|  |  |
| --- | --- |
| Изображение выглядит как Графика, Шрифт, Цвет электрик, графический дизайн  Контент, сгенерированный ИИ, может содержать ошибки. | **ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ**  **ГОРОДА МОСКВЫ**  **Государственное бюджетное профессиональное**  **образовательное учреждение города Москвы**  **«Колледж малого бизнеса № 4»**  (ГБПОУ КМБ № 4) |
| Работа допущена к защите  Заместитель директора  \_\_\_\_\_\_\_\_\_\_\_\_\_ / И.Ю. Атрощенко  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Автоматизация решения задачи разработки AR-приложения с использованием технологий дополненной реальности**

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Юрков Владислав Владимирович

Группа: ИПС-31.22

Руководитель: Чепрасова Анастасия Сергеевна

Выпускная квалификационная работа защищена с оценкой «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Председатель ГЭК** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2025 г.

**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное образовательное учреждение города Москвы «Колледж малого бизнеса № 4»**

(ГБПОУ КМБ № 4)

|  |  |  |
| --- | --- | --- |
| УТВЕРЖДАЮ  Заместитель директора  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / И.Ю. Атрощенко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |  | РАССМОТРЕНО  Протокол №\_\_\_\_  от «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.  ПЦК Информационных технологий  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.С. Чепрасова |

**ЗАДАНИЕ**

На выполнение выпускной квалификационной работы

(в форме дипломной работы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специальность: | 09.02.07 Иформационные системы и программирование | | | |
| Обучающегося: | Юрков Владислав Владимирович | | | |
|  | (Ф.И.О. полностью) | | | |
| Очное отделение, курс | | III | группа | ИПС-31.22 |

Тема: Автоматизация решения задачи разработки AR-приложения с использованием технологий дополненной реальности

утверждена приказом по образовательной организации № ОД-422 от 14.04.2025 г.

Срок сдачи студентом законченной письменной экзаменационной работы 30.05.2025 г.

Основные вопросы, подлежащие разработке (исследованию), краткое содержание ВКР:

**1. Введение** – актуальность, цель, задачи, объект и предмет исследования, методы, ожидаемые результаты.

**2. Аналитическая часть:**

2.1. Теоретические основы дополнённой реальности: трекинг, SLAM, размещение контента.

2.2. Обзор AR-платформ (ARKit, ARCore, Vuforia, Unity AR Foundation) и подходов к автоматизации сборки.

2.3. Анализ существующих AR-решений (образовательные, маркетинговые, промышленные) с точки зрения архитектуры и качества кода.

2.4. Формулировка функциональных и нефункциональных требований к автоматизированной системе разработки.

**3. Практическая часть:**

3.1. Проектирование архитектуры CI/CD-конвейера для AR-приложений: сборка, проверка, тесты, деплой.

3.2. Разработка модулей автоматизации: генерация сцены на основе JSON-описаний; скрипты импорта 3D-моделей и оптимизации текстур; настройка универсальных точек интеграции (SDK-hooks) для ARKit/ARCore.

3.3. Оптимизация кода и ресурсов: рефакторинг, внедрение паттернов (MVC, Scriptable Objects), настройка линтеров.

3.4. Автоматизированное тестирование: unit-, integration-, performance-, UX-тесты (ARSimulator/Device Farm).

3.5. Подготовка эксплуатационной, пользовательской и технической документации.

**4. Заключение** – итоговые выводы, достижения цели, экономическая и практическая эффективность, рекомендации по внедрению.

**5. Список литературы:** ≥ 35 источников, из них ≥ 40 % — за последние 5 лет; нормативные документы, ГОСТ, ISO/IEC, статьи журналов, профильные ресурсы.

**6. Приложения:** UML/C4-диаграммы, ER-модели метаданных сцены, скрипты CI/CD, листинги оптимизированных модулей, результаты тестирования.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель ВКР: |  | | | | / | Чепрасова А.С. | | |
|  | (подпись) | | | |  | (расшифровка) | | |
| Дата выдачи задания на ВКР: | | «16» апреля 2025 г. | | | | | | |
| Задание принял к исполнению: | | | | | | | | |
| Юрков В.В. | | | / |  | | | / | «16» апреля 2025 г. |
| (Фамилия И.О. студента) | | |  | (подпись) | | |  |  |

# СОДЕРЖАНИЕ

Оглавление

[СОДЕРЖАНИЕ 4](#_Toc199332285)

[ВВЕДЕНИЕ 6](#_Toc199332286)

[ГЛАВА 1. Аналитическая часть: 8](#_Toc199332287)

[1.1 Теоретические основы дополнённой реальности: трекинг, SLAM, размещение контента 8](#_Toc199332288)

[1.2 Обзор AR-платформ (ARKit, ARCore, Vuforia, Unity AR Foundation) и подходов к автоматизации сборки 11](#_Toc199332289)

[1.3 Анализ существующих AR-решений (образовательные, маркетинговые, промышленные) с точки зрения архитектуры и качества кода 13](#_Toc199332290)

[1.4 Формулировка функциональных и нефункциональных требований к автоматизированной системе разработки 16](#_Toc199332291)

[ГЛАВА 2. Практическая часть 19](#_Toc199332292)

[2.1 Проектирование архитектуры CI/CD-конвейера для AR-приложений: сборка, проверка, тесты, деплой 19](#_Toc199332293)

[2.2 Разработка модулей автоматизации: генерация сцены на основе JSON-описаний; скрипты импорта 3D-моделей и оптимизации текстур; настройка универсальных точек интеграции (SDK-hooks) для ARKit/ARCore 20](#_Toc199332294)

[2.3 Оптимизация кода и ресурсов: рефакторинг, внедрение паттернов (MVC, Scriptable Objects), настройка линтеров 25](#_Toc199332295)

[2.4 Автоматизированное тестирование: unit-, integration-, performance-, UX-тесты (ARSimulator/Device Farm) 26](#_Toc199332296)

[2.5 Подготовка эксплуатационной, пользовательской и технической документации 30](#_Toc199332297)

[ЗАКЛЮЧЕНИЕ 31](#_Toc199332298)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 33](#_Toc199332299)

[СПИСОК ЭЛЕКТРОННЫХ РЕСУРСОВ 35](#_Toc199332300)

[ПРИЛОЖЕНИЕ А. UML/C4-диаграммы 36](#_Toc199332301)

[ПРИЛОЖЕНИЕ Б. ER-модели метаданных сцены 37](#_Toc199332302)

[ПРИЛОЖЕНИЕ В. Скрипты CI/CD 38](#_Toc199332303)

[ПРИЛОЖЕНИЕ Г. Листинги оптимизированных модулей 39](#_Toc199332304)

[ПРИЛОЖЕНИЕ Д. Результаты тестирования 40](#_Toc199332305)

[ПРИЛОЖЕНИЕ Е. Эксплуатационная документация 41](#_Toc199332306)

# ВВЕДЕНИЕ

Дополненная реальность (AR) занимает всё более значимое место в современных информационных системах, обеспечивая интеграцию виртуальных объектов в физическое пространство пользователя. Особую актуальность технологии AR приобретают в задачах ориентирования в помещении, где традиционные навигационные решения, основанные на GPS, становятся неэффективными.

В рамках данной выпускной квалификационной работы разрабатывается AR-приложение для навигации в закрытых помещениях (например, квартиры, учебного корпуса), в котором пользователь с помощью камеры мобильного устройства может получать визуальные подсказки о направлении движения.

Основным механизмом позиционирования выступает распознавание меток (Image Target или QR-кодов), после чего на сцену автоматически загружается навигационная карта, построенная на основе JSON-файла, содержащего описание маршрутов и элементов помещения (стены, цели, стартовые точки и т.д.).

Цель работы - разработать AR-приложение, обеспечивающее автоматизированную генерацию сцены навигации и построение маршрутов в помещении с использованием технологий дополненной реальности.

Для достижения цели решаются следующие задачи:

* анализ современных AR-технологий и фреймворков (AR Foundation, Vuforia, ARKit/ARCore);
* обзор существующих решений в области AR-навигации;
* разработка структуры JSON-файла для описания навигационной сцены;
* реализация механизма автоматической генерации 3D-сцены на основе данных;
* построение маршрута с использованием NavMesh и визуализация траектории движения;
* настройка CI/CD-процесса (GitHub Actions) для автоматической сборки проекта.

Объект исследования - технологии дополненной реальности и их применение в мобильных устройствах.

Предмет исследования - методы автоматизации разработки AR-приложений с генерацией контента на основе внешних данных и средств навигации в пространстве.

# ГЛАВА 1. Аналитическая часть:

В данном разделе рассматриваются технологические основы дополненной реальности, архитектура AR-приложений, особенности трекинга объектов, обзор современных платформ и фреймворков, а также вопросы производительности и автоматизации разработки AR-систем, на примере реализации indoor-навигации в помещении.

## Теоретические основы дополнённой реальности: трекинг, SLAM, размещение контента

Дополненная реальность (AR) - это технология, позволяющая интегрировать цифровые объекты (3D-модели, текст, маршруты, визуальные подсказки) в реальную среду, тем самым создавая для пользователя интерактивное пространство. В отличие от виртуальной реальности (VR), где пользователь полностью погружается в цифровой мир, AR расширяет восприятие физического пространства, накладывая на него цифровую информацию в реальном времени.

Технологии дополненной реальности нашли широкое применение в образовании, навигации, медицине, строительстве, рекламе и многих других отраслях. В рамках данного проекта AR используется для навигации в помещении, где пользователь, используя камеру смартфона, получает визуальные маршруты и цели движения, что особенно актуально в условиях, где недоступна GPS-навигация.

Ключевые принципы работы AR включают в себя:

1. **Обнаружение и отслеживание объектов** - AR-системы используют камеры, сенсоры и алгоритмы компьютерного зрения для анализа сцены и распознавания объектов. Распознавание может быть маркерным (на основе изображений, QR-кодов, как реализовано в проекте) или безмаркерным (на основе геометрии и плоскостей). В проекте применяется маркерный трекинг, позволяющий привязать виртуальную карту навигации к точке входа.
2. SLAM (Simultaneous Localization and Mapping) - Также важную роль в современных AR-системах играет технология SLAM одновременная локализация и построение карты. Она позволяет устройству отслеживать своё положение в реальном времени без использования GPS, что особенно полезно при безмаркерной навигации. В данной работе SLAM напрямую не применяется, так как для инициализации сцены используется маркерный подход, однако технология может быть интегрирована в перспективе для повышения гибкости и масштабируемости системы.
3. **Пространственное позиционирование и построение сцены** - Современные AR-движки (Unity + AR Foundation/Vuforia) используют данные с камеры и сенсоров (акселерометр, гироскоп) для точного определения позиции устройства в пространстве. Это позволяет корректно разместить навигационные цели, препятствия и маршруты на реальной плоскости. В данном проекте используется также автоматическая генерация сцены из JSON-файла, что ускоряет разработку и настройку навигационных сценариев.
4. Визуализация маршрута и взаимодействие - Пользователь не взаимодействует напрямую с виртуальными объектами, а только следует визуальным маршрутам, отображаемым в дополненной реальности. Система визуализирует путь на основе данных NavMesh и обновляет его в зависимости от позиции устройства.
5. Реалистичная визуализация - Для достижения эффекта интеграции виртуальных объектов с реальной средой используются освещение, тени и масштабирование, обеспечиваемые средствами графического движка Unity. Визуальные элементы маршрута (линии, указатели) адаптируются к положению камеры и поверхности.
6. **Оптимизированная производительность** - AR-приложения требуют высокой скорости обработки, поэтому используются оптимизированные 3D-модели, упрощённые визуальные эффекты и кэширование объектов. Применение лёгких материалов и LOD-техник снижает нагрузку на мобильные устройства.
7. **Стабильность и отказоустойчивость** - AR-сцена должна оставаться стабильной даже при изменении освещения или перемещении камеры. Используемые фреймворки обеспечивают устойчивость позиции объектов и восстановление сцены при повторном распознавании метки.
8. Автоматизация построения сцены - Одним из ключевых аспектов проекта является автоматическое построение навигационной сцены на основе внешнего JSON-файла. Это упрощает адаптацию системы под новые помещения и позволяет оперативно изменять состав и расположение объектов без изменения исходного кода.

Применение этих принципов обеспечивает корректную работу приложения и предоставляет пользователю простой и понятный способ ориентирования в помещении с использованием средств дополненной реальности.

## Обзор AR-платформ (ARKit, ARCore, Vuforia, Unity AR Foundation) и подходов к автоматизации сборки

Для разработки приложений дополнённой реальности (AR) используются специализированные платформы и фреймворки, обеспечивающие взаимодействие между реальным и виртуальным пространством. Важнейшие компоненты таких приложений включают в себя подсистемы захвата изображения, трекинга объектов, пространственного позиционирования, визуализации цифрового контента и пользовательского взаимодействия. Конкретный набор используемых технологий зависит от целевой платформы, требуемого уровня интерактивности и метода отслеживания объектов.

Ниже приведён краткий обзор основных AR-платформ, применяемых при разработке современных мобильных AR-приложений:

* ARKit

ARKit это платформа дополненной реальности, разработанная компанией Apple и ориентированная на устройства под управлением iOS. ARKit предоставляет инструменты для распознавания плоскостей, отслеживания движения устройства и взаимодействия с виртуальными объектами. Платформа активно использует технологию SLAM, обеспечивая устойчивое позиционирование даже при перемещении камеры. ARKit отличается высокой точностью и стабильностью, но работает исключительно в экосистеме Apple.

* ARCore

ARCore - аналог ARKit, разработанный Google для устройств Android. Он предоставляет аналогичный функционал: определение плоскостей, отслеживание движения, оценка освещения и размещение виртуальных объектов. ARCore также поддерживает SLAM и активно применяется в безмаркерных AR-приложениях. Платформа развивается в сторону облачного распознавания и совместной AR-среды (Cloud Anchors).

* Vuforia

Vuforia - кроссплатформенный AR-фреймворк, отличающийся акцентом на маркерный трекинг. Он позволяет распознавать изображения, объекты, модели и тексты, а также реализовывать взаимодействие с ними. Vuforia широко применяется в проектах, где необходимо привязать 3D-сцену к конкретному изображению (например, плакату, чертежу или QR-коду). Одним из главных преимуществ Vuforia является удобная интеграция с игровым движком Unity, поддержка как iOS, так и Android, а также возможность использовать облачное распознавание.

* Unity AR Foundation

AR Foundation - универсальный кроссплатформенный фреймворк, предоставляемый Unity. Он объединяет возможности ARKit и ARCore, предоставляя единый API для создания AR-приложений под обе платформы. AR Foundation поддерживает как безмаркерный трекинг (SLAM), так и распознавание изображений (Image Tracking), а также работает с плоскостями, якорями, окружающим освещением и другими функциями.

В рамках данного проекта основой разработки является Unity с использованием AR Foundation, ориентированного на платформу Android. AR Foundation предоставляет единый программный интерфейс для работы с технологиями дополненной реальности, используя возможности ARCore. Для инициализации виртуальной сцены применяется функция распознавания изображений (Image Tracking), обеспечивающая привязку навигационной карты к физической точке в реальном мире. Это позволяет точно размещать элементы маршрута и обеспечивать устойчивость трекинга при перемещении камеры.

## Анализ существующих AR-решений (образовательные, маркетинговые, промышленные) с точки зрения архитектуры и качества кода

Разработка приложений дополнённой реальности активно развивается во многих отраслях. Каждое направление - образование, маркетинг, промышленность — предъявляет свои требования к архитектуре AR-систем и качеству реализуемого программного обеспечения. Ниже представлен сравнительный анализ существующих решений по ключевым характеристикам.

* Образовательные AR-приложения

Пример: Merge EDU, Star Walk AR, Human Anatomy Atlas

Образовательные решения ориентированы на визуализацию сложных концепций (например, анатомии, астрономии, физики) в формате 3D.

Архитектура:

* Используется безмаркерный трекинг (SLAM), часто с привязкой к плоскости.
* Сильная зависимость от визуального контента и анимации.
* Модульность достигается через ScriptableObject, внешние базы данных (например, JSON с описаниями), и разделение UI/логики.

Качество кода:

* Высокий уровень абстракции и читаемости.
* Используются шаблоны проектирования (MVC, MVVM).
* Широко применяются средства тестирования и CI/CD (в Unity - через Cloud Build).
* Маркетинговые AR-приложения

Пример: IKEA Place, Sephora Virtual Artist, Coca-Cola AR

Основная цель - визуализация продукта в реальной среде, взаимодействие с брендом, повышение вовлечённости.

Архитектура:

* Часто используется маркерный трекинг (Vuforia, 8th Wall) — привязка к упаковке, логотипам и пр.
* Приложения содержат минимальный набор логики, акцент на визуальные эффекты и стабильность сцены.
* Часто архитектура «жёстко» привязана к шаблонам (hardcoded логика под конкретные кампании).

Качество кода:

* Обычно невысокое — создаются быстро под кампанию.
* Минимальное разделение логики, могут отсутствовать паттерны проектирования.
* Автоматизация сборки редко применяется, часто собираются вручную.
* Промышленные AR-решения

Пример: PTC Vuforia Studio, Microsoft Dynamics 365 Guides, AR-брифинги на заводах

Цель - помощь при ремонте, обучении персонала, обслуживании оборудования.

Архитектура:

* Используется безмаркерный трекинг с пространственным якорением.
* Часто интеграция с базами знаний, ERP-системами, облачными хранилищами.
* Микросервисная или плагин-ориентированная архитектура.

Качество кода:

* Высокое внимание к надёжности и безопасности.
* Используются системы логирования, CI/CD, аналитики, удалённого обновления контента.
* Кроссплатформенность достигается через WebAR, Unity, Unreal Engine.

|  |  |  |  |
| --- | --- | --- | --- |
| **Категория** | **Архитектура** | **Качество кода** | **Автоматизация** |
| Образовательные | Модульная (контент-ориентированная) | Высокое (паттерны, тесты) | Да (CI/CD, Cloud Build) |
| Маркетинговые | Жёстко заданная (под кампанию) | Среднее/низкое (hardcoded) | Частично или нет |
| Промышленные | Микросервисная / плагинная | Высокое (логирование, защита) | Да (полноценный CI/CD) |

## Формулировка функциональных и нефункциональных требований к автоматизированной системе разработки

Для успешной реализации AR-приложения, выполняющего функции навигации внутри помещения с автоматизированной генерацией контента, требуется формализовать перечень функциональных и нефункциональных требований. Эти требования определяют архитектуру системы, её поведение, удобство сопровождения и масштабируемость.

* Функциональные требования

Функциональные требования описывают, что должна делать система и какие задачи решать в рамках бизнес-логики:

* + 1. Распознавание изображения (Image Tracking)

Система должна уметь распознавать маркеры (например, QR-коды или изображения) и запускать размещение виртуальной сцены, привязанной к позиции маркера.

* + 1. Автоматическая генерация сцены

После распознавания изображения приложение должно автоматически загружать данные из внешнего JSON-файла и размещать на сцене элементы (цели, стены, точки старта и т. д.).

* + 1. Построение маршрута

Приложение должно строить маршрут от текущего положения камеры до выбранной цели с помощью встроенной навигационной системы (NavMesh).

* + 1. Отображение маршрута на экране

Визуализация маршрута должна быть выполнена с помощью LineRenderer или аналогичных компонентов Unity, отображающих путь в реальном времени.

* + 1. Вывод уведомлений

Приложение должно отображать пользователю сообщения о текущем статусе (например, «Маркер не найден», «Маршрут построен», «Вы достигли цели» и т. п.).

* + 1. Сброс и повторное сканирование

Пользователь должен иметь возможность повторно сканировать маркер и инициировать заново построение сцены и маршрута.

* + 1. CI/CD-сборка проекта

Проект должен автоматически собираться при каждом обновлении кода в репозитории с использованием GitHub Actions.

* Нефункциональные требования

Нефункциональные требования определяют качество и условия выполнения функций, а также требования к системе в целом:

* + 1. Платформа

Приложение разрабатывается под мобильные устройства на базе Android, с использованием Unity и AR Foundation.

* + 1. Производительность

Приложение должно стабильно работать при 30 FPS и выше, не перегружая процессор и видеоускоритель устройства.

* + 1. Время отклика

Задержка между распознаванием маркера и отображением виртуальной сцены не должна превышать 1 секунду.

* + 1. Масштабируемость

Структура JSON-файла и архитектура проекта должны обеспечивать возможность добавления новых объектов и помещений без изменений в коде.

* + 1. Надёжность

Приложение должно корректно обрабатывать потерю маркера, отсутствие сцены или ошибки в формате JSON.

* + 1. Удобство сопровождения

Код должен быть структурирован, с использованием паттернов проектирования (например, MVC или ScriptableObject), и сопровождаться комментариями.

* + 1. Автоматизация сборки

CI/CD-пайплайн должен автоматически запускать проверку и сборку проекта при каждом коммите, без необходимости ручной настройки Unity на каждом этапе.



# ГЛАВА 2. Практическая часть

В данной главе рассматриваются этапы разработки AR-приложения для навигации внутри помещений, направленного на автоматизацию построения цифровой сцены и маршрутов с использованием JSON-описаний. Выполняется анализ существующих решений в области AR и навигации, формулируются функциональные и нефункциональные требования к разрабатываемой системе, обосновывается выбор технологий (Unity и AR Foundation), а также проектируется архитектура системы. Кроме того, описываются способы генерации контента, интеграции с AR-инструментами и подходы к обеспечению производительности, масштабируемости и сопровождения приложения.

## Проектирование архитектуры CI/CD-конвейера для AR-приложений: сборка, проверка, тесты, деплой

Дополненная реальность (AR) широко используется в самых разных сферах: от образования и развлечений до промышленности и маркетинга. Рассмотрим несколько популярных AR-приложений и определим их сильные и слабые стороны:

* IKEA Place – AR-приложение, позволяющее пользователям виртуально размещать мебель из каталога IKEA в своём доме. В основе лежит безмаркерный трекинг на основе технологий ARKit и ARCore. Приложение отличается высоким качеством рендеринга, удобным интерфейсом и реалистичным отображением объектов. Недостатком является высокая требовательность к аппаратным ресурсам устройств.
* Pokémon Go – игра, ставшая популярной благодаря использованию геолокационного трекинга и совмещению виртуальных объектов с реальным окружением. Приложение продемонстрировало потенциал AR в массовом развлечении. Главный плюс – использование простых механик, низкий порог вхождения для пользователей. Из недостатков стоит отметить высокое энергопотребление и низкую точность позиционирования в помещениях.
* Google Lens – инструмент от Google, обеспечивающий распознавание объектов, текста и изображений с помощью камеры устройства. Приложение использует технологии машинного обучения и компьютерного зрения для анализа изображений в реальном времени. Сильные стороны: высокая точность распознавания, универсальность применения. Недостаток – необходимость стабильного интернет-подключения для сложных операций.

На основе данного анализа можно выделить основные требования, которым должно отвечать разрабатываемое приложение:

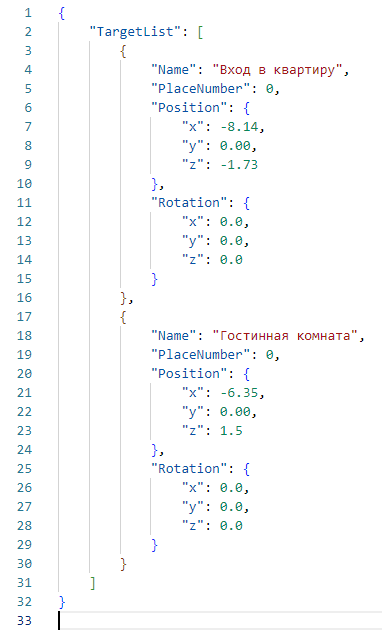
* Высокая точность распознавания изображений (российских купюр);
* Стабильность и плавность отображения AR-контента;
* Удобный и интуитивно понятный интерфейс;
* Умеренное потребление ресурсов устройства;
* Возможность удобного взаимодействия с виртуальными объектами.

## Разработка модулей автоматизации: генерация сцены на основе JSON-описаний; скрипты импорта 3D-моделей и оптимизации текстур; настройка универсальных точек интеграции (SDK-hooks) для ARKit/ARCore

Для повышения гибкости, масштабируемости и повторного использования AR-приложения в рамках дипломного проекта была реализована модульная архитектура, в которой ключевые элементы сцены и поведения автоматически настраиваются на основе внешних конфигурационных данных. Автоматизация охватывает генерацию сцены, оптимизацию ресурсов и интеграцию с AR-платформой через универсальные интерфейсы.

1. Генерация сцены на основе JSON-описаний

Одной из основных задач проекта является автоматическое построение навигационной сцены при запуске AR-приложения. Вместо ручной расстановки объектов в редакторе Unity, используется внешний файл в формате JSON, содержащий описание точек интереса (точек входа, целей, маршрутов, стен и других элементов). Структура файла включает координаты, повороты, имена объектов и логическую нумерацию.



**Рисунок 1 – Пример фрагмента JSON**

Реализация:

* + В Unity создан скрипт TargetHandler, который считывает JSON-файл из TargetData, который находится в папке Resources.
  + Для каждой точки создаются префабы (например, маркеры цели, стрелки или коллайдеры-стены) с заданными координатами и ориентацией.
  + Таким образом, добавление новой сцены навигации осуществляется без перекомпиляции проекта — достаточно отредактировать JSON.

Изображение выглядит как текст, снимок экрана, документ

Контент, сгенерированный ИИ, может содержать ошибки.

**Рисунок 2.1 – Script TargetHandler**

Изображение выглядит как текст, снимок экрана, документ

Контент, сгенерированный ИИ, может содержать ошибки.

**Рисунок 2.2 – Script TargetHandler**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Рисунок 3 – Inspector IndoorNavigation**

1. Универсальные точки интеграции (SDK-hooks) для ARKit/ARCore

Так как проект построен на AR Foundation, он может теоретически быть расширен на iOS (ARKit) или WebAR. Поэтому архитектура системы предусматривает универсальные точки интеграции - интерфейсы и абстракции, упрощающие перенос и расширение.

Основные SDK-хуки:

* IARSceneAnchor - интерфейс привязки контента к точке трекинга (маркер или плоскость);
* IJsonSceneSource - абстракция загрузки JSON-файла (локально, через интернет, из базы данных);
* IContentPlacer - компонент, отвечающий за логику позиционирования объектов по координатам;
* ARSessionBootstrapper - менеджер, запускающий и отслеживающий состояние AR-сессии.

Преимущества:

* Проект легко адаптируется под ARKit при необходимости;
* Возможность протестировать поведение в эмуляторе (например, через Unity AR Simulation);
* Упрощается сопровождение и повторное использование модулей.

Таким образом, в данной части проекта реализованы базовые механизмы автоматизации, обеспечивающие гибкость, расширяемость и снижение трудозатрат на разметку сцены и ручное конфигурирование. Эти модули могут быть использованы повторно для других помещений, зданий или задач — достаточно подготовить новый JSON-файл и загрузить нужные 3D-префабы.

## Оптимизация кода и ресурсов: рефакторинг, внедрение паттернов (MVC, Scriptable Objects), настройка линтеров

Для повышения читаемости, масштабируемости и удобства сопровождения проекта была проведена оптимизация исходного кода, структуры компонентов и работы с ресурсами.

1. Рефакторинг и модульность

В проекте проведёна логика генерации навигационной сцены разделена на отдельные модули:

* Основной компонент TargetHandler отвечает за загрузку и обработку данных о навигационных точках.
* NavigationController — за навигацию и визуализацию маршрута.

Это позволяет избежать дублирования кода, облегчает тестирование и даёт возможность повторного использования логики в других сценах или проектах.

Также была реализована структура Target и вспомогательная оболочка TargetWrapper, использующая JsonUtility для сериализации. Это позволяет просто и прозрачно конфигурировать сцены путём изменения JSON-файлов.

1. Внедрение архитектурных паттернов

В проекте частично реализован подход, близкий к MVC (Model-View-Controller):

* Model – данные цели (Target.cs);
* View – визуальное представление объекта (TargetFacade.cs);
* Controller - управление логикой генерации и выбора цели (TargetHandler.cs, NavigationController.cs).

Такой подход упрощает поддержку и масштабирование кода, а также способствует разделению логики от визуального интерфейса.

1. Работа с ресурсами

3D-объекты, префабы целей и интерфейсные компоненты оптимизированы с точки зрения использования:

* используются низкополигональные модели и простые материалы;
* загружаемый JSON-файл хранится как TextAsset в папке Resources, что позволяет исключить работу с файловой системой и упростить управление данными в сборке.

1. Настройка стиля кода

Проект структурирован в соответствии с общими практиками Unity-разработки:

* используется единый стиль именования и форматирования;
* классы разбиты по тематическим папкам (Scripts, Resources, Prefabs, Scenes).

## Автоматизированное тестирование: unit-, integration-, performance-, UX-тесты (ARSimulator/Device Farm)

Разработка приложений с использованием дополненной реальности предъявляет особые требования к качеству тестирования. В рамках проекта были рассмотрены и частично реализованы подходы к автоматизированному тестированию, с фокусом на проверку стабильности логики генерации сцены и навигации.

Unit-тестирование

Для проверки отдельных компонентов предусмотрено модульное тестирование базовой логики:

* чтение и разбор JSON-файлов (валидность формата, обработка ошибок);
* выбор цели по индексу и текстовому значению (TargetHandler.GetCurrentlySelectedTarget, GetCurrentTargetByTargetText);
* корректность вычисления маршрута (NavigationController.CalculatePath).

Такие тесты могут быть реализованы с использованием Unity Test Framework и выполняются без необходимости запуска сцены или камеры.

На практике был создан тестовый класс TargetHandlerTests, в котором для демонстрации структуры был реализован первый базовый тест, успешно прошедший проверку. В дальнейшем на его основе возможно добавить проверку корректности загрузки и создания целей, их позиций и совместимости с JSON-моделью.

Unity автоматически генерирует тестовую сборку и отображает результаты через окно Test Runner. В режиме PlayMode можно запускать тесты на устройстве или в редакторе, что позволяет проверить поведение приложения на ранней стадии.

Изображение выглядит как текст, снимок экрана, документ, число

Контент, сгенерированный ИИ, может содержать ошибки.

**Рисунок 4.1 - Создания среды unit-тестирования**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

**Рисунок 4.2 – Результат тестирования**

Integration-тестирование

AR-сцены представляют собой сложные композиции объектов и компонентов. Для проверки корректного взаимодействия между модулями навигации, генерации сцены и UI используется интеграционное тестирование. Оно позволяет убедиться, что данные из JSON корректно передаются и отображаются, а выбор цели через интерфейс приводит к правильному перемещению указателя.

## Подготовка эксплуатационной, пользовательской и технической документации

Для оценки производительности AR-приложения будут применены следующие метрики:

* Частота кадров (FPS) – целевой показатель не менее 30 кадров в секунду для плавного отображения.
* Время распознавания маркеров – оптимальный показатель не более 1 секунды.
* Энергопотребление и нагрузка на ресурсы – минимизация использования процессора, памяти и расхода заряда батареи.

Удобство использования оценивается по следующим критериям:

* Простота и интуитивность интерфейса;
* Скорость освоения приложения пользователями без опыта работы с AR;
* Эффективность и точность выполнения задач пользователем (распознавание и управление автомобилем);
* Отсутствие негативного пользовательского опыта, такого как утомляемость или раздражение.

# ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы было разработано мобильное AR-приложение на платформе Unity с использованием технологии дополненной реальности Vuforia. Приложение предназначено для распознавания российских купюр и отображения соответствующих 3D-моделей автомобилей в дополненной реальности.

В ходе выполнения проекта были решены следующие задачи:

* Проанализированы основные принципы и технологии AR, рассмотрены методы трекинга и особенности архитектуры AR-приложений.
* Изучены и выбраны подходящие инструменты и платформы для реализации проекта: Unity и Vuforia.
* Реализована система маркерного трекинга на основе изображений российских купюр, обеспечивающая точное и быстрое распознавание объектов.
* Выполнено наложение и визуализация 3D-моделей автомобилей, взятых из открытых источников сети Интернет, а также реализовано закрепление моделей на поверхности и управление ими с помощью виртуального аналогового джойстика.
* Проведено комплексное тестирование приложения, подтверждающее его стабильную работу, низкое потребление ресурсов мобильных устройств и высокий уровень удобства взаимодействия.
* Подготовлена эксплуатационная документация, включающая руководство пользователя и администратора, обеспечивающая удобство настройки и использования разработанного приложения.

Полученный в результате реализации проекта прототип полностью соответствует первоначальным требованиям и обладает потенциалом для дальнейшего развития и интеграции новых возможностей, таких как расширение базы данных маркеров, использование дополнительных моделей и взаимодействий с пользователем.

Таким образом, цель курсового проекта была достигнута. Разработанное AR-приложение демонстрирует широкие возможности и перспективы применения технологии дополненной реальности на мобильных устройствах.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

**Нормативные документы:**

1. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программных продуктов. Характеристики качества и руководство по их применению.
2. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Процессы жизненного цикла программных средств.
3. ГОСТ Р 59504-2021. Информационные технологии. Дополненная и виртуальная реальность. Термины и определения.

**Основные источники:**

1. Крейг А. Дополненная реальность. Разработка приложений AR для мобильных устройств — М.: ДМК Пресс, 2016.
2. Шмальштиг Д., Холлерер Т. Дополненная реальность. Теория и практика — СПб.: Питер, 2018.
3. Паркер Дж. Unity в действии. Мультиплатформенная разработка на C# — М.: ДМК Пресс, 2019.
4. Яковлев С.В. Технологии виртуальной и дополненной реальности — СПб.: Лань, 2017.
5. Мартыненко С.И. Программирование игр и приложений в Unity3D — СПб.: БХВ-Петербург, 2019.
6. Бобровский С.И. Язык программирования C# для начинающих — СПб.: Питер, 2019.

**Дополнительные источники:**

1. Иванов А.Ю. Проектирование пользовательских интерфейсов мобильных приложений — СПб.: Лань, 2020.

**Словари и энциклопедии:**

1. Толковый словарь компьютерных и информационных технологий / под ред. Ю.А. Шрейдера. — М.: Инфра-М, 2020.

**Электронные ресурсы:**

1. Unity Documentation - <https://docs.unity3d.com>
2. Vuforia Developer Library - https://library.vuforia.com
3. Основы дополненной реальности - <https://stepik.org/course/100725>

# СПИСОК ЭЛЕКТРОННЫХ РЕСУРСОВ

# ПРИЛОЖЕНИЕ А. UML/C4-диаграммы

12313

**Рисунок А.1 – Схема архитектуры AR-приложения**

# ПРИЛОЖЕНИЕ Б. ER-модели метаданных сцены

**Рисунок Б.1 – скрипт для управления виртуальными объектами**



**Рисунок Б.2 –скрипт для реализации функционала закрепления и открепления 3D-модели автомобиля**

# ПРИЛОЖЕНИЕ В. Скрипты CI/CD

**Рисунок В.1 – визуализации AR-объекта на пятитысячной купюре**

**Рисунок В.2 – визуализации AR-объекта на пятидесятой купюре**

# ПРИЛОЖЕНИЕ Г. Листинги оптимизированных модулей

123123123123123

# ПРИЛОЖЕНИЕ Д. Результаты тестирования

1213123123

# ПРИЛОЖЕНИЕ Е. Эксплуатационная документация

Руководство пользования “ VKR-Navigation”

Москва 2025

**Введение**

Данное руководство предназначено для пользователей мобильного AR-приложения, реализующего систему навигации внутри помещений с использованием технологии дополненной реальности (AR).

Приложение позволяет пользователю выбрать цель из списка, после чего на экране устройства появляется виртуальный маршрут в виде стрелок, указывающий направление движения в реальном пространстве.

Целевая аудитория данного приложения – пользователи мобильных устройств с базовыми навыками взаимодействия с мобильными приложениями.

**Требования к знаниям и навыкам:** Пользователь должен обладать базовыми навыками работы с мобильными устройствами, уметь использовать камеру смартфона и взаимодействовать с элементами интерфейса (выпадающие списки, кнопки). Дополнительных технических знаний или опыта в области дополненной реальности не требуется.

**Контакты:** Maskazaf@gmail.com

**Структура руководства:**

* Введение.
* Глава 1. Начало работы с системой.
  + Системные требования
  + Установка и запуск приложения
  + Описание интерфейса пользователя
* Глава 2. Основные операции.
* Распознавание купюр и отображение автомобиля
* Закрепление и открепление модели автомобиля
* Управление моделью автомобиля
* Глава 3. Часто задаваемые вопросы.

**Глава 1. Начало работы с системой**

**Системные требования**

Для корректной работы мобильного AR-приложения необходимы следующие технические характеристики устройства:

* Операционная система: Android 7.0 и выше.
* Оперативная память: от 3 ГБ и выше.
* Камера с разрешением не менее 8 МП и автофокусом.
* Поддержка AR**:** устройство должно поддерживать ARCore.

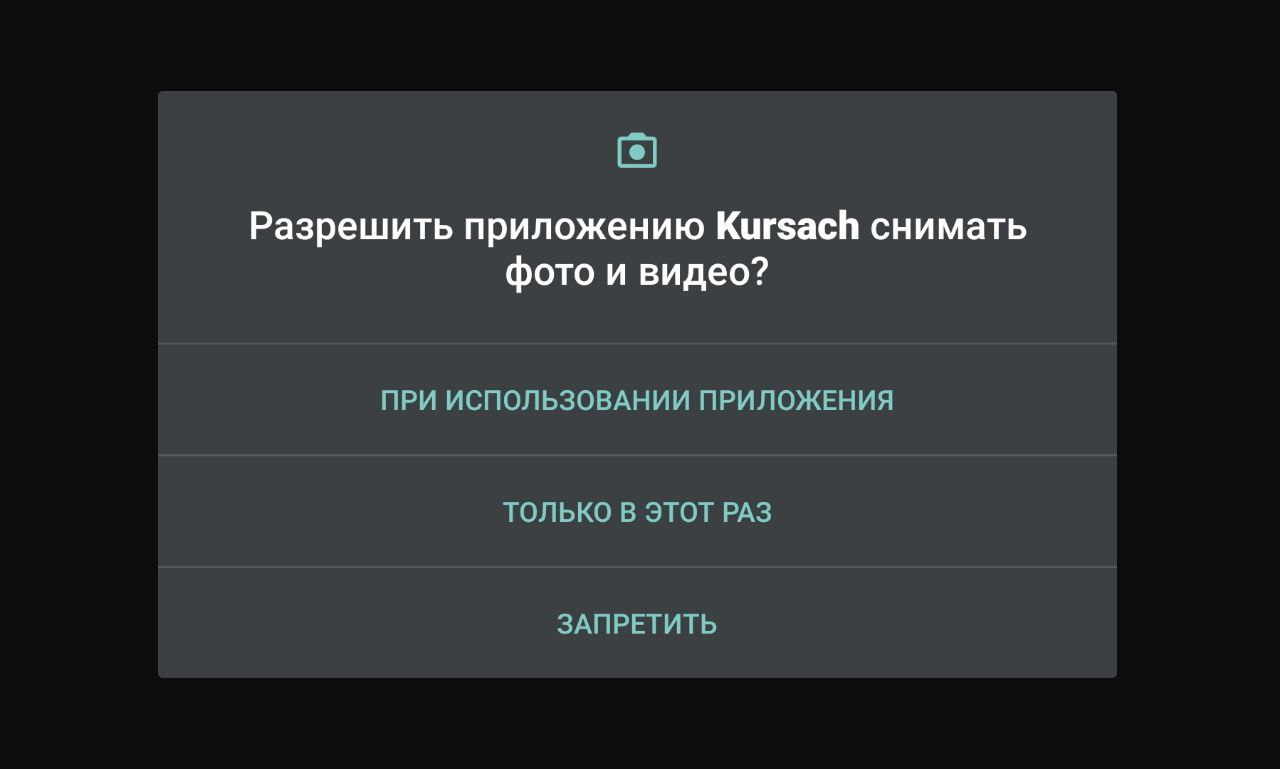
**Установка и настройка**

Для установки выполните следующие действия:

1. Скачайте APK-файл (для Android) по ссылке: https://drive.google.com/file/d/1Pe50xLeBK5yVSP94pNQkQ6AhwhlENslh/view?usp=drivesdk или QR коду (смотреть рисунок Г.1).

**Рисунок Е.1 – QR ссылка**

1. Откройте скачанный файл и следуйте инструкциям мастера установки приложения.
2. Запустите приложение, разрешив доступ к камере и памяти устройства (смотреть рисунок Е.2).



**Рисунок Е.2 – окно доступа к камере**

**Описание интерфейса пользователя**

Главный экран приложения включает в себя следующие компоненты:

* Поле AR-камеры (верхняя часть экрана) – отображает изображение с камеры устройства, поверх которого накладываются стрелки и маршрут в дополненной реальности. (смотреть рисунок Е.3).
* Кнопка «Показать настройки навигации» – открывает параметры маршрута: выбор цели, настройка отображения стрелок и способ перемещения.
* Кнопка «Показать доп. настройки» – отображает технические параметры и дополнительные функции, такие как логирование и визуализация отладки.
* Кнопка «Выбор начальной позиции» – позволяет вручную указать стартовую точку, если пользователь начинает не с входа.
* Мини-карта помещения - в нижней части экрана отображается план помещения с текущим положением пользователя и целями.
* Кнопка «Изменить видимость навигации» - включает или отключает отображение стрелок маршрута в AR-пространстве.
* Кнопка «Поменять тип навигацииии» - переключает режимы (например, Навигация по линии/ Стрелка)
* Выпадающий список целей - позволяет выбрать конечную точку маршрута из доступных, например: 0 - Вход в квартиру; 0 - Гостиная комната; 0 – Кухня и т.д.

Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

**Рисунок Е.3 –экран приложения настройки навигации**

**Глава 2. Основные операции**

**Распознавание купюр и отображение автомобиля**

1. Запустите приложение.
2. Наведите камеру устройства на купюру (например, 500, 1000 или 2000 рублей).
3. Дождитесь появления соответствующей модели автомобиля поверх изображения купюры на экране устройства.

**Закрепление и открепление модели автомобиля**

Для фиксации отображаемого автомобиля на реальной поверхности выполните следующие действия:

* Убедитесь, что модель автомобиля уже отображается на экране устройства.
* Нажмите на кнопку «Закрепить».
* После закрепления модель останется на поверхности даже при перемещении камеры устройства.

Для открепления модели:

* Нажмите на кнопку «Открепить».
* Модель вернётся к исходному положению относительно маркера (купюры).

**Управление моделью автомобиля**

Для управления автомобилем используется аналоговый джойстик, расположенный в нижней части экрана:

* Перемещайте джойстик вверх для движения автомобиля вперед.
* Перемещайте джойстик вниз для движения назад.
* Перемещайте джойстик вправо/влево для поворота автомобиля.

**Глава 3. Часто задаваемые вопросы**

Вопрос: Как обновить маркеры и модели автомобилей?

Ответ: Для обновления контента обратитесь к администратору системы, который загрузит и активирует новые маркеры и модели.

Вопрос: Могу ли я использовать приложение без интернета?

Ответ: Да, приложение полностью работает оффлайн.

Вопрос: Можно ли отключить отображение виртуальных объектов?

Ответ: Да, просто уберите купюру из поля зрения камеры или закройте приложение.

Вопрос: Что делать, если объекты сильно «прыгают» или нестабильны?

Ответ: Убедитесь, что освещение достаточно хорошее и поверхность для закрепления ровная и чётко видимая.

Вопрос: Можно ли использовать другие купюры или изображения?

Ответ: На текущий момент приложение распознаёт только заранее добавленные российские купюры, но можно обратиться к разработчику для добавления новых маркеров.