	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	И9	Систем управления и компьютерных технологий
	шифр	наименование
Дисциплина	Программирование на ЯВУ	

КУРСОВАЯ РАБОТА

на тему

СОЗДАНИЕ ИГРЫ “Labo!”

Выполнил студент группы И973

Глушков Дмитрий Сергеевич

Фамилия И.О.

РУКОВОДИТЕЛЬ

Палехова О. А.

Фамилия И.О.

Подпись

Оценка _____

« _____

» _____ 201_г.

САНКТ-ПЕТЕРБУРГ

2018 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Формулировка задания.....	4
2 Обоснование используемых иерархий классов.....	6
2.1 Иерархия, описывающая окна программы.....	6
2.2 Иерархия, описывающая объекты игрового процесса.....	7
3 Состав проекта.....	9
4 Описание реализации.....	10
4.1 Реализация множественного сохранения.....	10
4.2 Сохранение лучших результатов.....	10
4.3 Обновления сохранения и списка лидеров.....	10
4.4 Игровой процесс.....	11
5 Иллюстрации работы программы.....	13
ЗАКЛЮЧЕНИЕ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17
ПРИЛОЖЕНИЕ А.....	18

ВВЕДЕНИЕ

Целью курсовой работы является создание игры с использованием таких возможностей языка C++, как классы и наследование классов, то есть выполненной в соответствии с принципами ООП.

Сформулированная цель работы ставит следующие задачи:

- проанализировать поставленное задание;
- спроектировать и реализовать иерархию классов;
- на основании возможностей, предоставляемых библиотекой SDL 1.2, спроектировать и реализовать дружественный интерфейс, понятное управление.

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-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования [2].

1 Формулировка задания

Создать игру “Labo!”.

Правила игры: В каждом углу прямоугольного лабиринта находится шар уникального цвета. Игроку необходимо вывести из лабиринта последовательно каждый из шаров, при этом совершив минимальное количество шагов, то есть выбрав кратчайший маршрут. Игрок может переключаться между шарами посредством нажатий клавиш 1-4 на клавиатуре. Выход из лабиринта находится в центре экрана и занимает две центральные ячейки лабиринта. Движение по лабиринту осуществляется при нажатии клавиши-стрелки, указывающей в сторону, соответствующую стороне движения.

Реализовать заставку. Она должна выводиться на экран при запуске программ. На заставке должны быть отражены фамилия и номер учебной группы ее автора. По завершении заставки игрока просят ввести имя.

Во время игры должна быть возможность поставить её на паузу или завершить. Из паузы или идет продолжение, или выход в главное меню с завершением игрового процесса. При запуске программы после ввода имени пользователя проверяется наличие у него сохраненных игр.

Пять лучших результатов игры сохраняются в двоичном файле в виде структур, содержащих непосредственно результат и имя игрока, получившего его.

При выборе пункта меню «Таблица лидеров» на экран должно в табличном виде выводиться содержимое файла результатов, и подменю, позволяющее выполнить следующие действия: «обнулить» (при выборе данного пункта очищается файл результатов), «печать» (при выборе данного пункта создается текстовый файл с данными для печати), «вернуться в главное меню».

Управление движением управляемого игроком объекта осуществляется с клавиатуры.

Необходимо обеспечить достаточную скорость перерисовки динамически перемещаемых элементов.

Для безопасности данных и избегания стороннего вмешательства в файлы

игры запись списка лучших результатов реализовать в виде бинарного файла.

Реализовать меню со следующим пунктами:

- “начать игру” – начинается игровой процесс;
- “таблица лидеров” – на экран выводятся 5 лучших результатов;
- “правила” – на экран выводится окно, в котором описаны правила игры;
- “выход” – завершение работы программы.

2 Обоснование используемых иерархий классов

В программе реализованы две иерархии классов. Иерархии строились по принципу объединения по общему признаку. Первая иерархия классов представляет из себя совокупность всех окон, выводимых на экран во время работы программы, вторая отвечает за обработку элементов непосредственно игрового процесса.

При этом, объекты классов второй иерархии используются объектами класса первой иерархии. Для наглядной иллюстрации диаграмма классов представлена на рисунке 1.

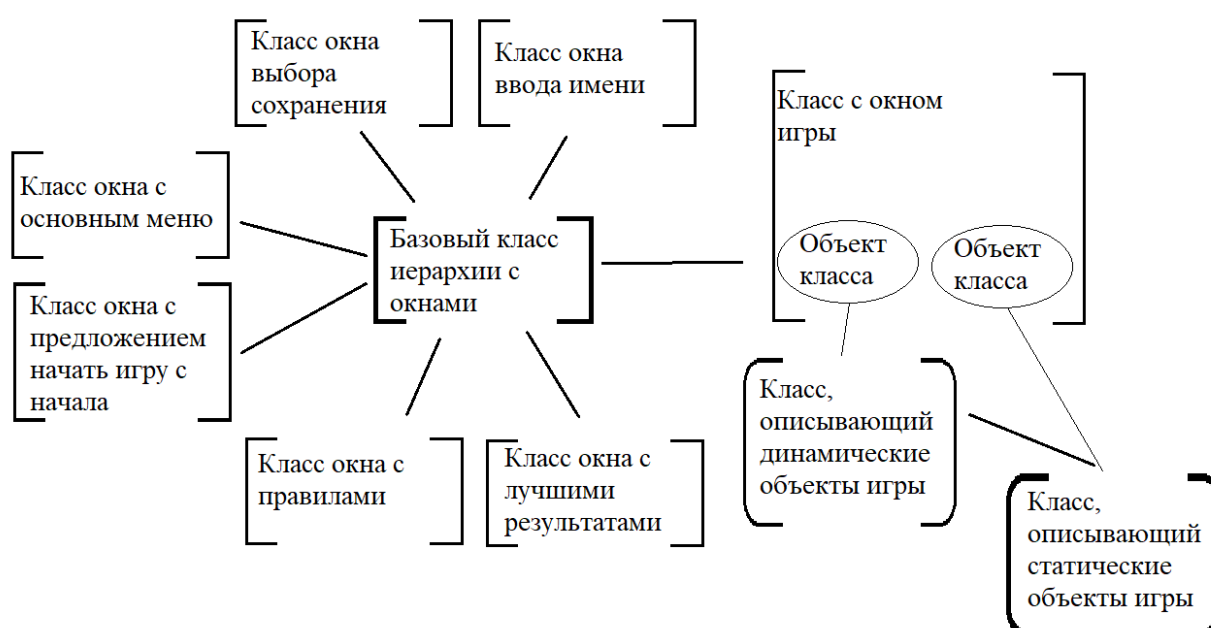


Рисунок 1 – Диаграмма классов

При проектировании и написании программы использовалась справочная литература [3], [4].

2.1 Иерархия, описывающая окна программы

Для составления полноценной иерархии окон, отображаемых на экране во время работы программы, у всех окон были выделены следующие общие элементы:

1) поверхность окна. Эта поверхность используется для обеспечения достаточной скорости перерисовки. Все элементы окна сначала изображаются

на этой поверхности, после чего переносятся на поверхность, отображаемую на экране;

2) текст. Для обеспечения понятного интерфейса каждое окно содержит в себе текстовые подсказки;

3) обработка событий. Каждое окно, выводимое на экран, должно позволять пользователю взаимодействовать с элементами окна, переключаться между окнами.

Эти элементы вынесены в базовый класс. В программе же используются производные классы, описание которых представлено ниже.

- Класс окна ввода имени. В этом окне пользователю предлагается ввести имя игрока, по которому строится имя файла сохранения, это имя будет записано в таблице лидеров.

- Класс окна выбора сохранения. В этом окне пользователю предоставляется возможность выбрать одно из существующих сохранений, либо создать новое.

- Класс окна с основным меню программы. В этом окне пользователь может вызвать начало игры, вызвать окно с правилами, вызвать окно с таблицей лидеров, выйти из программы.

- Класс окна с правилами игры. В этом окне пользователь может ознакомиться с правилами игры.

- Класс окна с таблицей лидеров. Окно с обновляемой таблицей, в которой представлены 5 лучших результатов. В этом окне пользователю предлагается удалить список лидеров и создать текстовый файл, в котором будут записаны 5 лучших результатов.

- Класс окна с игрой. Вызов этого окна означает начало непосредственно игрового процесса.

- Класс окна с предложением начать игру сначала. Это окно выводится на экран в том случае, если пользователь пытается начать игру, но на этом сохранении игра уже пройдена.

2.2 Иерархия, описывающая объекты игрового процесса.

Игровой процесс строится на движении объектов по графической матрице, ячейками которой являются участки окна размером 50 на 50 пикселей. Все объекты выводятся на экран в виде геометрических фигур: квадратов и кругов определенного цвета и размера. Значения этих характеристик определяются значением элементов матрицы, в которой хранится карта лабиринта. Все объекты игры можно разделить на 2 типа: статические и динамические. Статические элементы представляют из себя фундамент лабиринта – это поля, по которым передвигаются динамические объекты; а также поля, ограничивающие движение шаров – стенки лабиринта. При этом оба типа имеют следующие общие элементы:

- 1) высота объекта;
- 2) ширина объекта;
- 3) цвет объекта.

Так как помимо этих черт динамические объекты должны иметь возможность перемещаться по лабиринту, было решено сделать класс, описывающий динамические объекты, наследником класса, описывающего статические объекты.

3 Состав проекта

Проект состоит из следующих файлов:

- 1) main.cpp – файл, содержащий в себе описание и вызов основного меню программы, а также определение функций, отвечающих за заставку игры;
- 2) windows.cpp – файл, содержащий в себе иерархию классов, описывающую все окна программы;
- 3) objects.cpp – файл, содержащий в себе иерархию классов, описывающую объекты игры;

В ходе работы программы используются следующие вспомогательные файлы:

- 1) fonts.ttf – файл со шрифтами, заимствованный из источника [6];
- 2) logo.bmp – изображение, используемое в качестве логотипа игры;
- 3) top.bin – файл, содержащий в себе список лучших результатов;
- 4) map.txt – файл, содержащий в себе карту лабиринта с начальными положениями шаров.

4 Описание реализации

4.1 Реализация множественного сохранения

Имя файла сохранения имеет следующий вид:

`<имя_пользователя><номер_сохранения>.txt`

имя_пользователя – имя, вводимое игроком при запуске программы, номер сохранения – номер ячейки сохранения, выбранной пользователем.

После ввода имени игроком программа проверяет, существует ли хотя бы один файл сохранения, в котором в поле имя_пользователя будет значиться введенное имя. Если такой файл существует, в меню выбора сохранения в соответствующей ячейке появится надпись “сохранение <номер_сохранения+1>”. В противном случае ячейка будет содержать надпись: “Новая игра”.

4.2 Сохранение лучших результатов

В данной реализации в списке лучших результатов представлено только 5 лучших результатов.

При выборе пункта “Таблица лидеров” в основном меню программы, происходит считывание всех записанных в файл данных и вывод на экран в виде <результат_пользователя><имя_пользователя>. В случае, если в файл записано меньше 5 результатов, то оставшиеся ячейки окна содержат надпись: “Пусто”. В окне табличном виде представлены имена 5 лучших игроков с их результатами, информация о том, что игрок может очистить список лидеров, подготовить файл для печати, выйти в главное меню.

Информация о лучших результатах хранится в бинарном файле. Это сделано для того, чтобы пользователи не могли вписывать желаемые значения в файл с лучшими результатами игроков.

После подготовки файла к печати, то есть создания текстовой версии списка лидеров, изменения информации в текстовой версии списка лидеров никак не повлияет на список лидеров, выводимый в программе.

4.3 Обновления сохранения и списка лидеров

Обновление списка лидеров происходит в двух случаях:

1) Выбор действия “Обнулить” в соответствующем окне программы. В этом случае происходит полное удаление списка лидеров, теряются все записанные результаты.

2) Полное прохождение игры. В этом случае результат игрока будет сравниваться с уже хранящимися в списке лидеров результатами, если таковые имеются. Если этот результат превосходит один из результатов в списке лидеров, то последний элемент из списка удаляется, а текущий результат занимает соответствующее место, после чего файл перезаписывается.

Обновление файла с сохранением происходит исключительно при выборе пункта “Выход” в основном меню программы.

Во время работы программы вся информация о состоянии игрового процесса хранится в оперативной памяти компьютера.

4.4 Игровой процесс

Для удобства пояснения определим следующие понятия:

- Ячейка – элемент матрицы;
- Точка – участок лабиринта, на котором изображен отдельный объект игры.
- Исходная ячейка – ячейка матрицы, в которой записано значение текущего шара (от 2 до 5);
- Исходная точка – участок графически представленного лабиринта, занимаемый шаром;
- Конечная ячейка – ячейка матрицы, находящаяся ниже (нажатие стрелки, указывающей вниз), выше (нажатие стрелки, указывающей вверх), слева (нажатие стрелки, указывающей влево), или справа (нажатие стрелки, указывающей вправо) относительно исходной ячейки;
- Конечная точка - участок графически представленного лабиринта, находящийся в соответствующем нажатой клавише направлении;

Изображение элементов лабиринта реализовано при помощи матрицы, в которой записаны цифры в диапазоне от 0 до 9 которые в игре соответствуют следующие точки:

0 – пустая точка, по которой возможно движение;

1 – барьерная точка, ограничивающая поля, по которым возможно движение;

2,3,4,5 – обозначают точки, занимаемые шарами, которые необходимо вывести из лабиринта;

9 – означают точки, являющиеся выходами из лабиринта.

Движение по лабиринту осуществляется при нажатии клавиш-стрелок, соответствующих желаемому направлению. При нажатии на клавишу-стрелку в программе осуществляются проверка значения конечной ячейки.

а) В случае, если значение этой ячейки равно одному из значений диапазона 1-

5 никаких действий не происходит, так как конечная точка занята другим объектом.

б) В случае, если значение этой ячейки равно 0:

1) Значение конечной ячейки становится равно значению исходной ячейки;

2) Значение исходной ячейки становится равно 0;

3) Вызывается функция покадровой перерисовки для имитации движения шара из исходной точки в конечную.

с) В случае, если значение этой ячейки равно 9:

1) Значение исходной ячейки обнуляется;

2) Вызывается функция покадровой перерисовки для имитации движения шара из исходной точки в конечную.

5 Иллюстрации работы программы

На рисунке 2 представлено окно, в котором пользователю предлагается ввести свое имя. Поддерживаются только символы латинского алфавита. Это имя будет определять доступные игроку сохранения. Под этим именем в таблицу лучших результатов будет вноситься результат текущего игрока.



Рисунок 2 – Окно ввода имени

Окно выбора сохранения представлено на рисунке 3. В случае, если игрок с таким именем уже играл, то в выбранном им раннее слоте сохранения будет надпись: “Сохранение <номер сохранения>”



Рисунок 3 – Окно выбора сохранения

Основное меню программы представлено на рисунке 4. При помощи этого меню игрок может загрузить таблицу лидеров, окно с правилами, начать игру, выйти из игры.



Рисунок 4 – Основное меню программы

На рисунке 5 изображена иллюстрация игрового процесса. Нижняя полоса представляет собой полосу состояния, в которой указан текущий счет; активный шар; список шаров, которые еще в лабиринте.

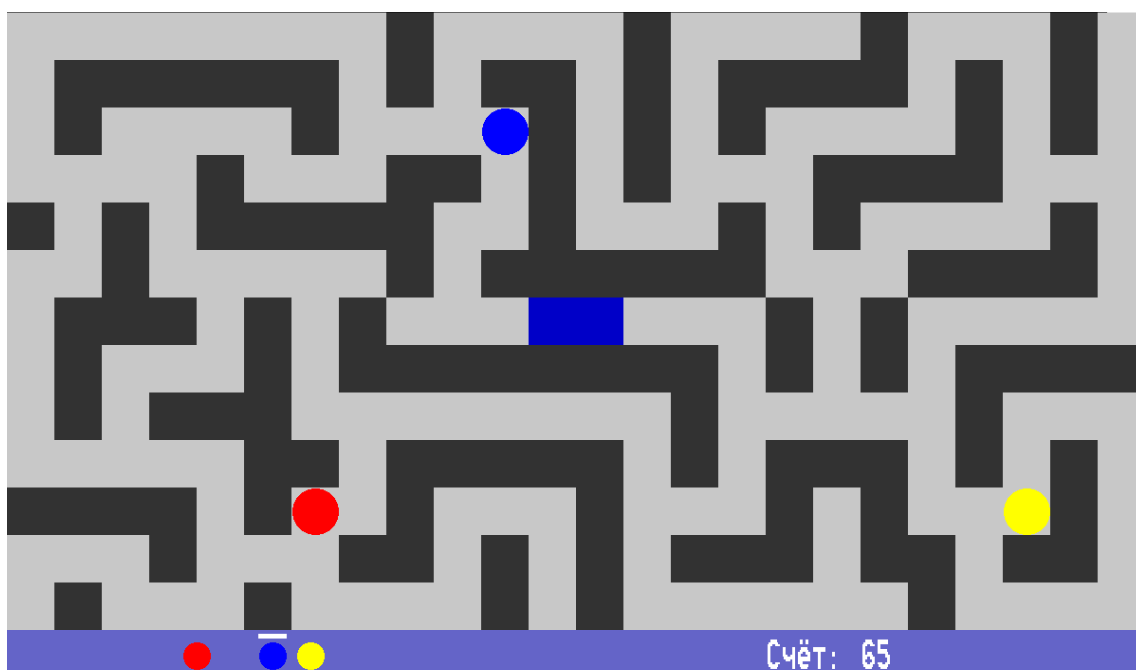


Рисунок 5 – Игровой процесс

На рисунке 6 изображено окно “Таблица лидеров”. В этом окне игрок может очистить список игроков нажатием клавиши 2, создать текстовый файл с лучшими результатами нажатием клавиши 1. Выход из этого меню осуществляется нажатием клавиши ESC.

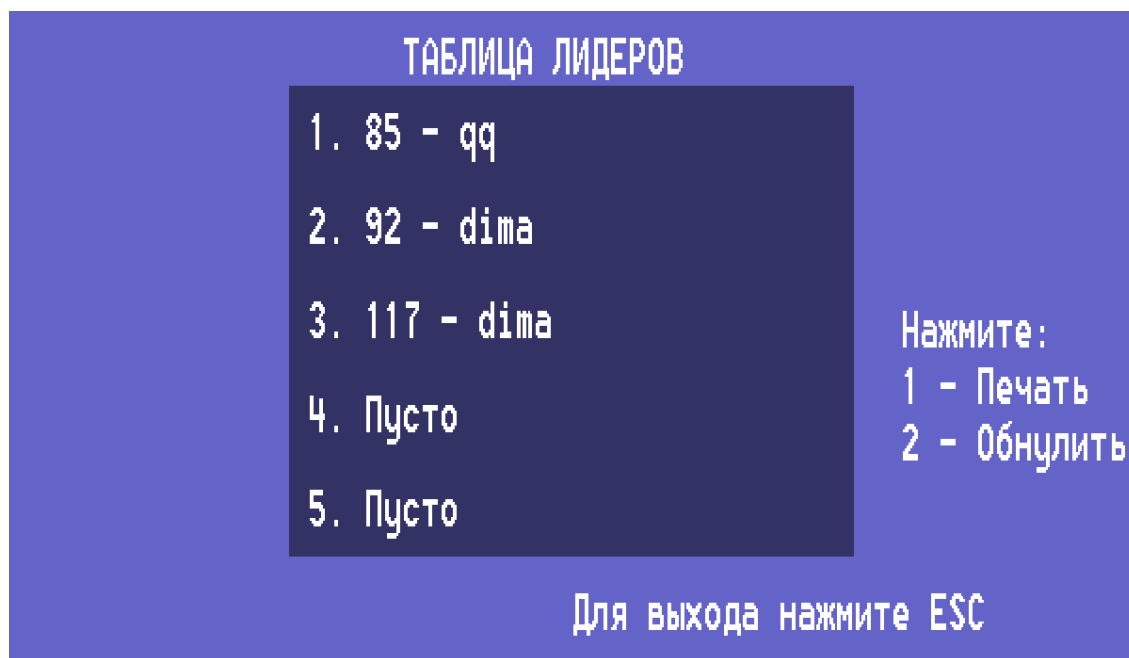


Рисунок 6 – Окно с таблицей лидеров

На рисунке 7 изображено окно с правилами. Возврат в основное меню программы происходит при нажатии клавиши ESC.

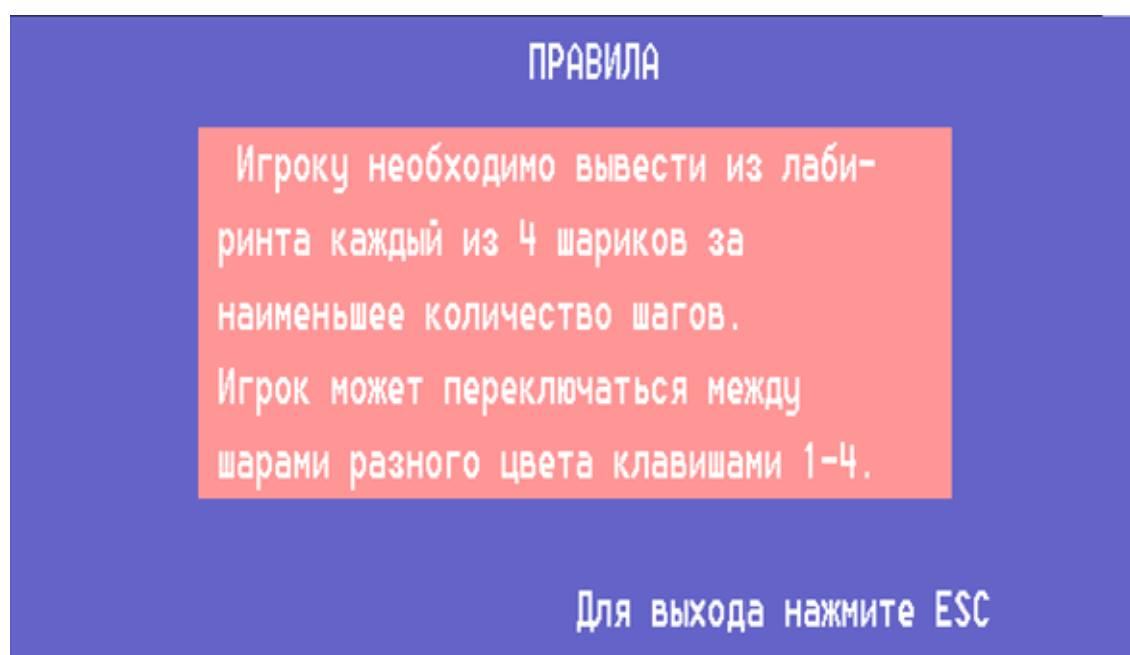


Рисунок 7 – Окно с правилами

ЗАКЛЮЧЕНИЕ

Таким образом, в процессе выполнения курсовой работы были выполнены все поставленные задачи, а именно:

- проанализировано задание;
- спроектирована и реализована иерархия классов;
- на основании возможностей, предоставляемых библиотекой SDL 1.2, спроектирован и реализован дружественный интерфейс, реализовано понятное управление.

Достигнута цель курсовой работы – разработана игра, предоставляющая пользователю следующие возможности:

- 1) Игровой процесс, представляющий из себя лабиринта на 4 элемента;
- 2) Множественное сохранение;
- 3) Таблица лучших игроков.

Игра разработана в соответствии с принципами объектно-ориентированного программирования. Графический интерфейс реализован при помощи возможностей, предоставляемых библиотекой SDL 1.2.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Simple DirectMedia Layer [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Simple_DirectMedia_Layer (дата обращения 15.06.2018)
2. Объектно-ориентированное программирование [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование (дата обращения 15.06.2018)
3. *Лафоре Р.* Объектно-ориентированное программирование в C++. 4 издание – СПб.: ПИТЕР, 2004. - 361 с.
4. Документация к SDL 1.2[Электронный ресурс] URL: <https://www.libsdl.org/release/SDL-1.2.15/docs/> (дата обращения 08.06.2018).
5. 7Fonts.ru – библиотека шрифтов. [Электронный ресурс] URL: http://7fonts.ru/lcd/font_zig_8-bit.html (дата обращения 25.04.2018).

1.

ПРИЛОЖЕНИЕ А. Текст программы

Текст программы прилагается на цифровом носителе.