Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего

образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Специальность (направление, профиль) Программная инженерия

Курсовой проект

защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Троицкий

(подпись руководителя проекта) (инициалы, фамилия)

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2024 г.

КУРСОВОЙ ПРОЕКТ

Разработка компьютерной игры-квест в жанре платформер

(тема курсового проекта )

Пояснительная записка

по дисциплине Программирование – 3 семестр

КП 09.03.04.1.000 ПЗ

(обозначение документа)

Студенты группы ПИ-22 Гартман С. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Карелин Н. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20.01.2024

Чепрасов А. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество) (подпись) (дата)

Руководитель

Проекта доцент, к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Троицкий

(должность, ученое звание) (подпись) (инициалы, фамилия)

БАРНАУЛ 2024

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный технический университет

имени И.И. Ползунова»

Факультет информационных технологий

Кафедра «Прикладная математика»

З А Д А Н И Е

на курсовой проект по дисциплине «Программирование»

студентам группы ПИ-22 Чепрасову Андрею Вячеславовичу, Гартману Степану Андреевичу, Карелину Никите Витальевичу

Тема курсового проекта: «Разработка компьютерной игры-квест в жанре платформер».

Календарный план работы:

|  |  |  |
| --- | --- | --- |
| № этапа | Содержание этапа | Недели семестра |
| 1 | Получение задания | 1 |
| 2 | Постановка задачи. Работа с документацией | 2 |
| 3 | Проектирование программы | 3-4 |
| 4 | Реализация программы | 5-13 |
| 5 | Оформление пояснительной записки | 14 |
| 6 | Защита курсового проекта | 15-16 |

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Троицкий В. С., доцент

подпись

Дата выдачи задания «04» сентября 2023 г.

число месяц год

Задание приняли к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гартман С. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Карелин Н. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чепрасов А. В.

подпись

**Содержание**

[Введение 1](#_Toc132842336)

[1. Обзор предметной области и постановка задачи 2](#_Toc132842337)

[1.1. Обзор предметной области 2](#_Toc132842338)

[1.2. Постановка задачи 2](#_Toc132842339)

[2. Проектирование 4](#_Toc132842340)

[2.1. Алгоритм решения 4](#_Toc132842341)

[2.2 Структура данных 4](#_Toc132842342)

[3. Реализация 5](#_Toc132842343)

[3.1. Выбор средств реализации 5](#_Toc132842344)

[3.2. Структура программы 5](#_Toc132842345)

[3.3. Состав программы 5](#_Toc132842346)

[Заключение 7](#_Toc132842347)

[Список использованных источников 8](#_Toc132842348)

[Приложение А Исходный код программы 9](#_Toc132842349)

[Приложение Б Результат работы программы 10](#_Toc132842351)

Введение

В настоящее время разработка видеоигр стала одной из наиболее популярных областей программирования. Создание гейм-движков, написание игр и управление её ресурсами - всё это требует глубокого понимания своего дела и не обходится без применения объектно-ориентированного программирования (ООП)

Цель данного курсового проекта - разработать 2D пиксельную приключенческую видеоигру с использованием принципов ООП. Этот проект предоставит возможность применить умения и знания ООП на практике, создавая игровые объекты, персонажей, сцены и игровую логику.

В данном отчете будут рассмотрены основные этапы разработки игры, включая проектирование классов, представленных в используемом движке блюпринтами (от слова blueprint), их взаимодействие и реализацию игровых механик.

Использование ООП в разработке видеоигр позволяет упростить поддержку кода, повысить его читаемость и расширяемость, что делает данную тему актуальной и интересной для изучения и практической реализации.

1. Обзор предметной области и постановка задачи
   1. Обзор предметной области

*Игровой движок* — базовое программное обеспечение компьютерной игры. Разделение игры и игрового движка часто расплывчато, и не всегда студии проводят чёткую границу между ними. Но в общем случае термин «игровой движок» применяется для того программного обеспечения, которое пригодно для повторного использования и расширения, и тем самым может быть рассмотрено как основание для разработки множества различных игр без существенных изменений. Он предоставляет все инструменты и материалы, необходимые для проработки игрового мира и его последующего функционирования, поэтому серьезно экономит время и деньги.

Игровой движок в играх отвечает практически за всё. Благодаря работе игрового движка на экране не только двигаются объекты, но и происходит расчет освещения, влияние действий на игровой мир, обработка команд, поступающих с контроллера игрока, переключение музыки и множество других действий.

Движок игры работает на настолько глубоком уровне, что именно он обеспечивает появление на экране необходимых моделей персонажей, задает им анимации и поведение, а также включает озвучку, музыку и звуковые эффекты, соответствующие происходящим событиям. Часто игровые движки используют наравне с обычными языками программирования ещё и визуальные.

*Блочное кодирование -* это язык программирования, который позволяет пользователям создавать программы, манипулируя программными элементами графически, а не указывая их как текст.

*Блюпринт*— система визуального программирования в Unreal Engine 4 и Unreal Engine 5 на основе нодов с данными: событиями и функциями. Их можно связывать между собой и формировать элементы геймплея. Различают несколько видов блюпринтов, рассчитанных на определённые задачи — от создания события на уровне до интерфейсов и макросов, которые можно использовать как основу для другого блюпринта.

Блюпринты не полная альтернатива программированию на C++. Всё зависит от самого проекта и возможностей разработчика. Кто-то делает игры целиком на C++, другие используют только блюпринты. Иногда применяется комплексный подход: прототип собирают на блюпринтах, а непосредственно сам проект пишут на C++.

В дискуссиях программисты часто упоминают, что проекты на C++ получаются более оптимизированными и в них проще делать отладки. К тому же рефакторинг блюпринтов реализован в несколько раз хуже, что критично для масштабных проектов.

1.2. Постановка задачи

Разработать игру в жанре квест с видом сбоку и пиксельной графикой. В игре представлена одна многоуровневая локация – общежитие с интерактивным внутренним миром.

Игра должна включать в себя:

1) Игрового персонажа, которым будет управлять игрок и проходить основной квест, также у персонажа должны быть сделаны анимации, чтобы картинка не выглядела статичной.

2) Игровая локация, которую будет исследовать игрок: в нашем случае – общежитие и его постоянные жильцы. Общежитие не должно быть пустым, иначе игроку будет неинтересно исследовать локацию.

3) Понятный интерфейс, чтобы игрок мог легко ориентироваться в игровом меню и побочных меню, которые будут в игре.

4) Полноценный квест, иначе игроку будет нечего делать на локации и пропадёт смысл исследовать её.

5) Защита от неожиданных сценариев, иначе игрок не поймёт изначальной задумки и пропустит весь сюжет.

Цель состоит в том, чтобы показать человеку, который не жил в общежитии, ситуации, с которыми может столкнуться первокурсник, который только заселился в общежитие. Наша игра погружает в дружественную атмосферу студенческого общежития, показывает важность помощи и общения в жизни человека.

2. Проектирование

2.1. Алгоритм решения

Программа работает в оконном и полноэкранном режимах, является полноценным приложением.

При входе в программу пользователь видит главное меню в оконном режиме, где он осуществляет выбор для дальнейшей работы:

* Начать играть;
* Полноэкранный режим;
* Выход.

При выборе меню «Начать играть», осуществляется переход непосредственно к самой игре.

При выборе меню «Полноэкранный режим» программа меняет режим, переходит из оконного в полноэкранный режим.

При выборе меню «Выход», вызывается меню, в котором у пользователя спрашивается, хочет ли он покинуть игру и предоставляется выбор:

* Да;
* Нет.

Соответственно, при выборе ответа «Да», программа завершает свою работу, при выборе ответа «Нет», программа возвращает пользователя в главное меню.

Как сказано было выше, при выборе в главном меню «Начать играть», осуществляется переход непосредственно к самой игре. Игра представляет собой многоуровневую локацию “Общежитие” с визуальным интерфейсом и открытым для взаимодействия миром. Пользователь является основным героем игры, которому необходимо взаимодействовать со второстепенными персонажами для прохождения игры. Целью игры является знакомство с жильцами общежития (второстепенными персонажами) посредством главного квеста, задачей которого является сбор подписей. Взаимодействие происходит путем общения главного и второстепенных персонажей.

При отказе выполнения квеста, пользователь досрочно заканчивает прохождение и перед ним появляется меню с выбором:

* Начать заново;
* Выйти в меню.

При выборе меню «Начать заново», осуществляется перезапуск игры, и пользователь начинает прохождение сначала. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

При успешном диалоге персонажей, пользователю открывается доступ к главному квесту по сбору подписей. Во время прохождения квеста, пользователь может рассмотреть обстановку общежития, познакомиться с его жильцами и соответственно выполнит главный квест.

Перемещение между этажами локации осуществляется через меню, которое вызывается при подходе к определенной двери в конце этажа с помощью клавиши Enter. В данном меню пользователь может сделать выбор, куда ему отправиться “вверх” или “вниз”, в зависимости от того, на каком этаже находиться пользователь.

Пользователь может в любой момент игрового процесса вызывать меню паузы нажатием клавиши Tab, в котором предоставляется выбор:

* Вернуться в игру;
* Выйти в меню.

При выборе меню «Вернуться в игру», пользователь возвращается в игру. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

Как говорилось выше, основной целью игры является выполнение главного квеста. При успешном выполнении данного квеста, пользователь заканчивает прохождение игры. Перед пользователем открывается победное меню с поздравлением и секретной наградой, как и в других меню, пользователю доступен выбор:

* Начать заново;
* Выйти в меню.

При выборе меню «Начать заново», осуществляется перезапуск игры, и пользователь начинает прохождение сначала. Соответственно, при выборе «Выйти в меню», пользователь попадает в главное меню.

При выборе определённой функции программа ждёт выполнения условий от пользователя и даёт соответствующую реакцию.

2.2 Структура данных

Для решения поставленной задачи реализованы следующие модули:

* MainTileMap – Карта, построенная из плиток;
* MainHero – Главный персонаж, которым управляет игрок;
* Menu – Карта меню запуска игры;
* MainMap – Игровая карта, основная часть игры;
* 2DSideScrollerGameMode – Игровой режим, который хранит информацию о игроке
* BT\_CharacterResponse – Класс, отвечающий за ответы игрока;
* BT\_EndDialog – Класс, отвечающий за выход из диалога
* BT\_NPCSpeak – Класс, отвечающий за речь второстепенных героев
* BT\_ResponseDecorator – Класс, отвечающий за проверку ответа героя
* Quest\_Item – Игровой предмет, который игрок должен поднимать
* Parent – Родительский класс для неиграбельных героев

3. Реализация

3.1. Выбор средств реализации

Данное ПО реализовано на языке Blueprint с использованием игрового движка Unreal Engine 4.27.2. Blueprint — система визуального программирования в Unreal Engine 4 и Unreal Engine 5 на основе нодов с данными: событиями и функциями.

Для полноценного функционирования программы требуется:

* ОС: Windows 10 версии 1909 или новее, Windows 11 версии 21Н2 или новее;
* Оперативная память: не менее 8 ГБ ОЗУ;
* Жёсткий диск: не менее 20 ГБ;
* Процессор: минимальное число ядер – 4, 2,5 ГГц или выше;
* Видеоадаптер: DirectX 11 или DirectX 12 совместимые графические карты;
* Наличие клавиатуры, мыши, монитора.

3.2. Структура программы

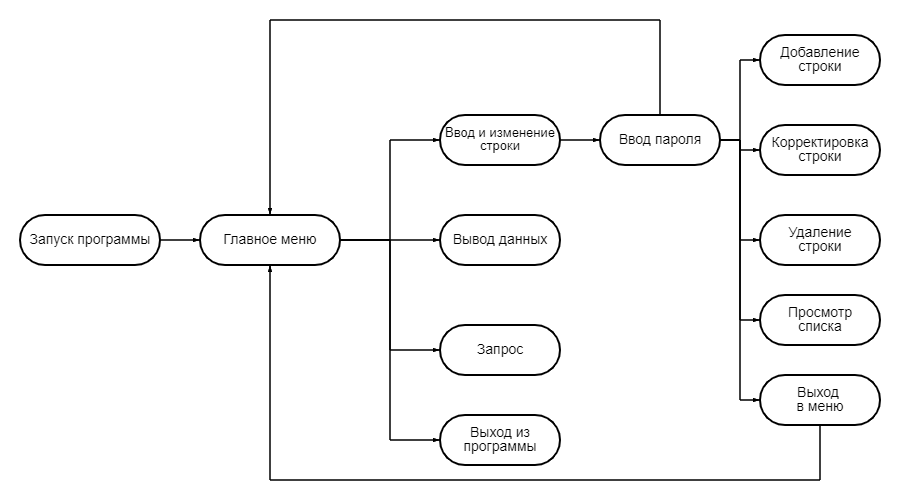


Рисунок 1 – Структура программы в виде иерархии модулей

3.3. Состав программы

Модули (функции) программы:

* *clean* – функция очистки входного потока
  + Прототип: int clean();
* *vvod –* функция для входа в меню ввода и изменения строки;
  + Прототип: int vvod();
* *vivod –* функция вывода базы данных на экран;
  + Прототип: int vivod();
* *vvod\_stroki –* функция для добавления строки в базу данных;
  + Прототип: int vvod\_stroki();
* *change\_str –* функция для корректировки отдельного элемента строки;
  + Прототип: int change\_str();
* *delete\_str* - функция для удаления строки из базы данных;
  + Прототип: int delete\_str();
* *ask* – функция для выполнения поиска по базе данных;
  + Прототип: int ask();
* *password* – функция для ввода пароля администратора;
  + Прототип: int password();

Заключение

В ходе работы был создан программный продукт, который позволяет эффективно работать с базой данных автовокзала после изучения документации. Реализованы защиты на ввод некорректных данных, несуществующих дат и времён, создание файла при его отсутствии, защита от чтения файла при его отсутствии , эффективная обработка запроса в реальном времени, а также добавление, изменение и удаление строк базы данных.

В процессе выполнения задания были подробно изучены методы работы с файлами в языке Си.

Преимущества приложения:

* Понятный для пользователя интерфейс;
* Быстрая и эффективная обработка запроса;
* Возможность редактирования/добавления/удаления записей для администраторов;
* Минималистичный дизайн приложения;
* Проверка файла перед его использованием;
* Создание файла для записи при его отсутствии.

Недостатки приложения:

* Отсутствие выбора расположения и имени файла;
* Консольный режим работы приложения;
* При повреждении файлов их необходимо создавать заново;
* Пароль администратора хранится в незашифрованном виде в тесте программы;

Возможны следующие усовершенствования приложения со временем:

* Работа с несколькими базами одновременно;
* Запись пароля в зашифрованном виде в отдельном файле;
* Возможность выбора имени файла и его расположения;
* Сохранения результатов запросов в отдельные файлы;
* Вывод списка файлов;
* Разработка визуального интерфейса;
* Добавление функций работы с базой данных.

Список использованных источников

1. Егорова Е.В. Программирование на языке СИ. Учебное пособие/Алт. Госуд. Технич. Ун-т им. И.И.Ползунова.- Барнаул: 2013. - 184 с.
2. Прата, Стивен. Язык программирования С. Лекции и упражнения, 6-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2015. — 928 с. : ил. - Парал, тит. англ

3. Керниган Б., Ритчи Д. Язык программирования Си = The C programming language. — 2-е изд. — М.: Вильямс, 2007. — 304 с.

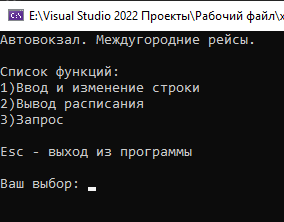
Приложение А

Исходный код программы

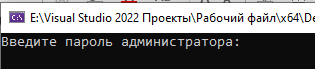
Приложение Б

Результат работы программы

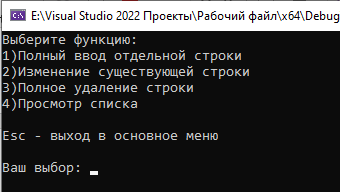
* Главное меню



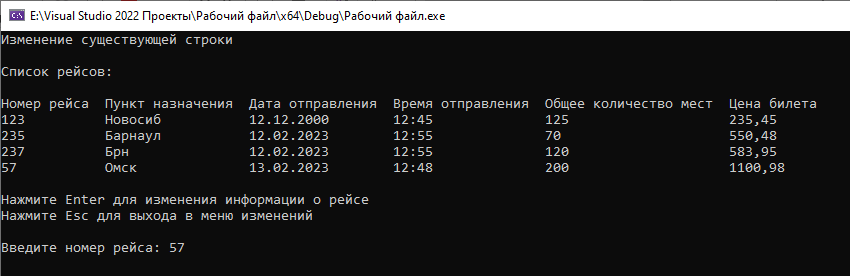
* Ввод пароля администратора

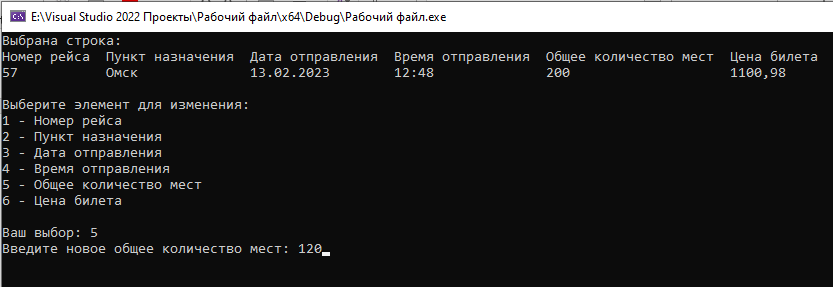


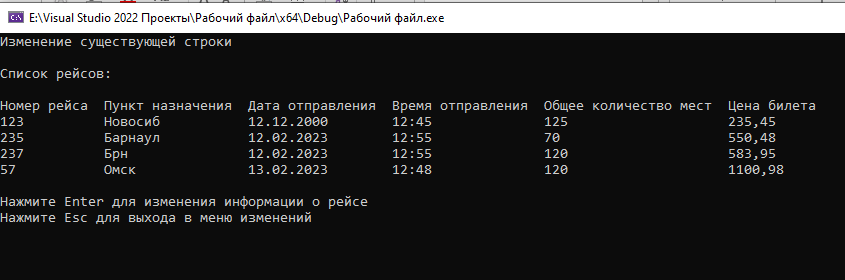
* Ввод и изменение строк



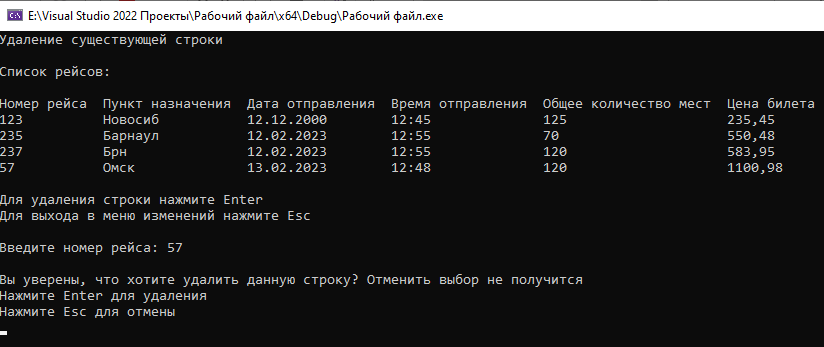
* Изменение строк

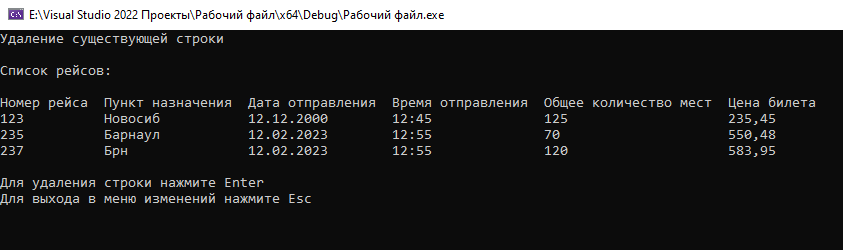




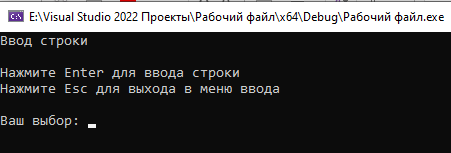


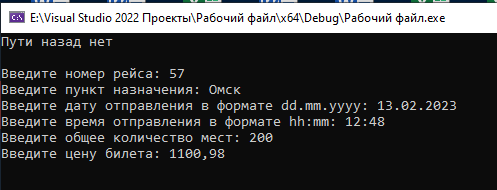
* Удаление строк



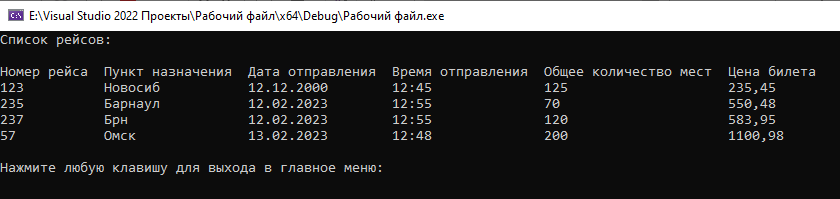


* Добавление строк





* Вывод данных



* Поиск по запросу

