СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| Введение | 8 |
| 1 Общая часть | 11 |
| 1.1 Постановка задачи | 11 |
| 1.1.1 Назначение задачи | 11 |
| 1.1.2 Требования к программе | 11 |
| 1.1.3 Требования к составу и параметрам технических средств | 12 |
| 1.2 Описание языка программирования | 14 |
| 1.2.1 Выбор языка программирования | 14 |
| 1.2.2 Элементы языка программирования | 22 |
| 2 Специальная часть | 24 |
| 2.1 Диаграмма предметной области | 24 |
| 2.2 Описание алгоритма | 25 |
| 2.2.1 Алгоритм работы системы | 25 |
| 2.2.2 Алгоритм работы подпрограмм | 25 |
| 2.3 Текст программы | 26 |
| 2.4 Инструкция работы с программой | 26 |
| 2.4.1 Общие сведения | 26 |
| 2.4.2 Вызов и загрузка | 27 |
| 2.4.3 Входные и выходные данные | 29 |
| 2.4.4 Сообщения программы | 30 |
| 2.5 Описание процесса отладки программы | 30 |
| 2.5.1 Методы отладки | 30 |
| 2.5.2 Тестирование и отладка | 31 |
| 2.5.3 Контрольный пример | 32 |

|  |  |
| --- | --- |
| 3 Экономическая часть | 35 |
| Заключение | 36 |
| Литература | 37 |

ВВЕДЕНИЕ

Современные технологии виртуальной реальности открывают новые возможности в сфере образования, позволяя создавать интерактивные и иммерсивные среды для изучения различных дисциплин. Особую значимость такие решения приобретают в контексте исторического образования, где визуализация событий прошлого способствует более глубокому пониманию материала. Разработка виртуального музея, посвященного Великой Отечественной войне, представляет собой актуальный проект, направленный на сохранение исторической памяти и повышение эффективности образовательного процесса.

Необходимость создания виртуального музея обусловлена рядом факторов, связанных как с образовательными потребностями, так и с технологическим прогрессом:

* 1. Сохранение исторической памяти. Современное поколение студентов все реже сталкивается с живыми свидетельствами событий Великой Отечественной войны. Виртуальный музей позволяет воссоздать ключевые моменты истории в доступной и наглядной форме.
  2. Интерактивность обучения. Традиционные методы преподавания истории часто ограничиваются лекциями и учебниками. VR–технологии дают возможность погрузиться в исторический контекст, взаимодействовать с экспонатами и даже моделировать сценарии событий.
  3. Доступность образовательных ресурсов. Не все учебные заведения имеют возможность организовать посещение реальных музеев или мемориальных комплексов. Виртуальный музей устраняет географические и финансовые барьеры.
  4. Развитие цифровых компетенций. Использование VR в обучении способствует формированию навыков работы с современными технологиями, что особенно важно для студентов IT–специальностей.

Основной целью проекта является создание интерактивного VR–приложения, которое позволит пользователям изучать историю Великой Отечественной войны через виртуальную экспозицию.

Для достижения этой цели необходимо решить следующие задачи:

* разработка концепции виртуального музея, включая структуру экспозиции, тематические зоны и способы взаимодействия с контентом;
* создание 3D–моделей исторических объектов, таких как военная техника, оружие, документы и реконструкции ключевых событий;
* реализация механизмов взаимодействия в VR–среде, включая навигацию, управление интерфейсом и работу с интерактивными элементами;
* интеграция мультимедийного контента, такого как аудиогиды, архивные фотографии и видеохроника, для расширения информационной составляющей;
* обеспечение производительности и стабильности приложения, чтобы оно могло работать на стандартном VR–оборудовании без чрезмерных требований к hardware.

Разработка виртуального музея имеет как теоретическую, так и прикладную ценность:

1. Научная новизна проекта заключается в адаптации VR–технологий для образовательных целей с учетом специфики исторического контента. В отличие от коммерческих VR–приложений, данный проект ориентирован на учебный процесс, что требует особого подхода к подаче информации и методике взаимодействия.
2. Практическая значимость выражается в возможности внедрения приложения в образовательные программы колледжа. Виртуальный музей может использоваться не только на занятиях по истории, но и в рамках междисциплинарных проектов, связанных с программированием, 3D–моделированием и дизайном.

Проект имеет важное социальное значение, так как способствует патриотическому воспитанию молодежи:

1. Формирование исторического сознания. Виртуальный музей помогает студентам осознать масштаб и трагизм событий войны, что трудно передать через сухие факты учебников.
2. Связь поколений. Возможность интерактивного изучения истории делает ее более близкой и понятной для современной молодежи, привыкшей к цифровым технологиям.
3. Популяризация истории. Использование современных технологий повышает интерес к изучению прошлого, особенно среди студентов технических специальностей.

Создание виртуального музея – это только первый этап в развитии образовательных VR–решений. В дальнейшем проект может быть расширен за счет следующих направлений:

* добавление новых экспозиций, посвященных другим историческим периодам или событиям регионального значения;
* разработка мультиплеерного режима, позволяющего студентам совместно посещать музей и участвовать в виртуальных экскурсиях;
* интеграция с мобильными платформами, что сделает приложение доступным для более широкой аудитории;
* использование технологий искусственного интеллекта для создания динамических экскурсий, адаптирующихся под уровень знаний пользователя.

Разработка виртуального музея Великой Отечественной войны представляет собой комплексный проект, объединяющий образовательные, технологические и социальные аспекты. Применение VR–технологий позволяет создать уникальную интерактивную среду, которая не только обогащает учебный процесс, но и способствует сохранению исторической памяти. Реализация данного проекта в стенах Владимирского политехнического колледжа станет важным шагом в развитии цифровых образовательных ресурсов и откроет новые возможности для изучения истории с помощью современных технологий.

1 ОБЩАЯ ЧАСТЬ

1.1 Постановка задачи

1.1.1 Назначение задачи

Разработка приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» направлена на создание интерактивной образовательной среды, позволяющей изучать исторические события в формате VR. Приложение предназначено для использования в учебном процессе ГАПОУ ВО «Владимирский политехнический колледж» с целью повышения наглядности и вовлеченности в изучение темы.

Основные функции приложения включают:

* воссоздание виртуальных экспозиций, посвященных ключевым событиям Великой Отечественной войны;
* обеспечение интерактивного взаимодействия с объектами музея;
* поддержка навигации в VR–пространстве с использованием контроллеров SteamVR;
* интеграция с базой данных для хранения и управления контентом.

Приложение разрабатывается с учетом всех требований современного образовательного процесса, обеспечивая доступность, удобство использования и техническую совместимость с самым новейшим и популярным VR – оборудованием.

1.1.2 Требования к программе

Функциональные требования:

* разработка VR–приложения с использованием Unity и SteamVR;
* создание интерактивной 3D–среды виртуального музея;
* реализация навигации по экспозициям с помощью VR–контроллеров;
* интеграция базы данных для хранения информации об экспонатах;
* обеспечение загрузки и отображения мультимедийного контента;
* реализация системы управления через интерфейс администратора.

Технические требования:

* поддержка VR–гарнитур, совместимых с SteamVR;
* использование C# для разработки логики приложения;
* применение WindowsForms для административного интерфейса;
* хранение данных в СУБД MySQL;
* совместимость с операционной системой Windows 10 и новее.

Интерфейсные требования:

* интуитивно понятное меню для пользователя;
* голосовые подсказки и текстовые инструкции;
* адаптация управления под стандартные схемы VR–взаимодействия.

Надежность и безопасность:

* обеспечение стабильной работы приложения без критических ошибок;
* защита базы данных от несанкционированного доступа;
* регулярное создание резервных копий данных.

Производительность:

* оптимизация 3D–моделей для плавного рендеринга в VR;
* минимальная задержка при загрузке контента;
* поддержка частоты кадров не ниже 90 FPS для комфортного использования.

Условия эксплуатации:

* наличие VR–оборудования и ПК с рекомендуемыми характеристиками;
* установленный SteamVR и актуальные драйверы;
* доступ к локальной сети или интернету для работы с базой данных.

Соблюдение перечисленных требований гарантирует создание функционального, VR–приложения, соответствующего поставленным задачам.

1.1.3 Требования к составу и параметрам технических средств

Для успешной разработки и функционирования приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» необходимо обеспечить соответствие технических средств следующим требованиям:

Таблица 1 – Минимальные требования для разработки

|  |  |
| --- | --- |
| Параметр | Требование |
| Операционная система | Windows 10 или новее |
| Процессор | Intel Core i5–4590 / AMD FX 8350 или эквивалент |
| Оперативная память | 8 ГБ и более |
| Графический адаптер | NVIDIA GTX 970 / AMD Radeon R9 290 и выше, с поддержкой DirectX 11 |
| Жесткий диск | SSD–накопитель объемом не менее 20 ГБ для установки Unity, SteamVR и сопутствующих инструментов |
| VR – гарнитура | HTC Vive, Oculus Rift, Valve Index или аналоги с контроллерами |
| Монитор | С разрешением Full HD 1920×1080 и выше |
| Программное обеспечение | SteamVR |
| Пространство | Пространство для Room–Scale VR |
| Доступ в интернет | Для загрузки обновлений и работы с облачными данными |
| Веб – сервер | Apache, Nginx с PHP 7.4+ и MySQL 5.7+/MariaDB |

Таблица 2 – Рекомендуемые системные требования для правильной и комфортной работы

|  |  |
| --- | --- |
| Параметр | Требование |
| Операционная система | Windows 10 или новее |
| Процессор | Intel Core i7–9700K / AMD Ryzen 7 3700X и выше |
| Оперативная память | 16 ГБ и более |
| Графический адаптер | NVIDIA RTX 2060 / AMD RX 5700 и выше |
| Жесткий диск | NVMe SSD объемом 500 ГБ и более |
| VR – гарнитура | HTC Vive, Oculus Rift, Valve Index или аналоги с контроллерами |
| Монитор | С разрешением Full HD 1920×1080 и выше |
| Программное обеспечение | SteamVR |
| Пространство | Пространство для Room–Scale VR |
| Доступ в интернет | Для загрузки обновлений и работы с облачными данными |
| Веб – сервер | Apache, Nginx с PHP 7.4+ и MySQL 5.7+/MariaDB |

Данные требования обеспечивают стабильную работу приложения на всех этапах разработки и эксплуатации.

1.2 Описание языка программирования

1.2.1 Выбор языка программирования

Для разработки приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» рассматриваются следующие языки программирования:

* C#;
* C++;
* Java;
* Python.

Каждый из них обладает своими особенностями, преимуществами и недостатками.

C# – это современный, объектно–ориентированный язык программирования, разработанный компанией Microsoft в рамках платформы .NET. Он сочетает в себе высокую производительность, безопасность типов и простоту синтаксиса, что делает его удобным для создания сложных приложений, включая игры и VR–проекты.[18]

Особенности C#:

* 1. C# является строго типизированным объектно–ориентированным языком. Это позволяет структурировать код, разделять логику на классы и использовать наследование, полиморфизм и инкапсуляцию. Это способствует удобному управлению игровыми объектами и взаимодействию с виртуальной средой.
  2. C# поддерживает асинхронное программирование. Это полезно для загрузки данных в VR–приложении без зависаний. Также язык обладает мощными инструментами для работы с коллекциями, событиями и делегатами. Это важно при проектировании сложных систем взаимодействия.
  3. Unity использует C# в качестве основного языка скриптов, предоставляя разработчику удобные API для управления игровыми объектами, физикой, анимацией и пользовательским интерфейсом. Благодаря этому C# становится естественным выбором для VR–разработки.
  4. C# работает с автоматическим управлением памятью. Это уменьшает вероятность утечек памяти в VR–приложениях. Также язык поддерживает безопасные типы данных. Это снижает риск ошибок, связанных с неправильной работой с памятью.
  5. C# в сочетании с .NET и Mono позволяет разрабатывать приложения для Windows, macOS, Linux и даже мобильных платформ. Это делает язык хорошим выбором для создания VR–приложений, которые могут быть портированы на разные устройства.
  6. C# поддерживается мощной экосистемой инструментов, включая Visual Studio, Rider и JetBrains, а также имеет огромную базу библиотек и фреймворков, таких как .NET Core, ASP.NET, Entity Framework.[6]

Преимущества C#:

* 1. C# является основным языком программирования для разработки на платформе Unity, которая используется в данном проекте. Благодаря встроенной поддержке C#, разработчики могут эффективно работать с игровыми объектами, сценами, анимацией и физическими процессами.
  2. C# поддерживает принципы объектно–ориентированного программирования. Это позволяет организовать код в виде модулей, обеспечивая повторное использование, удобство сопровождения и расширяемость.
  3. C# предлагает удобные средства работы с памятью, автоматическое управление памятью и защиту от ряда ошибок, связанных с указателями. Это снижает вероятность утечек памяти и ошибок доступа.
  4. Язык C# активно поддерживается Microsoft, обладает широкой документацией, обширным сообществом разработчиков, а также множеством библиотек и фреймворков. Это упрощает разработку сложных приложений.
  5. Использование C# в связке с платформой .NET обеспечивает простоту взаимодействия с базами данных, в том числе Microsoft SQL Server. Это критично для реализации данного проекта. Поддержка Entity Framework и ADO.NET позволяет работать с базами данных на высоком уровне абстракции.
  6. C# предлагает встроенные механизмы безопасности, такие как строгая типизация, исключения и защита от неконтролируемых операций с памятью. Это делает надежным языком для разработки сложных приложений.
  7. Современные версии C# и .NET позволяют создавать кроссплатформенные приложения, включая поддержку Windows, macOS и Linux, а также различных VR–устройств, таких как HTC Vive, Oculus Rift и другие.[14]

Недостатки С#:

1. C# требует среды выполнения .NET. Это ограничивает использование вне экосистемы Microsoft. Хотя существуют кроссплатформенные решения, такие как Mono или .NET Core, производительность и совместимость с различными ОС могут быть менее стабильными по сравнению с нативными языками, такими как C++.
2. Программы, написанные на C#, часто потребляют больше ресурсов по сравнению с аналогами на C++. Это связано с работой сборщика мусора, автоматически управляя памятью, но может вызывать кратковременные задержки в критически важных приложениях, например, в играх или VR–симуляциях.
3. Несмотря на оптимизацию и Just–In–Time компиляцию, C# медленнее нативного кода C++ в задачах, требующих высокой вычислительной мощности. Это может быть критично для VR–приложений, где важны высокая частота кадров и низкая задержка.
4. В отличие от C++, C# предоставляет ограниченный доступ к управлению памятью и системным ресурсам. Это затрудняет написание высокоэффективного кода для VR–устройств, требующего работы с низкоуровневыми API.
5. Хотя C# популярен среди разработчиков Unity, в сфере VR и AR больше специализированных решений основаны на C++, так как основные VR–движки предоставляют расширенные возможности именно для этого языка.[16]

С++ – это компилируемый, статически типизированный язык программирования общего назначения, разработанный Бьерном Страуструпом как расширение языка C. Язык поддерживает как процедурное, так и объектно–ориентированное программирование с популярными парадигмами, а также предоставляет механизмы для работы с низкоуровневыми операциями. Это делает гибким и производительным инструментом для разработки сложных программных решений.[17]

Особенности C++:

1. Поддержка классов, наследования, полиморфизма и инкапсуляции позволяет создавать масштабируемые и легко поддерживаемые программные системы.
2. Благодаря компиляции в машинный код и возможности управления памятью вручную, C++ обеспечивает максимальную эффективность работы программ.
3. Помогает выявлять ошибки на этапе компиляции. Это повышает надежность кода.
4. Язык поддерживает динамическое выделение памяти с помощью операторов new и delete, а также предоставляет умные указатели для автоматического управления ресурсами.
5. Позволяют писать обобщенный код, работающий с различными типами данных. Это снижает дублирование кода и повышает гибкость.
6. C++ включает библиотеку std::thread и механизмы синхронизации, позволяющие создавать многопоточные приложения.
7. Предоставляет эффективные контейнеры и алгоритмы для работы с данными.
8. Возможность взаимодействия с аппаратными ресурсами делает C++ подходящим для разработки системного программного обеспечения, драйверов и встроенных систем.
9. Программы на C++ могут компилироваться и выполняться на различных операционных системах при минимальных и не влияющих на работу изменениях кода.[10]

Преимущества C++:

1. C++ является компилируемым языком. Это позволяет создавать код, работающий напрямую с аппаратным обеспечением. В отличие от интерпретируемых языков, программы на C++ выполняются быстрее. Это критично для виртуальной реальности, где важна минимальная задержка при обработке данных и рендеринге.
2. C++ предоставляет разработчику полный контроль над памятью за счет использования указателей, выделения и освобождения памяти вручную. Это особенно важно при работе с VR, так как позволяет оптимизировать использование оперативной памяти и ресурсов видеокарты.
3. C++ поддерживает принципы ООП, включая инкапсуляцию, наследование и полиморфизм. Это упрощает разработку сложных систем, таких как виртуальный музей, где необходимо моделировать взаимодействие различных объектов.
4. C++ является основным языком для разработки на популярных игровых движках, таких как Unreal Engine и Unity. Это делает язык удобным выбором для создания VR–приложений, так как движки поддерживают работу с низкоуровневыми графическими API.
5. В C++ предусмотрена встроенная поддержка многопоточности. Это позволяет эффективно распределять вычисления, например, между рендерингом сцены и обработкой данных из базы данных.[19]

Недостатки C++:

1. C++ предоставляет низкоуровневый контроль над памятью и ресурсами. Это требует от разработчика большей ответственности за управление памятью, выделение и освобождение ресурсов. Это увеличивает сложность кода и возможность возникновения ошибок, таких как утечки памяти и использование неинициализированных указателей.
2. Из–за высокой сложности C++ разработка программ на этом языке занимает больше времени по сравнению с C# или Python. Использование автоматического управления памятью в C# значительно упрощает разработку и ускоряет процесс тестирования.
3. Unity изначально разрабатывался с поддержкой C# и .NET. Это делает язык использование с C# значительно удобнее. Хотя существуют плагины и способы интеграции C++ с Unity, это усложняет процесс разработки и требует дополнительных усилий.
4. C++ не имеет встроенной поддержки для работы с Microsoft SQL Server. В то время как в C# есть ADO.NET и Entity Framework, упрощающие взаимодействие с базой данных, в C++ необходимо использовать сторонние библиотеки. Это требует дополнительной настройки и усложняет процесс разработки.
5. C++ код сложнее поддерживать и переносить на другие платформы по сравнению с C#. В C# благодаря .NET и Unity обеспечивается кроссплатформенность. Это делает язык более удобным для разработки VR–приложений.[9]

Java – объектно–ориентированный язык с автоматическим управлением памятью, работающий на виртуальной машине.[21]

Особенности Java:

* кроссплатформенность;
* широкое применение в корпоративной разработке.[8]

Преимущества Java:

* + стабильность и безопасность;
  + большое количество фреймворков и библиотек;
  + поддержка многопоточности.[20]

Недостатки Java:

* + менее производителен, чем C# и C++;
  + ограниченная поддержка в Unity;
  + зависимость от JVM.[12]

Python – это высокоуровневый интерпретируемый язык программирования общего назначения, ориентированный на простоту кода и скорость разработки. Однако для VRприложений на Unity он не подходит из–за ограниченной поддержки и меньшей производительности в сравнении с C#.[13]

Особенности Python:

1. Благодаря лаконичному синтаксису Python значительно упрощает разработку. Это позволяет писать и поддерживать код быстрее, чем в более сложных языках, таких как C++ или Java.
2. Python обладает богатой экосистемой, включающей модули для работы с базами данных, такие как sqlite3, SQLAlchemy, а также библиотеки для интеграции с Unity, например, Unity ML–Agents.
3. Python поддерживается на Windows, Linux и macOS. Это делает удобным для работы в разных средах.
4. Python часто используется для написания скриптов автоматизации. Это упрощает администрирование базы данных и защиту данных, например, с помощью регулярного резервного копирования.
5. Язык поддерживает взаимодействие с REST API. Это важно для разработки распределенных систем и облачного хранения данных.
6. Использование asyncio, threading и multiprocessing позволяет обрабатывать большие объемы данных в реальном времени.
7. Наличие библиотек TensorFlow, PyTorch, scikit–learn делает Python идеальным выбором для анализа данных и интеллектуальной обработки информации в БД.[15]

Преимущества Python:

1. Python отличается лаконичным и понятным синтаксисом. Это упрощает разработку и поддержку программного кода. В сравнении с C++ и C#, язык требует меньше строк кода для выполнения аналогичных задач. Это ускоряет процесс разработки.
2. Python обладает обширной библиотекой модулей, которые могут быть полезны при работе с базами данных.
3. Python поддерживает работу на различных операционных системах. Это делает удобным для кроссплатформенной разработки. В отличие от C#, Python можно использовать без привязки к конкретной ОС.
4. В отличие от C++, где управление памятью требует особого внимания, в Python реализован автоматический сборщик мусора, снижающий вероятность утечек памяти и упрощающий разработку.
5. Python имеет одно из крупнейших сообществ разработчиков. Это обеспечивает доступ к множеству учебных материалов, документации и форумов поддержки.[7]

Недостатки Python:

* + - 1. Python является интерпретируемым языком. Это делает значительно медленнее по сравнению с компилируемыми языками, такими как C# и C++. В задачах, связанных с 3D–графикой и виртуальной реальностью, производительность критически важна, а Python может не справляться с высокими нагрузками.
      2. Хотя Python широко применяется в сфере научных вычислений, анализа данных и автоматизации, поддержка в Unity отсутствует из коробки. В отличие от C#, не является основным языком для Unity, использование Python потребует дополнительных библиотек и костылей. Это усложнит разработку.
      3. Из–за Global Interpreter Lock Python ограничен в возможностях параллельного выполнения кода. Это критично при разработке VR–приложений, где активно используется многопоточная обработка.
      4. Python является динамически типизированным языком. Это делает менее удобным для разработки крупных проектов. Ошибки, связанные с типами данных, могут проявляться только во время выполнения. Это усложняет отладку и снижает надежность кода.
      5. Python использует автоматический сборщик мусора. Сборщик может вызывать непредсказуемые задержки при работе VR–приложений. В C# и C++ управление памятью осуществляется более эффективно. Это позволяет избегать неожиданных пауз в работе программы.[11]

Для данного проекта выбран C#, так как он:

* оптимален для разработки в Unity;
* имеет хорошую интеграцию с SteamVR;
* обеспечивает баланс между производительностью и удобством разработки;
* поддерживает WindowsForms для вспомогательных утилит.

Остальные языки либо менее подходят для VR, либо требуют более сложной разработки.

1.2.2 Элементы языка программирования

Элементы языка программирования С# представлены в соответствии с таблицей 3.

Таблица 3 – Элементы языка программирования C#

|  |  |
| --- | --- |
| Элемент языка C# | Описание |
| 1 | 2 |
| Классы (class) | Базовая структура для создания объектов, инкапсулирующая данные и методы. |
| Методы (void Foo()) | Функции, выполняющие определенные действия. Могут возвращать значения. |
| Циклы (for, while) | Управляющие конструкции для повторения кода. |
| Условные операторы (if, switch) | Проверка условий и ветвление логики программы. |
| Свойства (get; set;) | Контролируемый доступ к полям класса. |
| Делегаты (delegate) | Типы, представляющие ссылки на методы. Используются для событий и callback’ов. |
| События (event) | Механизм оповещения о действиях. |
| Интерфейсы (interface) | Контракты, определяющие поведение классов без реализации. |
| Асинхронность (async/await) | Позволяет выполнять длительные операции без блокировки основного потока. |
| Лямбда–выражения (=>) | Упрощенный синтаксис для анонимных методов. |

Продолжение таблицы 3

|  |  |
| --- | --- |
| 1 | 2 |
| LINQ (Where, Select) | Язык запросов к коллекциям и базам данных. |
| Null–безопасность (?., ??) | Операторы для безопасной работы с nullable–типами. |
| Кортежи ((int, string)) | Группировка нескольких значений в одну структуру. |
| Паттерны (switch expression) | Упрощенное сопоставление с образцом. |
| Record–типы (record) | Неизменяемые структуры данных с автоматической реализацией Equals, ToString. |
| Пространства имен (namespace) | Логическая группировка классов для избежания конфликтов имен. |
| Работа с файлами (System.IO) | Чтение и запись файлов через File, StreamReader и др. |

Язык программирования C# предоставляет широкий набор инструментов для разработки программного обеспечения, включая работу с базами данных и виртуальной реальностью. В основе лежит строгая типизация и объектно – ориентированный подход. Это делает язык удобным для разработки масштабируемых и структурированных приложений.

Разнообразие типов данных позволяет эффективно управлять различными видами информации, а использование операторов и условных конструкций обеспечивает гибкость в написании логики программы. Циклы и массивы дают возможность обрабатывать большие объемы данных. Это критично при создании базы данных и работы с коллекциями.

Функции и методы в C# поддерживают перегрузку и инкапсуляцию. Это способствует созданию модульного кода. Наличие структур и коллекций делает работу с данными более удобной и производительной. Комментарии позволяют документировать код. Это важно для командной разработки и дальнейшего сопровождения проекта.

В целом, элементы C# обеспечивают мощный инструментарий для реализации поставленных задач в рамках разработки базы данных и приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны».

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Диаграмма предметной области

Виртуальный музей времен Великой Отечественной войны представляет собой интерактивное приложение, позволяющее пользователям взаимодействовать с историческими экспонатами в VR–пространстве.

Основные сущности:

* посетитель виртуального музея, взаимодействующий с системой через VR–устройство;
* программный комплекс, разработанный на Unity с использованием SteamVR, обеспечивающий визуализацию и навигацию;
* 3D–модели исторических артефактов, документов, оружия и других объектов, связанных с Великой Отечественной войной;
* хранилище информации об экспонатах, включающее описание, историческую справку и мультимедийные материалы;
* система меню и навигации, реализованная на C# и Windows Forms, для взаимодействия пользователя с приложением.

Взаимодействия:

* пользователь запускает VR–приложение через SteamVR;
* приложение загружает данные из базы данных и отображает виртуальное пространство музея;
* пользователь перемещается по залам, взаимодействует с экспонатами, получая дополнительную информацию.

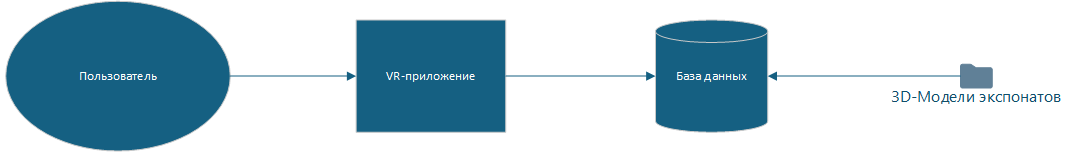


Рисунок 1 – Диаграмма предметной области

Данная структура отражает основные компоненты системы и их взаимосвязи, обеспечивая корректную работу виртуального музея.

2.2 Описание алгоритма

2.2.1 Алгоритм работы системы

Приложение виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» функционирует на основе последовательного выполнения ключевых этапов. После запуска инициализируются Unity и SteamVR, загружаются сцены и ресурсы. При наличии системы авторизации проверяются учетные данные через PHPMyAdmin. В главном меню пользователь выбирает экспозиции или интерактивные сцены, после чего загружаются 3D–модели, текстуры и мультимедийный контент. В VR–режиме обеспечивается взаимодействие с объектами: перемещение, осмотр экспонатов, прослушивание аудиогидов. При необходимости данные сохраняются в базу данных. Завершение работы включает освобождение ресурсов и корректное отключение от VR–устройств.

Схема взаимодействия модулей приведена на листе формата А1 с шифром ДП 09.02.07 ИСП–421/2к.23.25 01 С1, прилагаемом к дипломному проекту.

2.2.2 Алгоритм работы подпрограмм

Алгоритм работы «WWII» представлен в соответствии с рисунком 2.



Рисунок 2 – Алгоритм работы подпрограмм

Спецификации подпрограмм представлены в соответствии с таблицей 4.

Таблица 4 – Спецификация подпрограмм

|  |  |  |  |
| --- | --- | --- | --- |
| Название подпрограммы | Входные данные | Выходные данные | Функциональное назначение |
| Инициализация VR–среды | Конфигурационный файл | Готовая VR–сцена | Подготовка окружения для работы в виртуальной реальности |
| Загрузка контента | Запрос к БД | 3D–модели, текстуры, аудио | Получение и отображение музейных экспонатов |
| Взаимодействие с экспонатами | Позиция контроллера, триггеры | Контекстное меню, журнал действий | Обработка пользовательского ввода и отображение информации |
| Выход из приложения | Команда закрытия | Отчет о сеансе | Корректное завершение работы программы |

Блок–схема программы приведена на листе формата А1 с шифром ДП 09.02.07 ИСП–421/2к.23.25 01 С1, прилагаемый к дипломному проекту.

Диаграма классов приведена на листе формата А1 с шифром ДП 09.02.07 ИСП–421/2к.23.25 01 С1, прилагаемый к дипломному проекту.

2.3 Текст программы

Листинг программы – это полный или фрагментированный исходный код программного продукта, представленный в текстовом виде с пояснениями.

Листинг программы представлен в соответствии с приложением A.

2.4 Инструкция работы с программой

2.4.1 Общие сведения

Приложение «Виртуальный музей времен Великой Отечественной войны» разработано для ГАПОУ ВО «Владимирский политехнический колледж» с целью создания интерактивного образовательного пространства в формате виртуальной реальности. Программа позволяет пользователям взаимодействовать с историческими экспонатами, просматривать информационные материалы и погружаться в атмосферу событий 1941–1945 годов.

Приложение реализовано на платформе Unity с использованием SteamVR для обеспечения совместимости с VR–устройствами. Взаимодействие с базой данных организовано через PHPMyAdmin, что позволяет хранить и управлять информацией об экспонатах, пользователях и их действиях. Интерфейс разработан на C# с применением WindowsForms для администрирования контента.

Основные функции включают:

* навигацию по виртуальным залам музея;
* просмотр 3D–моделей экспонатов с детализированными описаниями;
* интерактивные сценарии, воссоздающие ключевые события войны;
* систему управления контентом для администраторов.

Для работы требуется VR–гарнитура, ПК с операционной системой Windows 10/11 и установленным SteamVR. Интерфейс администратора доступен через отдельное приложение на C#, взаимодействующее с базой данных MySQL.

2.4.2 Вызов и загрузка

Для запуска приложения необходимо выполнить следующие действия:

* Открыть клиент Steam и запустить SteamVR в соответствии с рисунками 3–7;
* Открыть исполняемый файл приложения в соответствии с рисунками 7–8.



Рисунок 3 – Запуск Steam

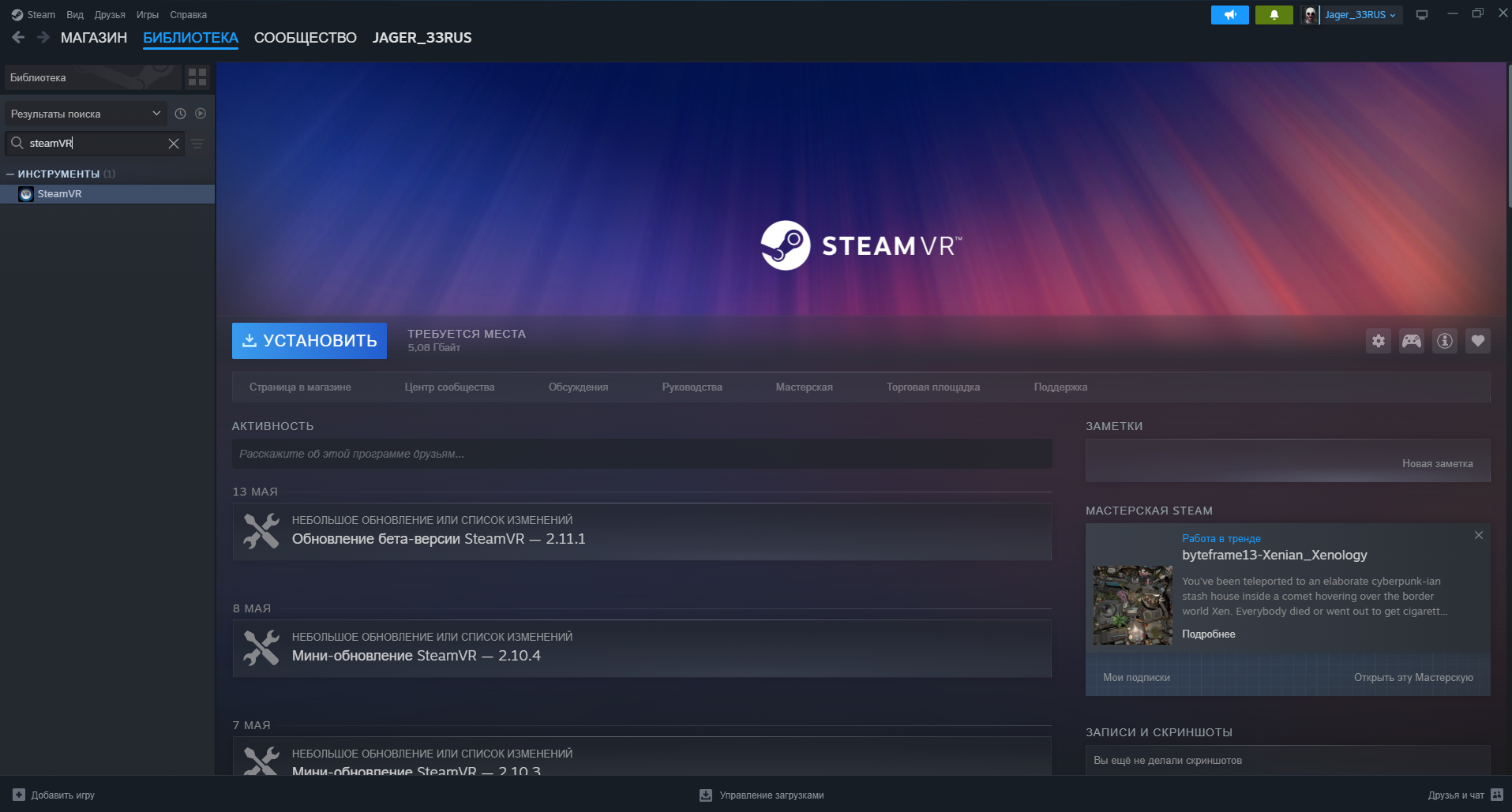


Рисунок 4 – Интерфейс Steam

Рисунок 5 – Запуск SteamVR

Рисунок 6 – Интерфейс SteamVR



Рисунок 7 – Место хранения игры



Рисунок 8 – Запуск игры

Запуск приложения виртуальной реальности требует предварительной активации SteamVR через клиент Steam. После успешного запуска SteamVR выполняется открытие исполняемого файла приложения. Процесс загрузки и инициализации виртуальной среды подтверждается корректным отображением интерфейсов Steam и SteamVR. Указанные шаги обеспечивают стабильную работу приложения и готовность к взаимодействию с виртуальным музеем.

2.4.3 Входные и выходные данные

В процессе работы в приложении может быть получена следующая выходная информация:

* 3D–модели музейных экспонатов и виртуальной среды;
* текстовые и аудиовизуальные материалы;
* сообщения о состоянии системы;
* готовые SQL–запросы и результаты их выполнения;
* изменения в интерфейсе.

Входной информацией в приложении может выступить следующие типы данных:

* параметры подключения к базе данных;
* данные аутентификации пользователя;
* команды управления, передаваемые через устройства VR;
* данные о местоположении пользователя в виртуальном пространстве;
* запросы к базе данных;
* настройки графики и управления.

В ходе анализа входных и выходных данных приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» определены ключевые типы информации, обеспечивающие его функционирование.

Выходные данные включают 3D–модели экспонатов и виртуальной среды, текстовые и аудиовизуальные материалы, системные сообщения, сформированные SQL–запросы с результатами их выполнения, а также изменения интерфейса в ответ на действия пользователя.

Входные данные представлены параметрами подключения к базе данных, учетными данными для аутентификации, командами управления с VR–устройств, координатами пользователя в виртуальном пространстве, запросами к СУБД и настройками графики.

Определение структуры входных и выходных данных позволяет обеспечить корректное взаимодействие пользователя с системой, стабильную работу приложения и эффективную обработку информации.

2.4.4 Сообщения программы

В процессе взаимодействия с приложением «Виртуальный музей времен Великой Отечественной войны» система выводит информационные сообщения в зависимости от выбранного объекта:

* сообщения об оружии в соответствии с рисунком 9;
* сообщения о технике в соответствии с рисунком 10.

Рисунок 9 – Сообщение об оружии

Рисунок 10 – Сообщение о технике

Сообщения формируются динамически в зависимости от взаимодействия с элементами виртуального пространства, что способствует удобному и интуитивно понятному изучению материалов.

2.5 Описание процесса отладки программы

2.5.1 Методы отладки

Существует три подхода к тестированию программного обеспечения: тестирование белого, серого и черного ящиков. Подходы к тестированию рассматривает процесс с точки зрения, и не может быть использован в качестве единственного подхода.

Завершение всех трех этапов гарантирует качество продукта.

Для тестирования программы будут использоваться методы белого и черного ящиков.

2.5.2 Тестирование и отладка

Тестирование приложения методом черного ящика представлено в соответствии с таблицей 5.

Таблица 5 – Тестирование приложения методом черного ящика

|  |  |  |
| --- | --- | --- |
| Действие | Что произошло | Соответствует ожиданию |
| 1 | 2 | 3 |
| Запуск программы «WWII.exe» | Открытие приложения, сцена музея | Да, соответствует |
| Зажатие верхней части стика | Появляется луч на соответствующем контроллере | Да, соответствует |
| Нажатие на кнопку информации об оружии | Выводится информация с описанием 3D–модели соответствующего оружия | Да, соответствует |
| Нажатие на кнопку тематического видео | Запускается воспроизведение тематического видео в VR–среде | Да, соответствует |
| Нажатие на кнопку «В ангар» | Переход на сцену с техникой времен ВОВ | Да, соответствует |
| Нажатие на кнопку «Выйти» | Корректное закрытие приложения, освобождение ресурсов | Да, соответствует |

Тестирование приложения методом черного ящика представлено в соответствии с таблицей 6.

Таблица 6 – Тестирование приложения методом черного ящика

|  |  |  |
| --- | --- | --- |
| Действие | Что произошло | Соответствует ожиданию |
| 1 | 2 | 3 |
| Запуск программы «WWII.exe» | Происходит инициализация сцены «Музей» | Да, соответствует |
| Зажатие верхней части стика | Перемещение по музейному пространству | Да, соответствует |
| Нажатие на кнопку информации об оружии | Отображение исторической справки (GetInfo()) | Да, соответствует |
| Нажатие на кнопку тематического видео | Запуск видео с документальными материалами (PlayVideo()) | Да, соответствует |
| Нажатие на кнопку «В ангар» | Загрузка новой сцены с военной техникой, возможность осмотра (GoToScene()) | Да, соответствует |
| Нажатие на кнопку «Выйти» | Сохранение состояния, выход в операционную ситстему (Exit()) | Да, соответствует |

Выявленные ошибки устранены. Приложение стабильно работает в целевой среде, соответствует функциональным и техническим требованиям.

2.5.3 Контрольный пример

Для проверки корректности работы приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» разработан контрольный пример, включающий последовательность действий и ожидаемые результаты.

Для открытия программы, необходимо дважды нажать на значок, формата .exe с названием «WWII» левой кнопкой мыши, расположенный в корневой папке с игрой в соответствии с рисунком 11.



Рисунок 11 – Запуск программы

После, откроется сцена с музеем, в соответствии с рисунком 12.

Рисунок 12 – Музей

Для перемещения по музею, необходимо нажать на верхнюю часть стика контроллера, навестись на место, в которое необходимо переместиться и отпустить верхнюю часть стика контроллера, в соответствии с рисунками 13 – 14.

Рисунок 13 – Луч перемещения

Рисунок 14 – Перемещение

Для просмотра информации об экспонатах, необходимо нажать на кнопку, соответствующую экспонату, в соответствии с рисунками 15 – 16.

Рисунок 15 – Кнопка просмотра информации об экспонате

Рисунок 16 – Показ информации об экспонате

Для просмотра тематического видео, необходимо нажать на соответствующую кнопку, в соответствии с рисунками 17 – 18.

Рисунок 17 – Кнопка просмотра видео

Рисунок 18 – Показ тематического видео

Для перемещения в ангар с военной техникой, необходимо нажать на соответствующую кнопку, в соответствии с рисунками 19–20

Рисунок 19 – Кнопка перемещения в ангар

Рисунок 20 – Ангар

Для того, чтобы выйти из приложения, необходимо нажать на соответствующую кнопку, в соответствии с рисунком 21.

Рисунок 21 – Кнопка выхода из приложения

Для проверки корректности работы приложения виртуальной реальности «Виртуальный музей времен Великой Отечественной войны» разработан контрольный пример, включающий последовательность действий и ожидаемые результаты.

Запуск программы осуществляется двойным нажатием левой кнопки мыши на исполняемый файл формата .exe с названием «WWII». После запуска открывается сцена с виртуальным музеем.

Перемещение по музею выполняется с помощью контроллера: нажатие на верхнюю часть стика активирует луч перемещения, а отпускание кнопки подтверждает перемещение в выбранную точку.

Просмотр информации об экспонатах осуществляется нажатием на соответствующую кнопку, после чего отображается детальная информация. Для воспроизведения тематического видео также предусмотрена отдельная кнопка.

В приложении реализована возможность перехода в ангар с военной техникой через соответствующую кнопку. Завершение работы программы выполняется нажатием кнопки выхода.

Все функции приложения протестированы и работают в соответствии с заявленными требованиями. Контрольный пример подтверждает корректность выполнения основных сценариев взаимодействия пользователя с программой.

3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

А

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта была разработана VR–система «Виртуальный музей времен Великой Отечественной войны» для ГАПОУ ВО «Владимирский политехнический колледж». Приложение реализовано с использованием современных технологий, включая Unity, SteamVR, C# и Windows Forms, что обеспечило интерактивность и удобство взаимодействия пользователя с виртуальной экспозицией.

В рамках проекта проведен анализ требований, определены функциональные возможности системы, разработана архитектура приложения и реализованы ключевые модули. Использование PHPMyAdmin позволило организовать хранение и управление данными, обеспечив гибкость и масштабируемость решения.

Тестирование подтвердило корректность работы системы, включая взаимодействие с VR–оборудованием, навигацию по музею и отображение экспонатов. Применение методов отладки и контрольного примера позволило выявить и устранить возможные ошибки.

Разработанное приложение может быть использовано в образовательных целях, способствуя сохранению исторической памяти и повышая интерес к изучению событий Великой Отечественной войны с помощью инновационных технологий.

Проект демонстрирует эффективность выбранных инструментов и подходов к разработке VR–приложений, а также подтверждает возможность их практического применения в учебных заведениях.

ЛИТЕРАТУРА

1. ГОСТ 19.105 – 78 Единая система программной документации: Общие требования к программным документам. – М: Изд–во стандартов, 1994.
2. ГОСТ 19.106 – 78 Единая система программной документации: Требования к печатным документам, выполненным печатным способом. – М: Изд–во стандартов, 1994.
3. ГОСТ 19.402 – 78 Единая система программной документации: Описание программы. – М: Изд–во стандартов, 1994.
4. ГОСТ 19.404 – 79 Единая система программной документации: Пояснительная записка. Требования к содержанию и оформлению. – М: Изд–во стандартов, 1994.
5. ГОСТ 19.504 – 79 Единая система программной документации: Руководство программиста. Требования к содержанию и оформлению. – М: Изд–во стандартов, 1994.
6. Албахари Д.: C# 11. Справочник. Полное описание языка 2024 – 976 стр.
7. Бейдер Д.: Python для анализа данных в VR–средах 2024 – 490 стр.
8. Блох Д.: Java: эффективное программирование 2024 – 416 стр.
9. Грин К.: Современный C++ для геймдева. Оптимизация VR–приложений 2025 – 534 стр.
10. Липпман С., Лажойе Ж.: \*C++20. Начала программирования\* 2024 – 672 стр.
11. Маккинни У.: Python и базы данных. Работа с MySQL и PHPMyAdmin 2023 – 356 стр.
12. Марченко А.: Java для VR–разработки. Интеграция с Unity 2025 – 278 стр.
13. Матиз Э.: Изучаем Python. Программирование игр, визуализация данных, веб–приложения 2023 – 704 стр.
14. Прайс М.: Unity в действии. Разработка VR–приложений на C# 2023 – 480 стр.
15. Рамальо Л.: Python. К вершинам мастерства 2022 – 608 стр.
16. Скит Д.: Программирование на C# для начинающих. Подробное руководство 2022 – 352 стр.
17. Страуструп Б.: Язык программирования C++. Краткий курс 2023 – 320 стр.
18. Троелсен Э., Джепикс Ф.: C# 10 и .NET 6. Современная кроссплатформенная разработка 2023 – 1024 стр.
19. Уильямс Э.: C++ для виртуальной реальности. Разработка под SteamVR 2023 – 418 стр.
20. Уолш Д.: \*Java и веб–разработка. Spring Boot 3\* 2023 – 512 стр.
21. Хорстман К.: Java. Библиотека профессионала. Том 1. Основы 2023 – 864 стр.