Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Петрозаводский государственный университет»

Физико-технический институт

Кафедра информационно-измерительных систем и физической электроники

Курсовой проект по дисциплине «Технология программирования»

Авторы работы:

студенты группы 21312

Мирон Алексей

Пушко Денис

Пименов Артем

Кобылин Дмитрий

Научный руководитель:

канд. физ.-мат. наук

А.В. Бульба

Петрозаводск 2023

Содержание

[**Цель работы.** 3](#_Toc136532471)

[**Программная реализация.** 3](#_Toc136532472)

[**Пошаговое описание процесса разработки.** 3](#_Toc136532473)

[**Краткое словесное описание сюжета.** 3](#_Toc136532474)

[**Список вариантов использования.** 3](#_Toc136532475)

[**Описание вариантов использования.** 4](#_Toc136532476)

[**Список существительных** 4](#_Toc136532477)

[**Диаграмма классов.** 5](#_Toc136532478)

[**Заголовочные файлы.** 5](#_Toc136532479)

[**Исходные файлы.** 7](#_Toc136532480)

[**История проекта на GitHub.** 34](#_Toc136532481)

[**Заключение.** 35](#_Toc136532482)

# **Цель работы.**

Используя возможности языка программирования С++ (IDE Qt Creator (Qt 5.4.2)) и библиотеки (Simple and Fast Multimedia Library) написать простую 2D игру.

В качестве примера был выбрана игра Ведьмак 4 в стиле средневековой РПГ с видом сверху.

# **Программная реализация.**

Данная программа написана в среде разработки Qt Creator (Community) с использованием языка программирования С++ и библиотеки SFML. В результате работы были сделаны следующие заголовочные и файлы их реализации:

1. map.h – в данном заголовочном файле содержится карта 1 комнаты для игры «Ведьмак 4»;
2. map1.h – в данном заголовочном файле содержится карта 2 комнаты для игры «Ведьмак 4»
3. main.cpp – в данном файле содержится основная логика игры.

## **Пошаговое описание процесса разработки.**

### **Краткое словесное описание сюжета.**

Молодой авантюрист-ведьмак просыпается на кровати в темном подземелье и видит старого странного мужчину. Он говорит, что ему необходимо принести каменное сердце с сундука, а также победить монстра, чтобы они могли выйти наружу. Ведьмак обладает силой огня, может выпускать фаерболы, а также может подбирать с земли камни и метаться ими во врагов.

### **Список вариантов использования.**

На рисунке 1 представлена диаграмма вариантов использования.

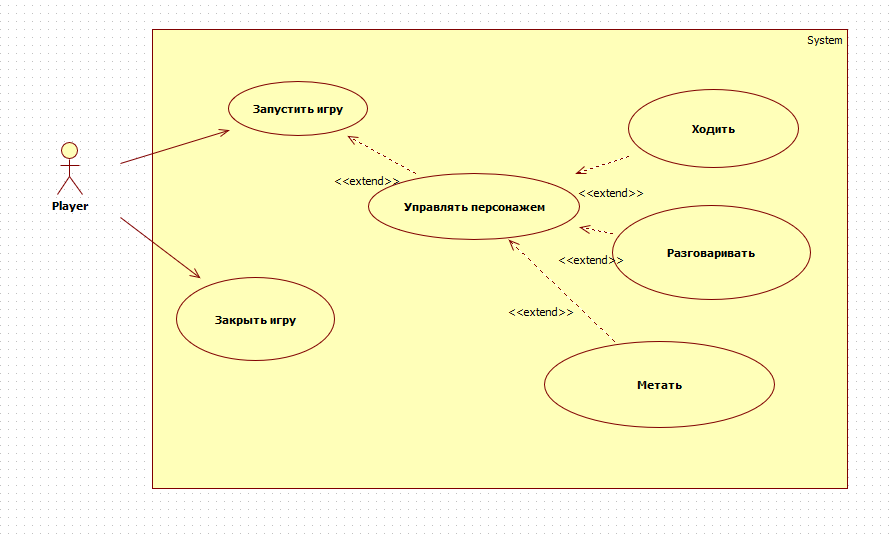


Рис. 1. Диаграмма вариантов использования.

### **Описание вариантов использования.**

- Описание варианта использования «Запустить игру».

Когда пользователь запускает игру, на экране появляется уровень игры, где пользователь может начать управлять персонажем.

- Описание варианта использования «Управлять персонажем».

После загрузки уровня и появление игрока и врагов, игрок может начать управлять персонажем.

- Описание варианта использования «Ходить».

Игрок может перемещать персонажа с помощью WASD.

- Описание варианта использования «Разговаривать».

Игрок может слушать слова старика, пролистывая диалог с помощью Space

- Описание варианта использования «Метать».

Игрок может метать фаерболы и камни с помощью P и выбирать, что конкретно метать с помощью I.

- Описание варианта использования «Закрыть игру».

Пользователь в любое время может нажать красный крестик в правом верхнем углу окна, и игра закроется.

### **Список существительных**

Класс Entity – базовый класс, от которого наследуются классы Player, Enemy, Bullet, Boss, Stone и Effect.

Класс Player – класс игрока. Содержит в себе состояния персонажа, его координаты, здоровье, ману, состояние(жив ли), направление движения и текстуры.

Класс Enemy – класс врага. Содержит в себе состояния врага, его координаты, состояние, направление движения и текстуры.

Класс Bullet – класс пули. Содержит в себе координаты на игровом поле, направление полета пули.

Класс Boss – класс босса. Содержит в себе состояния босса, его здоровье, координаты, состояние, направление движения и текстуры.

Класс Stone - класс камня. Содержит в себе координаты на игровом поле, направление полета камня.

Класс Effect - класс камня. Содержит в себе координаты на игровом поле, время жизни эффекта.

### **Диаграмма классов.**

Диаграмма классов для игры «Ведьмак 4» представлена на рисунке 2.

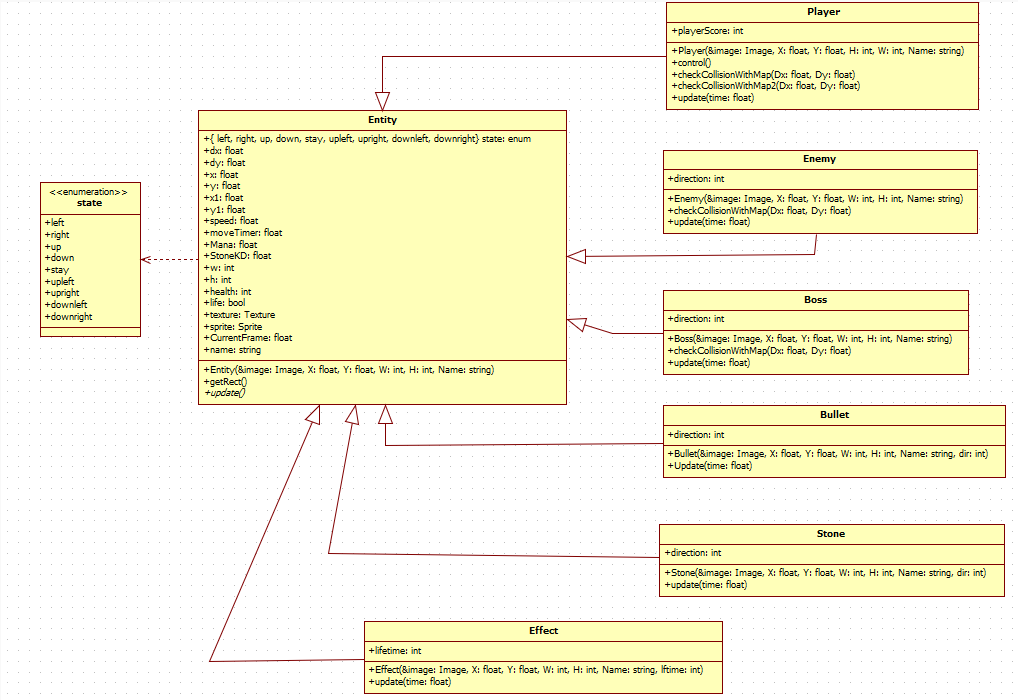
****

Рис. 2. Диаграмма классов для игры «Ведьмак 4»

### **Заголовочные файлы.**

Код заголовочного файл map.h представлен на рисунке 3.

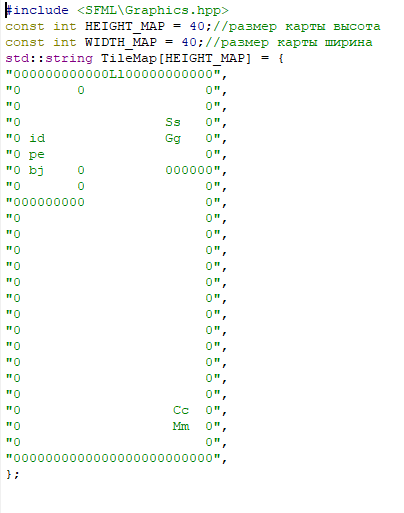


Рис. 3. Код заголовочного файла map.h

Код заголовочного файл map1.h представлен на рисунке 4.

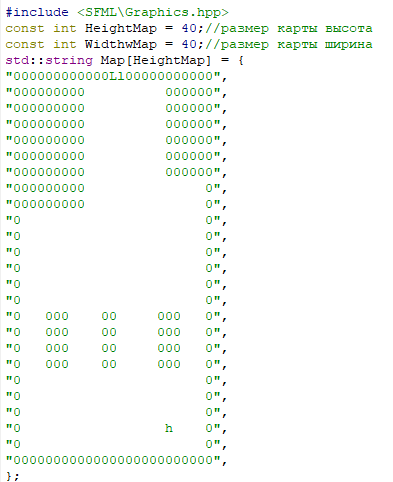


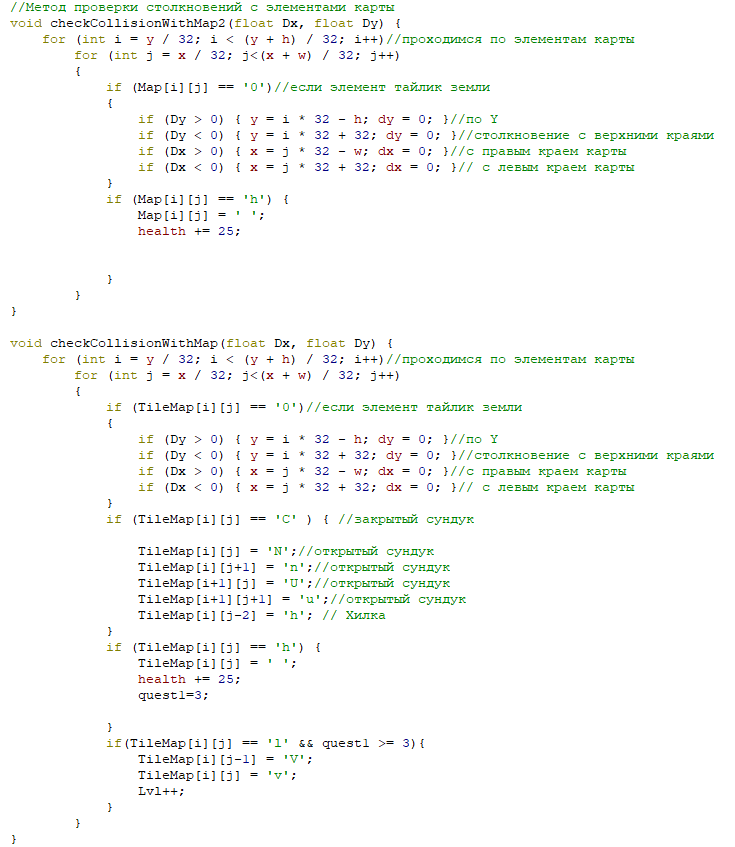
Рис. 4. Код заголовочного файла map1.h.

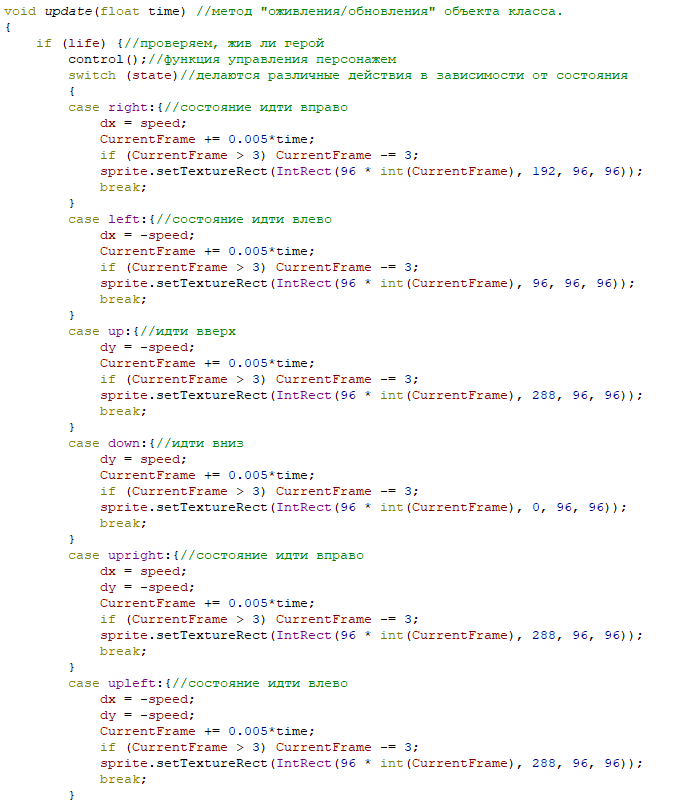
### **Исходные файлы.**

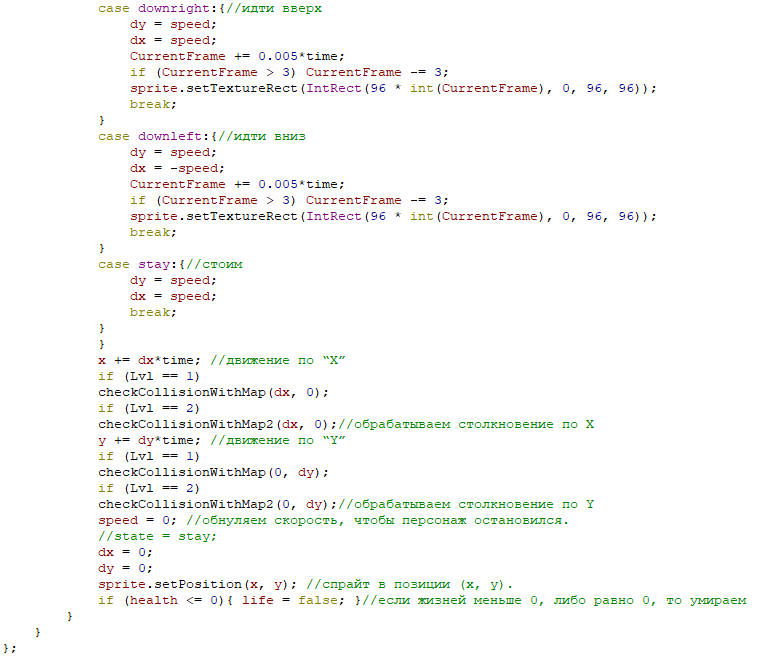
Исходный код файл main.cpp представлен на рисунке 5.











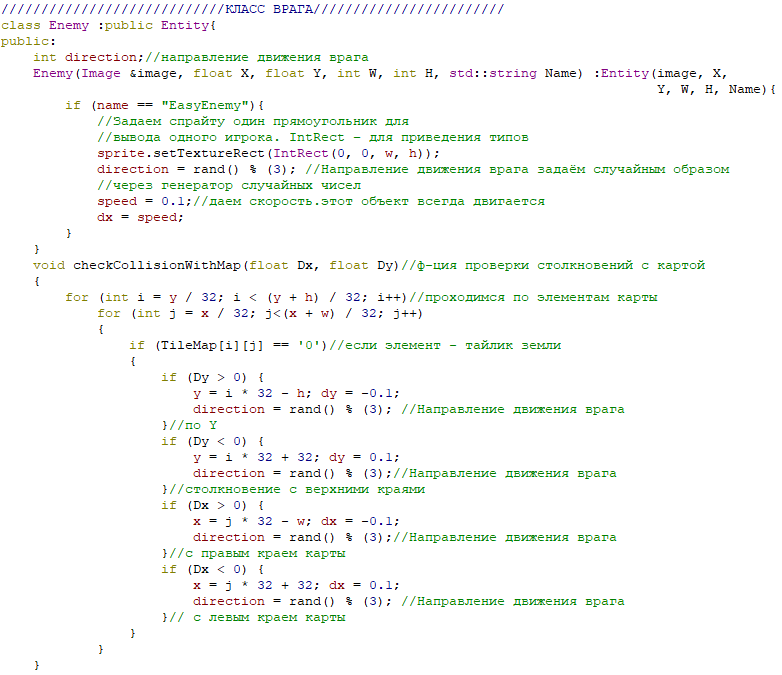
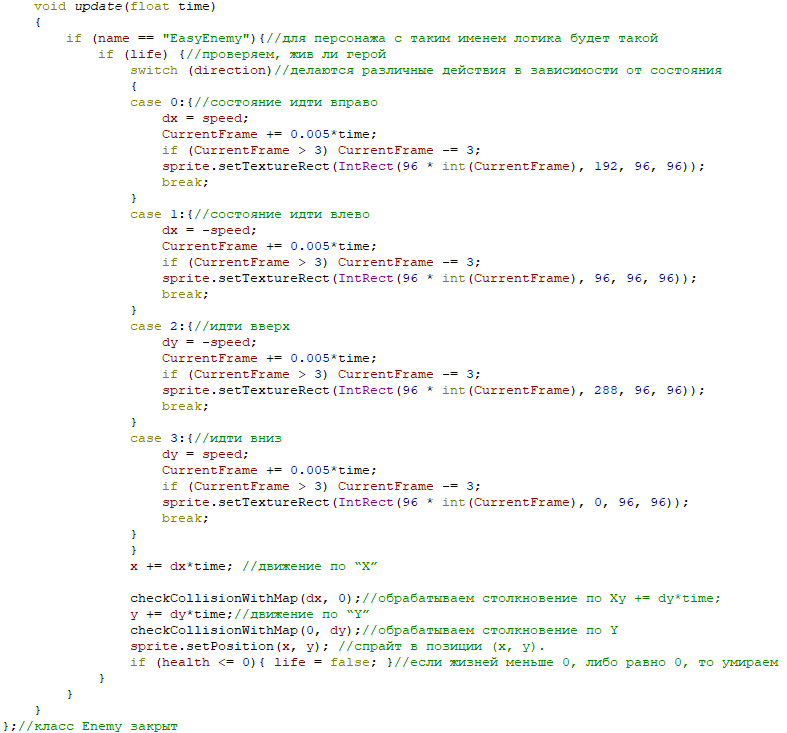
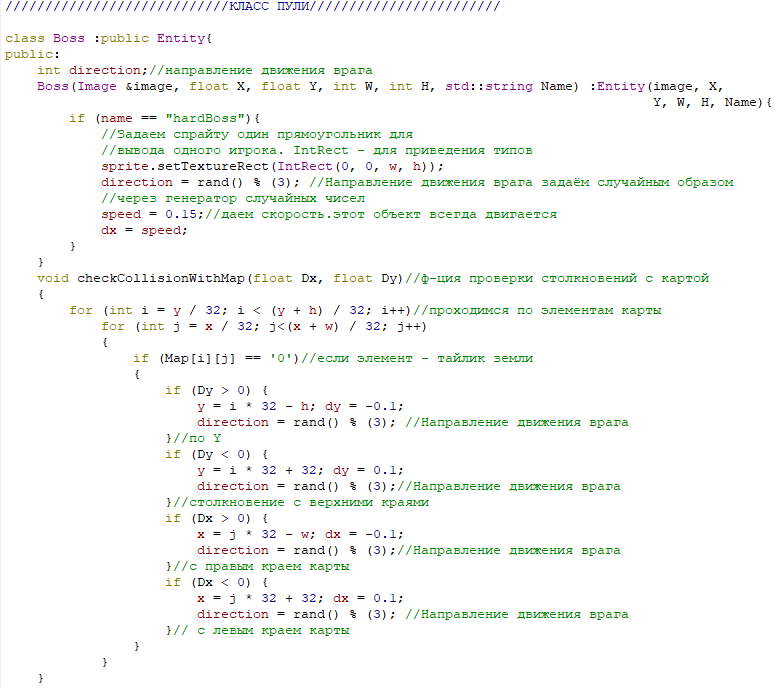
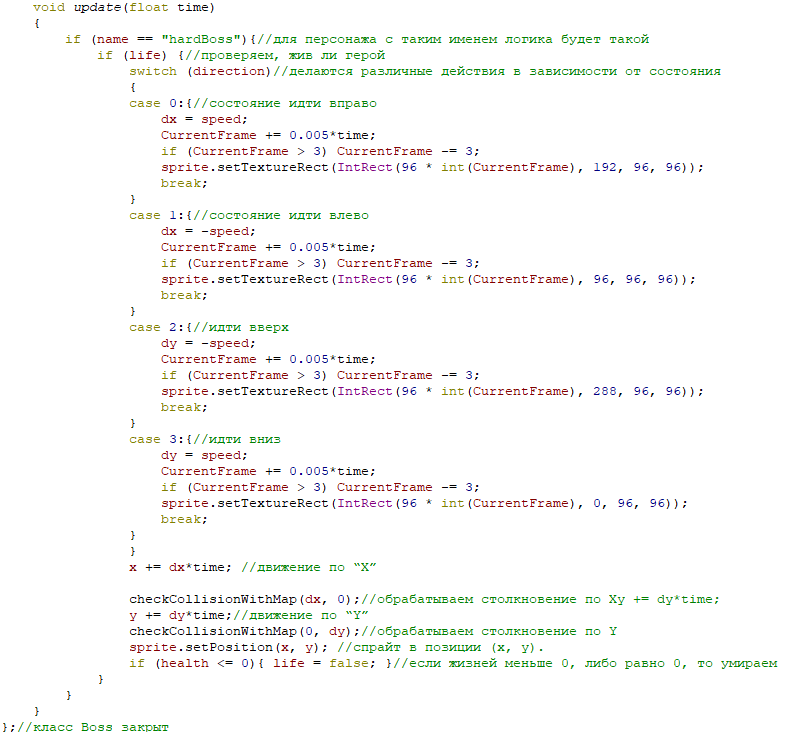
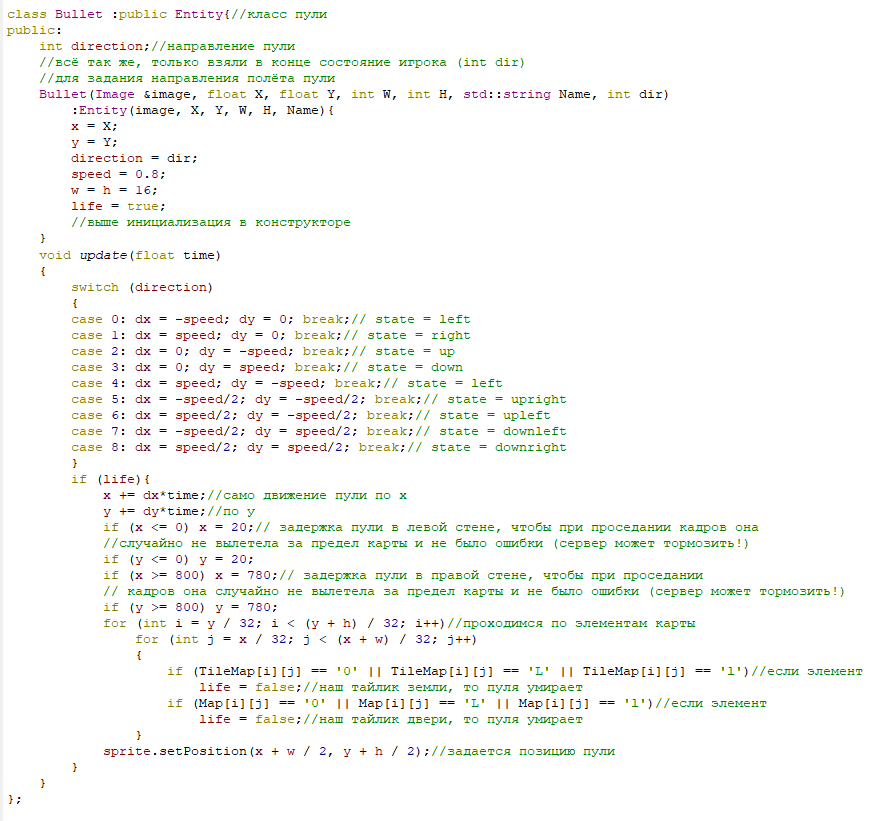
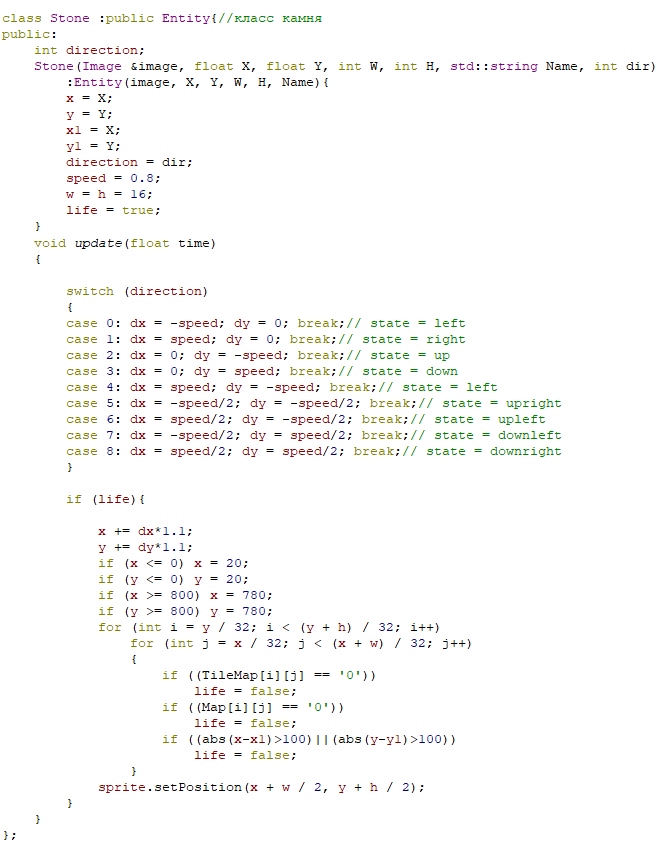
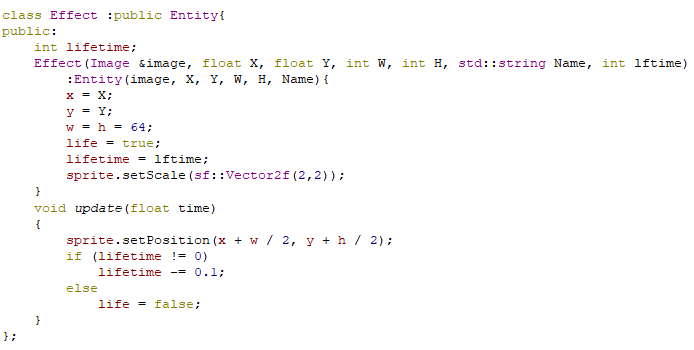
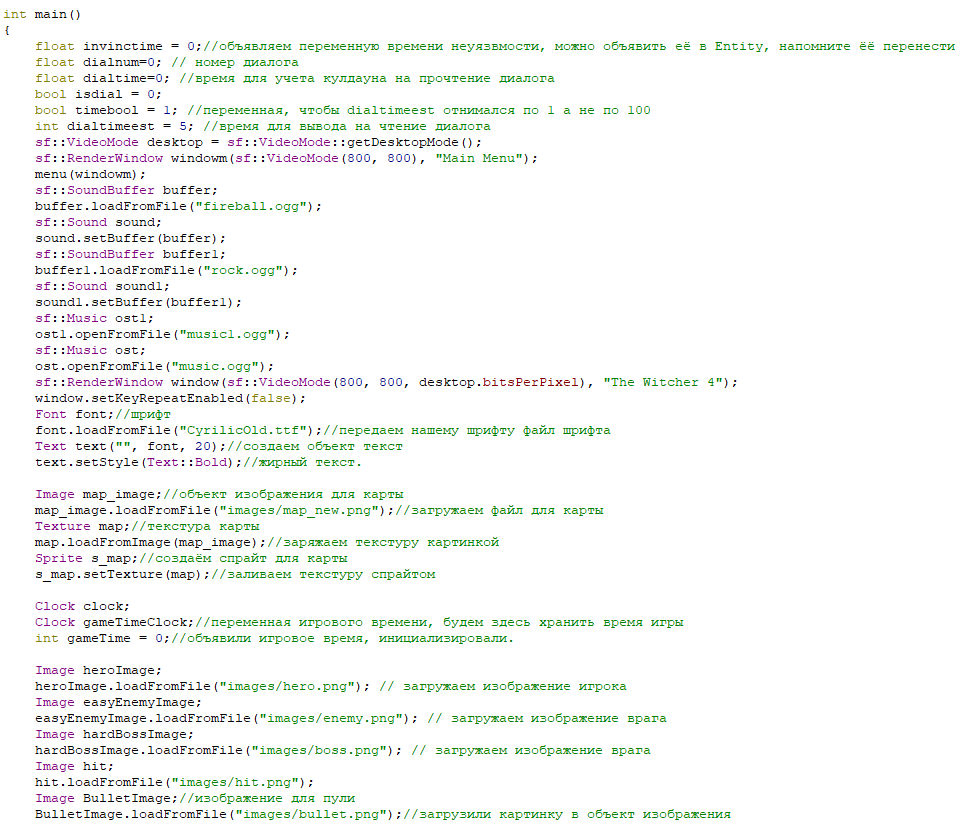
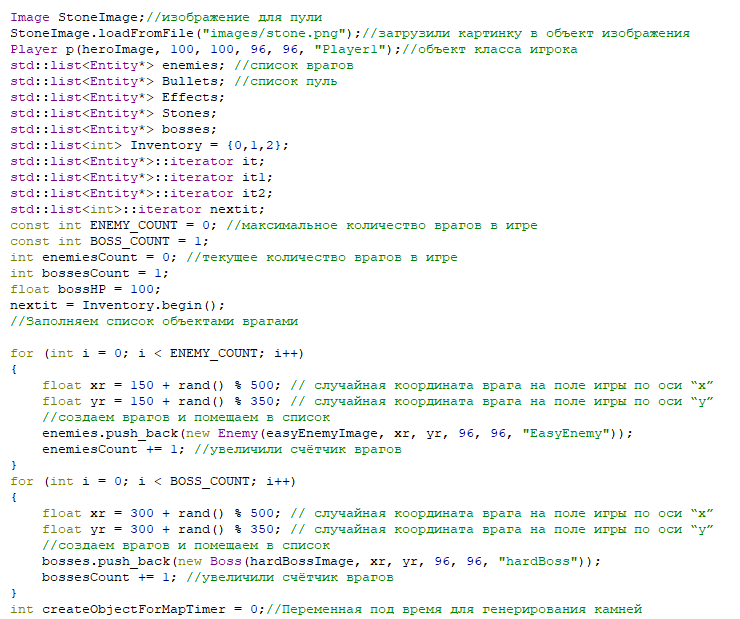
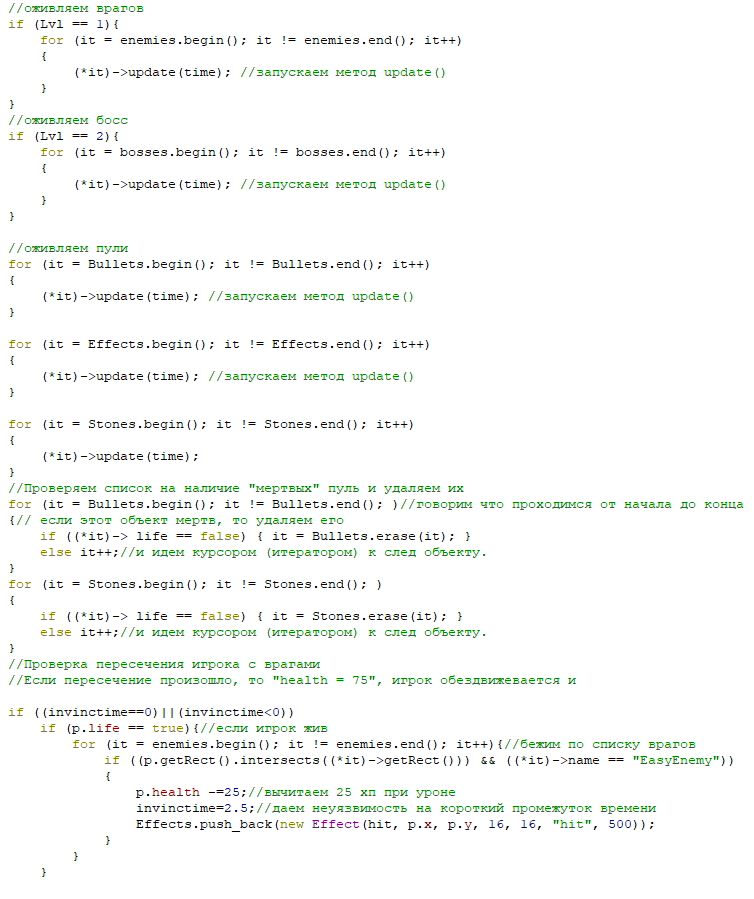
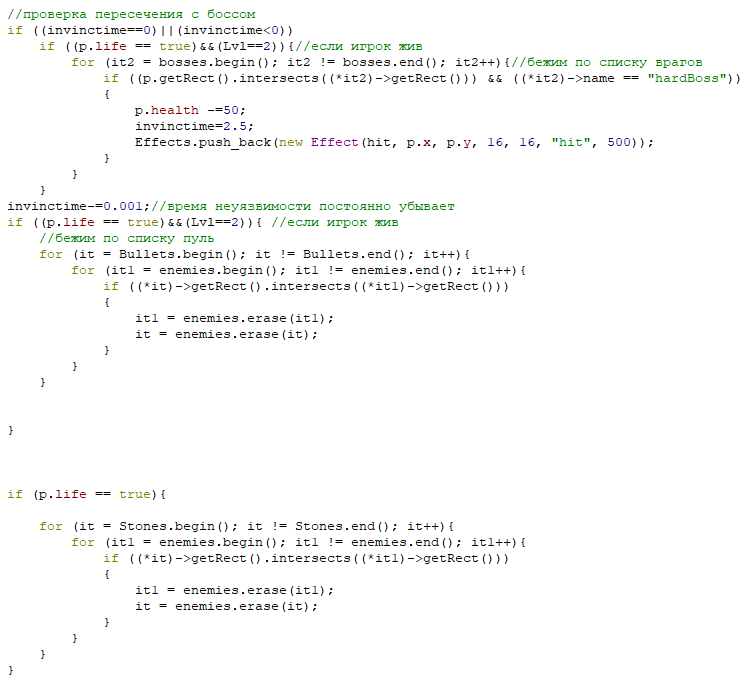
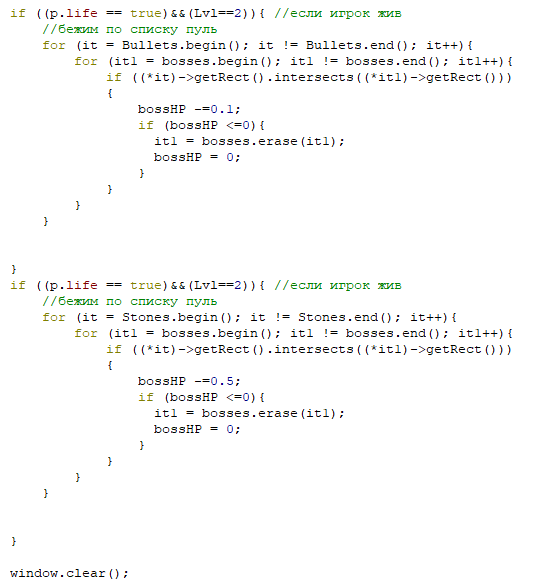
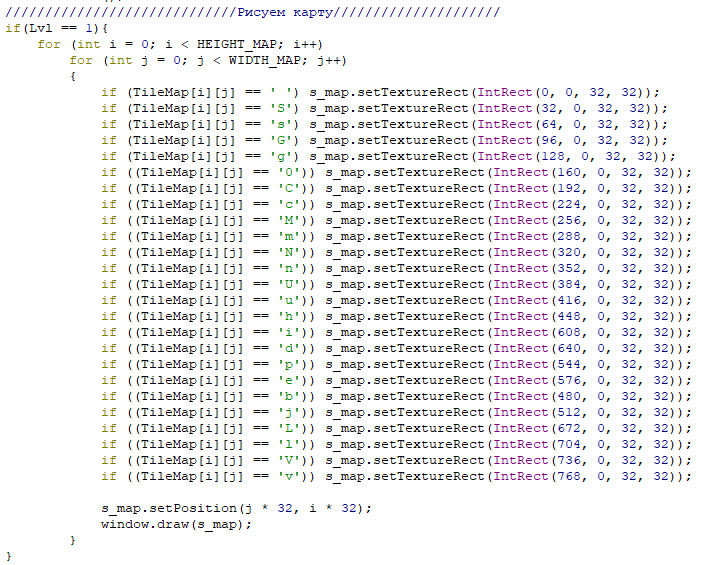
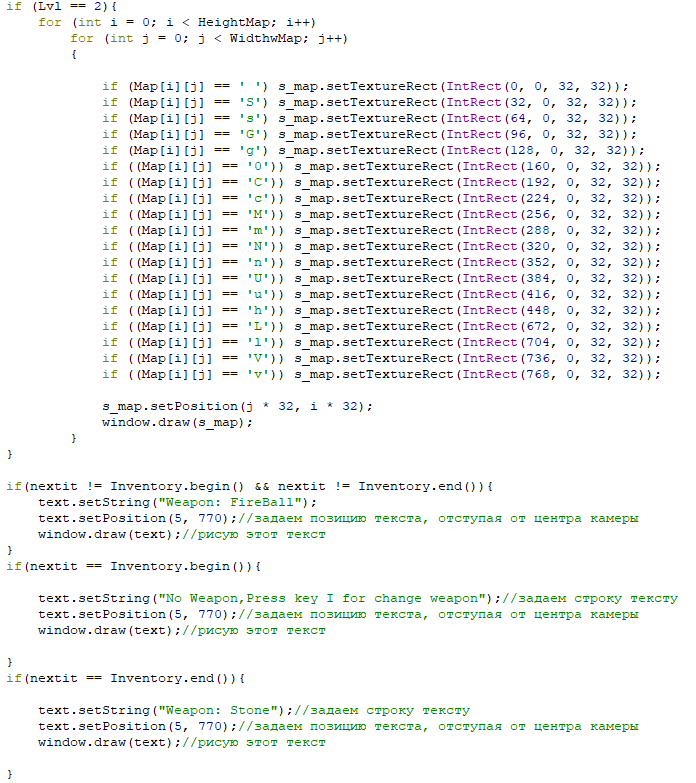
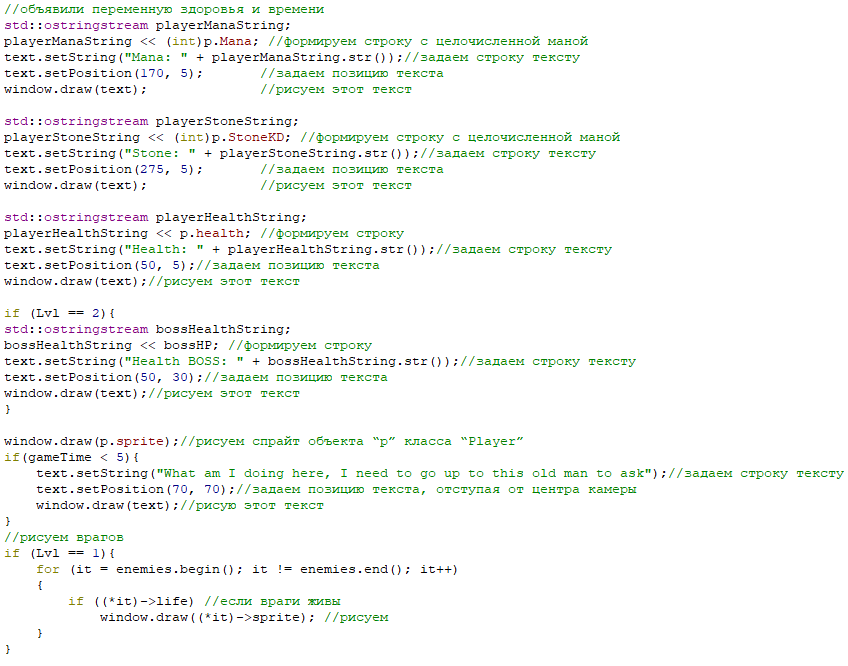
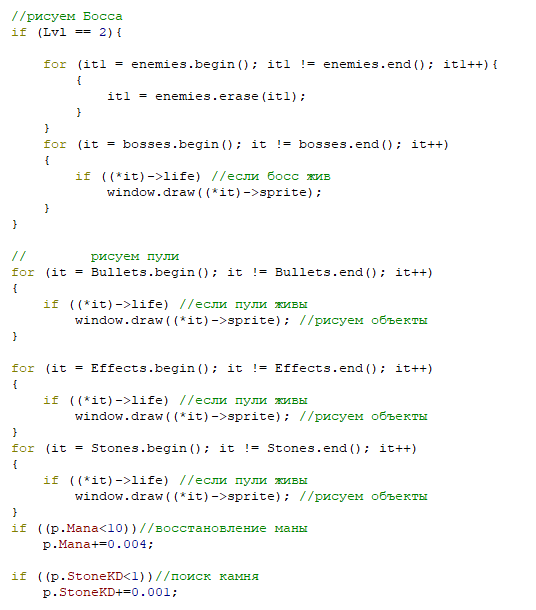
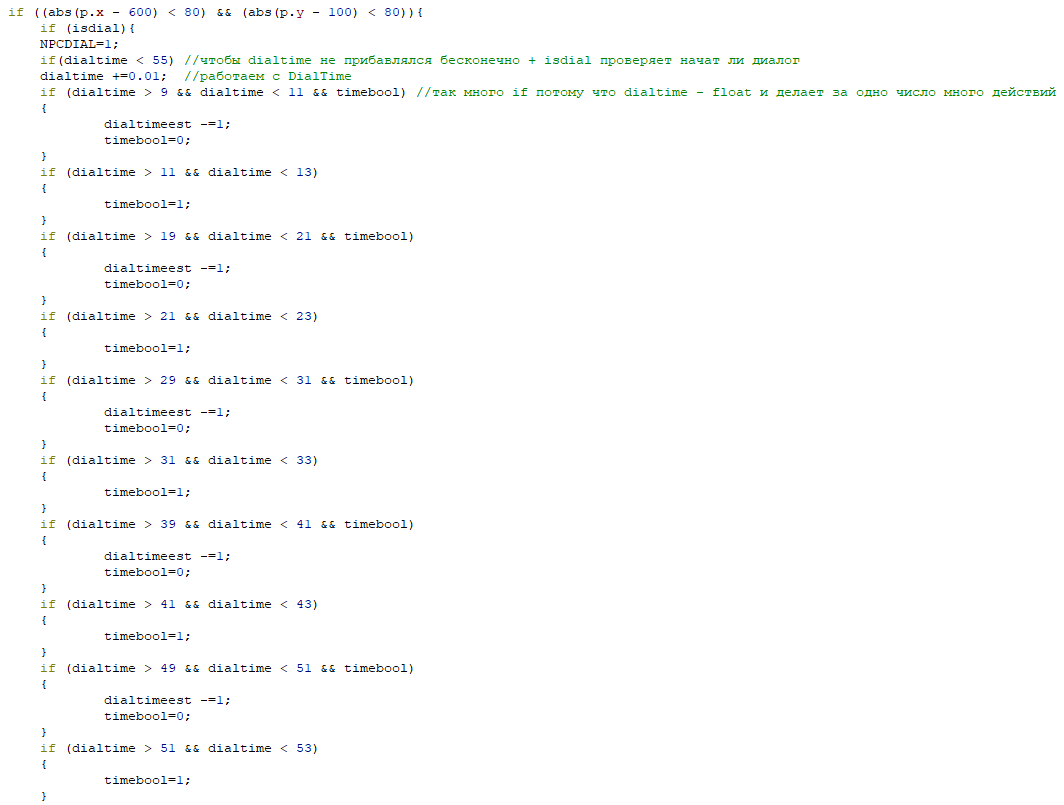
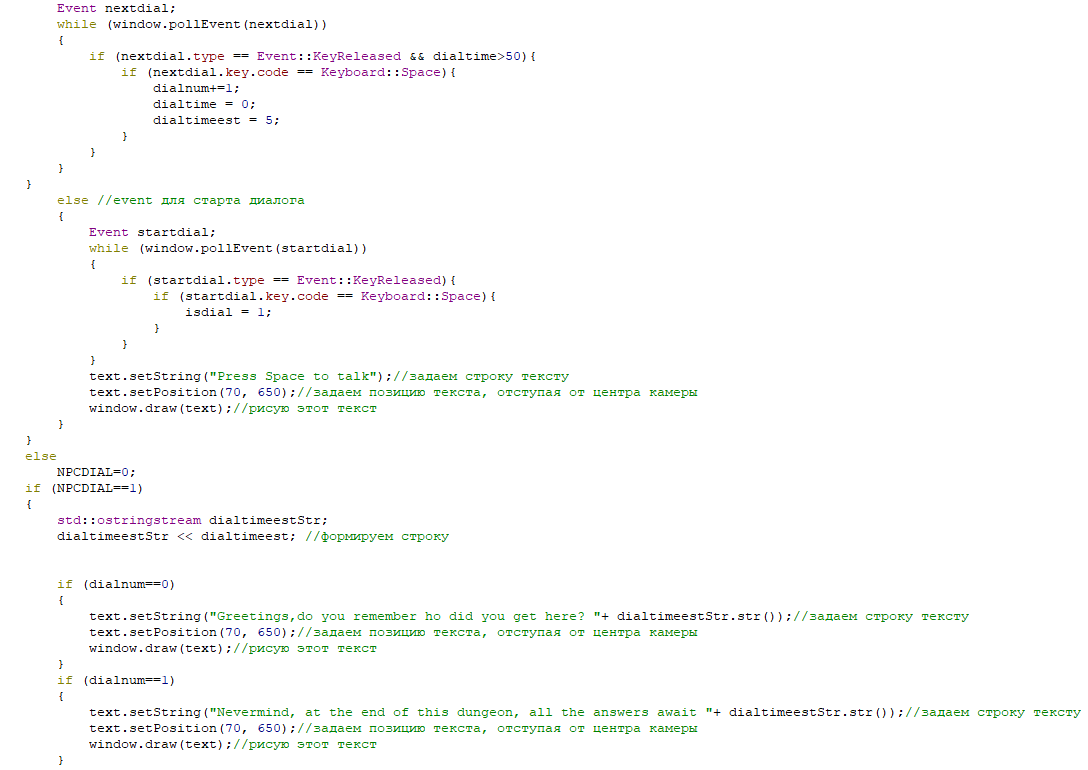
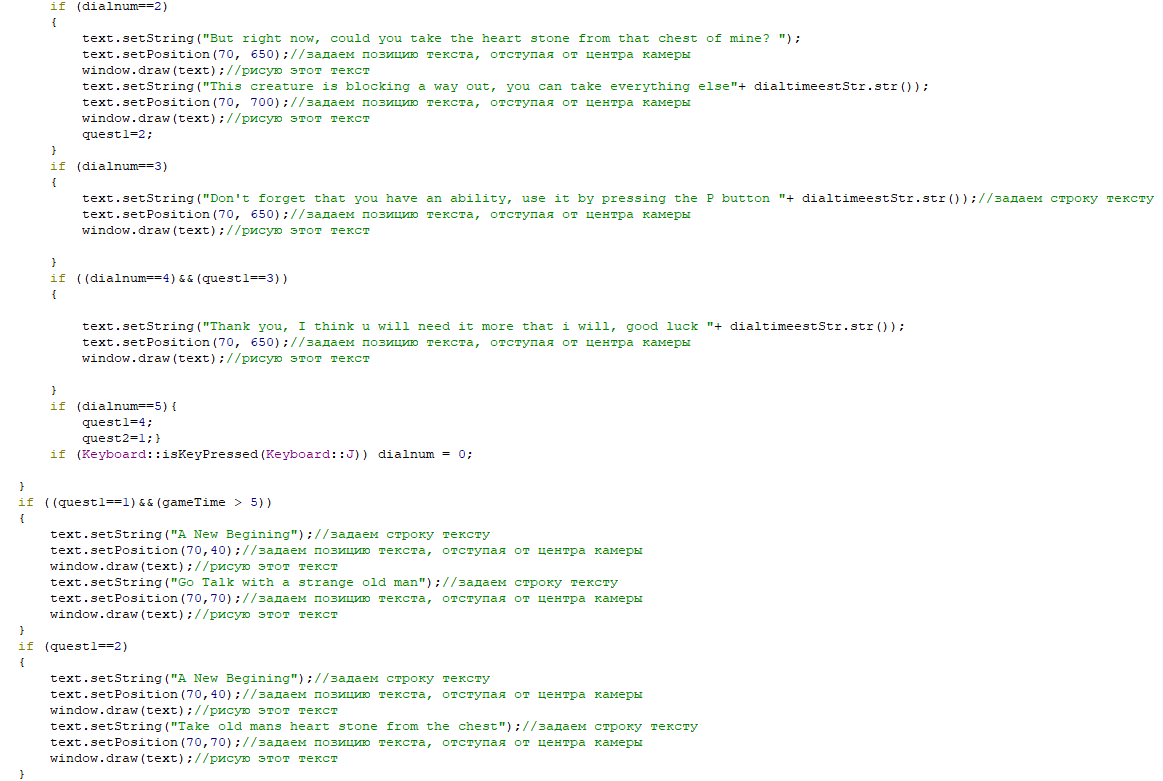
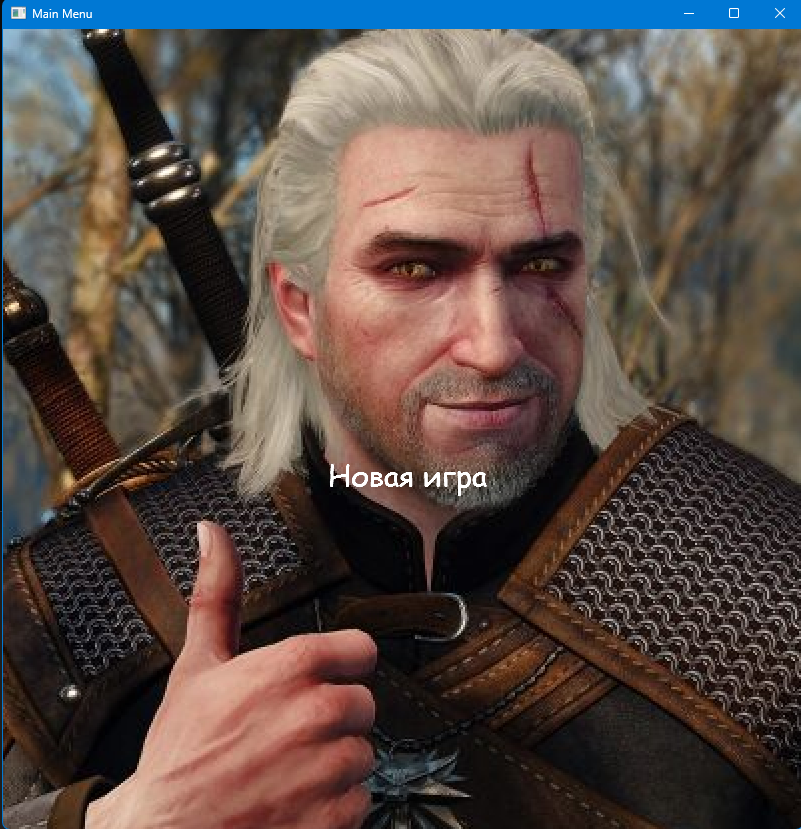
                     

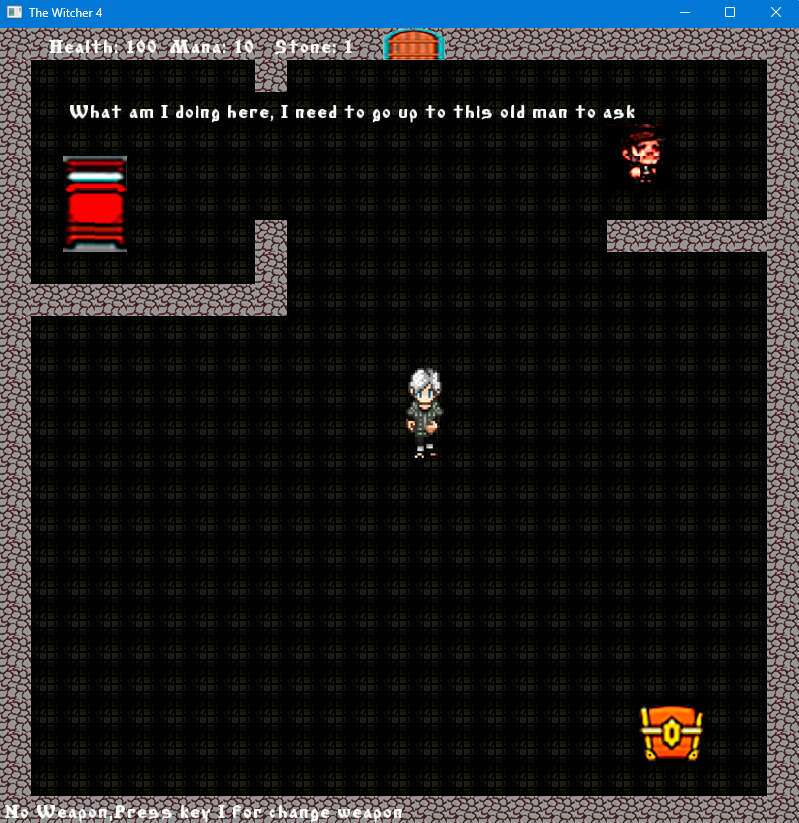
Рис.5 Код исходного файла main.cpp.

* 1. **Руководство пользователя.**

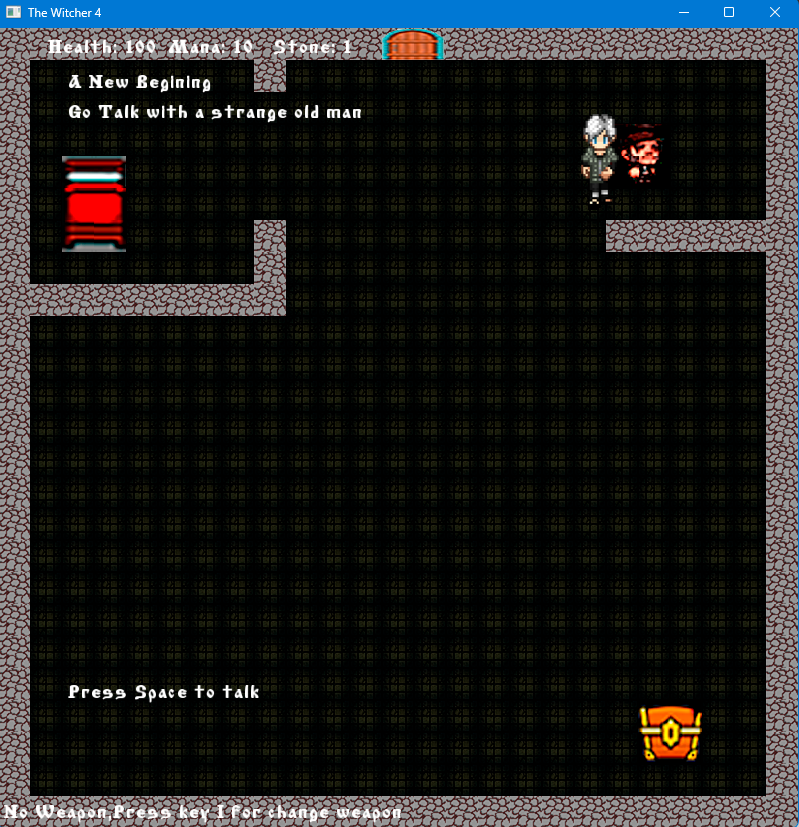
1. При запуске игры необходимо нажать New Game для начала

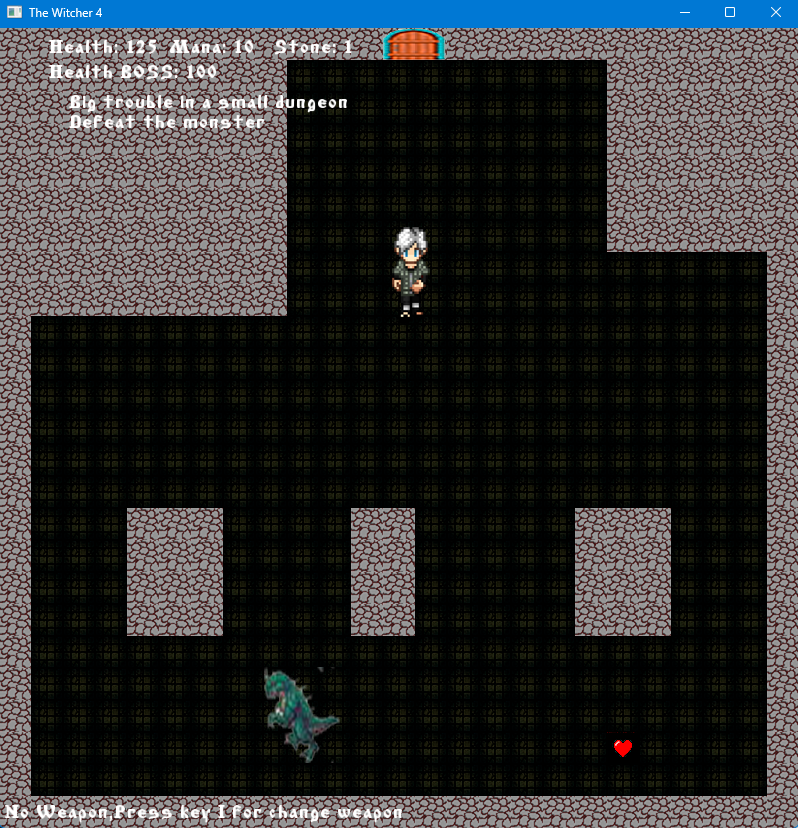
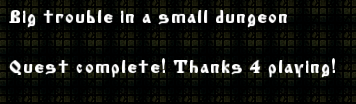


1. Появляется поле с локацией и объектами, а также интерфейс сверху.
2. Нажимая WASD персонаж может перемещаться



1. Игроку необходимо подойти к старику и, нажимая пробел, послушать диалог.

1. Игроку необходимо принести с сундука сердце для старика, для этого достаточно подойти к сундуку и подобрать выпавшее сердце
2. Отнеся сердце старику можно направляться к двери для встречи с боссом
3. Босса нужно победить с помощью фаерболов и камней, для переключения необходимо использовать кнопку I, а для стрельбы кнопку P.
4. После убийства босса игрока встретит надпись 

## **История проекта на GitHub.**

Данная игра располагается по ссылке <https://github.com/ShoshaAlexis/Witcher-2.0>

На рисунке 6 представлена история коммитов на GitHub.com

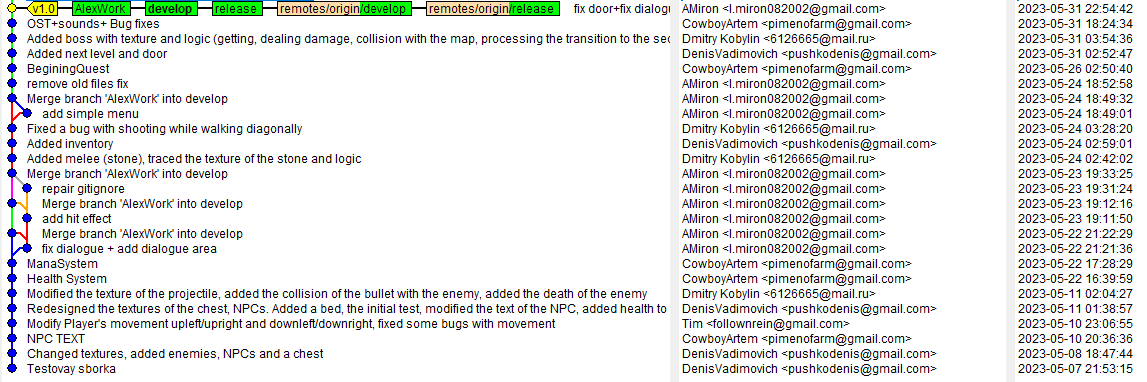
****

Рис.6. История коммитов игры «Ведьмак 4»

Старт разработки игры начался с того, что Денис Пушко создал проект в среде разработки Qt Creator (Qt 5.4.2).

После этого Денис добавил основу игры, текстуры, противника, НПС и сундук.

После этого Артем Пименов добавил начальную версию диалога со стариком, которая была подготовкой к появлению задания.

Дальше Тимофей Шестаков улучшил движение персонажа по диагонали, а также починил другие баги, связанные с движением.

Денис Пушко обновил текстуры для сундука и НПС, а также добавил кровать, обновил текст для НПС и добавил игроку здоровье и сердце, которое восстанавливает здоровье игроку.

Потом Дима Кобылин изменил текстуру для пули, добавил коллизию пули с врагами, а также добавил врагам здоровье.

Артем Пименов добавил полноценные системы здоровья и Маны для персонажа.

После этого Алексей Мирон улучшил систему диалогов, а также добавил конкретную зону для проигрывания диалога.

Также Алексей Мирон добавил эффект попадания врага по персонажу.

Дмитрий Кобылин добавил новое оружие – камень, и дописал для него логику и добавил текстуры.

После этого Денис Пушко добавил систему инвентаря.

Дмитрий Кобылин починил баг со стрельбой при ходьбе по диагонали.

После этого Алексей Мирон добавил в начало игры простое меню и починил git, который не учитывал gitignore.

Потом Артем Пименов добавил начальное задание.

После этого Денис Пушко добавил второй уровень с боссом и дверь для прохода туда.

Дмитрий Кобылин добавил босса, добавил текстуру и прописал логику.

Затем Артем Пименов добавил в игру звуковые эффекты и музыкальное сопровождение, а также пофиксил некоторые баги.

В заключении Алексей Мирон пофиксил дверь, которая «ловила» пули, добавил в диалоги откат на пропуск и выложил проект в релиз v1.0

# **Заключение.**

Игра «Ведьмак 4» была разработана с использованием возможностей языка программирования С++ (IDE Qt Creator (Qt 5.4.2)) и библиотеки (Simple and Fast Multimedia Library).

Для хранения истории изменений и резервного копирования во время разработки программы использовалась система контроля версий GIT. Во время работы игры сбои и зависания не наблюдались. В данной работе приложены следующие UML диаграммы: диаграмма прецедентов и диаграмма классов.

Считаю, что все поставленные задачи были выполнены в полном объеме.