

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования

«Уральский федеральный университет имени первого Президента  
России Б.Н. Ельцина»

Кафедра интеллектуальных информационных технологий

Оценка работы \_\_\_\_\_

Руководитель от УрФУ \_\_\_\_\_

Игра «Portal2D»

Отчет по проекту

Студенты *Костюченко А.А., Наговицин М.Д., Меньшиков Ю.А., Жуков В.В.*

Специальность (направление подготовки) *Программная инженерия*

Группа *ФО-160001*

Екатеринбург 2017

## Оглавление

<b>1 Постановка задачи</b>	<b>3</b>
<b>1.2 Ограничения</b>	<b>3</b>
<b>1.3 Задачи</b>	<b>3</b>
<b>2 Проект решения</b>	<b>4</b>
<b>2.1 Общее описание решения</b>	<b>4</b>
3.1 Модуль Map	4
3.2 Модуль Instruments	4
3.3 Модуль Gameplay	5
3.4 Модуль Menu	6
3.5 Records.cpp	7
3.6 Queue.h	7
3.7 SortingMethods.h	7
3.8 List.cpp	7

# **1 Постановка задачи**

## **1.1 Цель**

Создать консольное приложение, ознакомиться с основными алгоритмами и структурами данных.

## **1.2 Ограничения**

- по срокам выполнения работы
- по использованию языков программирования
- по использованию объектно-ориентированных средств языка программирования

## **1.3 Задачи**

- изучить литературу, касающуюся языка программирования C++
- написать программу, реализующую игру «Portal2D»
- закомментировать код
- написать отчет по написанной программе
- научиться работать в команде

## 2 Проект решения

### 2.1 Общее описание решения

В качестве приложения была выбрана игра «Portal2D», в качестве средства разработки – интегрированная среда разработки Microsoft Visual Studio 2015.

## 3 Описание решения

### 3.1 Модуль Map

Состоит из одной функции

```
MapShell** createMap(char* levelName, Hero* hero, Aim* aim, RedPortal* redPortal, BluePortal* bluePortal);
```

Функция отвечает за создание карты (считывая уровень из текстового файла, функция заполняет структурный массив необходимой информацией) Также функция заполняет поля в структурных переменных, хранящих координаты объектов на карте.

Также модуль содержит описания структур: MapShell, Hero, Aim, RedPortal, BluePortal.

Каждая из структур содержит поля для хранения различной информации: координат, очков здоровья (в случае Hero) и информации о содержимом ячейки на карте (MapShell)

### 3.2 Модуль Instruments

Состоит из двух функций:

```
void drawFrame(MapShell** map);
```

Данная функция, принимая на вход массив ячеек карты, выводит на экран содержимое каждой ячейки в зависимости от содержимого.

```
void clearScreen();
```

Данная функция отвечает за очистку экрана, отключает мерцание экрана

### 3.3 Модуль Gameplay

Модуль отвечает за процесс игры, за взаимодействие объектов на карте.

Запуск уровня осуществляется с помощью функции `levelOne()`. В ней содержится объявление структур (герой, прицел, красный портал, синий портал). Из функции осуществляются вызовы следующих функций:

- 1) Создание карты (считывание из файла): `createMap("Lvl_1.txt", hero, aim, redPortal, bluePortal);`
- 2) Отрисовка кадра `drawFrame(map)`
- 3) Запуск главной функции модуля: `performAnAction(map, hero, aim, redPortal, bluePortal)`

Модуль содержит в себе 14 функций, главной функцией является `performAnAction("Название_уровня.txt", hero, aim, redPortal, bluePortal)`. В данной функции запущен бесконечный цикл, который слушает нажатия пользователя. В зависимости от нажатой клавиши, происходит перемещение игрока или прицела.

Перемещение осуществляется при помощи функций: `moveLeft(HERO, hero, map)`, `moveRight(HERO, hero, map)`, `moveUp(AIM_DOT, aim, map)`, `moveDown(AIM_DOT, aim, map)`, `jump(hero, map)`. Данные функции переносят модель и координаты объекта в соответствующую направлению ячейку карты. А также есть функция, отвечающая за гравитацию: `gravity(map, hero)`. Если под персонажем отсутствует непроходимый блок, то его модель и координаты перемещаются на ячейку карты вниз.

В игре есть возможность устанавливать порталы, осуществляется это посредством функции: `setPortal(TYPE_OF_PORTAL, portal, aim, map)`. Считываются

координаты прицела, по которым устанавливается один из порталов.

### 3.4 Модуль Menu

Постановленная задача: создание многовкладочного меню.

Реализация: были создано 3 функции: Srand(), menu1(), menu2(). Вторая и третья функция представляют собой рекурсивную раскадровку нашего меню.

Были переработаны все функции, меню, конструкция (if ... else) заменена конструкцией (switch), устранен бесконечный цикл. Через перечисление были установлены константные значения клавиш, а также значений цветов. Функции меню были разбиты на мелкие задачи, отвечающие за начальный вывод на экран, отслеживание нажатия клавиш, а также передача значения в функцию выбора игрока. Была устранена экстренная передача управления в теле цикла.

Srand()- функция отвечающая за установку курсора на заданные координаты, а также за выбранный цвет.

Point(), Point 2(), Point 3()- функции отвечающие за начальный вывод на экран пунктов меню.

MoveToMenu (), MoveToMenu2 (), MoveToMenu 3()-функции отвечающие за отслеживания нажатых клавиш, а также подсчет перемещения пользователя.

RewriterCaseUpPoint(),RewriterCaseUpPoint2(),RewriterCaseUpPoint3()-функции отвечающие за перерисовку (смену цвета), двух действующих пунктов меню, для клавиши CaseUp.

RewriterCaseDownPoint(),RewriterCaseDownPoint2(),RewriterCaseDownPoint 3()-функции отвечающие за перерисовку (смену цвета), двух действующих пунктов меню, для клавиши CaseDown.

Hat()-функции отвечающая за вывод “шапки” меню.

UserSelection(), UserSelection2() ,UserSelection3()- функции отвечающие за определение установленного пункта меню, и вызов соответствующей за ним функции.

### **3.5 Records.cpp**

Модуль отвечает за всю работу с рекордами в игре.

Состоит из 7 функций в пространстве имён “records”, которые объявлены в Records.h. Также пространство “records” содержит структурный тип DataAboutTheChampion.

### **3.6 Queue.h**

Модуль отвечает за работу с двусвязной очередью.

Реализация функций находится в header файле из-за невозможности объявления в cpp файле, по причине использования template. На данный момент состоит из 3 функций в пространстве имён “queue”, которое также содержит 2 структурных типа Node и Queue.

### **3.7 SortingMethods.h**

Модуль отвечает за работу с алгоритмами сортировки.

Реализация функций находится в header файле из-за невозможности объявления в cpp файле, по причине использования template. На данный момент состоит из 4 функций в пространстве имён “sorting”.

### **3.8 List.cpp**

Модуль отвечает за работу с односвязным списком.

Состоит из 4 функций в пространстве имён “list”, которые объявлены в Lists.h. Также пространство “list” содержит структурный тип List.