»

База проведения исследований Digital-агентство F5 (ИП Овсянкин И.П.)

Перечень обязательных приложений к работе

Приложение А Листинг программного кода

Перечень графического материала

Консультанты по работе

по разделу №3	_____	_____	<u>Пархимович М.Н., ст. преподаватель</u>
по разделу	_____	_____	_____
по разделу	_____	_____	_____
	(дата)	(подпись)	(ФИО, должность)

Дата выдачи задания «15» сентября 2022 г.

Руководитель ВКР	_____	<u>Е.А. Деменкова</u>
	(подпись)	(инициалы, фамилия)

Задание принял к исполнению «15» сентября 2022 г.

Обучающийся (-аяся)	_____	<u>Орлов Д.Д.</u>
---------------------	-------	-------------------

РЕФЕРАТ

Орлов Д.Д. Разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта».

Руководитель ВКР – доцент кафедры информационных систем и информационной безопасности, кандидат технических наук Деменкова Е.А.
Консультант ВКР – старший преподаватель кафедры информационных систем и информационной безопасности Пархимович М.Н.

Выпускная квалификационная работа объемом 76 страниц содержит 26 рисунков, 2 таблицу, 36 источника, 1 приложение.

Ключевые слова: AR, VR, A-frame, ARCore, ARkit, модули виртуальной и дополненной реальности для печатных изданий.

Целью данной выпускной квалификационной работы является разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта».

Структура ВКР: состоит из введения, трех глав, заключения, списка использованных источников и приложения.

Во введении обоснована актуальность темы, поставлена цель и задачи выпускной квалификационной работы, определены объект и предмет исследования.

В первой главе исследованы технологии AR и VR, их текущий уровень развития и доступности, сферы применения.

Во второй главе проведен анализ существующих технологий и сред разработки для реализации AR и VR модулей.

В третьей главе описаны этапы разработки конструктора AR и VR сцен, модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта».

В заключении подведены итоги выполненной работы, сформированы окончательные выводы по рассматриваемой теме.

Дата _____

Подпись _____

ОГЛАВЛЕНИЕ

Определения, обозначения и сокращения	6
Введение.....	7
1 Дополненная и виртуальная реальность	9
1.1 Понятие дополненной реальности	9
1.2 Понятие виртуальной реальности	11
1.3 Мировые тенденции в области развития технологий VR/AR.....	12
1.4 Сферы применения технологий AR и VR	14
2 Выбор технологий и инструментов реализации VR/AR.....	17
2.1 Исследование технологий AR с отслеживанием плоскости	17
2.1.1 Фреймворк A-frame	17
2.1.2 Платформа ARCore и технологии WebXR, model-viewer, scene viewer...	20
2.1.3 Платформа ARkit и технология Quick Look.	27
2.2 Исследование технологий и инструментов AR с отслеживанием метки	29
2.2.1 Сервис Stories AR.....	29
2.2.2 Сервис Artar	30
2.2.3 Сервис ARstudio	31
2.2.4 Библиотека AR.js	32
2.2.5 Библиотека MindAR	36
2.3 Технологии VR	38
2.4 Выбор инструментов и технологий для реализации модулей VR/AR в печатных изданиях.....	38
2.5 Выбор среды разработки	40
3 Разработка модулей виртуальной и дополненной реальности для печатного издания романа «Ключ от лабиринта».....	42
3.1 Реализация стартовой страницы модулей	42
3.2 Разработка конструктора для генерации стартовых страниц и тестирования разрабатываемых моделей	42
3.3 Разработка VR сцен для романа «Ключ от лабиринта»	46
Заключение	51
Список использованных источников	53
Приложение А (обязательное) Листинг программного кода	57

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем текстовом документе применяются следующие определения, обозначения и сокращения:

AR — Augmented Reality, дополненная реальность.

VR — virtual reality, виртуальная реальность.

WebAR — web augmented reality, дополненная реальность в браузере.

HTML — Hyper Text Markup Language.

CSS — Cascading Style Sheets.

PHP — Hypertext Preprocessor.

ВЕБ (WEB) — (World Wide Web) — всемирная паутина.

ОС — Операционная система.

QR-код — это сокращение от Quick Response, в переводе с английского «быстрый отклик». QR-код позволяет мгновенно получать доступ к данным.

Иммерсивный — это способ восприятия, создающий эффект погружения в искусственно созданную среду.

ВВЕДЕНИЕ

Современная литература всегда интересна читателям, ведь она отражает текущий мир и ставит насущные проблемы. Авторы рассказывают о социальных, культурных, экономических и политических вопросах, которые волнуют сегодняшнее общество, отображают моральные и этические дилеммы, с которыми мы сталкиваемся в нашей жизни. А то, что каждый автор предлагает свой уникальный взгляд на мир, делает его произведения еще более уникальными и интересными. Кроме того, современная литература оказывает сильное влияние на развитие культуры и взгляды общества, может вызывать дискуссии и споры, вносить новые идеи и подходы, и даже изменять наше отношение к определенным вещам.

С появлением новых форм развлечения, таких как видеоигры, потоковое видео и социальные медиа, люди все больше предпочитают проводить свое время именно за этими активностями вместо чтения книг. Интерактивные и визуальные формы развлечения становятся более привлекательными и притягательными для молодых поколений, потому чтобы сблизить читателя с художественным текстом, заинтересовать его, сегодня требуется адаптировать произведения для различных форматов и видов искусства, вовлекать в него новые формы и форматы.

Стремительное развитие информационных технологий и повышение доступности устройств, поддерживающих технологии виртуальной (VR) и дополненной (AR) реальности, позволяют использовать их возможности в различных сферах, в том числе в художественных произведениях и различного рода печатных изданиях. Практически каждый современный смартфон позволяет «прикоснуться» к дополненной и виртуальной реальности, разнообразить, улучшить взаимодействие с реальным миром, добавляя в него виртуальные объекты.

Опыт применения VR/AR технологий все чаще встречается в работах ученых и исследователей в различных сферах жизни: в образовании (Иванько А.Ф., Бурцевой М.Б. [1], Коннова З.И., Семеновой Г.В. [3]), экономике и

финансах (Козловой Н.П., Петуховой А.В. [2]), машиностроении и производственных процессах (Яхричева В.В [4]). Однако выявление наиболее эффективных инструментов, применимых для создания модулей виртуальной и дополненной реальности в печатных изданиях, учитывающих особенности данной среды, остается актуальным.

Целью данной выпускной квалификационной работы стала разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта».

Для достижения поставленной цели были выявлены следующие задачи:

- анализ возможности использования в печатных изданиях технологий виртуальной и дополненной реальности;
- анализ существующих решений в сфере применения в печатных изданиях технологий виртуальной и дополненной реальности;
- анализ и выбор оптимальных технологий и инструментов для реализации;
- разработка модулей виртуальной и дополненной реальности для печатного издания романа «Ключ от лабиринта».

Объектом исследования являются печатные издания, а предметом исследования — возможности применения AR и VR технологий в печатных изданиях. В ходе работы используются такие методы исследования как анализ литературы по теме исследования, сравнительный анализ средств и технологий виртуальной и дополненной реальности, а также моделирование и эксперимент при разработке модулей виртуальной и дополненной реальности для романа «Ключ от лабиринта».

Результаты проведенного в ходе исследования анализа технологий и инструментов разработки модулей виртуальной и дополненной реальности могут быть применимы в различного рода печатных изданиях, а также в образовательных пособиях, виртуальных музеях и онлайн-турах.

1 ДОПОЛНЕННАЯ И ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ

1.1 Понятие дополненной реальности

Дополненная реальность (AR) — это компьютерная технология, которая объединяет виртуальные объекты и информацию с реальным окружением, создавая таким образом гибридное восприятие мира [5]. В AR виртуальные элементы, такие как изображения и 3D-модели, накладываются на реальные объекты, что позволяет пользователям взаимодействовать с этими элементами и воспринимать их как часть окружающей среды.

Одним из наиболее известных примеров использования AR являются мобильные приложения, которые позволяют видеть на экране своего смартфона виртуальные объекты, наложенные на реальный мир. Это может быть что угодно от дополненной информации о местоположении или погоде до интерактивных игровых элементов [6].

С развитием AR появляются новые устройства и методы технологий, которые улучшают и расширяют ее возможности. Например, очки дополненной реальности позволяют пользователям смотреть на мир через прозрачный дисплей, на котором отображаются виртуальные объекты. Это делает AR более непосредственным и интуитивным, поскольку пользователь может видеть и взаимодействовать с виртуальными элементами, не отвлекаясь от реального мира [7].

AR имеет огромный потенциал для изменения способа, которым мы получаем информацию. От улучшенного путешествия и навигации до расширения возможностей обучения и создания новых форм развлечения — дополненная реальность открывает новые горизонты для инноваций и творчества.

Среди технологий, которые применяются сейчас для построения AR-объектов, выделяют технологии AR с отслеживанием плоскости и технологии AR с отслеживанием метки (маркерная технология).

Технологии AR с отслеживанием плоскости — это метод распознавания поверхностей в реальном мире с помощью камеры на смартфоне или другом устройстве с AR-возможностями. Он использует компьютерное зрение и алгоритмы для определения плоской поверхности, такой как стол или пол, на которую можно проецировать виртуальные объекты.

Отслеживание плоскости является важным аспектом AR, поскольку оно позволяет виртуальным объектам «прикрепляться» к физическим поверхностям, таким как столы, полы, стены и другие плоские поверхности. Это обеспечивает более реалистичное взаимодействие между виртуальными и реальными объектами в AR-приложениях.

Технологии дополненной реальности с отслеживанием метки — это метод создания AR-приложений, который использует распознавание изображений. Метка — это уникальное изображение или QR-код, который распознается камерой устройства и используется для отображения виртуального контента, связанного с этим маркером.

Основной плюс маркерной технологии — широкое покрытие устройств и стабильность работы. Она отлично подходит для решений, рассчитанных на широкую аудиторию, так как не требует большой производительности.

Также следует отметить технологию реализации AR в браузере. WebAR — это технология, которая позволяет создавать и отображать дополненную реальность (AR) прямо в веб-браузере, без необходимости установки специальных приложений или плагинов. Она сочетает возможности веб-технологий, таких как HTML, CSS и JavaScript, с функциональностью AR, что делает ее более доступной и широко применимой.

В целом, WebAR — это эффективная и доступная технология, которая расширяет возможности дополненной реальности, делая ее более широко распространенной и интуитивно понятной для пользователей.

1.2 Понятие виртуальной реальности

Виртуальная реальность (VR) — это компьютерная технология, создающая иммерсионное восприятие искусственного окружения, полностью отделяющего пользователя от реального мира и погружающего его в виртуальное пространство [8]. В отличие от дополненной реальности, где виртуальные элементы наслаиваются на реальный мир, VR создает полностью симулированное окружение, которое пользователь может исследовать и взаимодействовать с ним.

Одним из ключевых компонентов VR является гарнитура виртуальной реальности, которая надевается на голову пользователя. Гарнитура обычно включает в себя дисплеи, обеспечивающие изображение для каждого глаза, а также датчики, отслеживающие движения головы и позволяющие пользователю взаимодействовать с виртуальным окружением [9].

Одним из главных преимуществ VR является возможность создания ситуаций и сред, которые в реальности могут быть опасными, дорогостоящими или недоступными. Пользователи могут исследовать виртуальные миры, взаимодействовать с виртуальными объектами и испытывать эмоции и ощущения, которые в реальной жизни были бы недоступны. Это делает VR мощным инструментом для развлечения, образования и тренировки [10].

Однако, несмотря на преимущества, VR также имеет свои ограничения. Некоторые люди могут испытывать дискомфорт при использовании гарнитуры VR. Также существует риск отрыва от реальности и возможность возникновения психологических эффектов [11].

С течением времени VR продолжает развиваться и совершенствоваться. Большинство современных систем VR становятся все более доступными и удобными в использовании, а разработчики создают все более захватывающие и реалистичные виртуальные миры. Ожидается, что эта технология будет продолжать преобразовывать наши взаимодействия социальной и физической

средой, открывая новые возможности для развлечения, образования и многих других областей [12].

1.3 Мировые тенденции в области развития технологий VR/AR

Бурное развитие технологий VR/AR, снижение стоимости оборудования и расширение рынка приложений и контента с поддержкой VR/AR обеспечило рост доступности дополненной реальности и виртуальной реальности, которые стали заметным трендом в последние годы.

В отношении AR, мобильные устройства сыграли ключевую роль в увеличении его доступности. Большинство современных смартфонов оснащены камерами и датчиками, что позволяет использовать AR-приложения без необходимости в дополнительных устройствах. Согласно статистике сайта «Statista», количество активных устройств, поддерживающих AR в 2023 году, насчитывает 1,4 миллиарда, а к 2024 составит 1,73 миллиарда [13]. Статистика представлена на рисунке 1.1.

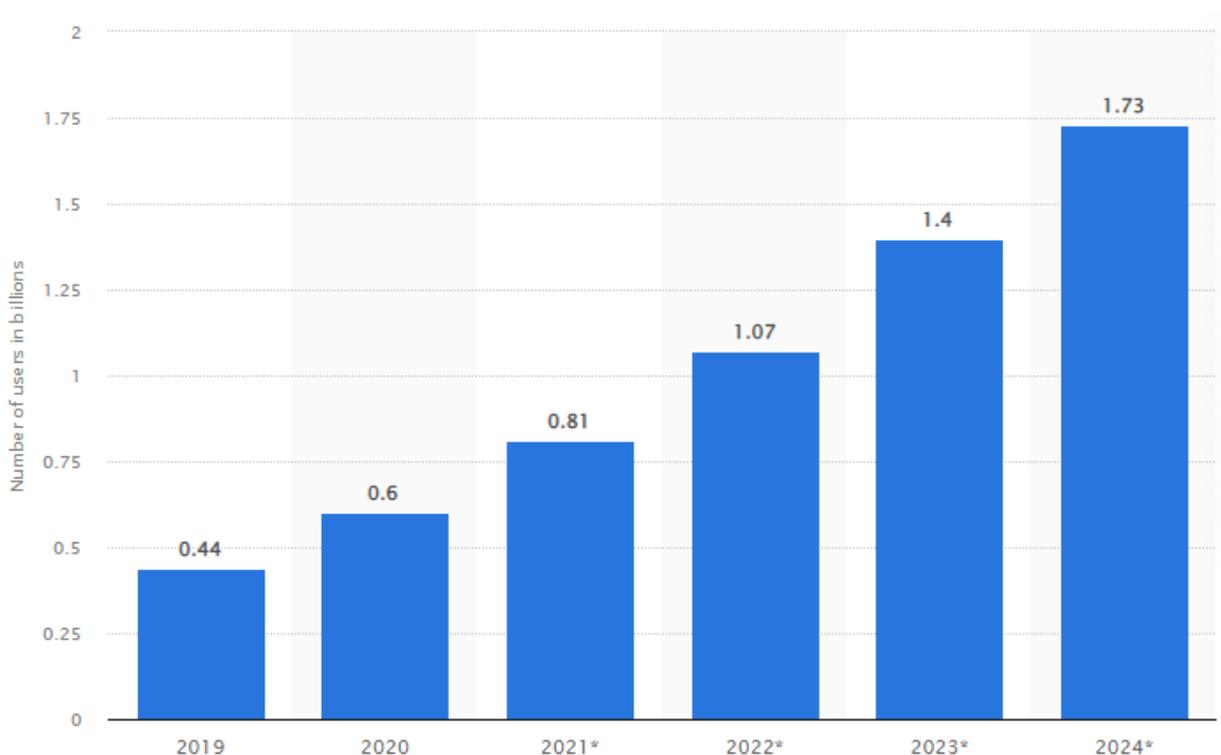


Рисунок 1.1 - Статистика количества устройств, поддерживающих AR

Все это делает AR доступным для широкой аудитории, пользователи могут использовать свои существующие устройства без дополнительных затрат.

В случае VR рост доступности также наблюдается. Технологические компании разработали и выпустили различные модели VR-гарнитур, начиная от более доступных вариантов, таких как Oculus Quest, и заканчивая более продвинутыми и дорогими устройствами, такими как HTC Vive Pro или Valve Index. Это разнообразие вариантов позволяет пользователям выбирать подходящую для них цену и функциональность. Кроме того, цены на VR-гарнитур постепенно снижаются с течением времени, что делает их более доступными для широкой аудитории.

Так же, как и в случае с AR, пользователи могут обойтись и без специальной гарнитуры. Можно использовать Google Cardboard («Картон») — стандарт компании Google в области виртуальной реальности, в основе которого лежит шлем, главной частью которого служит обычный смартфон. Название связано с простейшим вариантом шлема виртуальной реальности, который, по замыслу разработчиков, можно собрать из двух линз, застёжек и картона [14]. С помощью данного шлема можно прикоснуться к VR с помощью своего смартфона, а при использовании вместе со шлемом специальных контроллеров, можно ощутить новый опыт погружения в VR.

Еще одним важным фактором роста доступности AR и VR является расширение контента и приложений. Разработчики создают все больше впечатляющих и захватывающих AR- и VR-приложений в различных сферах. Это делает эти технологии более привлекательными для широкой аудитории и расширяет их применение в различных отраслях.

Ожидается, что рост доступности AR и VR будет продолжаться, поскольку технологии становятся более интегрированными, разработчики создают все более удобные и доступные устройства, а спрос на AR и VR контент продолжает расти. Эти технологии обещают изменить способ,

которым мы взаимодействуем социально-цифровым миром и открыть новые возможности для образования, развлечений и промышленности.

1.4 Сферы применения технологий AR и VR

Технологии AR и VR позволяют пользователям окунуться в уникальные виртуальные миры или дополнять реальность виртуальными элементами. Они создают глубокий и погружающий опыт, который позволяет людям полностью погрузиться в виртуальную среду.

Сегодня AR и VR технологии можно встретить в различных сферах деятельности: образовании, туризме, производстве, маркетинге и рекламе и конечно же в сфере развлечений.

Использование AR и VR могут значительно улучшить процесс обучения. VR позволяет учащимся практиковать навыки в безопасной и контролируемой среде. Например, студенты медицинских учебных заведений могут проводить виртуальные операции, архитекторы — создавать виртуальные модели зданий, а историки — исследовать воссозданные исторические события и объекты. Так приложение компании «Увлекательная реальность» для учебника по физике 7 класса делает страницы учебника маркерами дополненной реальности, на которых, при просмотре через мобильное устройство, разворачиваются анимированные трехмерные сцены и живые демонстрации процессов и явлений по изучаемой теме [15].

Технологии AR и VR могут кардинально изменить способ путешествий и туризма. Путешественники могут исследовать виртуальные модели мест и достопримечательностей, а также погрузиться в виртуальные туры и путешествия, чтобы получить уникальный опыт и переживания. Это позволяет путешествовать, не выходя из дома, а для людей с ограниченными возможностями может стать единственной возможностью посетить новые интересные места. Примером использования VR в туризме является платформа «TrueVirtualTours» [16]. Она представляет собой хостинг для

панорамных фото и видео, любой человек может загрузить или просмотреть панораму любого места.

На производстве AR и VR имеет огромный потенциал для оптимизации процессов и повышения эффективности персонала. С помощью виртуальных моделей и инструкций на экране пользователи могут видеть, как правильно собирать или обслуживать сложные устройства и машины. Также с помощью AR и VR можно создавать виртуальные прототипы и модели, которые могут быть проверены и протестированы до физического производства, что значительно сократит как трудовые, так и материальные затраты. Компания «СИБУР» активно включает в свое производство современные технологии, в том числе и дополненную реальность. AR-очки крепятся на каску работника и управляются голосом. На экран очков при консультации выводится вся необходимая информация, и сотрудник может максимально оперативно получать помощь в решении той или иной задачи, не отрываясь от работы [17].

AR и VR предоставляют новые возможности для маркетинга и рекламы. Дополненная реальность позволяет компаниям создавать интерактивные и увлекательные рекламные кампании, которые могут привлекать больше внимания и вовлекать потребителей. Виртуальная реальность может предоставлять потенциальным клиентам возможность примерить продукты или погрузиться в виртуальные магазины. Как пример можно вспомнить «живую этикетку» от «Heinz», где под этикеткой «Heinz» скрывается целая книга рецептов, которые можно приготовить с использованием этого продукта [18].

И конечно же AR и VR преобразовывают индустрию развлечений и игр, предоставляя уникальные игровые и развлекательные возможности. Виртуальная реальность позволяет пользователям стать частью виртуального мира и полностью погрузиться в игровой процесс. AR дополняет реальный мир интерактивными элементами, создавая увлекательные игры. Платформа «Steam» насчитывает в своем каталоге на сегодняшний день порядка 4700 приложений, разработанных под VR-устройства [19]. В них входят игры

различных жанров, среди самых популярных можно выделить шутеры, аркады, головоломки, а также приложения для образования и так называемые «песочницы».

Использование виртуальной и дополненной реальности открывает новые перспективы для художников и создателей художественных произведений. Эти технологии позволяют переносить искусство в цифровое пространство и предлагают уникальные возможности для взаимодействия и экспериментирования с художественными идеями. VR позволяет художникам создавать искусство в виртуальной среде, полностью погружаясь в виртуальное пространство. Они могут рисовать, моделировать, создавать анимацию и визуализировать свои творческие идеи в трехмерном пространстве. VR-очки позволяют художникам совершенно новым образом взаимодействовать с цифровыми инструментами и материалами, создавая уникальные художественные работы.

Дополненная и виртуальная реальность может привести новые возможности в печатные издания и изменить способ, которым мы взаимодействуем с печатным контентом.

AR позволяет добавлять дополнительные слои информации к печатным страницам. Читатели могут использовать мобильные устройства или специальные очки, чтобы увидеть дополнительные изображения, видео, звуки или интерактивные элементы, которые дополняют текст на странице. Это может быть особенно полезно для печатных изданий, связанных с искусством, модой, дизайном или научными открытиями, где визуальные и интерактивные элементы являются важными.

VR позволяет создавать полноценные виртуальные миры, в которых читатели могут погрузиться. Это может быть использовано для создания иммерсивных виртуальных книг или журналов, где пользователи могут «проживать историю» или участвовать в интерактивных сценариях.

2 ВЫБОР ТЕХНОЛОГИЙ И ИНСТРУМЕНТОВ РЕАЛИЗАЦИИ VR/AR

2.1 Исследование технологий AR с отслеживанием плоскости

Для реализации AR существует множество инструментов и технологий, проанализируем наиболее перспективные и привлекательные варианты. Первоначально разберем технологии AR, построенные на отслеживании плоскости.

2.1.1 Фреймворк A-frame

Одним из самых популярных и перспективных способов применения технологии AR, построенные на отслеживании плоскости, является использование фреймворка A-frame.

A-frame — это системный фреймворк сущностных компонентов для Three.js, где разработчики могут создавать 3D- и WebVR-сцены с использованием HTML. HTML предоставляет знакомый инструмент разработки для веб-разработчиков и дизайнеров, в то же время включая популярную модель разработки игр, используемую такими движками, как Unity [20].

Фреймворк A-frame может использоваться совместно с дополнением «aframe-extras». Это дополнение включает:

- компоненты для элементов управления;
- компоненты для загрузки моделей;
- компоненты для поиска путей по навигационной сетке;
- несколько новых примитивов.

Фреймворк подключается как JS файл на HTML странице. Для создания AR-проекта задается так называемая сцена с главным объектом «a-scene», внутри которого находятся все остальные элементы. В этом главном объекте задаются настройки сцены, включены ли будут тени, а также убрана или нет кнопка перехода в режим виртуальной реальности. Базовый объект «a-assets»

позволяет добавить все текстуры и модели, которые будут использованы в данной сцене. Добавление же самих объектов происходит с помощью базового объекта «a-entity», в котором задается, что это будет 3d модель, её размеры и положение. Объекты освещения «a-light» задают тип и интенсивность света. Примера кода такой сцены представлен в листинге 2.1.

Листинг 2.1 – Код сцены A-frame

```
<body>
  <script src="https://aframe.io/releases/1.4.0/aframe.min.js">
  </script>
  <a-scene cursor="rayOrigin: mouse" vr-mode-ui="enabled: true">
    <a-assets>
      <a-asset-item id="Chronicler"
src="../model/Chronicler_LightTexture.glb"></a-asset-item>
    </a-assets>
    <a-entity gltf-model="#Chronicler" scale="1 1 1"></a-entity>
    <a-light type="ambient" intensity="1"></a-light>
  </a-scene>
</body>
```

Эксперименты показали, что к преимуществам данного фреймворка можно отнести скорость и простоту разработки, благодаря возможности использовать готовые компоненты. Также A-frame позволяет создать взаимодействия со сценой благодаря возможности обработки различных событий. К недостаткам же — небольшая нестабильность отслеживания плоскости. Кроме того, не все смартфоны поддерживают такой AR ввиду ряда технических причин. Пример работы фреймворка A-frame представлен на рисунках 2.1 и 2.2.



Рисунок 2.1 – Пример стартовой страницы объекта в A-frame AR



Рисунок 2.2– Пример работы A-frame AR

2.1.2 Платформа ARCore и технологии WebXR, model-viewer, scene viewer

Такие IT гиганты как Google и Apple активно ведут разработку в области AR. Для работ с дополненной реальностью Google разработал свою платформу ARCore. Используя различные API, ARCore позволяет телефону ощущать окружающую среду, понимать мир и взаимодействовать с информацией [21]. Однако не все устройства, работающие на Android, поддерживают ARCore ввиду различий в аппаратном обеспечении.

Наиболее популярными и функциональными среди технологий платформы ARCore являются модуль расширенной реальности WebXR, веб-компонент model-viewer, средство просмотра моделей в дополненной реальности scene viewer. Рассмотрим их подробнее и протестируем в работе.

Модуль расширенной реальности WebXR позволяет привести виртуальный контент в соответствие с реальной средой перед его отображением пользователям, он использует ARCore для расширения возможностей дополненной реальности в браузере Google Chrome на устройствах Android [22].

Чтобы работать с WebXR к файлу необходимо подключить несколько JS скриптов. Подключение скриптов представлено в листинге 2.2.

Листинг 2.2 – Подключение WebXR

```
<script
src="https://unpkg.com/three@0.126.0/build/three.js"></script>
<script
src="https://unpkg.com/three@0.126.0/examples/js/loaders/GLTFLoader.js
"></script>
```

Для того, чтобы создать AR-проект, в WebXR инициализируется функция «activateXR()», которая будет создавать холст для отображения виртуального контента «canvas» и WebGL-контекст, который совместим с WebXR.

Затем создается сцена, применяя библиотеку Three.js, и в нее добавляется направленный источник света. Осуществляется передача в функцию «THREE.WebGLRenderer» контекста «WebGL» и нового элемента

«canvas», создается сессия «WebXR» с помощью вызова функции «navigator.xr.requestSession», где в качестве типа данных задается элемент «immersive-ar».

Далее, с помощью события «session.requestReferenceSpace» создается «локальное» пространство и пространство зрителя. Эта функция создает цикл рендеринга, который использует данные пространства для масштабирования и позиционирования объектов Three.js в дополненной реальности.

Также создается функция загрузки модели «THREE.GLTFLoader», которая позволяет размещать объект в мировом пространстве через событие «session.addEventListener». Функция также рисует прицельную сетку на основе «frame.getHitTestResults».

Эксперименты показали, что плюсами использования данного расширенного метода является свобода действий — возможность реализовать практически любое взаимодействие со сценой и моделью, однако это требует детальной настройки и проработки программного кода и его параметров. Кроме того, стабильность модели, созданной с применением данного решения, уступает другим аналогичным решениям Google. Пример работы ARCore WebXR представлен на рисунках 2.3 и 2.4.

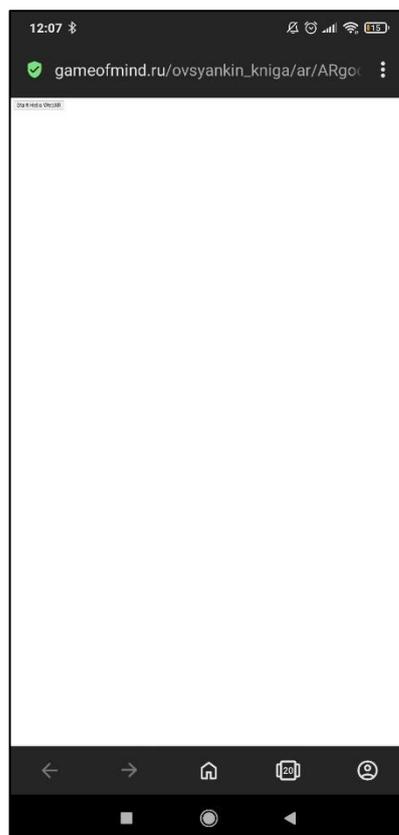


Рисунок 2.3 – Пример страницы ARCore WebXR



Рисунок 2.4 – Пример работы ARCore WebXR

Еще одним способом реализации AR через ARCore является «model viewer». Веб-компонент «model viewer» можно использовать для просмотра и взаимодействия с 3D моделями [23].

Компонент подключается как JS файл. Настройка компонента осуществляется с помощью атрибутов. Атрибут «ar» компонента сообщает браузеру, что данная модель может быть показана в режиме AR. Также внутри компонента есть «button», который позволяет активировать режим AR. «camera-controls» предоставляет пользователю возможность управлять поведением камеры. Другие атрибуты, такие как «alt», «scale», «shadow-intensity», и «exposure» позволяют настроить модель. Пример кода представлен в листинге 2.3.

Листинг 2.3 – Код ARCore model-viewer

```
<script type="module" src="https://unpkg.com/@google/model-viewer/dist/model-viewer.min.js"></script>
<model-viewer id="model" src="../model/Chronicler_LightTexture.glb"
  alt="Модель"
  ar
  auto-rotate
  camera-controls
  shadow-intensity="4"
  exposure="0.4"
  environment-image="legacy"
  scale="0.5 0.5 0.5">
  <button slot="ar-button" style="background-color:
gray; border-radius: 4px; border: none; position: absolute; top: 16px;
right: 16px; width: 100px; height: 100px;">Activate AR</button>
</model-viewer>
```

Эксперименты показали, что к плюсам данного решения можно отнести простоту реализации и стабильность отслеживания плоскости, а к минусам малую возможность взаимодействия с моделью. Пример работы ARCore «model viewer» представлен на рисунках 2.5 и 2.6.

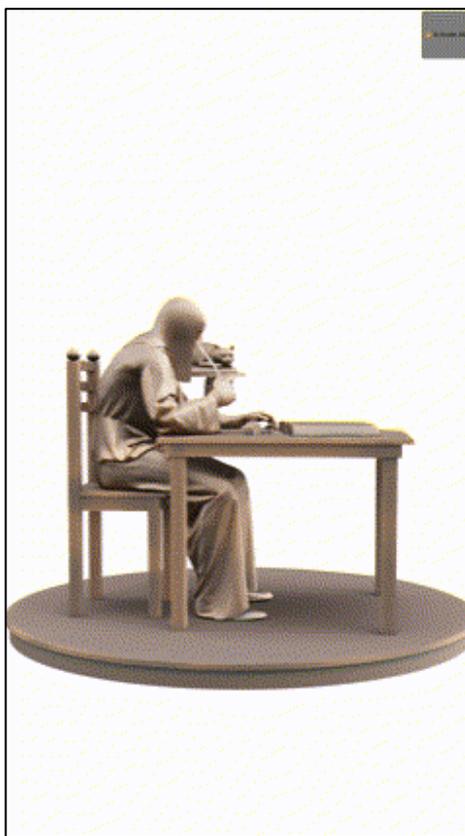


Рисунок 2.5 – Пример страницы ARCore model-viewer

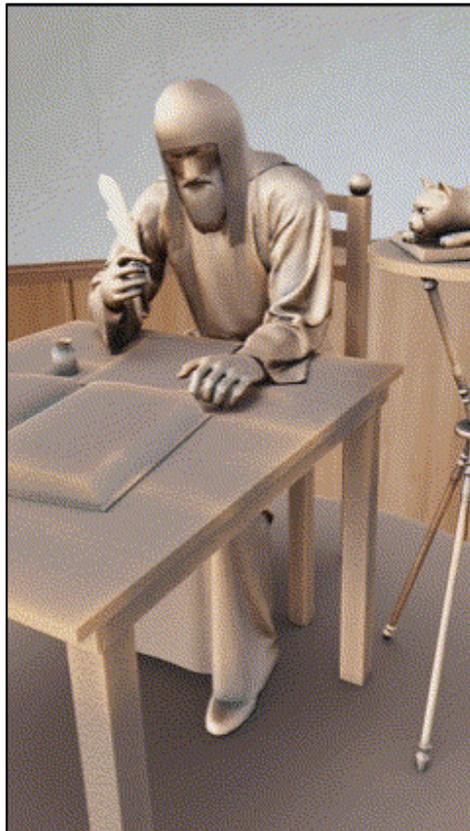


Рисунок 2.6 – Пример работы ARCore model-viewer

Последним способом, предоставляемым Google, является использование scene viewer. Scene viewer — это средство просмотра с эффектом погружения, которое позволяет работать с 3D и дополненной реальностью на веб-сайте или в приложении для Android [24].

Scene viewer, в отличие от других решений Google не требует отдельного подключения. Для создания AR-модуля сначала задается гиперссылка, при клике на которую будет открывается «scene viewer» - приложение Google для просмотра 3D-объектов в дополненной реальности. В атрибут «file» указывается URL-адрес модели в формате glb. А в атрибут «browser_fallback_url» указывает браузеру, что нужно открыть, если приложение «scene viewer» не может быть запущено. Пример кода, созданного для scene viewer представлен в листинге 2.4.

Листинг 2.4 – Код ARCore scene viewer

```
<a href="intent://arvr.google.com/scene-viewer/1.0?file=*ссылка*  
#Intent;scheme=https;package=com.google.android.googlequicksearchbox;action=android.intent.action.VIEW;S.browser_fallback_url=*ссылка*;end;"  
>Ссылка</a>
```

Эксперимент показал, что к преимуществам метода можно отнести высокую стабильность отображения модели и возможность делать фото и видео. Недостатки же аналогичны предыдущему способу, это малая возможность взаимодействия с моделью. Пример работы ARCore scene viewer представлен на рисунках 2.7, 2.8 и 2.9.

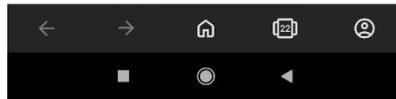
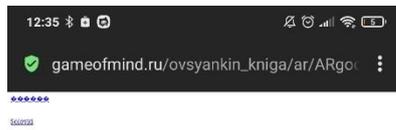


Рисунок 2.7 – Пример страницы ARCore scene viewer



Рисунок 2.8 – Пример промежуточной страницы ARCore scene viewer



Рисунок 2.9 – Пример работы ARCore scene viewer

2.1.3 Платформа ARkit и технология Quick Look

Apple разработали свою платформу ARkit для работы с AR. ARkit — это платформа дополненной реальности от Apple, разработанная для мобильных устройств, которая позволяет создавать интерактивные и захватывающие приложения AR на устройствах iPhone и iPad. Для работы с AR используется Quick Look для отображения используемых файлов виртуальных объектов в 3D или AR [25].

Для работы с Quick Look не требуется подключать дополнительные файлы. Для работы с платформой задается гиперссылка с атрибутом «rel» равным значению «ar», чтобы сообщить браузеру, что данная ссылка относится к приложению AR. Дальше в атрибут «href» необходимо указать URL-адрес модели в формате USDZ. Пример кода представлен в листинге 2.5.

Листинг 2.5 – Код ARkit

```
<a rel="ar"
href="https://gameofmind.ru/ovsyankin_kniga/ar/model/Chronicler_LightT
exture.usdz"></a>
```

Эксперименты показали, что к преимуществам способа можно отнести высокую стабильность отображения модели и возможность делать фото и видео. Среди минусов — малая возможность взаимодействия с моделью. Пример работы ARkit представлен на рисунках 2.10 и 2.11.

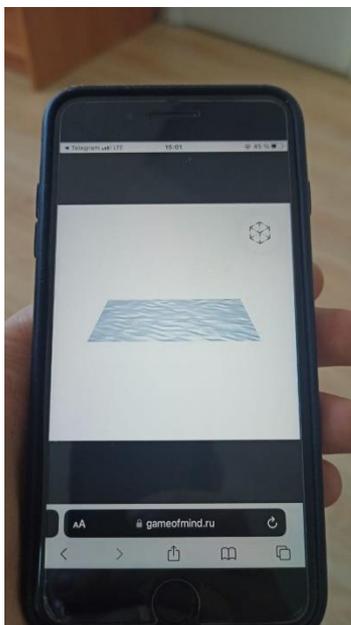


Рисунок 2.10 – Пример страницы ARkit



Рисунок 2.11 – Пример работы ARkit

2.2 Исследование технологий и инструментов AR с отслеживанием метки

Использование технологии AR с отслеживанием плоскости имеет один существенный общий недостаток — не все смартфоны на сегодняшний день способны поддерживать данную технологию. Этого недостатка лишены технологии AR с отслеживанием метки. Проанализируем основные существующие на сегодняшний день варианты решений с использованием данного подхода, как конструкторов, так и библиотек.

2.2.1 Сервис Stories AR

Конструктор Stories AR — «сервис создания оживающих фото с дополненной реальностью» [26]. Он позволяет отслеживать изображения и открывать поверх него заданное видео. Для этого достаточно загрузить на сайт изображение — метку, а также видео, которое будет открыто при наведении на метку.

Пример работы сервиса представлен на рисунке 2.12.



Рисунок 2.12 – Пример работы сервиса Stories Ar

К преимуществам данного инструмента можно отнести простоту и интуитивность работы с сервисом, стабильность отслеживания метки, однако с использованием данного сервиса возможно отображение только видеоматериалов, например, использовать 3d объекты или же изображения не получится.

2.2.2 Сервис Artar

Artar — это «Web AR-конструктор для создания интерактивной дополненной реальности без специальных знаний и необходимости устанавливать мобильное приложение» [27]. Он позволяет отслеживать заданные изображения-метки или qr-коды и открывать поверх них 3d или 2d объекты. Для удобства создания AR-моделей присутствует встроенный редактор 3D сцены, который позволяет точно настроить размер и положение объекта относительно метки.

Среди выявленных преимуществ — стабильность отслеживания меток и возможность использовать в качестве нее любое изображение, к недостаткам можно отнести ограниченность сферы применения, сервис ориентирован под разработку визитных карточек, из-за этого интерфейс инструмента не удобен для создания других видов проектов. Пример работы программы представлен на рисунке 2.13.

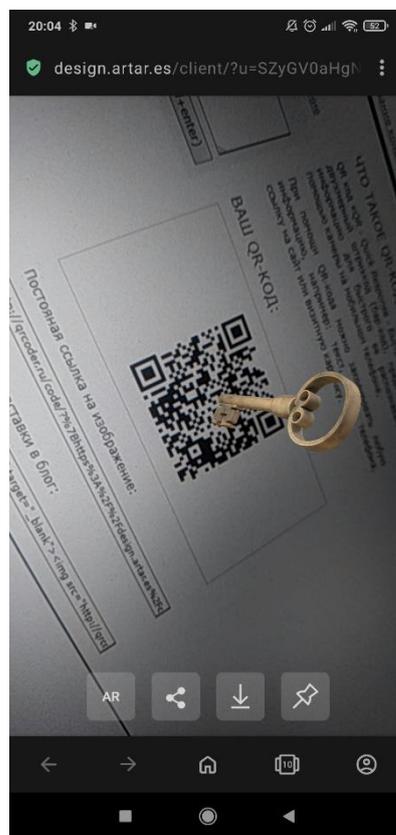


Рисунок 2.13 – Пример работы сервиса Artar

2.2.3 Сервис ARstudio

ARstudio — это «онлайн-конструктор Web AR и AR проектов, позволяющий создавать оживающие фото, альбомы и другие проекты с дополненной реальностью без кода и навыков работы с 3D» [28].

Он разрешает использовать в качестве метки QR-код, а при наличии платной подписки и любые изображения. В качестве демонстрируемых объектов могут выступать 3d или 2d объекты. ARstudio имеет свой встроенный редактор 3d сцены, позволяющий точно настроить размер и положение объекта относительно метки. Проекты, созданные с использованием данного конструктора, имеют хорошую стабильность отслеживания, однако функционал бесплатной версии среды достаточно ограничен. Пример работы сервиса представлен на рисунке 2.14.

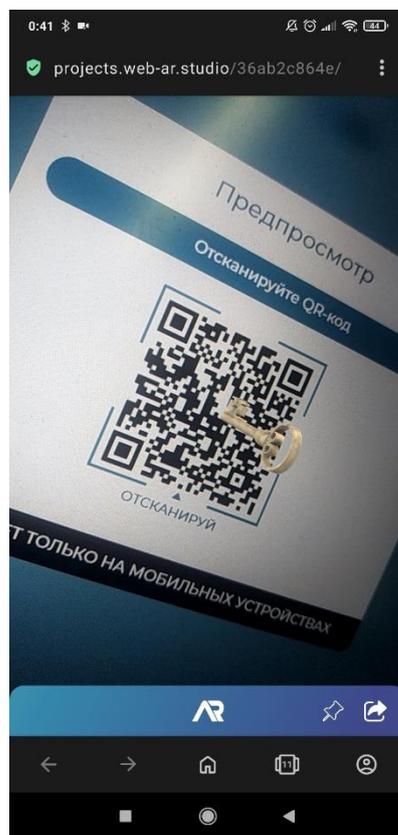


Рисунок 2.14 – Пример работы сервиса ARstudio

2.2.4 Библиотека AR.js

Помимо применения конструкторов для Web AR и AR проектов могут использоваться различные библиотеки, позволяющие реализовать собственный AR с отслеживанием метки.

AR.js — это библиотека дополненной реальности с открытым исходным кодом, которая позволяет создавать приложения AR для веб-браузеров. Она была разработана для упрощения процесса создания AR-приложений, которые доступны пользователям с помощью любого устройства, имеющего веб-браузер с камерой [29]. Библиотека основывается на фреймворке A-frame.

AR.js предлагает несколько способов реализации AR. Один из них это использование в качестве метки любого изображения.

Для работы с данным способом помимо A-frame необходимо подключить еще и файл библиотеки. Все подключения представлены в листинге 2.6.

Листинг 2.6 – Код подключения AR.js с отслеживанием изображения

```
<script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar-nft.js"></script>  
<script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
```

Чтобы начать работу мы добавляем на A-frame сцену новый элемент «a-nft», который используется для отображения содержимого дополненной реальности (AR).

Атрибут `type` устанавливает тип элемента как «nft», указывая на использование маркера для отображения модели. Атрибут «url» содержит URL-адрес маркера, который определяет положение и ориентацию модели AR, параметр «smooth» включает плавное перемещение модели для более приятного визуального эффекта. Параметры «smoothCount», «smoothTolerance» и «smoothThreshold» определяют количество кадров, допустимую погрешность и пороговое значение соответственно для плавного перемещения модели. Пример кода с применением данного подхода представлен в листинге 2.7.

Листинг 2.7 – Код AR.js с отслеживанием изображений

```
<a-nft  
  type="nft"  
  url="https://gameofmind.ru/ovsyankin_kniga/ar/ARjs/metka"  
  smooth="true"  
  smoothCount="15"  
  smoothTolerance=".01"  
  smoothThreshold="5"  
>
```

Возможность использования в качестве метки любого изображения открывают широкий простор для творчества, однако эксперименты показали, что данный способ имеет один существенный недостаток — крайняя нестабильность отслеживания изображения, что, несомненно, нивелирует все преимущества. Пример работы представлен на рисунке 2.15.

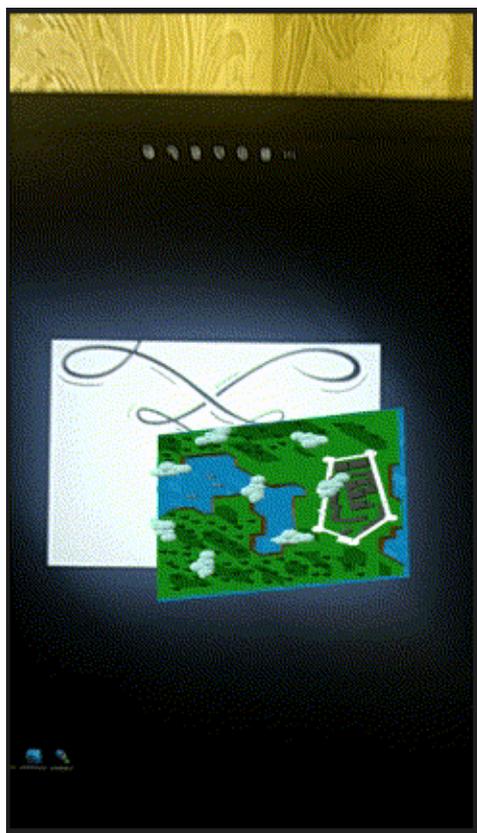


Рисунок 2.15 – Пример работы AR.js с отслеживанием изображений

Также библиотека AR.js позволяет использовать изображения, обладающих определенными характеристиками, так называемых «меток». «Метка» представляет собой Qr-код или изображение с толстой черной рамкой.

Так же, как и в предыдущем варианте необходимо подключить A-frame и саму библиотеку. Все подключения представлены в листинге 2.8.

Листинг 2.8 – Код подключения AR.js с отслеживанием «метки»

```
<script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>  
<script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
```

AR реализуется с помощью нового элемента «a-marker». Атрибут «type» установлен на значение «pattern», что указывает на использование маркера с использованием образца. Атрибут «url» указывает на URL-адрес файла с образцом маркера, в данном случае, «pattern50.patt». Внутри элемента «a-marker» находится элемент «a-entity», который определяет отображаемую сущность в AR. Пример кода представлен в листинге 2.9.

Листинг 2.9 – Код AR.js с отслеживанием «метки»

```
<a-marker type = 'pattern' url="pattern50.patt" smooth="true">  
  <a-entity  
    position="0 0 0"  
    scale="0.05 0.05 0.05"  
    gltf-model="../model/newSolovki.glb"  
  ></a-entity>  
</a-marker>
```

В отличие от использования любых изображений такой подход имеет хорошую стабильность отслеживания и плавность перемещения. Пример работы представлен на рисунке 2.16, а пример «метки» представлен на рисунке 2.17.

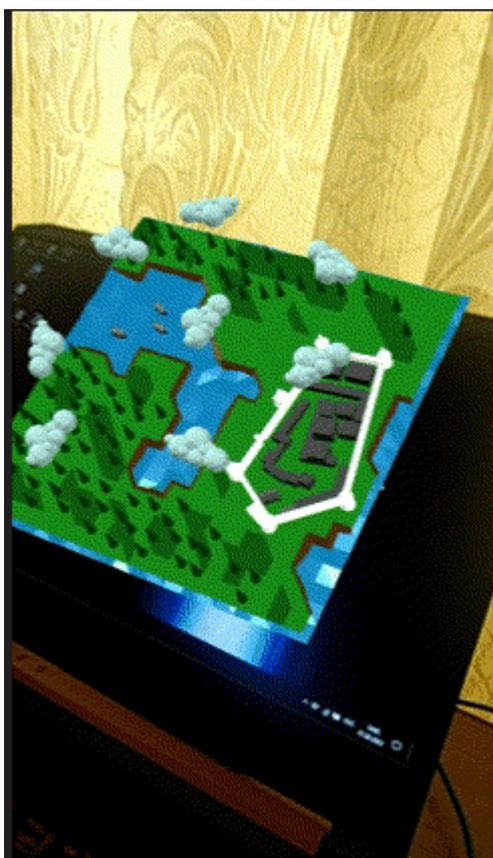


Рисунок 2.16 – Пример работы AR.js с отслеживанием «метки»

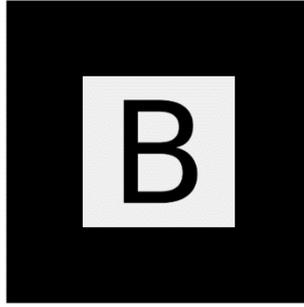


Рисунок 2.17 – Пример «метки»

2.2.5 Библиотека MindAR

Часто встречающаяся проблема многих инструментов разработки AR-моделей, заключающаяся в сложности достижения хорошего уровня стабильности изображения может быть решена с помощью библиотек, основанных на A-Frame, одной из них является MindAR.

MindAR — это веб-библиотека дополненной реальности с открытым исходным кодом, работающая в симбиозе с A-Frame [30].

Так как библиотека работает в симбиозе с A-frame, необходимо подключить как саму библиотеку, так и A-frame.

Для этого к элементу «a-scene», внутри которого располагаются другие элементы для настройки сцены AR, необходимо добавить атрибут «mindar-image», определяющий параметры обработки изображения и указывает путь к изображению-маркеру (например, "mind/targets.mind").

Атрибуты «vr-mode-ui» и «device-orientation-permission-ui» необходимы для отключения интерфейса виртуальной реальности и запроса разрешения на использование датчика ориентации устройства соответственно. Далее необходимо задать элемент «a-entity» с атрибутом «mindar-image-target», указывающим индекс маркера. Внутри добавляется элемент «a-gltf-model», который отображает модель, задавая ее поворот, позицию и масштаб. Пример кода представлен в листинге 2.10.

Листинг 2.10 – Код MindAR

```
<script src="mind-ar-js-master/dist/mindar-image.prod.js"></script>
<script src="aframe.min.js"></script>
<script src="mind-ar-js-master/dist/mindar-image-
aframe.prod.js"></script>
<a-scene mindar-image=" filterMinCF:0.01; filterBeta: 1000;
imageTargetSrc: mind/targets.mind;" vr-mode-ui="enabled: false"
device-orientation-permission-ui="enabled: false">
  <a-camera position="0 0 0" look-controls="enabled: false"></a-
camera>
  <a-assets>
    <a-asset-item id="avatarModel" src="model/newSolovki.glb"></a-
asset-item>
  </a-assets>
  <a-entity mindar-image-target="targetIndex: 0">
    <a-gltf-model rotation="90 0 0" position="0 0 0" scale="0.02
0.02 0.02" src='#avatarModel'>
  </a-entity>
</a-scene>
```

Главное преимущество, которое дает эта библиотека — возможность использовать в качестве метки любое изображение, причем стабильность отслеживания при ее использовании получается выше, чем у аналогичного решения на AR.js. Пример работы представлен на рисунке 2.18.



Рисунок 2.18 – Пример работы MindAR

2.3 Технологии VR

Виртуальную реальность, как и дополненную, можно реализовать средствами фреймворка A-frame, который практически не имеет достойных конкурентов. Он использует технологию WebVR для создания интерактивных VR-приложений, которые можно запускать непосредственно в веб-браузере.

В целом, A-frame предоставляет простой и доступный способ создания VR-сцен в веб-браузере, позволяя разработчикам использовать знакомые веб-технологии для создания захватывающих виртуальных реальностей. Также A-frame имеет встроенный редактор сцены, который упрощает разработку. Редактор сцены представлен на рисунке 2.19.

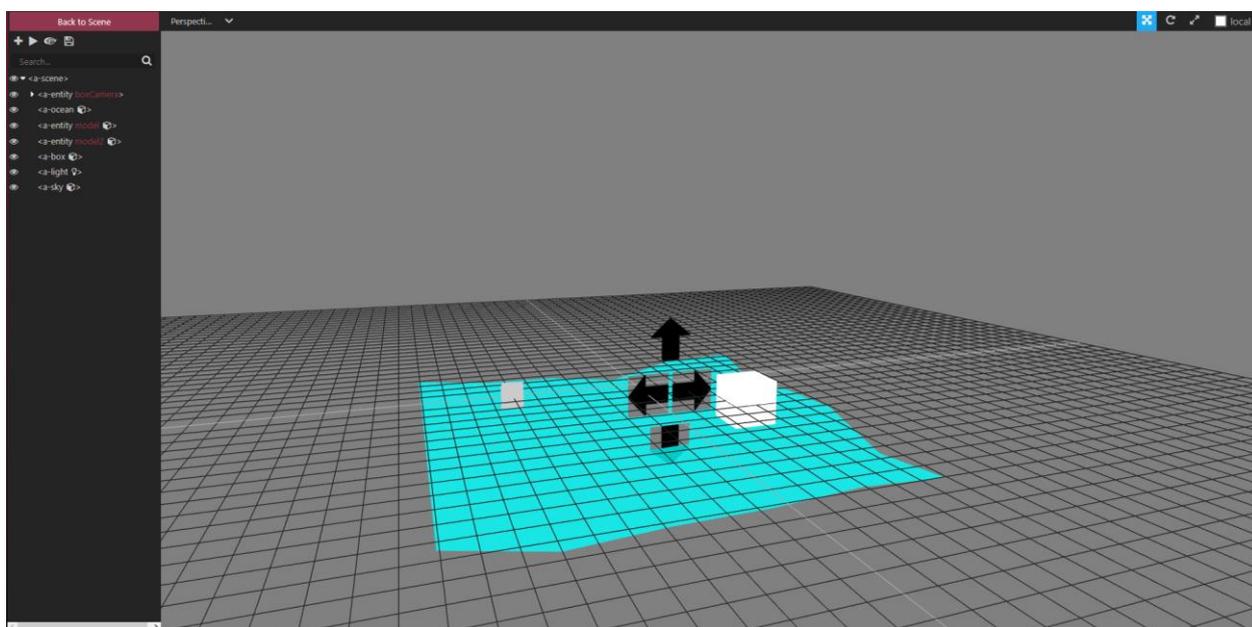


Рисунок 2.19 – Редактор сцены в A-frame

2.4 Выбор инструментов и технологий для реализации модулей VR/AR в печатных изданиях

Основываясь на вышеперечисленных экспериментах, можно составить сравнительную таблицу технологий AR. В ней будут следующие критерии сложности разработки, возможность отслеживать любое изображение, стабильность отслеживания и возможность записи фото и видео.

Сложность разработки будет оценена по трехбалльной системе, где «1» — реализация не требует настройки кодом, «2» — реализация требует настройку небольшого количества кода, «3» — реализация требует детальной настройки кода.

Стабильность отслеживания также по трехбалльной системе, где «1» — модель пропадает более чем на 10% от общего времени просмотра и/или модель абсолютно не задерживается на одном месте, «2» — модель пропадает более чем на 5%, но менее 10% от общего времени просмотра и/или модель задерживается на одном месте с колебаниями, «3» — модель пропадает менее чем на 5% от общего времени просмотра и модель задерживается на одном месте с незначительными отклонениями.

Сравнение технологий AR представлено в таблице 2.1.

Таблица 2.2 – Сравнение средств разработки

Технология	Сложность разработки (от 1 до 3)	Возможность отслеживать любое изображение	Стабильность отслеживания (от 1 до 3)	Возможность записи фото и видео
A-frame	2	-	2	-
ARCore WebXR	3	-	2	-
ARCore model-viewer	2	-	3	-
ARCore scene viewer	1	-	3	+
ARkit Quick Look	1	-	3	+
Stories AR	1	+	3	-
Artar	1	+	2	-
ARstudio	1	+	2	-
AR.js с отслеживанием изображения	3	+	1	-
AR.js с отслеживанием «метки»	2	-	2	-
MindAR	2	+	2	-

Технологии AR с отслеживанием изображений в ходе экспериментов показали меньшей показатель стабильности отслеживания, потому было решено выбирать из технологий AR с отслеживанием плоскости, но для открытия AR проекта пользователю необходимо перейти на сайт, на котором будет размещена модель в дополненной реальности с отслеживанием

плоскости. Для этого в печатных изданиях будут предусмотрены QR-коды для перехода на сайт с AR проектом.

Кроме того, использование AR в печатных изданиях накладывает свои требования к качеству. Так как метка или QR-код будут расположены на бумаге, то плохо пропечатанные они могут плохо считываться или считываться с ошибками. Излишне глянцевая поверхность издания также может негативно сказаться на скорости и качестве работы технологии и не рекомендуется. Таким образом, в качестве запуска AR проекта с печатных листов будут применяться QR-коды, имеющие встроенную проверку целостности и восстановление в случае небольших потерь данных. Размер QR-кодов при этом будет использоваться достаточно большим, чтобы снизить потери качества при печати.

Таким образом, проанализировав и апробировав инструменты для реализации AR с отслеживанием плоскости, были выбраны технологии ARCore «Scene viewer» и ARkit «Quick Look» для Android и iOS соответственно. Выбор обусловлен простотой реализации, стабильностью отслеживания плоскости и возможностью записи фото и видео в данных технологиях. Для реализации VR будет использован A-frame.

2.5 Выбор среды разработки

Выбор инструментов разработки играет важную роль в быстрой и качественной реализации проектов. Рынок программных средств предоставляет широкие возможности для выбора необходимого инструментария.

Один из самых популярных на сегодняшний день редакторов исходного кода — Visual Studio Code (VS Code). Его разработал Microsoft для всех популярных операционных систем: Windows, Linux и macOS. Визуальный редактор кода позволяет разработчикам создавать, редактировать и отлаживать код различных языков программирования с удобством и эффективностью [31].

Еще один возможный вариант для работы — IntelliJ IDEA, IDE, интегрированная среда разработки (комплекс программных средств, который используется для написания, исполнения, отладки и оптимизации кода) для Java, JavaScript, Python и других языков программирования от компании JetBrains. Она отличается обширным набором инструментов для рефакторинга (перепроектирования) и оптимизации кода [32].

Рассмотрим также PhpStorm. PhpStorm — это среда разработки, которая мгновенно предотвращает ошибки, предоставляет точное автодополнение и безопасные рефакторинги, а также возможность редактирования кода на HTML, CSS и JavaScript [33].

Для анализа и выбора оптимального инструмента разработки было проведено сравнение рассмотренных редакторов, оно представлено в таблице 2.2.

Таблица 2.2 – Сравнение средств разработки

Среда разработки	Цена	Производительность	Функциональность	Простота настройки и использования
VS Code	Бесплатно	+	+	+
IntelliJ IDEA	Бесплатно	+	-	+
PhpStorm	9.90 us/мес	+	+	-

В результате был выбран VS Code ввиду его простоты и функциональности, кроме того, он является полностью бесплатным решением.

3 РАЗРАБОТКА МОДУЛЕЙ ВИРТУАЛЬНОЙ И ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ ПЕЧАТНОГО ИЗДАНИЯ РОМАНА «КЛЮЧ ОТ ЛАБИРИНТА»

3.1 Реализация стартовой страницы модулей

На основе приложения ARcore «scene viewer» и фреймворка ARkit была реализована стартовая страница с кнопкой, открывающей ту или иную ссылку в зависимости от используемой на устройстве пользователя операционной системы IOS или Android.

С помощью метода «match» из системной строки «\$_SERVER['HTTP_USER_AGENT']» выделяется информация об ОС пользователя. Далее в зависимости от ОС пользователя выбирается ссылка необходимая ссылка на AR. Код функции по выбору ссылки представлен в листинге 3.1.

Листинг 3.1 – Код функции по выбору ссылки

```
<?php
preg_match("/iPhone|Android|iPad|iPod|webOS/",
$_SERVER['HTTP_USER_AGENT'], $matches);
$os = current($matches);

switch($os){
    case 'iPhone': $link_app = '**ссылка на модель**'; break;
    case 'iPad': $link_app = '**ссылка на модель**'; break;
    case 'iPod': $link_app = '**ссылка на модель**'; break;
    case 'Android': $link_app = 'intent://arvr.google.com/scene-
viewer/1.0?file=**ссылка на
модель**#Intent;scheme=https;package=com.google.android.googlequicksea
rchbox;action=android.intent.action.VIEW;S.browser_fallback_url=https:
//developers.google.com/ar;end;'; break;
default: $link_app = '';
}
echo $link_app;?>
```

3.2 Разработка конструктора для генерации стартовых страниц и тестирования разрабатываемых моделей

Для упрощения тестирования моделей в дополненной реальности был разработан конструктор, позволяющий генерировать стартовую страницу с заданными параметрами. Для этого в него достаточно загрузить модели

формата glb и/или usdz, а результатом будет являться ссылка на стартовую страницу с заданными моделями.

Главная страница конструктора генерирует POST запрос к файлу генерации. В теле запроса передаются файлы моделей, а выходных данных ссылка на сгенерированный проект. Код запроса представлен в листинге 3.2.

Листинг 3.2 – Код POST запрос к файлу генерации

```
var formData = new FormData();

formData.append('google', document.getElementById("input_google").files[0]);

formData.append('ios', document.getElementById("input_ios").files[0]);
$.ajax({
  type: "POST",
  url: 'creator.php',
  cache: false,
  contentType: false,
  processData: false,
  data: formData,
  dataType : 'html',
  success: function(data) {
    document.getElementById("qr").innerHTML = '';
    new QRCode(document.getElementById("qr"), data);
    document.getElementById("url").innerText = (data);
  }
});
```

Файл генерации первоначально создает папку с названием, состоящим из достаточно длинного набора случайных цифр, что позволит отказаться от необходимости создания базы данных проектов. Далее в созданную папку копируется шаблон стартовой страницы, а также создается папка для хранения моделей. Код данной части кода представлен в листинге 3.3.

Листинг 3.3 – Код генерации проекта

```
$allow = array('glb', 'usdz');
$file = 'teemplate/index.php';
$rand = rand();
mkdir('../'.$rand);
mkdir('../'.$rand.'/model');
$path = '../'.$rand.'/model/';
$newfile = '../'.$rand.'/index.php';
copy($file, $newfile);
echo "https://gameofmind.ru/ovsyankin_kniga/ar/".$rand;
```

После этого происходит проверка файлов, полученных через POST запрос. Первоначально необходимо убедиться существует ли файл, далее проверить соответствие файла необходимому формату, и, если все условия

соблюдены, файл сохраняется в папку для моделей. Код проверки glb файла представлен в листинге 3.4, для проверки файлов usdz код аналогичен.

Листинг 3.4 – Код проверки файлов

```
if (!isset($_FILES['google'])) {
    $error = 'Файл не загружен.';
} else {
    $file = $_FILES['google'];
    $pattern = "[^a-zA-яё0-9,~!@#%^-_\\$\\?\\(\\)\\{\\}\\[\\]\\.]*";
    $name = mb_ereg_replace($pattern, '-', $file['name']);
    $name = mb_ereg_replace('[-]+', '-', $name);
    $parts = pathinfo($name);

    if (empty($name) || empty($parts['extension'])) {
        $error = 'Недопустимый тип файла';
    } elseif (!empty($allow) &&
!in_array(strtolower($parts['extension']), $allow)) {
        $error = 'Недопустимый тип файла';
    } else {
        if (move_uploaded_file($file['tmp_name'], $path .
$name)) {
            $success = '<p style="color: green">Файл «' .
$name . '» успешно загружен.</p>';
        } else {
            $error = 'Не удалось загрузить файл.';
        }
    }
}
}
```

Сам шаблон стартовой страницы представляет собой заголовку макета стартовой страницы, в начале которой получаем список содержимого папки с моделями и генерируем ссылку с использованием найденных файлов. Код генератора ссылок представлен в листинге 3.5.

Листинг 3.5 – Код генератора ссылок

```
<?php
$files = scandir('model');
$google = '';
$ios = '';
foreach ($files as $file) {
    if(strpos($file, '.glb')) $google =
'intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/vlad_model'.s
trrchr(str_replace('\\', '/', getcwd()),
'/').'/model/'.$file.'#Intent;scheme=https;package=com.google.android.
googlequicksearchbox;action=android.intent.action.VIEW;S.browser_fallb
ack_url=https://developers.google.com/ar;end;';
    if(strpos($file, '.usdz')) $ios =
'https://gameofmind.ru/ovsyankin_kniga/ar/vlad_model'.strrchr(str_repl
ace('\\', '/', getcwd()), '/') .'/model/'.$file;
}
?>
```

Интерфейс конструктора представлен на рисунке 3.1, а сгенерированная страница на рисунке 3.2.



Рисунок 3.1 – Конструктор AR

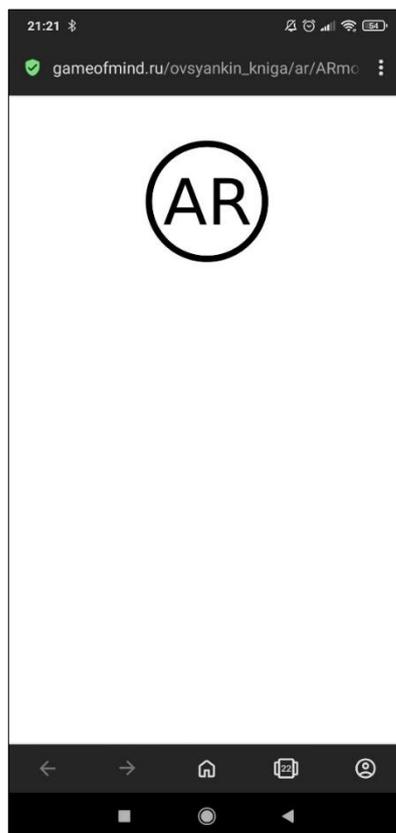


Рисунок 3.2 – Страница, сгенерированная конструктором

3.3 Разработка VR сцен для романа «Ключ от лабиринта»

На основе библиотеки A-frame был реализован ряд VR сцен для книги романа «Ключ от лабиринта». Рассмотрим несколько из них.

Первая VR сцена — это сцена с анимацией. По сюжету романа Петр Первый на Голанских кораблях подплывает к Соловецкой крепости. Но в крепости их корабли путают с кораблями шведов и открывают огонь. Петр Первый разворачивает корабли и отплывает от крепости. Сцена представлена на рисунке 3.3.



Рисунок 3.3 – Сцена с крепостью

Рассмотрим код данной сцены. Вся сцена представляет собой объект «a-scene», внутри которого находятся все остальные элементы. Также в этом объекте задаются настройки сцены, включены тени, а также убрана кнопка перехода в режим дополненной реальности. После в объект «a-assets» добавляются все текстуры и модели, которые будут использованы в данной сцене. Код «a-assets» представлен в листинге 3.6.

Листинг 3.6 – Код «a-assets»

```
<a-assets>
  <a-asset-item id="Solovki" src="../model/solovki.glb"></a-
asset-item>
</a-assets>
```

Камера представлена объектом «a-camera», в нем указываются некоторые параметры, разрешен поворот камеры, но запрещено её перемещение. Для работы с камерой в режиме виртуальной реальности необходимо её поместить в контейнер «a-entity». Код камеры представлен в листинге 3.7.

Листинг 3.7 – Код камеры

```
<a-entity id="boxCamera" position="11.837 4.218 -6.068" rotation="0 0 0">  
  <a-camera id="camera" look-controls wasd-controls-enabled="false" position="0 1.6 0">  
    </a-camera>  
  </a-entity>
```

Добавление объектов также происходит с помощью объекта «a-entity», в нем указано что это будет 3d модель, её размеры и положение, а также добавлен модификатор «animation mixer», который позволяет использовать анимацию, встроенную в модель. Код объекта представлен в листинге 3.8.

Листинг 3.8 – Код объекта

```
<a-entity gltf-model="#Solovki" scale="1.25 1.25 1.25" position="0 0 0" animation-mixer></a-entity>
```

Далее идут объекты освещения «a-light», для них указывается тип и интенсивность. Код освещения представлен в листинге 3.9.

Листинг 3.9 – Код освещения

```
<a-light type="ambient" intensity="0.2"></a-light>  
<a-light type="point" intensity="1.8" position="16.516 94.542 -27.404"></a-light >
```

Последним задается задний фон, объект «a-sky» с указанием цвета заднего фона. Код заднего фона представлен в листинге 3.10.

Листинг 3.10 – Код сцены A-frame

```
<a-sky color="#74b6d2"></a-sky>
```

Другая VR-сцена представляет собой модель летописца, которую пользователь должен рассмотреть с разных сторон. Сцена представлена на рисунке 3.4.

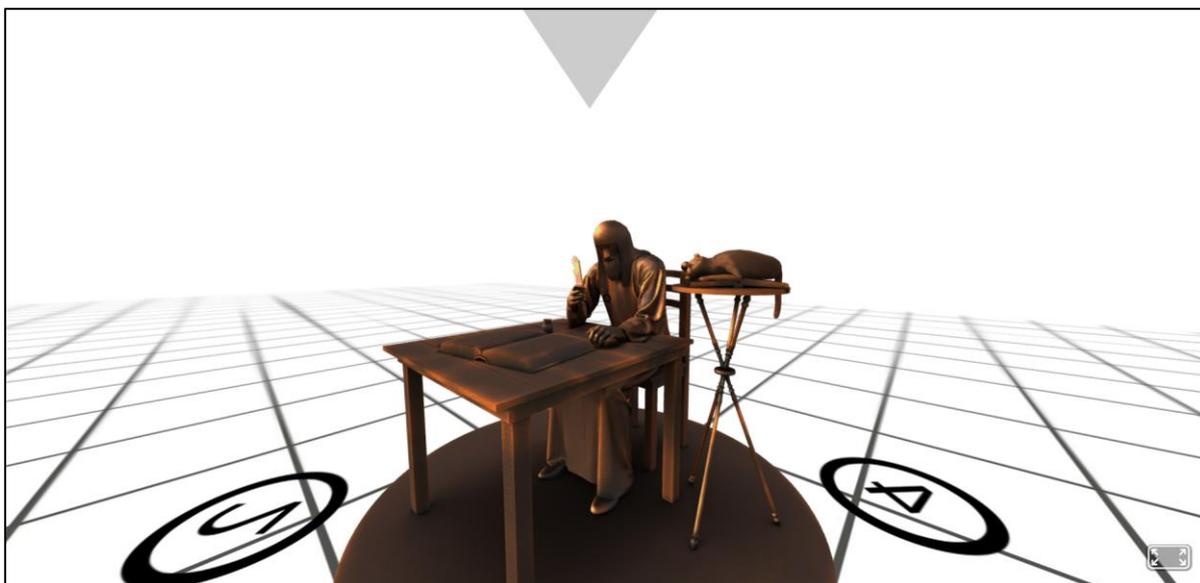


Рисунок 3.4 – Сцена с летописцем

Для перемещения пользователя по сцене было реализовано два способа. Первый — это перемещение по точкам. Когда пользователь смотрит определенное время на точку, представленную в виде круга на полу или пирамиды над летописцем, он перемещается в эту точку. Второй способ — это «панель с кнопками» перед пользователем. При удержании взгляда на кнопке пользователь «переносится» в определенную точку. Интерфейс второго способа представлен на рисунке 3.5.

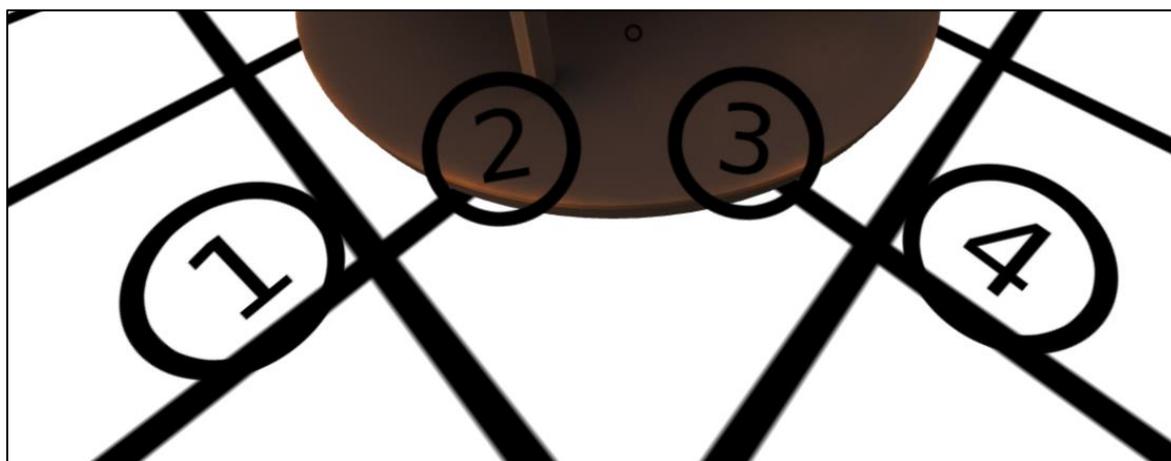


Рисунок 3.5 – «Панель с кнопками»

Код у обоих способов схож, поэтому будет рассмотрен одного из них. Был создан новый модификатор «tr» для объектов, который используется для

всех точек перемещения. Каждому объекту добавлен обработчик события клика по нему, который возвращает точку, из которой уходит пользователь.

Задаёт ей изначальный вид, устанавливает ее видимость на значение «true» и меняет класс «tp-box» на «tp». А точку, в которую перемещается пользователь, наоборот делает невидимой и соответственно меняет класс «tp» на «tp-box». После происходит само перемещение камеры в нужную точку. Код модификатора представлен в листинге 3.11.

Листинг 3.11 – Код модификатора перемещения

```
AFRAME.registerComponent('tp', {
  init: function () {
    this.el.addEventListener("click", () => {
      var els = document.querySelectorAll('.tp-box');
      for (var i = 0; i < els.length; i++) {
        els[i].setAttribute('visible', true);
        els[i].classList.add('tp');
        els[i].classList.remove('tp-box');
      }
      this.el.setAttribute('visible', false);
      this.el.classList.remove('tp');
      this.el.classList.add('tp-box');

      document.querySelector('#boxCamera').setAttribute('position', {
        x: this.el.getAttribute('position').x,
        y: this.el.getAttribute('position').y-1,
        z: this.el.getAttribute('position').z});});});});});
```

Еще одна реализованная сцена «Лабиринт с ключом», где на земле расположен лабиринт, а над ним вращается ключ. Сцена представлена на рисунке 3.6.



Рисунок 3.6 – Сцена «Лабиринт с ключом»

Для симуляции перемещения был реализован модификатор «povorot2d». Он перемещает модель в зависимости от того, куда смотрит пользователь. Код модификатора поворота представлен в листинге 3.12.

Листинг 3.12 – Код модификатора поворота

```
AFRAME.registerComponent('povorot2d', {
  init: function () {
    setInterval(() => {
      this.el.setAttribute('position', {
        x: 4 * Math.sin(-
document.querySelector('#camera').getAttribute('rotation').y * Math.PI
/180),
        y: this.el.getAttribute('position').y,
        z: -4 * Math.cos(-
document.querySelector('#camera').getAttribute('rotation').y * Math.PI
/180),
      });
    }, 1 );
  }
});
```

Таким образом был реализован конструктор AR для упрощения генерации стартовых страниц модулей дополненной и виртуальной и тестирования моделей в дополненной реальности. Были разработаны несколько VR-сцен. Доступ к VR и AR сценам будет интегрироваться в печатное издание посредством QR-кодов, отсканировав которые пользователь перейдет на стартовую страницу. QR-коды будут расположены на специально выделенных для этого страницах. Данные модули также можно дополнить, используя технологии AR с отслеживанием изображения, тогда можно будет использовать иллюстрации из книги в качестве меток.

ЗАКЛЮЧЕНИЕ

В современном мире книги уже, к сожалению, уступают другим способам времяпрепровождения. Молодому поколению становятся более привлекательны интерактивные и визуальные развлечения. Однако применение современных технологий, таких как виртуальная и дополненная реальность, позволяет художественным произведениям быть интересными и увлекательными для юных читателей, открывает новые возможности для демонстрации различных сюжетных линий, сцен и персонажей.

Целью выпускной квалификационной работы стала разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта».

Для достижения поставленной цели были проанализированы существующие на сегодняшний день способы реализации виртуальной и дополненной реальности, выделены их основные преимущества и недостатки. Были рассмотрены и опробованы основные возможности как онлайн-конструкторов, такие как Stories Ar, Artar и ARstudio, так и библиотеки и фреймворки, позволяющие создать собственные VR/AR проекты. Среди них фреймворк A-frame, его дополнение «aframe-extras», а также библиотеки, основанные на фреймворке A-frame – AR.js и MindAR. Кроме того, проанализированы возможности платформ, создаваемых крупнейшими ИТ-гигантами как Google (ARCore) и Apple (ARKit) и сделаны выводы о ситуациях, когда применимы те или иные библиотеки и технологии, используемые для печатных изданий, онлайн музеев и туров, а также в сфере маркетинга. Так, если не требуется взаимодействие со сценой, то можно использовать ARCore scene viewer и ARKit Quick Look. Если в качестве метки необходимо использовать изображение, то подойдет MindAR или онлайн-конструктор ARstudio. Для проектов, в которых необходимо осуществлять взаимодействие пользователя со сценой, наиболее эффективным при текущем уровне развития технологий будет применение A-frame.

Результатом выполнения выпускной квалификационной работы стал конструктор AR проектов, который упрощает процесс генерации и тестирования модулей виртуальной и дополненной реальности, а также набор VR-сцен к печатному изданию романа «Ключ от лабиринта». Созданные сцены сделают роман более визуализированным и интерактивным, предоставят читателю дополнительную информацию об персонажах, событиях и месте действий романа, подарят читателям новый иммерсивный опыт.

Результаты, полученные в ходе выполнения работы, могут быть применимы при реализации модулей виртуальной и дополненной реальности для других художественных произведений, обучающей литературы, а также в проектах, связанных с онлайн-туризмом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Использование технологий виртуальной реальности (vr) и дополненной реальности (ar) в финансовой сфере [Электронный ресурс]. – Режим доступа: <https://vestnik.volbi.ru/upload/numbers/158/article-158-3253.pdf>, свободный (дата обращения 10.06.2023).

2. Технологии дополненной и виртуальной реальностей: инновации в обучении иностранным языкам в вузе [Электронный ресурс]. – Режим доступа: <http://rrpedagogy.ru/journal/article/2488/>, свободный (дата обращения 10.06.2023).

3. Повышение уровня безопасности производственных процессов машиностроения посредством технологий виртуальной и дополненной реальности [Электронный ресурс]. – Режим доступа: <https://vestnik.vogu35.ru/docs/2020/tekhnich/2/17.pdf>, свободный (дата обращения 10.06.2023).

4. The Benefits of Augmented Reality vs. Virtual Reality [Электронный ресурс]. – Режим доступа: [https://vr-app.ru/blog/the-benefits-of-augmented-reality-vs-virtual-reality/#:~:text=Дополненная%20реальность%20\(AR\)%20-%20это,смартфонов%2C%20планшетов%2C%20AR-очков%20или%20гарнитур](https://vr-app.ru/blog/the-benefits-of-augmented-reality-vs-virtual-reality/#:~:text=Дополненная%20реальность%20(AR)%20-%20это,смартфонов%2C%20планшетов%2C%20AR-очков%20или%20гарнитур), свободный (дата обращения 11.06.2023).

5. AR — Дополненная Реальность (статья плюс ролик) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/419437/>, свободный (дата обращения 11.06.2023).

6. Очки дополненной реальности: где мы сейчас? (статья плюс ролик) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/503630/>, свободный (дата обращения 11.06.2023).

7. Куда нас погружают иммерсивные технологии (статья плюс ролик) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/vtb/articles/463707/>, свободный (дата обращения 11.06.2023).

8. Как работает VR? Разбор [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/droider/articles/538748/>, свободный (дата обращения 11.06.2023).

9. Плюсы и минусы виртуальной реальности и дополненной реальности [Электронный ресурс]. – Режим доступа: <https://vr-app.ru/blog/pliusy-i-minusy-virtualnoi-realnosti-i-dopolnennoi-realnosti/>, свободный (дата обращения 11.06.2023).

10. Проблемы безопасности при использовании VR: возможные риски и способы их предотвращения. [Электронный ресурс]. – Режим доступа: <https://vr-app.ru/blog/problemy-bezopasnosti-pri-ispolzovanii-vr-vozmoznye-riski-i-sposoby-ix-predotvrashheniia/>, свободный (дата обращения 11.06.2023).

11. Введение в виртуальную реальность и дополненную реальность. [Электронный ресурс]. – Режим доступа: <https://vr-app.ru/blog/vvedenie-v-virtualnuiu-realnost-i-dopolnennuiu-realnost/>, свободный (дата обращения 11.06.2023).

12. Дополненная и виртуальная реальность в образовании [Электронный ресурс]. – Режим доступа: <https://moluch.ru/archive/223/52655/>, свободный (дата обращения 10.06.2023).

13. Number of mobile augmented reality (AR) active user devices worldwide from 2019 to 2024 [Электронный ресурс]. – Режим доступа: <https://www.statista.com/statistics/1098630/global-mobile-augmented-reality-ar-users/>, свободный (дата обращения 05.06.2023).

14. Cardboard [Электронный ресурс]. – Режим доступа: https://arvr.google.com/intl/en_en/cardboard/, свободный (дата обращения 16.03.2023).

15. Учебник физики с дополненной реальностью [Электронный ресурс]. – Режим доступа: http://funreality.ru/product/ar_textbook/, свободный (дата обращения 10.06.2023).

16. TrueVirtualTours [Электронный ресурс]. – Режим доступа: <https://truevirtualtours.com/ru/>, свободный (дата обращения 10.06.2023).

17. Сибур диджитал [Электронный ресурс]. – Режим доступа: <https://www.sibur.digital/projects/ar>, свободный (дата обращения 10.06.2023).

18. Дополненная реальность в маркетинге: 15+ примеров использования [Электронный ресурс]. – Режим доступа: <https://texterra.ru/blog/dopolnennaya-realnost-v-marketinge-primery-ispolzovaniya.html>, свободный (дата обращения 10.06.2023).

19. Продукты для vr [Электронный ресурс]. – Режим доступа: <https://store.steampowered.com/vr/>, свободный (дата обращения 10.06.2023).

20. A-frame [Электронный ресурс]. – Режим доступа: <https://aframe.io>, свободный (дата обращения 19.04.2023).

21. Обзор ARCore и поддерживаемых сред разработки [Электронный ресурс]. – Режим доступа: <https://developers.google.com/ar/develop?hl=ru>, свободный (дата обращения 21.03.2023).

22. WebXR [Электронный ресурс]. – Режим доступа: <https://developers.google.com/ar/develop/webxr/model-viewer?hl=ru>, свободный (дата обращения 22.03.2023).

23. Дополненная реальность с <model-viewer> [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/47876/>, свободный (дата обращения 21.03.2023).

24. Использование Scene Viewer для отображения интерактивных 3D-моделей в дополненной реальности из приложения или браузера для Android. [Электронный ресурс]. – Режим доступа: <https://developers.google.com/ar/develop/scene-viewer?hl=ru>, свободный (дата обращения 21.03.2023).

25. Previewing a Model with AR Quick Look реальность [Электронный ресурс]. – Режим доступа: https://developer.apple.com/documentation/arkit/previewing_a_model_with_ar_quick_look, свободный (дата обращения 22.03.2023).

26. Storiesar [Электронный ресурс]. – Режим доступа: <https://stories-ar.com>, свободный (дата обращения 06.06.2023).

27. artar [Электронный ресурс]. – Режим доступа: <https://design.artar.es>, свободный (дата обращения 06.06.2023).

28. AR studio [Электронный ресурс]. – Режим доступа: <https://web-ar.studio/ru/>, свободный (дата обращения 06.06.2023).

29. AR.js - Augmented Reality on the Web [Электронный ресурс]. – Режим доступа: <https://ar-js-org.github.io/AR.js-Docs/>, свободный (дата обращения 21.03.2023).

30. MindAR [Электронный ресурс]. – Режим доступа: <https://hiukim.github.io/mind-ar-js-doc/>, свободный (дата обращения 21.03.2023).

31. Гайд по Visual Studio Code: как работать с популярным инструментом разработчиков [Электронный ресурс]. – Режим доступа: <https://practicum.yandex.ru/blog/vsyo-o-visual-studio-code/>, свободный (дата обращения 16.04.2023).

32. IntelliJ IDEA [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/intellij-idea/>, свободный (дата обращения 19.04.2023).

33. PhpStorm [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/ru-ru/phpstorm/>, свободный (дата обращения 12.06.2023).

34. Папагианнис Хелен Дополненная реальность. Все, что вы хотели узнать о технологии будущего [Текст] : общенаучное знание и теории/ Папагианнис Хелен; БОМБОРА, 2019. – 288 с.

35. Линовес Джонатан Виртуальная реальность в Unity [Текст] : другие издания / Линовес Джонатан; ДМК Пресс, 2016. – 316 с.

36. Кириченко Александр Аполлонович Справочник HTML. Кратко, быстро, под рукой [Текст] : Справочник / Кириченко Александр Аполлонович; Наука и техника, 2021. – 288 с.

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программного кода

Листинг А.1 – Код AR с использованием A-frame

```
<html>
  <head>
    <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
    </head>
    <body>
      <a-scene vr-mode-ui="enabled: true" webxr="optionalFeatures: hit-
test;" ar-hit-test="target:#model;">
        <!-- <a-entity id="camera" camera look-controls position="0 1.4
2.5"></a-entity> -->
        <a-entity ar-hit-test="doHitTest:false" visible="false">
        <!-- Reticle HTML goes here ... -->
        </a-entity>
        <a-entity id="boxCamera" position="0 0 1.2">
          <a-camera id="camera" look-controls position="0 1.6 0"></a-
camera>
        </a-entity>
        <a-assets>
          <a-asset-item id="Chronicler"
src="../model/Chronicler_LightTexture.glb"></a-asset-item>
        </a-assets>
        <a-entity id="model" gltf-model="#Chronicler" scale="0.25 0.25
0.25"></a-entity>
        <a-light type="ambient" intensity="1"></a-light>
      </a-scene>
    </body>
  </script>

  AFRAME.registerComponent('log', {
    schema: {type: 'string'},
    init: function () {
      var stringToLog = this.data;
      this.el.addEventListener("click", () => {
        console.log(stringToLog)
      })
    }
  });

  AFRAME.registerComponent('hide-in-ar-mode', {
    init: function () {
      this.el.sceneEl.addEventListener('enter-vr', (ev) => {
        this.wasVisible = this.el.getAttribute('visible');
        if (this.el.sceneEl.is('ar-mode')) {
          this.el.setAttribute('visible', false);
        }
      });
      this.el.sceneEl.addEventListener('exit-vr', (ev) => {
        if (this.wasVisible) {
          this.el.setAttribute('visible', true);
        }
      });
    }
  });
```

```

    }
  });
</script>
</html>

```

Листинг А.2 – Код AR с использованием ARCore «WebXR»

```

<!doctype html>
<html>
<head>
  <script
src="https://unpkg.com/three@0.126.0/build/three.js"></script>
  <script
src="https://unpkg.com/three@0.126.0/examples/js/loaders/GLTFLoader.js
"></script>
</head>
<body>
  <!-- Кнопка для старта AR режима (WebXR требует взаимодействия с
пользователем, чтобы начать сеанс.)-->
<button onclick="activateXR()">Start Hello WebXR</button>
<script>
async function activateXR() {
  // Add a canvas element and initialize a WebGL context that is
compatible with WebXR.
  const canvas = document.createElement("canvas");
  document.body.appendChild(canvas);
  const gl = canvas.getContext("webgl", {xrCompatible: true});

  const scene = new THREE.Scene();

  const directionalLight = new THREE.DirectionalLight(0xffffff, 0.3);
  directionalLight.position.set(10, 15, 10);
  scene.add(directionalLight);

  // Set up the WebGLRenderer, which handles rendering to the
session's base layer.
  const renderer = new THREE.WebGLRenderer({
    alpha: true,
    preserveDrawingBuffer: true,
    canvas: canvas,
    context: gl
  });
  renderer.autoClear = false;

  // The API directly updates the camera matrices.
  // Disable matrix auto updates so three.js doesn't attempt
  // to handle the matrices independently.
  const camera = new THREE.PerspectiveCamera();
  camera.matrixAutoUpdate = false;

  // Initialize a WebXR session using "immersive-ar".
  const session = await navigator.xr.requestSession("immersive-ar",
{requiredFeatures: ['hit-test']});
  session.updateRenderState({
    baseLayer: new XRWebGLLayer(session, gl)
  });

  // A 'local' reference space has a native origin that is located
  // near the viewer's position at the time the session was created.

```

```

const referenceSpace = await session.requestReferenceSpace('local');

// Create another XRReferenceSpace that has the viewer as the
origin.
const viewerSpace = await session.requestReferenceSpace('viewer');
// Perform hit testing using the viewer as origin.
const hitTestSource = await session.requestHitTestSource({ space:
viewerSpace });
// загрузка модели
const loader = new THREE.GLTFLoader();
//Возможно невидимая модель для прицеливания
let reticle;
loader.load("https://immersive-web.github.io/webxr-
samples/media/gltf/reticle/reticle.gltf", function(gltf) {
    reticle = gltf.scene;
    reticle.visible = false;
    scene.add(reticle);
})

let Chronicler;
loader.load("../model/Chronicler_LightTexture.glb", function(gltf)
{
    Chronicler = gltf.scene;
});
//Размещать объект по нажатию
session.addEventListener("select", (event) => {
    if (Chronicler) {
        const clone = Chronicler.clone();
        clone.position.copy(reticle.position);
        scene.add(clone);
    }
});

// Create a render loop that allows us to draw on the AR view.
const onXRFrame = (time, frame) => {
    // Queue up the next draw request.
    session.requestAnimationFrame(onXRFrame);

    // Bind the graphics framebuffer to the baseLayer's framebuffer
    gl.bindFramebuffer(gl.FRAMEBUFFER,
session.renderState.baseLayer.framebuffer)

    // Retrieve the pose of the device.
    // XRFrame.getViewerPose can return null while the session
attempts to establish tracking.
    const pose = frame.getViewerPose(referenceSpace);
    if (pose) {
        // In mobile AR, we only have one view.
        const view = pose.views[0];

        const viewport =
session.renderState.baseLayer.getViewport(view);
        renderer.setSize(viewport.width, viewport.height)

        // Use the view's transform matrix and projection matrix to
configure the THREE.camera.
        camera.matrix.fromArray(view.transform.matrix)
        camera.projectionMatrix.fromArray(view.projectionMatrix);

```

```

        camera.updateMatrixWorld(true);

        //Рисуем прицельную сетку
        const hitTestResults = frame.getHitTestResults(hitTestSource);
        if (hitTestResults.length > 0 && reticle) {
            const hitPose = hitTestResults[0].getPose(referenceSpace);
            reticle.visible = true;
            reticle.position.set(hitPose.transform.position.x,
hitPose.transform.position.y, hitPose.transform.position.z)
            reticle.updateMatrixWorld(true);
        }

        // Render the scene with THREE.WebGLRenderer.
        renderer.render(scene, camera)
    }
}
session.requestAnimationFrame(onXRFrame);
}
</script>
</body>
</html>

```

Листинг А.3 – Код AR с использованием ARCore «model-viewer»

```

<!doctype html>
<html>
<head>
    <script type="module" src="https://unpkg.com/@google/model-
viewer/dist/model-viewer.min.js"></script>
</head>
<body>
    <style>
        #model{
            margin: 0px;
            padding:0px;
            width: 100vw;
            height: 100vh;
        }
        body{
            margin: 0px;
            padding:0px;
        }
    </style>
    <model-viewer id="model" src="../../../model/Chronicler_LightTexture.glb"
        alt="Модель"
        ar
        auto-rotate
        camera-controls
        shadow-intensity="4"
        exposure="0.4"
        environment-image="legacy"
        scale="0.5 0.5 0.5">
        <button slot="ar-button" style="background-color:
gray; border-radius: 4px; border: none; position: absolute; top: 16px;
right: 16px; width: 100px; height: 100px;">⌵ Activate AR</button>
    </model-viewer>
</body>
</html>

```

Листинг А.4 – Код AR с использованием ARCore «scene view»

```
<!doctype html>
<html>
<head>
</head>
<body>
<a href="intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/model/Chronic
ler_LightTexture.glb#Intent;scheme=https;package=com.google.android.go
oglequicksearchbox;action=android.intent.action.VIEW;S.browser_fallbac
k_url=https://developers.google.com/ar;end;">Ссылка</a>
<br>
<br>
<br>
<a href="intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/model/newSolo
vki.glb#Intent;scheme=https;package=com.google.android.googlequicksear
chbox;action=android.intent.action.VIEW;S.browser_fallback_url=https://
developers.google.com/ar;end;">Solovki</a>
</body>
</html>
```

Листинг А.5 – Код AR с использованием ARkit

```
<!doctype html>
<html>
<head>
  <style>
    img{
      width: 100%;
      height: auto;
    }
    a{
      width: 33%;
      margin-top: 10%;
      text-align: center;
      font-size: 3vw;
      text-decoration: none;
      font-family:sans-serif;
      color: black;
      font-weight:bold
    }
    body{
      display: flex;
      flex-direction: column;
      align-items: center;
    }
  </style>
</head>
<?php
preg_match("/iPhone|Android|iPad|iPod|webOS/",
$_SERVER['HTTP_USER_AGENT'], $matches);
$os = current($matches);

switch($os){
  case 'iPhone': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/zachita/Chronicler_LightText
ure.usdz'; break;
```

```

    case 'iPad': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/zachita/Chronicler_LightText
ure.usdz'; break;
    case 'iPod': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/zachita/Chronicler_LightText
ure.usdz'; break;
    case 'Android': $link_app = 'intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/zachita/Chron
icler_LightTexture.glb#Intent;scheme=https;package=com.google.android.
googlequicksearchbox;action=android.intent.action.VIEW;S.browser_fallb
ack_url=https://developers.google.com/ar;end;'; break;
    default: $link_app = '';
}
?>
<body>
<a rel="ar" href="<?echo $link_app;?>"></a>
<!-- <a href="">С МЕТКОЙ</a> -->
</body>
</html>

```

Листинг А.6 – Код AR с использованием AR.js с отслеживанием изображения

```

<script src="aframe-master.min.js"></script>
<script src="https://raw.githack.com/AR-js-
org/AR.js/master/aframe/build/aframe-ar-nft.js"></script>
<script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>

<!-- style for the loader -->
<style>
  .arjs-loader {
    height: 100%;
    width: 100%;
    position: absolute;
    top: 0;
    left: 0;
    background-color: rgba(0, 0, 0, 0.8);
    z-index: 9999;
    display: flex;
    justify-content: center;
    align-items: center;
  }

  .arjs-loader div {
    text-align: center;
    font-size: 1.25em;
    color: white;
  }
</style>

<body style="margin : 0px; overflow: hidden;">
  <!-- minimal loader shown until image descriptors are loaded.
Loading may take a while according to the device computational power -
->
  <div class="arjs-loader">
    <div>Loading, please wait...</div>
  </div>

  <!-- a-frame scene -->
  <a-scene

```

```

vr-mode-ui="enabled: false;"
renderer="logarithmicDepthBuffer: true;"
embedded
arjs="trackingMethod: best; sourceType: webcam; debugUIEnabled:
false;"
>
  <!-- a-nft is the anchor that defines an Image Tracking entity -->
  <!-- on 'url' use the path to the Image Descriptors created
before. -->
  <!-- the path should end with the name without the extension e.g.
if file is 'pinball.fset' the path should end with 'pinball' -->
  <a-nft
    type="nft"
    url="https://gameofmind.ru/ovsyankin_kniga/ar/ARjs/metka"
    smooth="true"
    smoothCount="15"
    smoothTolerance=".01"
    smoothThreshold="5"
  >
    <!-- as a child of the a-nft entity, you can define the content
to show. here's a GLTF model entity -->
    <a-entity
      gltf-
model="https://gameofmind.ru/ovsyankin_kniga/ar/model/newSolovki.glb"
      scale="100 100 100"
      position="100 0 -100"
      clickable
    >
    </a-entity>
  </a-nft>
  <!-- static camera that moves according to the device movemenents
-->
  <a-entity camera></a-entity>
</a-scene>
</body>
<script>
  AFRAME.registerComponent('clickable', {
    init: function () {
      this.el.addEventListener("click", () => {
        this.el.setAttribute('scale', {
          x: 10,
          y: 10,
          z: 10
        });
      });
    }
  });
</script>

```

Листинг А.7 – Код AR с использованием AR.js с отслеживанием «метки»

```

<html>
  <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
  <script src="https://raw.githack.com/AR-js-
org/AR.js/master/aframe/build/aframe-ar.js"></script>
  <body style="margin : 0px; overflow: hidden;">

```

```

    <a-scene vr-mode-ui='enabled: false;' embedded
arjs='trackingMethod: best; sourceType: webcam; debugUIEnabled: false;
patternRatio: 0.5;' renderer='logarithmicDepthBuffer: true;'>
    <a-marker type = 'pattern' url="pattern50.patt" smooth="true">
        <a-entity
            position="0 0 0"
            scale="0.05 0.05 0.05"
            gltf-model="../model/newSolovki.glb"
        ></a-entity>
    </a-marker>
    <a-marker type = 'pattern' url="pattern50B.patt" smooth="true">
        <a-entity
            position="0 0 0"
            scale="0.05 0.05 0.05"
            gltf-model="../model/newSolovki.glb"
        ></a-entity>
    </a-marker>
    <a-entity camera></a-entity>
</a-scene>
</body>
</html>

```

Листинг А.8 – Код AR с использованием MindAR

```

<!DOCTYPE HTML>
<html lang = "ru">
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1"
charset="utf-8"/>
    <script src="mind-ar-js-master/dist/mindar-image.prod.js"></script>
    <script src="aframe.min.js"></script>
    <script src="mind-ar-js-master/dist/mindar-image-
aframe.prod.js"></script>
</head>
<body>
    <a-scene mindar-image=" filterMinCF:0.01; filterBeta: 1000;
imageTargetSrc: mind/targets.mind;" vr-mode-ui="enabled: false"
device-orientation-permission-ui="enabled: false">
        <a-camera position="0 0 0" look-controls="enabled: false"></a-
camera>
        <a-assets>
            <a-asset-item id="avatarModel" src="model/newSolovki.glb"></a-
asset-item>
        </a-assets>
        <a-entity mindar-image-target="targetIndex: 0">
            <a-gltf-model rotation="90 0 0" position="0 0 0" scale="0.02
0.02 0.02" src='#avatarModel'>
        </a-entity>
    </a-scene>
</body>
</html>

```

Листинг А.9 – Код стартовой страницы

```

<!doctype html>
<html>
<head>
    <style>
        img{

```

```

        width: 100%;
        height: auto;
    }
    a{
        width: 33%;
        margin-top: 10%;
        text-align: center;
        font-size: 3vw;
        text-decoration: none;
        font-family:sans-serif;
        color: black;
        font-weight:bold
    }
    body{
        display: flex;
        flex-direction: column;
        align-items: center;
    }
</style>
</head>
<body>
<a rel="ar" href="<?php
preg_match("/iPhone|Android|iPad|iPod|webOS/",
$_SERVER['HTTP_USER_AGENT'], $matches);
$os = current($matches);

switch($os){
    case 'iPhone': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/model/test2d.usdz'; break;
    case 'iPad': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/model/test2d.usdz'; break;
    case 'iPod': $link_app =
'https://gameofmind.ru/ovsyankin_kniga/ar/model/test2d.usdz'; break;
    case 'Android': $link_app = 'intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/model/W_hlmau
s.glb#Intent;scheme=https;package=com.google.android.googlequicksearch
box;action=android.intent.action.VIEW;S.browser_fallback_url=https://d
evelopers.google.com/ar;end;'; break;
    default: $link_app =
'https://yandex.ru/search/?text=switch+php&lr=20&clid=2456107';
}
    echo $link_app;
?>"></a>
</body>
</html>

```

Листинг А.10 – Код главной страницы конструктора

```

<!doctype html>
<html>
<head>
    <style>
        img{
            width: 100%;
            height: auto;
        }
        a{
            width: 33%;
            text-align: center;

```

```

    }
    body{
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
    }
    div{
        display: flex;
        flex-direction: row;
        align-items: center;
        justify-content: center;
        font-size: 3vw;
        text-decoration: none;
        font-family:sans-serif;
        color: black;
        font-weight:bold
    }
</style>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"
></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.
js"></script>
</head>
<body>
    GLB: <input id="input_google" type="file" accept=".glb"/>
    USDZ: <input id="input_ios" type="file" accept=".usdz"/>
    <input type="submit" value="Загрузить файлы" onclick="TEST2()"
/>

    <div style="flex-direction: column;">
        <div id="qr"></div>
        <div id="url"></div>
    </div>
</body>
</html>

<script>
    function TEST2(){
        var formData = new FormData();

formData.append('google',document.getElementById("input_google").files
[0]);

formData.append('ios',document.getElementById("input_ios").files[0]);
$.ajax({
    type: "POST",
    url: 'creator.php',
    cache: false,
    contentType: false,
    processData: false,
    data: formData,
    dataType : 'html',
    success: function(data){
        document.getElementById("qr").innerHTML = '';
        new QRCode(document.getElementById("qr"), data);
    }
});

```

```

        document.getElementById("url").innerText = (data);
    }
    });
}
</script>

```

Листинг А.11 – Код генератора проектов конструктора

```

<?php
// Разрешенные расширения файлов.
$allow = array('glb', 'usdz');

$file = 'teemplate/index.php';
$rand = rand();
mkdir('../' . $rand);
mkdir('../' . $rand . '/model');
// Директория куда будут загружаться файлы.
$path = '../' . $rand . '/model/';
$newfile = '../' . $rand . '/index.php';
copy($file, $newfile);
echo "https://gameofmind.ru/ovsyankin_kniga/ar/" . $rand;

$error = $success = '';
if (!isset($_FILES['google'])) {
    $error = 'Файл не загружен.';
} else {
    $file = $_FILES['google'];
    // Оставляем в имени файла только буквы, цифры и некоторые
СИМВОЛЫ.
    $pattern = "[^a-zA-яё0-9,~!@#%^-_\$?\(\\)\{\}\[\]\. ]";
    $name = mb_ereg_replace($pattern, '-', $file['name']);
    $name = mb_ereg_replace('[-]+', '-', $name);
    $parts = pathinfo($name);

    if (empty($name) || empty($parts['extension'])) {
        $error = 'Недопустимый тип файла';
    } elseif (!empty($allow) &&
!in_array(strtolower($parts['extension']), $allow)) {
        $error = 'Недопустимый тип файла';
    } else {
        // Перемещаем файл в директорию.
        if (move_uploaded_file($file['tmp_name'], $path .
$name)) {
            // Далее можно сохранить название файла в БД и
т.п.
            $success = '<p style="color: green">Файл «' .
$name . '» успешно загружен.</p>';
        } else {
            $error = 'Не удалось загрузить файл.';
        }
    }
}
if (!isset($_FILES['ios'])) {
    $error = 'Файл не загружен.';
} else {
    $file = $_FILES['ios'];
    // Оставляем в имени файла только буквы, цифры и некоторые
СИМВОЛЫ.
    $pattern = "[^a-zA-яё0-9,~!@#%^-_\$?\(\\)\{\}\[\]\. ]";

```

```

$name = mb_ereg_replace($pattern, '-', $file['name']);
$name = mb_ereg_replace('[-]+', '-', $name);
$parts = pathinfo($name);

if (empty($name) || empty($parts['extension'])) {
    $error = 'Недопустимый тип файла';
} elseif (!empty($allow) &&
!in_array(strtolower($parts['extension']), $allow)) {
    $error = 'Недопустимый тип файла';
} else {
    // Перемещаем файл в директорию.
    if (move_uploaded_file($file['tmp_name'], $path .
$name)) {
        $success = '<p style="color: green">Файл «' .
$name . '» успешно загружен.</p>';
    } else {
        $error = 'Не удалось загрузить файл.';
    }
}
}

```

Листинг А.12 – Код шаблона проекта конструктора

```

<?php
$files = scandir('model');
$google = '';
$ios = '';
foreach ($files as $file) {
    if(strpos($file, '.glb')) $google =
'intent://arvr.google.com/scene-
viewer/1.0?file=https://gameofmind.ru/ovsyankin_kniga/ar/vlad_model'.s
trrchr(str_replace('\', '/', getcwd()),
'/').'/model/'.$file.'#Intent;scheme=https;package=com.google.android.
googlequicksearchbox;action=android.intent.action.VIEW;S.browser_fallb
ack_url=https://developers.google.com/ar;end;';
    if(strpos($file, '.usdz')) $ios =
'https://gameofmind.ru/ovsyankin_kniga/ar/vlad_model'.strrchr(str_repl
ace('\', '/', getcwd()), '/').'/model/'.$file;
}
?>

<!doctype html>
<html>
<head>
    <style>
        img{
            width: 100%;
            height: auto;
        }
        a{
            width: 33%;
            margin-top: 10%;
            text-align: center;
            font-size: 3vw;
            text-decoration: none;
            font-family:sans-serif;
            color: black;
            font-weight:bold

```

```

    }
    body{
        display: flex;
        flex-direction: column;
        align-items: center;
    }
</style>
</head>
<body>
<a rel="ar" href="<?php
preg_match("/iPhone|Android|iPad|iPod|webOS/",
$_SERVER['HTTP_USER_AGENT'], $matches);
$os = current($matches);

switch($os){
    case 'iPhone': $link_app = $ios; break;
    case 'iPad': $link_app = $ios; break;
    case 'iPod': $link_app = $ios; break;
    case 'Android': $link_app = $google; break;
    default: $link_app = '';
}
    echo $link_app;
?>"></a>
</body>
</html>

```

Листинг А.13 – Код VR-сцены «Соловецкая крепость»

```

<html>
  <head>
    <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
    <script src="../../../aframe-extras-master/dist/aframe-
extras.min.js"></script>
  </head>
  <body>
    <a-scene shadow="type: pcf" vr-mode-ui="enabled: true;
enterARButton: #enterARButton">
      <a id='enterARButton'></a>
      <a-assets>
        <a-asset-item id="Solovki"
src="../../../zachita2/solovki.glb"></a-asset-item>
      </a-assets>

      <a-entity id="boxCamera" position="11.837 4.218 -6.068"
rotation="0 0 0">
        <a-camera id="camera" look-controls wasd-controls-
enabled="false" position="0 1.6 0">
          </a-camera>
        </a-entity>
        <a-entity gltf-model="#Solovki" scale="1.25 1.25 1.25"
position="0 0 0" animation-mixer></a-entity>

      <a-light type="ambient" intensity="0.2"></a-light>

      <a-light type="point" intensity="1.8" position="16.516 94.542 -
27.404"></a-entity>
      <a-sky color="#74b6d2"></a-sky>

```

```

    </a-scene>
  </body>
</script>
  AFRAME.registerComponent('log', {
    schema: {type: 'string'},
    init: function () {
      var stringToLog = this.data;
      this.el.addEventListener("click", () => {
        console.log(stringToLog)
      })
    }
  });
</script>
</html>

```

Листинг А.14 – Код VR-сцены «Летописец» управление плитами

```

<!--Модель лабиринта с анимацией ключа -->
<html>
  <head>
    <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene vr-mode-ui="enabled: true; enterARButton:
#enterARButton">
      <a id='enterARButton'></a>
      <a-assets>
        <a-asset-item id="Chronicler"
src="../model/Chronicler_LightTexture.glb"></a-asset-item>
        <a-asset-item id="piramid" src="../model/piramid.glb"></a-
asset-item>
        
        
        
        
        
      </a-assets>

      <a-entity id="boxCamera" position="1.5 0 1.5" rotation="0 45 0">
        <a-camera id="camera" look-controls wasd-controls-
enabled="false" position="0 1.6 0">
          <a-cursor raycaster="objects: .tp" fuse="true" position="0
0 -0.01" scale="0.0075 0.0075 0.0075"
          animation__fusing="property: scale; from:0.0075 0.0075
0.0075; to:0.001 0.001 0.001; startEvents: fusing; dur: 1500"
          animation__end="property: scale; to:0.0075 0.0075
0.0075; startEvents: animationcomplete__fusing; dur: 1"
          animation__mouseleave="property: scale; to:0.0075 0.0075
0.0075; startEvents: mouseleave; dur: 1">
            </a-cursor>
          </a-camera>
        </a-entity>

        <a-image src="#pol" position="16 0 9" scale="16 9 0"
rotation="90 0 0"></a-image>
        <a-image src="#pol" position="0 0 9" scale="16 9 0" rotation="90
0 0"></a-image>

```

```

    <a-image src="#pol" position="-16 0 9" scale="16 9 0"
rotation="90 0 0"></a-image>
    <a-image src="#pol" position="16 0 0" scale="16 9 0"
rotation="90 0 0"></a-image>
    <a-image src="#pol" position="0 0 0" scale="16 9 0" rotation="90
0 0"></a-image>
    <a-image src="#pol" position="-16 0 0" scale="16 9 0"
rotation="90 0 0"></a-image>
    <a-image src="#pol" position="16 0 -9" scale="16 9 0"
rotation="90 0 0"></a-image>
    <a-image src="#pol" position="0 0 -9" scale="16 9 0"
rotation="90 0 0"></a-image>
    <a-image src="#pol" position="-16 0 -9" scale="16 9 0"
rotation="90 0 0"></a-image>

    <a-image class="tp" src="#one" position="1.5 0.001 1.5" scale="1
1" rotation="90 45 0" tp></a-image>
    <a-image class="tp" src="#two" position="-1.5 0.001 1.5"
scale="1 1" rotation="90 -45 0" tp></a-image>
    <a-image class="tp" src="#three" position="-1.5 0.001 -1.5"
scale="1 1" rotation="90 -135 0" tp></a-image>
    <a-image class="tp" src="#four" position="1.5 0.001 -1.5"
scale="1 1" rotation="90 135 0" tp></a-image>

    <!-- <a-box class="tp" scale="1 1 1" visible="false" position="-
0.07 1.9 0.33" tp></a-box> -->
    <a-entity class="tp" gltf-model="#pyramid" scale="0.5 0.5 0.5"
position="0 3 0" tp></a-entity>

    <a-entity gltf-model="#Chronicler" scale="1.25 1.25 1.25"></a-
entity>
    <a-light type="ambient" intensity="1"></a-light>
    <a-sky color="FFFFFF"></a-sky>
</a-scene>
</body>
<script>
    AFRAME.registerComponent('log', {
      schema: {type: 'string'},
      init: function () {
        var stringToLog = this.data;
        this.el.addEventListener("click", () => {
          console.log(stringToLog)
        })
      }
    });

    AFRAME.registerComponent('tp', {
      init: function () {
        this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
          x: this.el.getAttribute('position').x,
          z: this.el.getAttribute('position').z});
        })
      }
    });
</script>
</html>

```

Листинг А.15 – Код VR-сцены «Летописец» управление кнопками

```
<html>
  <head>
    <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
    <script src="../aframe-extras-master/dist/aframe-
extras.min.js"></script>
  </head>
  <body>
    <a-scene vr-mode-ui="enabled: true; enterARButton:
#enterARButton">
      <a id='enterARButton'></a>
      <a-assets>
        <a-asset-item id="Chronicler"
src="../model/Chronicler_LightTexture.glb"></a-asset-item>
        
        
        
        
        
      </a-assets>

      <a-entity id="boxCamera" position="1.5 0 1.5" rotation="0 45 0">
        <a-camera id="camera" look-controls wasd-controls-
enabled="false" position="0 1.6 0">
          <a-cursor raycaster="objects: .tp" fuse="true"
animation__fusing="property: scale; from:1 1 1; to:0.1 0.1
0.1; startEvents: fusing; dur: 1500"
animation__end="property: scale; to:1 1 1; startEvents:
animationcomplete__fusing; dur: 1"
animation__mouseleave="property: scale; to:1 1 1;
startEvents: mouseleave; dur: 1">
            </a-cursor>
          </a-camera>
          <a-image class="tp" src="#one" position="-0.2 1.3 -0.125"
scale="0.1 0.1" rotation="-45 50" tp_one></a-image>
          <a-image class="tp" src="#two" position="-0.075 1.3 -0.2"
scale="0.1 0.1" rotation="-45 10 0" tp_two></a-image>
          <a-image class="tp" src="#three" position="0.075 1.3 -0.2"
scale="0.1 0.1" rotation="-45 -10 0" tp_three></a-image>
          <a-image class="tp" src="#four" position="0.2 1.3 -0.125"
scale="0.1 0.1" rotation="-45 -50" tp_four></a-image>
        </a-entity>
        <a-grid src="pol.png"></a-grid>
        <a-entity gltf-model="#Chronicler" scale="1.25 1.25 1.25"></a-
entity>
        <a-light type="ambient" intensity="1"></a-light>
        <a-sky color="FFFFFF"></a-sky>
      </a-scene>
    </body>
    <script>
      AFRAME.registerComponent('log', {
        schema: {type: 'string'},
        init: function () {
          var stringToLog = this.data;
          this.el.addEventListener("click", () => {
            console.log(stringToLog)
          })
        }
      })
    </script>
  </html>
```

```

    })
  }
});
AFRAME.registerComponent('tp_one', {
  init: function () {
    this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
  x: 1.5,
  z: 1.5});

document.querySelector('#boxCamera').setAttribute('rotation', {
  y: 45});
    })
  }
});

AFRAME.registerComponent('tp_two', {
  init: function () {
    this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
  x: -1.5,
  z: 1.5});

document.querySelector('#boxCamera').setAttribute('rotation', {
  y: -45});
    })
  }
});

AFRAME.registerComponent('tp_three', {
  init: function () {
    this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
  x: -1.5,
  z: -1.5});

document.querySelector('#boxCamera').setAttribute('rotation', {
  y: -135});
    })
  }
});

AFRAME.registerComponent('tp_four', {
  init: function () {
    this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
  x: 1.5,
  z: -1.5});

document.querySelector('#boxCamera').setAttribute('rotation', {
  y: 135});
    })
  }
});

```

```
</script>
</html>
```

Листинг А.16 – Код VR-сцены «Лабиринт»

```
<html>
  <head>
    <script
src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
    <script src="../aframe-extras-master/dist/aframe-
extras.min.js"></script>
  </head>
  <body>
    <a-scene vr-mode-ui="enabled: true; enterARButton:
#enterARButton">
      <a id='enterARButton'></a>
      <a-assets>
        <a-asset-item id="Labyrinth"
src="../model/Labyrinth.glb"></a-asset-item>
        <a-asset-item id="OldKey" src="../model/OldKey.glb"></a-
asset-item>
        
        
        
        
        
      </a-assets>

      <a-entity id="boxCamera" position="0 0 0" rotation="0 0 0">
        <a-camera id="camera" look-controls wasd-controls-
enabled="false" position="0 1.6 0">
          <a-cursor raycaster="objects: .tp" fuse="true" position="0
0 -0.01" scale="0.0075 0.0075 0.0075"
            animation__fusing="property: scale; from:0.0075 0.0075
0.0075; to:0.001 0.001 0.001; startEvents: fusing; dur: 1500"
            animation__end="property: scale; to:0.0075 0.0075
0.0075; startEvents: animationcomplete__fusing; dur: 1"
            animation__mouseleave="property: scale; to:0.0075 0.0075
0.0075; startEvents: mouseleave; dur: 1">
          </a-cursor>
        </a-camera>
      </a-entity>
      <a-grid src="pol.png"></a-grid>

      <a-entity gltf-model="#Labyrinth" scale="1 1 1" position="0 0 0"
povorot2d></a-entity>
      <a-entity gltf-model="#OldKey" scale="0.8 0.8 0.8" position="0 3
0" rotation="0 0 0" povorot2d animation="property: rotation; to: 0 360
0; loop: true; dur: 5000; easing: linear"></a-entity>

      <a-light type="ambient" intensity="1"></a-light>
      <a-sky color="FFFFFF"></a-sky>
    </a-scene>
  </body>
  <script>
    AFRAME.registerComponent('log', {
      schema: {type: 'string'},
      init: function () {
```

```

        var stringToLog = this.data;
        this.el.addEventListener("click", () => {
            console.log(stringToLog)
        })
    }
});

AFRAME.registerComponent('tp', {
    init: function () {
        this.el.addEventListener("click", () => {
document.querySelector('#boxCamera').setAttribute('position', {
            x: this.el.getAttribute('position').x,
            z: this.el.getAttribute('position').z});
        })
    }
});

AFRAME.registerComponent('povorot2d', {
    init: function () {
        setInterval(() => {
            this.el.setAttribute('position', {
                x: 4 * Math.sin(-
document.querySelector('#camera').getAttribute('rotation').y * Math.PI
/180),
                y: this.el.getAttribute('position').y,
                z: -4 * Math.cos(-
document.querySelector('#camera').getAttribute('rotation').y * Math.PI
/180),
            });
        }, 1 );
    }
});

</script>
</html>

```

Сведения о самостоятельности выполнения работы

Выпускная квалификационная работа «Разработка модулей виртуальной и дополненной реальности для печатных изданий на примере романа «Ключ от лабиринта»» выполнена мной самостоятельно.

Используемые в работе материалы и концепции из публикуемой литературы и других источников имеют ссылки на них.

Электронный экземпляр выпускной квалификационной работы в формате pdf размещен на странице онлайн-курса «ГИА_09.03.03 Разработка Web и мультимедийных приложений (ОФО) (22-23)»

«__» _____ 2023г.

(подпись)

(И.О. Фамилия)