



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 ЯВУ	

КУРСОВАЯ РАБОТА

НА ТЕМУ

«Игра РасМан»

Выполнил студент группы И596

Орехов Руслан Вячеславович

Фамилия И.О.

Орлов О. В.

Фамилия И.О.

Подпись

Оценка

« _____ »

2020г.

САНКТ-ПЕТЕРБУРГ

2020 г

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
ВВЕДЕНИЕ	3
ОСНОВНАЯ ЧАСТЬ.....	4
1. Постановка задачи	4
2. Результат работы	4
3. Иерархия классов.....	8
ЗАКЛЮЧЕНИЕ.....	9
СКРИНШОТЫ ИГРЫ	10

ВВЕДЕНИЕ

Видео игры прочно заняли свою позицию в современной индустрии развлечений. Существуют попытки выделить компьютерные игры как отдельную область искусства, наряду с театром, кино и т.п. Разработка игр может оказаться не только увлекательным, но и прибыльным делом, примеров этому предостаточно в истории.

Первые примитивные компьютерные и видео игры были разработаны в 1950-х и 1960-х годах. Они работали на таких платформах, как осциллографы, университетские мейнфреймы и компьютеры EDSAC. Самой первой компьютерной игрой стал симулятор ракеты, созданный в 1942 году Томасом Голдсмитом Младшим (англ. Thomas T. Goldsmith Jr.) и Истл Рей Менном (англ. Estle Ray Mann). Позже, в 1952 году, появилась программа "ОХО", имитирующая игру "крестики-нолики", созданная А.С. Дугласом как часть его докторской диссертации в Кембриджском Университете. Игра работала на большом университетском компьютере, известном как EDSAC (Electronic Delay Storage Automatic Calculator).

В настоящее время, разработка игры – это многомиллионный процесс, в котором задействована целая команда разработчиков, сложные современные технологии и даже маркетинговые ходы.

С помощью своей курсовой работы я хотел познакомиться с процессом создания игры, узнать тонкости их реализации, сделать своё первое графическое приложение для Windows ОС.

ОСНОВНАЯ ЧАСТЬ

1. Постановка задачи

Требования к работе:

- использование C++;
- использование SDL версии 1.2.15.
- решение задачи методом ООП.

Основные требования к программе:

- графический интерфейс.
- наличие нескольких уровней.
- хранение списка рекордов в отдельном файле.

В меню игры должен предоставляться выбор между началом игры (кнопка «Play»), настройкой текущего уровня (кнопка «Level»), просмотром игровой статистики (кнопка «Score»), меню с основной информации об игре (кнопка «Help»), и выходом из игры (кнопка «Exit»). У игры должны быть простые правила, но при этом она должна быть интересной.

2. Результат работы

На рисунке 1 представлено полученное в результате разработки меню игры.

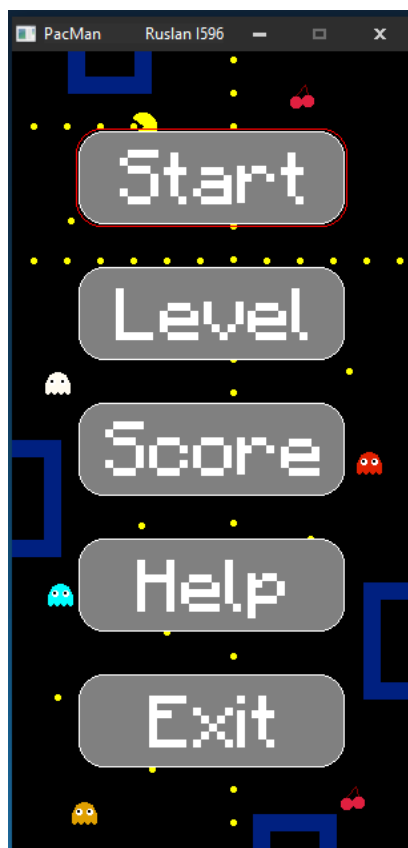


Рисунок 1 – Меню игры

Навигация по меню осуществляется с помощью клавиш ‘w’ и ‘s’. Кнопкой ‘enter’ активируется активный пункт меню (выделяется красной рамкой) и идет перенаправление.

После запуска игры появляется окно с выбранным лабиринтом. Изменение направления движения главного персонажа в игре осуществляется с помощью кнопок ‘w’, ‘a’, ‘s’ и ‘d’. Кнопкой ‘p’ можно поставить игру на паузу. Кнопка ‘esc’ возвращает игрока в главное меню. Цель игры – съесть все монетки и не умереть от противников. В игре присутствует 4 противника-призрака. У них есть 2 состояния. В первом случае главный персонаж умирает при попытке съесть их (в этом случае призраки разных цветов), а во втором случае они умирают от главного персонажа. Это происходит если растап съел ягодку (в этом случае призраки перекрашиваются в серый). Через некоторое время призраки становятся снова “обычными”, и главному герою приходится снова от них убегать. За съеденную монетку дается 100 очков, за ягодку – 200 а за призрака – 300.

На рисунке 2 и 3 предоставлены скриншоты двух разных уровней, один из которых можно выбрать в меню “Level”.

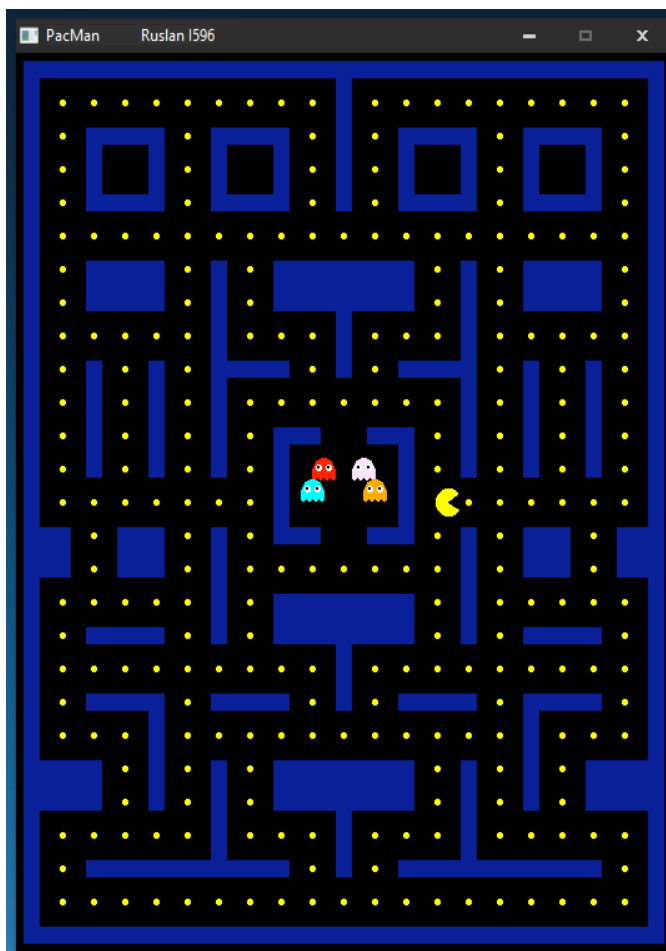


Рисунок 2 – уровень “Small”

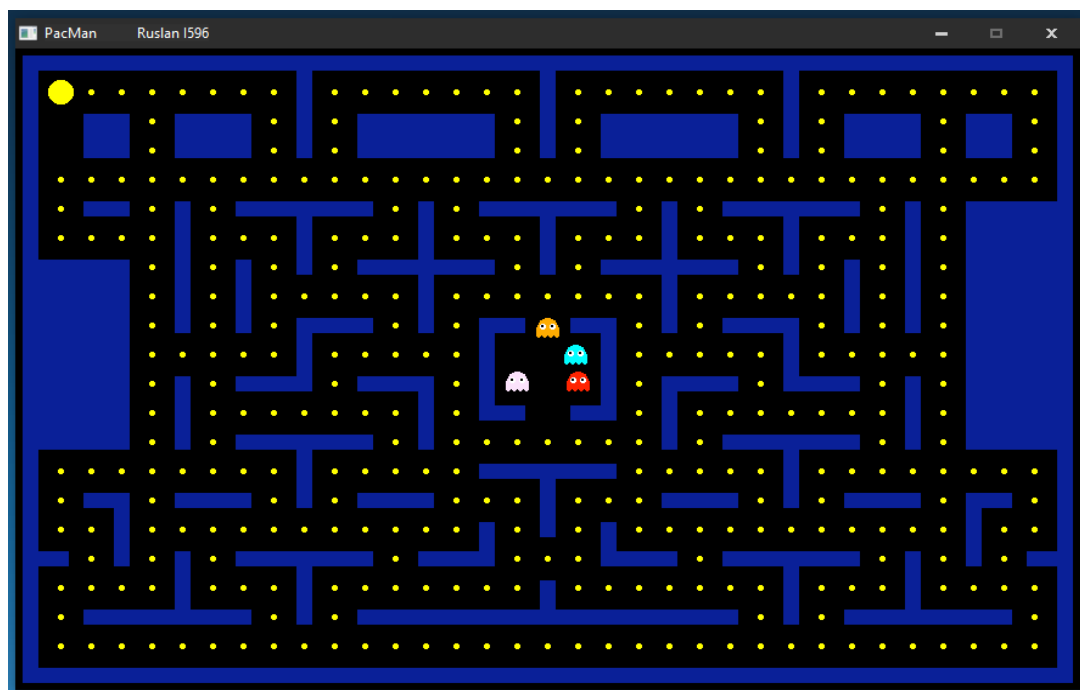


Рисунок 3 – уровень “Big”

На рисунке 4 приведен пример сообщения о том, что игра приостановлена.

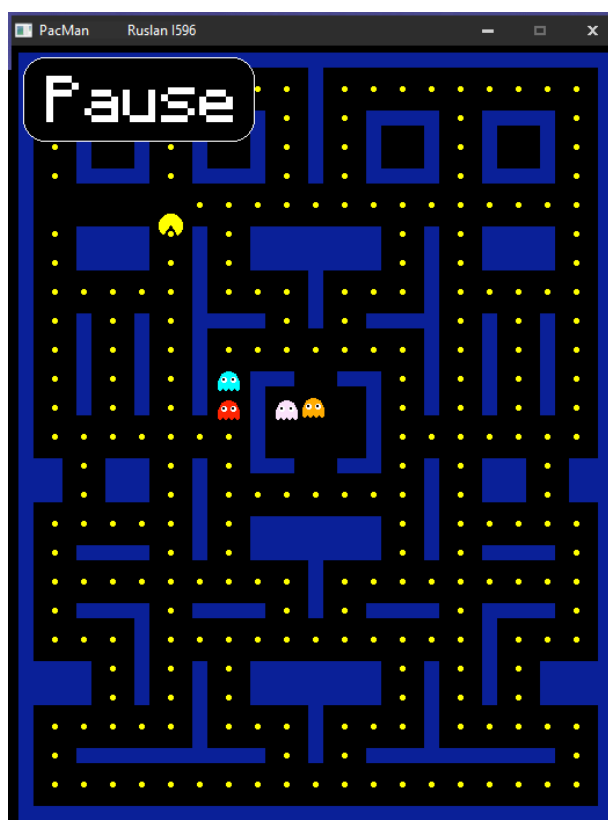


Рисунок 4 – приостановленная игры на уровне “Small”

На рисунке 5 приведен пример сообщения об окончании игры по причине того, что растап попытался съесть призрака.



Рисунок 5 – конец игры

На рисунке 6 приведено подменю “Level” - в нем выбирается уровень, подменю “Score” – в нем отображается список рекордов (для каждого уровня свой) а также подменю “Help” – в нем содержится краткая информация об игре.



Рисунок 6 – подменю “Level”, “Score” и “Help”

3. Иерархия классов

класс **TEXTURES** - хранит текстуры.

класс **MAZE** - хранит лабиринт, монетки и ягоды.

класс **ENTITY** – основа для пакмена и приведения.

класс **HERO**, наследуется от **ENTITY** – хранит информацию о пакмане.

класс **ENEMY**, наследуется от **ENTITY** – хранит информацию об одном призраке.

класс **LOGICS** – содержит в себе все, перечисленные классы, кроме **TEXTURES** и хранит информацию об игре

ЗАКЛЮЧЕНИЕ

С помощью SDL удалось реализовать графический интерфейс, а с помощью навыков объектно-ориентированного программирования - гибкость кода и простую возможность дальнейшего расширения функционала.

Благодаря этой работе была создана игра на C/C++ с использованием классов, значит задача, поставленная перед этой работой - решена.

Также, в ходе создания игры были получены ценные навыки проектирование графических приложений, который будут несомненно полезны в дальнейшей профессиональной деятельности.

СКРИНШОТЫ ИГРЫ

На рисунках 8 и 9 предоставлены скриншоты игры на разных уровнях.

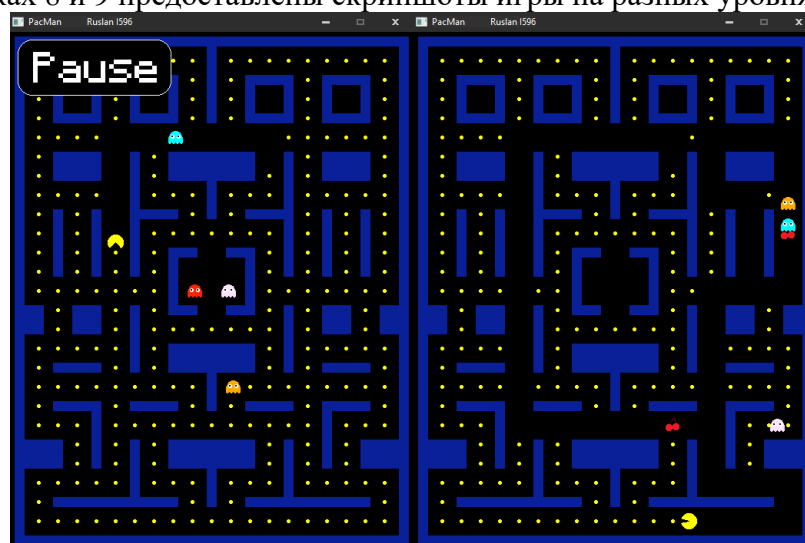


Рисунок 8 – геймплей на уровне “Small”

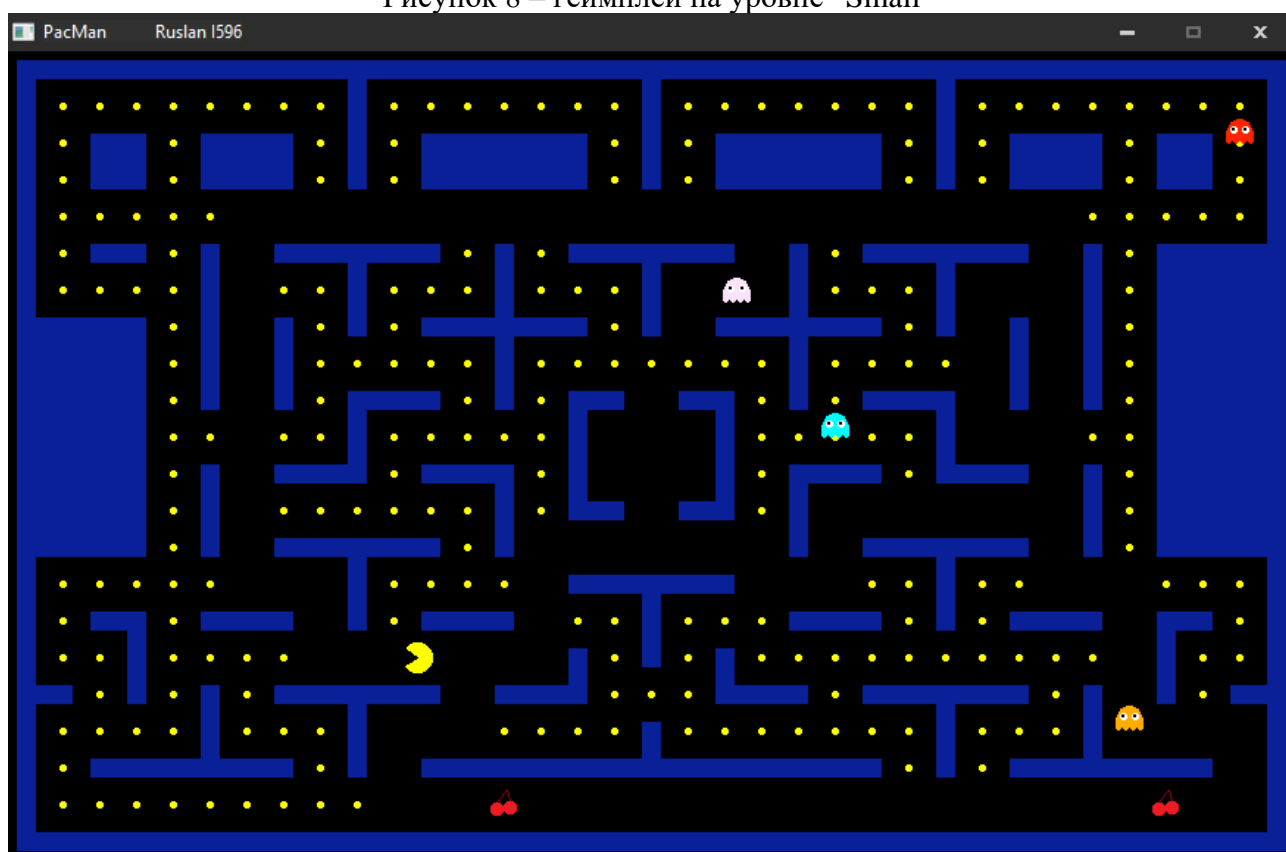


Рисунок 9 – геймплей на уровне “Big”

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ:

1. О.В. Арипова, О.А. Палехова, А.Н. Гущин. Программирование на языке высокого уровня. Лабораторный практикум.// Санкт-Петербург, 2014. -94 с.
2. Документация библиотеки SDL - <https://wiki.libsdl.org/> (дата обращения 05.07.2020)

Приложение А:

Приложение содержит в себе исходные файлы проекта.

Имя файла: «Приложение.rag».

Для разработки программы используется среда программирования Code::Blocks версии 1.2.15.

Для написания программы подключены основная библиотека SDL и дополнительные библиотеки для работы с текстом SDL_ttf.

Проект включает в себя файлы с текстом программы:

1. «main.cpp» – описывает последовательность действий по игре.
2. Файлы «ENTITY.h» и «ENTITY.cpp» содержат класс «ENTITY».
3. Файлы «HERO.h» и «HERO.cpp» содержат класс «HERO».
4. Файлы «ENEMY.h» и «ENEMY.cpp» содержат класс «ENEMY».
5. Файлы «MAZE.h» и «MAZE.cpp» содержат класс «MAZE».
6. Файлы «TESTURES.h» и «TESTURES.cpp» содержат класс «TESTURES».
7. Файлы «LOGICS.h» и «LOGICS.cpp» содержат класс «LOGICS».
7. Файл «menu_full.bmp» содержит фон всех пунктов меню.
8. Файл «level_1.bin» содержит информацию о маленьком лабиринте.
9. Файл «level_1.bin» содержит информацию о большом лабиринте.
10. Файл «score_1.bin» содержит информацию о рекордах для маленького лабиринта.
11. Файл «score_2.bin» содержит информацию о рекордах для большого лабиринта.
12. Файл «PacMan.exe» - итоговое приложение.
13. Вспомогательные файлы для работы SDL и итогового приложения:
 - 13.1 «libfreetype-6.dll».
 - 13.2 «SDL.dll».
 - 13.3 «SDL_draw.dll».
 - 13.4 «SDL-ttf.dll».
 - 13.5 «zlib.dll».
 - 13.6 «Pixel.otf».