Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт технологий открытого образования

**ДОПУСТИТЬ К ЗАЩИТЕ**

Руководитель образовательной программы

09.03.03 Прикладная информатика

 И. Н. Обабков .

 (подпись) (Ф.И.О.)

«20» июня 2024 г.

**Проектный практикум**

**Отчет**

**1C24S.GD. Образовательная игра**

Куратор \_\_\_\_\_\_\_\_\_\_\_ А. Е. Хрушков

Студенты группы РИЗ-130045д \_\_\_\_\_\_\_\_\_\_\_ А. А. Осетров

 \_\_\_\_\_\_\_\_\_\_\_ Т. Г. Раюллина

 \_\_\_\_\_\_\_\_\_\_\_ А. А. Свалухин

 \_\_\_\_\_\_\_\_\_\_\_ В. В. Черных

Екатеринбург 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт технологий открытого образования

Направление (специальность) 09.03.03 Прикладная информатика

**УТВЕРЖДАЮ**

Руководитель образовательной программы 09.03.03 Прикладная информатика

 И. Н. Обабков .

 (подпись) (Ф.И.О.)

«12» февраля 2024 г.

**ЗАДАНИЕ**

на выполнение проектного практикума

обучающихся Осетрова Артема Анатольевича, Раюллиной Татьяны Глебовны, Свалухина Алексея Анатольевича, Черных Василия Викторовича группы РИЗ-130045д

 (фамилия, имя, отчество)

1 Тема работы

 Разработка образовательной игры

2 Куратор Хрушков Артем Евгеньевич, внешний эксперт

(Ф.И.О., должность, ученое степень, ученая звание)

3 Исходные данные к работе

 Список требований к приложению и этапов к выполнению в рамках проектного практикума.

4 Содержание пояснительной записки (перечень подлежащих разработке вопросов)

 1. Анализ предметной области.

 2. Проектирование программного продукта.

 3. Реализация программного продукта.

5 Перечень демонстрационных материалов

 Презентация в MS PowerPoint.

6 Календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этаповвыполнения работы | Срок выполнения этапов работы |  | Отметкао выполнении |
| Анализ предметной области | 06.05.2024 — 12.05.2024 |  | выполнено |
| Проектирование программного продукта | 13.05.2024 — 19.05.2024 |  | выполнено |
| Реализация программного продукта | 20.05.2024 — 26.05.2024 |  | выполнено |
| Подготовка отчета и презентационных материалов | 27.05.2024 — 22.06.2024 |  | выполнено |

Куратор . . Хрушков Артем Евгеньевич .

 (подпись) (Ф.И.О.)

Задание приняли к исполнению 12.02.2024 г.

 (дата) (подпись обучающегося)

 (подпись обучающегося)

 (подпись обучающегося)

 (подпись обучающегося)

7 Работа по проекту закончена «20» июня 2024 г.

Отчет и все материалы просмотрены

Считаю возможным допустить . Осетрова Артема Анатольевича, Раюллину Татьяну Глебовну, Свалухина Алексея Анатольевича, Черных Василия Викторовича к защите их проектной работы в комиссии.

Руководитель . .

 (подпись)

8 Допустить . Осетрова Артема Анатольевича, Раюллину Татьяну Глебовну, Свалухина Алексея Анатольевича, Черных Василия Викторовича к защите проектной работы в комиссии.

Руководитель образовательной программы

09.03.03 Прикладная информатика . . . . И. Н. Обабков

 (подпись) (Ф.И.О.)

Содержание

[Содержание 6](#_Toc169796939)

[Введение 9](#_Toc169796940)

[1 Анализ предметной области 11](#_Toc169796941)

[1.1 Анализ технического задания 11](#_Toc169796942)

[1.2 Анализ средств разработки приложения 12](#_Toc169796943)

[2 Проектирование архитектуры системы 16](#_Toc169796944)

[2.1 Архитектура приложения 16](#_Toc169796945)

[2.2 Логика взаимодействия пользователя с компонентами игры 18](#_Toc169796946)

[3 Реализация системы 21](#_Toc169796947)

[3.1 Текущее состояние игры 21](#_Toc169796948)

[3.2 Дальнейшее развитие игры 22](#_Toc169796949)

[Заключение 23](#_Toc169796950)

Введение

 В современном мире программирование становится все более востребованным навыком, необходимым для различных областей профессиональной деятельности. Одним из ключевых аспектов программирования является понимание и применение принципов объектно-ориентированного программирования (ООП). Язык C# (C-Sharp) является одним из популярных языков программирования, активно используемым в разработке программного обеспечения, особенно в среде .NET. Образовательные игры представляют собой эффективный инструмент для обучения сложным концепциям, таким как ООП, благодаря интерактивности и вовлеченности учащихся. Разработка образовательной игры, обучающей ООП на языке C#, позволяет сочетать теоретические знания с практическим применением, что способствует более глубокому пониманию и усвоению материала.

Актуальность работы заключается в необходимости разработки образовательной игры для обучения ООП на языке C#, которая позволит улучшить качество образования и подготовку специалистов в области программирования. Пользователи узнают основы объектно-ориентированного программирования, научаться применять знания в написании кода, пользуясь материалами и упраженниями в разработанном приложении.

Наиболее предпочтительным форматом сервиса будет именно desktop-приложение, потому что этот формат позволяет максимально использовать технические возможности и обеспечивать высокое качество обучения, надежность и удобство для пользователей. Для реализации такого решения потребуется разработка полноценного приложения, включая его архитектуру, пользовательский интерфейс, учебный контент. Основываясь на требованиях, поставленных в техническом задании, и проведя их анализ, можно будет выявить требования к стеку технологий и подобрать подходящие инструменты.

Объектом исследования является разработка desktop приложения для обучения в игровой форме.

Предметом является разработка образовательной игры для обучения принципам ООП на языке C#.

Целью проекта является разработка и внедрение образовательной игры, которая поможет учащимся освоить основные концепции и принципы объектно-ориентированного программирования на языке C#. Для достижения цели необходимо выполнить задачи:

1. Провести анализ существующих образовательных ресурсов и игр для обучения ООП на языке C#.
2. Разработать концепцию и сценарий образовательной игры, направленной на обучение основным принципам ООП.
3. Реализовать разработанную концепцию в виде интерактивной образовательной игры.
4. Провести тестирование игры с участием учащихся для оценки ее эффективности в обучении ООП.
5. Анализировать результаты тестирования и внести необходимые улучшения в игру.

При выполнении работы использовались следующие методы: анализ, исследования и сравнения.

Для выполнения проектной работы были использованы техническая документация, техническое задание от аналитика.

Результатом работы будет desktop приложение для обучения основам объектно-ориентированного программирования на языке C#.

1. Анализ предметной области

Анализ технического задания

В первую очередь нужно провести анализ технического задания. На основе списка требований к сервису можно определить необходимые технологии и составить перечень задач для разработки приложения. Только после этого можно приступать к выбору инструментов разработки и к непосредственной разработке продукта.

В постановке задач требовалось, чтобы игра была разработана на движке Unity 3D, обладала возможностью для дальнейшего развития (добавления новых уровней и сюжетных линий), имела удобный инструмент по наполнению и редактированию контента и содержала следующие разделы:

* главное меню;
* таблицу достижений;
* игровые локации для каждого уровня;
* консоли для выполнения каждого типа заданий и просмотра теории.

Игра должна содержать элементы с теоритическими лекциями в текстовой и аудио форме, тестами на закрепление теории, задачами на написание кода на языке C# с автоматической проверкой.

Наличие теоретической части подразумевает наполнение игры учебным материалом, для чего необходимо верно подобрать его источники, где содержится корректная информация. Для этого можно воспользоваться материалами из различных онлайн-курсов и обучающих сайтов (например, Ulearn, METANIT.COM) и официальной документацией Microsoft Learn.

Тесты на закрепление теории могут содержать несколько вариантов ответа и, как правило, требуют хотя бы одного ответа от пользователя. Данную функциональность можно реализовать с помощью наличия базы вопросов с ответами и встроенного в игру окна с тестом.

Задачи с автоматической проверкой требуют от пользователя самостоятельного написания кода на языке C#. Удобно проверять правильность ответа при помощи встроенного компилятора, работающего на платформе Roslyn.

Для реализации игровых локаций необходимы различные графические материалы и структуры, которые возможно получить из открытых источников и доработать, при необходимости, в различных графических редакторах. Для управления графикой необходимо разработать менеджер пользовательского интерфейса.

Программный код должен представлять собой менеджер управления игрой, который должен включать в себя ряд элементов: для загрузки сцен – менеджер сцен, для озвучивания игры – аудио менеджер, позволяющий воспроизводить звук в несколько потоков, а также менеджеры для отдельных частей игры, таких как, меню загрузки, задания с компилированием кода, задания с тестами.

Наиболее подходящим форматом сервиса будет desktop приложение, так как этот формат обладает большей производительностью и дает доступ к использованию большего числа ресурсов, чем, например, веб-версия. Кроме того, desktop-приложения не зависят от постоянного подключения к Интернету, что обеспечивает пользователям возможность учиться в любое время и в любом месте, без риска потери данных или прерывания процесса обучения из-за проблем с сетью.

В ходе анализа технического задания были выявлены и скорректированы основные задачи на разработку, определен формат разрабатываемого приложения. Далее можно приступать к выбору технологического стека и инструментов разработки.

Анализ средств разработки приложения

Исходя из анализа технического задания можно выявить требования к инструментам разработки приложения. Выбранный стек технологий должен содержать:

* возможность удобного управления контентом;
* возможность развития приложения;
* встроенный компилятор дл проверки кода, вводимого игроком;
* гибкую работу с графическими моделями;
* механизм сохранения результатов пользователей.

 Кроме того, для организации работы команды необходимы средства для хранения и обмена информацией, постановки задач и проектного управления.

 Исходя из требований, для разработки desktop-приложения образовательной игры по обучению ООП на языке C# был выбран следующий стек технологий:

1. Управление проектом и задачами

* RedMine: Это мощная система управления проектами и отслеживания задач, которая будет использоваться для планирования, мониторинга и управления всеми этапами разработки. RedMine поддерживает создание и управление задачами, настройку рабочих процессов и отслеживание прогресса.

2. Контроль версий и совместная разработка

* GitLab: Платформа для управления репозиториями Git, которая также предлагает интеграцию с системами CI/CD. GitLab будет использоваться для хранения кода, управления версиями, а также для автоматизации сборки и тестирования приложения.
* Git Extensions: Это графический интерфейс для работы с Git, который облегчит управление репозиториями для разработчиков, предоставляя удобные инструменты для просмотра изменений, коммитов и истории проектов.

3. Разработка и программирование

* Unity: Мощный движок для разработки игр и интерактивных приложений. Unity поддерживает C# как основной язык программирования и будет использоваться для создания интерактивных элементов, визуализации и геймплея образовательной игры.
* Visual Studio: Основная среда разработки (IDE) для написания и отладки кода на языке C#. Visual Studio предлагает мощные инструменты для программирования, тестирования и отладки, что делает его идеальным выбором для разработки C# приложений.

4. Дизайн и прототипирование

* Figma: Онлайн-инструмент для совместного создания прототипов и интерфейсов. Figma будет использоваться для разработки макетов пользовательского интерфейса, что позволяет дизайнерам и разработчикам работать вместе в реальном времени, обеспечивая согласованность и интеграцию дизайна.

5. Графика и иллюстрации

* Inkscape: Бесплатный векторный графический редактор, который будет использоваться для создания и редактирования векторной графики, такой как иконки и интерфейсные элементы.
* GIMP: Бесплатный растровый графический редактор, аналогичный Adobe Photoshop, используемый для редактирования изображений, создания графики и текстур, которые будут использоваться в игре.

6. 3D Моделирование

* Blender: Бесплатный и мощный инструмент для 3D-моделирования, анимации и рендеринга. Blender будет использоваться для создания трехмерных моделей и анимаций, которые могут быть интегрированы в Unity для более интерактивного и визуально привлекательного образовательного опыта.

7. Хранение и обмен файлами

* Google Drive: Облачное хранилище, которое будет использоваться для хранения документов, ресурсов и совместной работы над файлами. Google Drive обеспечит доступ к необходимым материалам всем членам команды, независимо от их местоположения.

8. Коммуникация и совместная работа

* Discord: Платформа для голосового, видео и текстового общения, используемая для повседневной коммуникации команды. Discord будет полезен для проведения встреч, обсуждений и оперативного взаимодействия между участниками проекта.
* Telegram: Мессенджер для быстрого обмена сообщениями и уведомлений. Telegram будет использоваться для обмена важной информацией, документами и поддержания связи в реальном времени.

 Этот стек технологий обеспечивает всестороннюю поддержку всех аспектов разработки десктоп-приложения, от управления проектом и контроля версий до программирования, дизайна, моделирования и коммуникации. Такой подход позволит эффективно организовать работу команды и создать качественное образовательное приложение.

1. Проектирование архитектуры системы

Архитектура приложения

Приложение представляет собой десктопную образовательную игру, разработанную на языке C# с использованием Unity. В приложении используется несколько ключевых компонентов, которые взаимодействуют друг с другом для обеспечения полной функциональности игры.



Рисунок 1 − Схема архитектуры системы

Ключевые компоненты программного продукта:

1. Основная сцена (scn\_main)

scn\_main (стартовая сцена): Это начальная сцена приложения, с которой начинается работа пользователя.

2. Game Manager

Game Manager: Это центральный компонент, отвечающий за управление состояниями игры. Он обрабатывает следующие состояния:

* Starting
* Loading
* Hangar
* Battle
* EndBattle

Для каждого состояния определяется свой набор сцен и окон.

3. Управляющие компоненты

Scene Manager: Управляет переключением между различными сценами в игре.

UI Manager: Управляет интерфейсными элементами, обеспечивая их правильное отображение и взаимодействие с пользователем.

Audio Manager: Управляет звуковыми эффектами и музыкой в игре.

4. Контексты

Hangar Context: Содержит ангарную сцену с набором ангарных окон. В зависимости от нажатых кнопок может менять состояние в Game Manager. Hangar Context обеспечивает переходы между разными функциями внутри ангара.

Battle Context: Содержит боевую сцену (scn\_battle) и отвечает за управление игровыми действиями. Включает следующие компоненты:

* Battle Controller: Основной контроллер игры.
* Act\_1, Act\_2, Act\_3: Различные этапы игры, которые могут включать разные сценарии или задачи.
* Test Manager: Управляет тестами, которые игрок должен пройти.
* Dynamic Compiler Roslyn: Инструмент для динамической компиляции кода, обеспечивающий возможность выполнения пользовательского кода на лету.

5. Dependency Injection

Zenject Framework: Фреймворк для внедрения зависимостей, используемый для упрощения управления зависимостями между компонентами и повышения модульности кода.

Логика взаимодействия пользователя с компонентами игры

 Пользователь взаимодействует с игрой через графический интерфейс и игровые элементы, обеспеченные различными компонентами приложения.

Приложение состоит из следующих компонентов:

* стартовый экран;
* 3 игровых акта;
* главный персонаж;
* вспомогательный персонаж-робот;
* модуль с тестом и теорией;
* модуль с задачей на кодирование и компилятором;
* таблица достижений;
* аудио и диалоги.

Рассмотрим основные этапы взаимодействия пользователя с игрой и как различные модули обеспечивают эти взаимодействия.

В начале игры пользователю нужно нажать кнопку «Начать игру» для того, чтобы приступить к игровому процессу. Игра ведется от третьего лица, игрок управляет главным персонажем. По ходу игры главного персонажа сопровождает вспомогательный персонаж-робот, который ведет его по локациям и рассказывает сюжет игры: суть сюжета в том, что главный персонаж проходит стажировку в IT-организации по производству роботов-андроидов, но в ней произошел сбой в системе, из-за чего производство нарушилось и двери заблокированы; для прохождения игры необходимо пройти через все локации, открыть двери, для чего нужно проходить задания с помощью знаний из теоретических материалов, добраться до главного сервера и исправить сбой в системе. Также робот-помощник объясняет часть обучающего материала.

На данный момент игра состоит из трех игровых локаций, каждая из которых соответствует своему ировому акту (предусмотрена возможность развития игры в будущем и добавление новых локаций): вход и склад, комната для сборки и испытаний роботов, серверная комната.

В первом акте, после начала игры, персонаж оказывается на первой локации. Его приветствует робот-помощник и вводит его в ход игрового сюжета, а также рассказывает часть теоретического материала по начальным основам ООП. Задание первого акта состоит в том, чтобы открыть дверь, пройдя тест по пройденному материалу. Теория также доступна в текстовом виде при выполнении задания.

Во втором акте, пройдя первую дверь, персонаж оказывается в комнате по производству роботов. Он видит за стеклом несколько экземпляров андроидов, они могли бы открыть следующую дверь, но вышли из строя. Задача второго акта в том, чтобы починить роботов, чтобы те открыли следующую дверь. Для этого нужно изучить теоретический материал и пройти задание с написанием кода: данное задание проверяет знания по разделу ООП – инкапсуляция.

Третий акт также подразумевает изучение материала и выполнение задания по программированию, в данном акте изучается материал по разделу ООП наследование. Локацией является серверная комната. При успешном выполнении, по сюжету игры, главная задача, – исправить сбой в системе, решена, и игра пройдена.

Таким образом, обучающий материал в игре включает в себя:

1. теоретические материалы:
* аудио сопровождение;
* текстовые материалы.
1. практические материалы:
* тесты на закрепление теории:
	+ вопрос с выбором одного ответа,
	+ вопрос с выбором нескольких ответов.
* задачи с автоматической проверкой кода.

Для тестовых вопросов созданы модели, содержащие в себе отметку о том, какой это тип вопроса (с одним правильным отвтеом или несколькими), варианты ответов с флагами верно или неверно. Вопросы выводятся в окне теста с полями для текста вопроса и ответов, адаптирующимися к количеству вариантов. Окно теста содержит также кнопку с вызовом карточки с теорией.

Для задач с автоматической проверкой кода реализована консоль со встроенным компилятором. На экране консоли есть поля для текста задачи и поле для ввода кода, которое может содержать уже подготовленную структуру кода для облегечения выполнения задачи и ориентации игрока. Также есть поле вывода результата компиляции, при отправке кода на проверку игрок видит результат прохождения им проверочных тестов.

Такая архитектура курса позволяет быстро наполнять раздел новыми темами и практическими тестами – например, чтобы создать новый тестовый вопрос нужно всего лишь добавить новую модель вопроса в соответствующую папку в редакторе Unity.

Таким образом, взаимодействие игрока с игрой построено на тесной интеграции обучающих материалов и игровых элементов, что позволяет эффективно совмещать процесс обучения с увлекательным игровым процессом.

1. Реализация системы

Текущее состояние игры

**Задачи в рамках проекта**

|  |  |
| --- | --- |
| Разработать ядро игры. | **Выполнено** |
| Сформировать и наполнить объектами локации. | **Выполнено** |
| Создать анимированный персонаж и робота-помошника. | **Выполнено** |
| Включить в игру основные обучающие блоки:теория, тест, написание кода с проверкой его работоспособности (встроенный компилятор). | **Выполнено** |
| Реализовать одну сюжетную линию состоящую из трёх актов. | **Выполнено** |
| Предусмотреть возможность для будущего развития игры. | **Учтено** |
| Встроенный функционал позволяет наполнять игру новым контентом: добавлять теоретический материал, новые тесты, создавать задания, предполагающие написание кода и проверку его работоспособности. |

**Основные составляющие игры**

|  |  |
| --- | --- |
| Менеджер сцен и механизм перехода между ними | **Есть** |
| Аудио менеджер, многопоточный | **Есть** |
| Менеджер окон и UI элементов | **Есть** |
| Менеджер заданий | **Есть** |
| Контроллеры дверей, роботов и персонажей | **Есть** |
| Механизм создания тестовых заданий и запуска тестов в игре | **Есть** |
| Механизм динамической компиляции и создания заданий, требующих написания и запуска кода | **Есть** |

Дальнейшее развитие игры

**Следующие полгода:**

* дополнительные уровни с продолжением сюжета;
* ветвление сюжета;
* авторизация пользователей в игре;
* система рейтинга игроков.

**Среднесрочные задачи:**

* возможность выбора персонажа мальчик / девочка;
* более детализированная анимация и озвучка;
* доработка компилятора для возможности публикации как WebGL приложения на сайте URFU.PRO, в VK, ОК.

**Долгосрочные задачи:**

* индивидуализация персонажа;
* мультиплеер для уровней с парным кодированием.

Заключение

В рамках проектного практикума была достигнута следующая цель – разработан прототип (MVP)[[1]](#footnote-1) образовательной игры на тему «объектно-ориентированное программирование».

В результате анализа предметной области были сформулированы технические требования к ключевому функционалу приложения, выделены требования к стеку технологий. Исходя из полученных требований был выбран наиболее подходящий стек.

На этапе проектирования были выделены основные сущности для обеспечения игрового процесса, определена их логика и разработана структура взаимодействия.

На этапе реализации игровые объекты были собраны в сцены и была сформирована сюжетная линия первых уровней игры. Был заполнен теоретический раздел, создан тест для проверки базовых знаний. В рамках второй части этапа в игру был встроен компилятор, что позволяет давать игрокам задания по написанию кода и проверять его работособность. Было сформулировано первое задание, требующее написания кода.

Проект полностью реализован на платформе Unity с использованием C# в качестве основного языка программирования.

Результат выполненной работы полностью соответствует поставленной цели – был разработан прототип (MVP) образовательной игры на тему «объектно-ориентированное программирование». Игра имеет сюжетную линию, в которую встроен теоретический материал для изучения, тест для проверки базовых знаний и задание, требующее написание кода на C#.

Из преимуществ продукта можно выделить следующее:

- единая линия игрового сюжета, связанные локации и повествование;

- персонаж-помощник, который объясняет материал и помогает с прохождением игры;

- концентрация внимания на отдельной фундаментальной теме программирования;

- изучение общих концепций программирования, а не только синтаксиса отдельных языков.

Недостатки продукта: игра находится на стадии прототипа. Реализован минимальный набор заданий. Игровой мир и персонаж собраны из объектов, находящихся в открытом доступе, нет индивидуализации персонажа и объектов. Сюжет не имеет разных концовок, не реализована система рейтингования игроков.

Игра готова к тестированию со стороны первых пользователей. Исходя из их поведения и интересов в нее будут добавлены новые материалы, направленные на более конкретные запросы, и новые разделы для расширения и разнообразия функционала. Также будут проведены мероприятия по публикации игры онлайн и адаптации к другим платформам.

1. MVP (minimal viable product) – минимально жизнеспособный продукт - продукт, обладающий минимальными, но достаточными для удовлетворения первых потребителей функциями. [↑](#footnote-ref-1)