|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** | | | | |
| БГТУ.СМК-Ф-4.2-К5-01 | | | | |
| Факультет | |  | И |  | Информационные и управляющие системы |
|  | |  | шифр |  | наименование |
| Кафедра | |  | И5 |  | Информационные системы и программная инженерия |
|  | |  | шифр |  | наименование |
| Дисциплина | |  | Программирование на языке высокого уровня | | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
| Объектно-ориентированная разработка программ с графическим |
| пользовательским интерфейсом «снизу-вверх»: объединение |
| разработанных классов в иерархии. Создание игры «Охота» |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И983 |
| Гаврилов A.A. | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
| Вальштейн К.В | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2019 г. |

САНКТ-ПЕТЕРБУРГ

2019 г.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 3

1 Постановка задачи 4

2 Описание классов 6

3 Описание программы 10

4 Пример работы программы 12

ЗАКЛЮЧЕНИЕ 17  
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18  
ПРИЛОЖЕНИЕ А 19

ВВЕДЕНИЕ

Целью данной курсовой работы является создание игры на языке C++ с использованием библиотеки SDL 1.2 и применением объектно-ориентированного программирования. Для достижения данной цели были поставлены задачи:

- проанализировать постановку задания;

- создать иерархию классов;

- реализовать дружественный интерфейс и понятное управление;

- написать текст программы на языке С++.

Simple DirectMedia Layer (SDL) — это свободная кроссплатформенная мультимедийная библиотека, реализующая единый программный интерфейс к графической подсистеме, звуковым устройствам и средствам ввода для широкого спектра платформ. Данная библиотека активно используется при написании кроссплатформенных мультимедийных программ (в основном игр) [1].

Объектно-ориентированный подход (ООП) к программированию используется при разработке программных проектов. Вместо того чтобы рассматривать программу как набор последовательно выполняемых инструкций, в ООП программа представляется в виде совокупности объектов, обладающих сходными свойствами и набором действий, которые можно с ними производить [2].

**1 Постановка задачи**

Разработать игру «Охота», в которой игрок, используя клавиши управления и мышь должен убивая животных, зарабатывать очки.

Требования к возможностям программы и процессу ее выполнения:

- Программа при выполнении должна иметь дружественный интерфейс, включающий заставку, меню, пояснения и удобные средства для управления программой.

- Заставка должна выводится на экран при запуске программы и давать представление об игре. На заставке должны быть отражены название игры, фамилия и номер учебной группы ее автора.

- Главное меню должно в обязательном порядке включать в себя пункты «Игра», «Правила игры», «Выход».

- Управление движением управляемого игроком объекта осуществляется с клавиатуры или мышью.

- Необходимо обеспечить достаточную скорость перерисовки динамически перемещаемых элементов.

- Во время игры должна быть возможность поставить её на паузу или завершить.

Требования к реализации программы.

- Выполнение курсовой работы должно быть основано на принципах объектно-ориентированного программирования. В работе должна быть построена и обоснована иерархия классов, описывающих предметную область конкретной задачи.

- Программа должна являться графическим приложением для Windows 10, написанным на языке C++ (с элементами C) с использованием графической библиотеки SDL версии 1.2.

**2 Описание классов**

Для реализации игры в ней должны присутствовать две группы классов- группа “Графические элементы”, предоставляющая классы для игровых объектов, включая методы взаимодействия между этими объектами, а также их прорисовку и перемещение, и группа “Меню и диалоговые окна”, включающая в себя классы для пользовательского интерфейса. Это поможет структурировать программу и воспользоваться принципом наследования, что позволит не прописывать одни и те же методы для различных элементов.

Таким образом, все объекты будут поделены на две группы:

1) Игровые элементы: в этой группе будут динамическиe объекты – объекты, которые могут изменять свои координаты. Такими объектами являются: игрок и враги.

2) Диалоговые окна: в этой группе будут только статические объекты – объекты, которые не изменяют свое положение. К ним относятся текстовые поля и меню.

Для игровых элементов были созданы классы Dynamic и Static:

Dynamic – абстрактный класс, содержит чисто виртуальный метод Move, реалиующий передвижение объектов по экрану и поле speed, задающее скорость передвижения. Static – класс хранящий информацию о положении объекта (x, y, w, h) и поверхность с изображением объекта Sprite, а также метод Draw, реализующий отрисовку.

От этих двух классов наследуются три класса: класс Hunter, класс Animal и класс Bullet. Класс Hunter хранит в себе все данные об игроке: время перезарядки ружья – cooldown, количество заработанных очков – points, вектор выпущенных пуль - Bullets. Содержит метод получения данных посредством ссылочного доступа GetPoints (получение итогового количества очков), методы задания направления и скорости передвижения SetSpeed, обеспечивающие передвижение в нужном направлении, метод Compute, реализующий обработку всех событий касательно персонажа и обработку выпущенных пуль и метод DrawBullets, отрисовывающий все выпущенные пули. Класс Animal хранит информацию о направлении передвижения (direct), количество очков, передаваемое игроку при смерти, а также поля, определяющие смену кадров при анимации (clip, next\_clip\_time). Клас содержит методы обработки передвижения (Compute), отрисовки (Draw), метод получения данных (GetPoints) и метод, обрабатывающий столкновения с пулями (Collides). Класс Bullet хранит в себе информацию о необходимости обработки объекта (on\_screen), а также метод инициализации, задающий начальную скорость и координаты (Init) и метод обработки передвижения и взаимодействия (Compute).

Для реализации интерфейса посредством использования притивов и библиотеки SDL\_draw создан класс Field. Реализует отрисовку прямоугольного поля заданного цвета (поле bg\_color) с текстом (поле info) заданного цвета (структура SDL\_Color) , с координатами (поля x, y) и размером (поля w, h) на указанной поверхности. Отрисовка происходит за счет функции Draw. Для реализации ввода текста создан класс TextField, наследуемый от класса Field. Класс реализует ввод текста посредством метода Input. Также добавлено поле hint – подсказка, выводимая в поле перед вводом текста. Класс Field наследует класс Button, отвечающий за отрисовку кнопок. Класс Button использует для работы несколько методов: Focus – подсвечивает кнопку, на которую наведена мышь, UnFocus – убирает подсвечивание кнопки, MouseOver – определяет наведен ли курсор на кнопку.  
Для организации меню создан класс Menu наследуемый от класса Button. Полями класса является вектор, хранящий несколько кнопок – buttons и поле last\_focused, хранящее информацию о кнопке, на которую последней была наведена мышь. Для работы класса добавлены 2 метода: Draw – отрисовка меню, GetKey – при помощи маски получает номер кнопки исходя из координат мыши.

Для упрощения процесса разработки и дальнейшей работы создан класс Game. Класс реализует основной игровой цикл (метод MainCycle), контроль количества кадров в секунду (next\_frame\_time и delay), обработку (Compute), отрисовку всех динамических объектов и их взаимодействий (Draw). Содержит метод обработки событий, обеспечивающий управление персонажем (Event). также членами класса являются несколько объектов класса Field, организующих внутриигровой интерес (timer, username, points). Так же в классе содержатся поля, определяющие инициализацию объектов класса Animal (next\_spawn\_time) и поля, ограничивающие работу программы по времени (elapsed, ticks). Кроме того членами данного класса является объект класса Player (player) и вектор объектов класса Animal (Animals), при помощи которых и обрабатывается взаимодействие между охотником и животными.

Схема наследования классов представлена на рисунке 1.

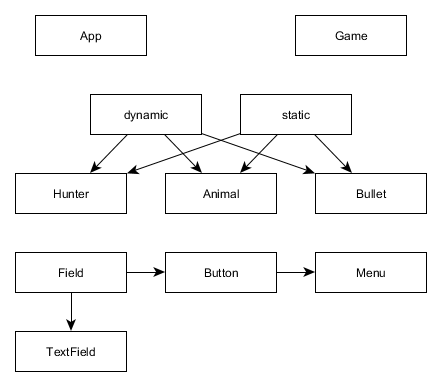


Рисунок 1 – Классы

**3 Описание программы**

Программа разработана в среде программирования Code::Blocks и написана на языке C++ (с элементами C) с использованием графической библиотеки SDL версии 1.2. При написании программы использовались источники [2], [3], [4].

Проект состоит из следующих файлов:

1. «App.h»
2. «Game.h»
3. «Object.h»
4. «UI.h»
5. «App.cpp»
6. «Game.cpp»
7. «Object.cpp»
8. «UI.cpp»
9. «main.cpp»

Главным из них является файл «App.cpp», т.к. в нем находятся методы класса App, описывающие всю работу программы

В файлах с расширением «.h» содержится объявление одноимённых классов, а в файлах с расширением «.cpp» содержится определение методов одноименных классов.

Класс App – основной класс. Он содержит следующие методы и поля:

SDL\_Surface \*display;

SDL\_Surface \*background;

SDL\_Event event;

TTF\_Font \*font;

User current;

vector<User\*> results;

Поле display – основная поверхность. Поле background – поверхность, содержащая фоновое изображение. Поле event позволяет обрабатывать внешние события. font – основной шрифт. current – данные о текущем пользователе (имя и количество очков). Вектор объектов класса User (results), реализующий список лучших 10 игроков.

В игре используется шрифт CharisSil, взятый из источника [5]. Изображения игрока и животных взяты из источников [6] и [7] соответственно. Фоновое изображение взято из источника [8].

**4 Пример работы программы**

На рисунке 2 представлена заставка, содержащая название игры, имя автора и номер группы. Также во время заставки пользователю предлагается ввести имя.

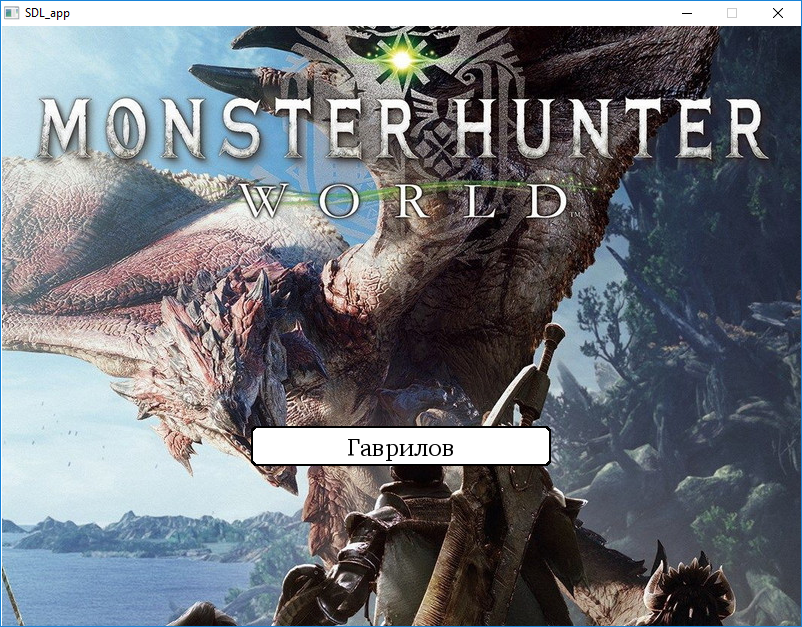


Рисунок 2 – Заставка и предложение ввести имя

На рисунке 3 представлено главное меню игры.



Рисунок 3 – Главное меню

Здесь, пункт Игра позволяет начать игру, пункт Смена игрока позволяет сменить имя, пункт Таблица результатов выводит таблицу 10 лучших результатов, пункт Правила вызывает окно справки, пункт Выход позволяет корректно завершить игру.

На рисунке 4 изображена таблица лучших результатов и подменю. Пункт Clear позволяет очистить таблицу, пункт зволяет вернуться в главное меню.

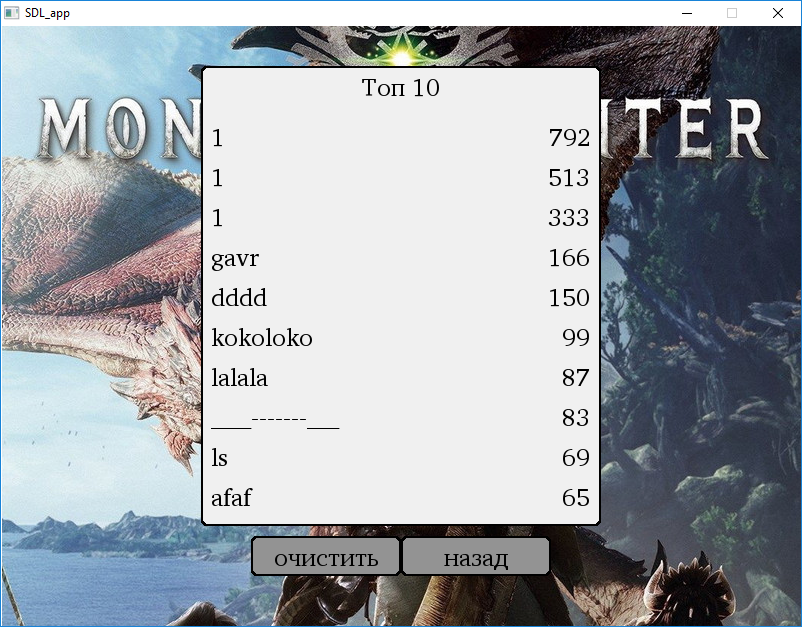


Рисунок 4 – Таблица результатов

Вывод справки изображен на рисунке 5.

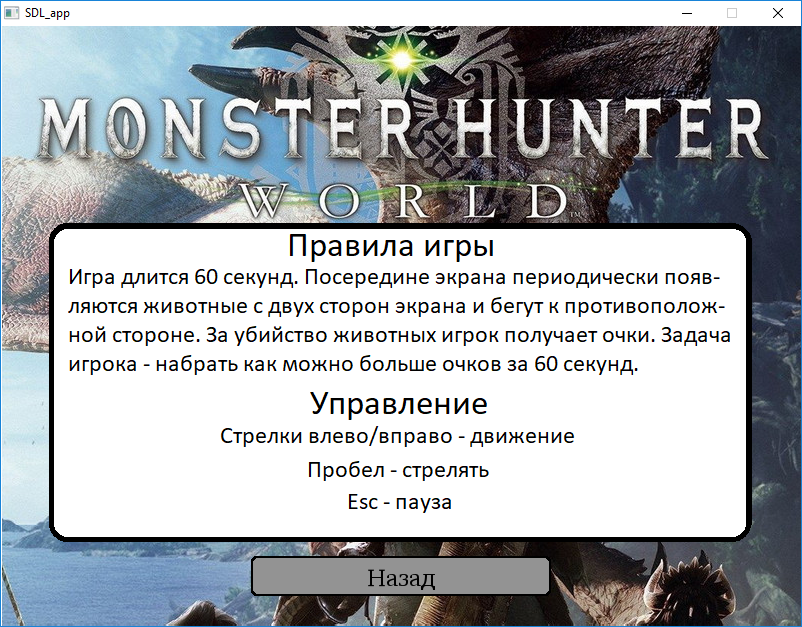


Рисунок 5 – Справка

На рисунке 6 изображен игровой процесс. Строка слева сверху– имя игрока. Число посередине сверху – оставшееся время. Число справа сверху – заработанные очки.

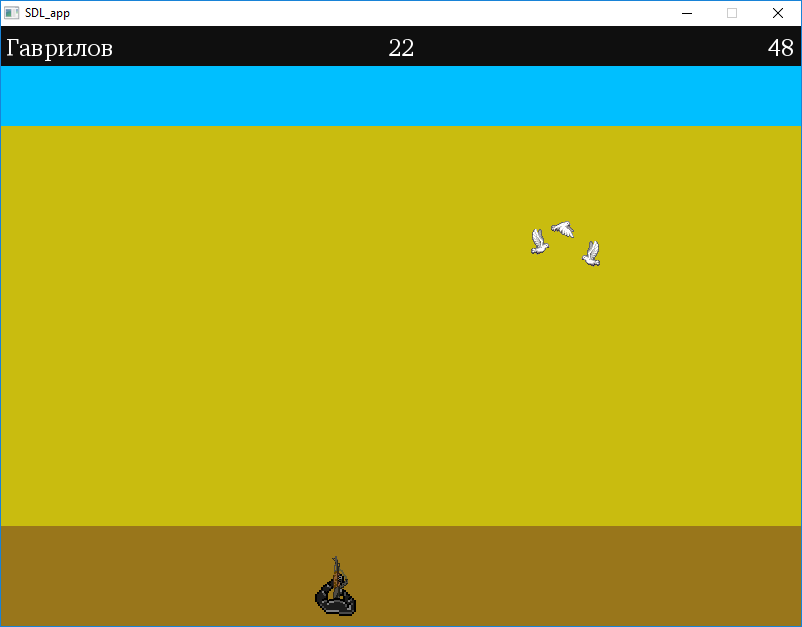


Рисунок 6 – Игровой процесс

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы по дисциплине «Программирование на языке высокого уровня» была реализована игра «Лабиринт». Проанализирована постановка задания. Реализованы иерархии классов, дружественный интерфейс и понятное управление. Все поставленные задачи выполнены, цель достигнута.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Simple DirectMedia Layer [Интернет-ресурс]  
   <https://ru.wikipedia.org/wiki/Simple_DirectMedia_Layer/> (дата обращения 25.06.2018)
2. Уроки по SDL [Интернет-ресурс]

<http://www.lazyfoo.net/SDL_tutorials/>

1. Арипова, О. В. Программирование на языке высокого уровня [Лабораторный практикум]/О. В. Арипова, А. Н. Гущин, О. А. Палехова —СПб. : БГТУ «Военмех» им. Д. Ф. Устинова, 2014. — 183 с.
2. Гущин А.Н. Применение библиотеки SDL для разработки программ на языке С. - СПб.: БГТУ «Военмех» им. Д.Ф. Устинова, 2014 – 183 стр.
3. CharisSil [Интернет-ресурс]  
   <http://allfont.ru/download/charis-sil/>
4. Изображение игрока [Интернет-ресурс]

<http://traffic-club.info/2018timage-top-down-shooter-sprites.awp>

1. Изображение животных [Интернет-ресурс]

<https://www.pinterest.ru/pin/661044051531968897/?lp=true>

1. Фоновое изображение [Интернет-ресурс]

<https://www.mobygames.com/game/monster-hunter-world/cover-art/gameCoverId,530723/>

**ПРИЛОЖЕНИЕ А**

Текст программы хранится на электронном носителе .