Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

Курсовой проект по дисциплине

“Объектно-ориентированное программирование ”

Выполнил студент гр. 3530904/80003 Крайнов А.К.

Руководитель Коптев Д.А.

«22» Декабря 2019 г.

СОДЕРЖАНИЕ

[1. Введение 3](#_Toc28009468)

[**1.1** **Цель работы:** 3](#_Toc28009469)

[**1.2** **Задачи** 3](#_Toc28009470)

[**1.3** **Основная информация о программе** 3](#_Toc28009471)

[2. Основная часть работы 4](#_Toc28009472)

[**2.1** **Архитектура программы** 4](#_Toc28009473)

[**2.2** **Пользовательское взаимодействие с программой** 5](#_Toc28009474)

[**2.3** **Код работы** 5](#_Toc28009475)

[**2.4** **Необходимые зависимости** 6](#_Toc28009476)

[**2.5** **Скриншоты** 7](#_Toc28009477)

[3. Заключение 8](#_Toc28009478)

[4. Список использованной литературы 8](#_Toc28009479)

1. Введение

## **Цель работы:**

Разработка объектно-ориентированной программы с графическим интерфейсом.

## **Задачи**

1. Разработка архитектуры (структуры классов) программы. К каждому заданию предлагается один из вариантов архитектуры.
2. Реализация классов, составляющих архитектуру программы.
3. Отладка и тестирование программы.
4. Документирование программы и написание отчета о проделанной работе.

## **Основная информация о программе**

Курсовой проект представляет собой приложение для поиска пути на двумерной плоскости, поделенной на клетки, с графикой.

Запускается с аргументами "M N S", где M - количество столбцов на плоскости, N - количесво рядов на плоскости, S - строка dfs или astar - для выбора используемого алгоритма поиска пути. Параметр astar позволит запустить программу с эффективным алгоритмом поиска пути A\*, dfs с неэффективным алгоритмом поиска в глубину.

После запуска программы отображается окно с инструкциями для пользователя, можно добавлять/удалять/изменять местоположение объектов.

При написании программы использовался 14 стандарт С++ Программа создана с применением библиотеки SFML, поэтому для сборки необходимо ее подключение и линковка.

1. Основная часть работы

## **Архитектура программы**

Исходный код програмы состоит из :

* Файлов PathFinder.hpp, PathFinder.cpp в папке PF – файлы базового класса алгоритма поиска пути. PathFinder содержит в себе std::unordered\_set пар координат препятствий на плоскости, и основным методом его взаимодействия с пользователем является метод pathBetween, возвращающий std::list координат точек пути.
* Файлов Astar.hpp, Astar.cpp класса Astar, алгоритма поиска пути методом А\*, реализующего класс PathFinder. В качестве эвристики для этого алгоритма выбрано эвклидово расстояние по прямой, которое задано в методе calculateHeuristics. При запуске программы с параметром astar запустится этот алгоритм. Путь построенный таким алгоритмом будет оптимален.
* Файлов Dfs.hpp, Dfs.cpp класса Dfs, алгоритма поиска пути методом обхода графа путей в глубину, реализующего класс PathFinder. Этот алгоритм чрезвычайно неэффективен и имеет демонстративную службу. При запуске программы с параметром dfs запустится этот алгоритм, причем можно будет заметить что программа, как правило, обойдет большинство свободных точек на плоскости прежде чем найдет конечную точку.
* Папки resorses, с файлами strings.hpp – строковыми константами программы, и OpenSansRegular.ttf – файла шрифта OpenSans для програмы.
* Файлов Grid.hpp, Grid.cpp –класса Grid отвечающего за прорисовку графики и пользовательского интерфейса. В нем отрисовывается необходимого размера сетка (метод drawGrid) , метки начальной, конечной точки (метод drawPoint), препятствий (метод drawObstacles), пути (метод drawPath), и пользовательского интерфейса (метод drawUI). Внутри него лежит enum state который представляет из себя конечный автомат состояний класса Grid для отрисовки UI выбора режима выставления объектов.
* Файлов InputListener.hpp, InputListener.cpp – класса InputListener, который отвечает за обработку пользовательского ввода
* Файла main.cpp – основного файла программы в котором происходит инициализация необходимых ресурсов, создание окна и необходимых классов и основной цикл программы.

## **Пользовательское взаимодействие с программой**

Пользователь запускает программу с параметрами с тремя параметрами – первый отвечает за ширину сетки поиска пути, второй за длину, третий за выбранный алгоритм поиска пути. Первый параметр варьируется от 1 до 38, второй от 1 до 15, третий параметр может быть строкой astar либо строкой dfs.

После того как пользователь задал начальные параметры запускается окно в котором можно выбрать режим взаимодействия клавишами 1,2,3.

* Клавиаша 1 соответствует режиму расстановки объектов: По нажатию левой кнопки мыши объект поставится на плоскость, левой – уберется
* Клавиша 2 соответствует режиму установки начальной точи. По нажатию левой кнопки мыши на плоскость поставится начальная точки.
* Клавиша 3 соответствует режиму установки конечной точки. По нажатию левой кнопки мыши на плосткость поставится конечная точка.

Пользователь может включить отображение пути нажав на клавишу пробел или выйти из программы нажав на клавишу Esc.

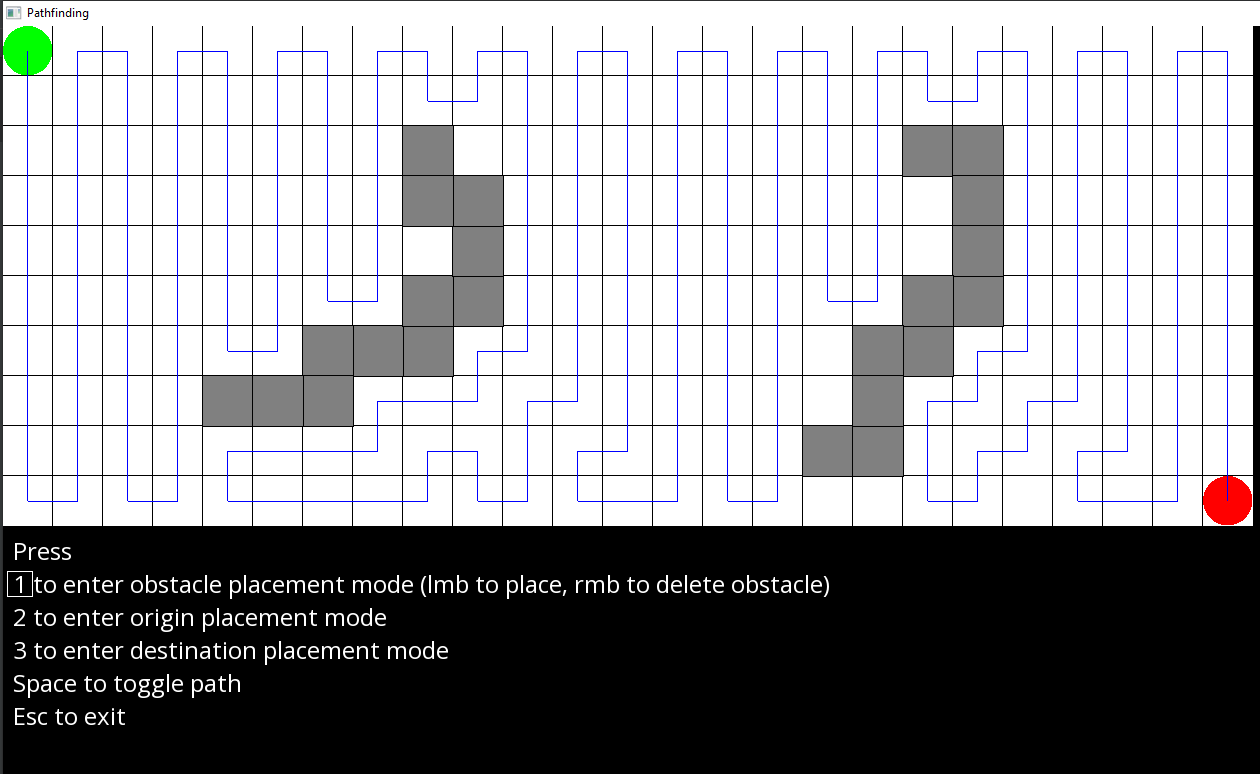
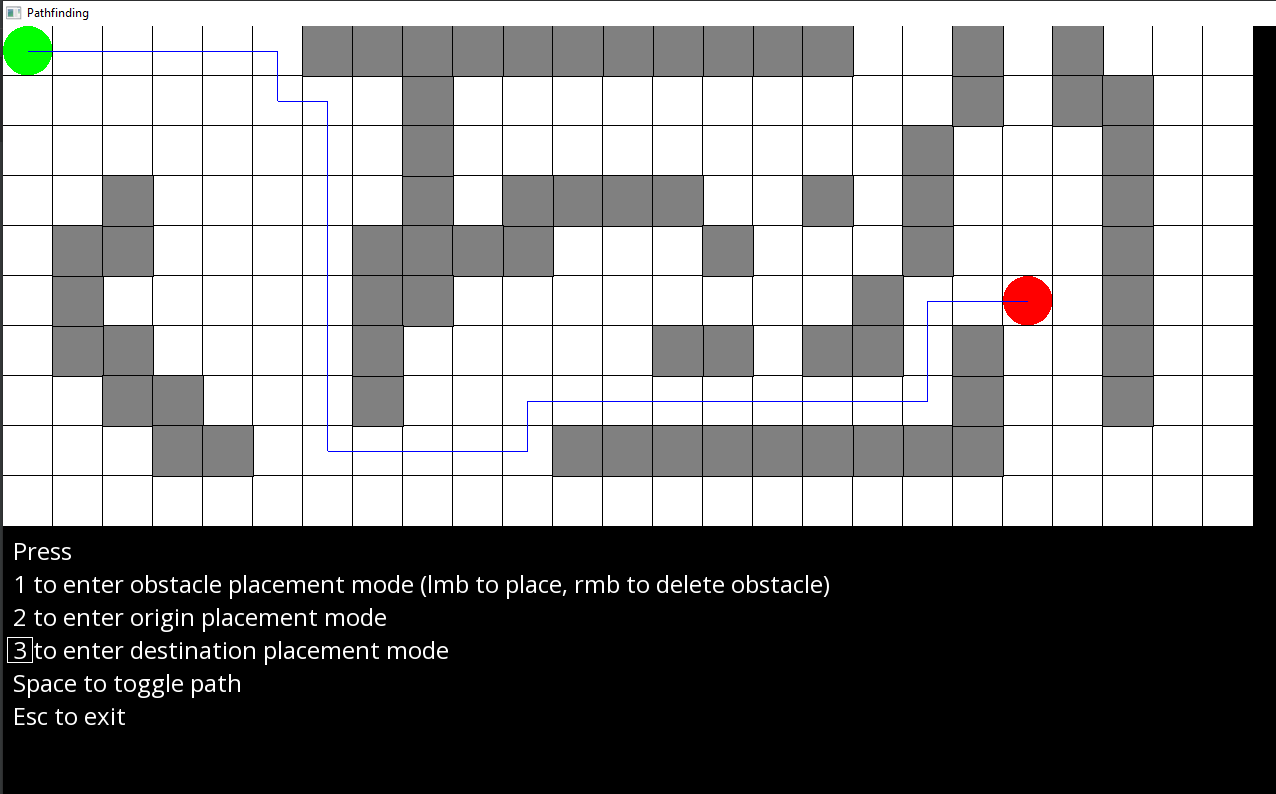
## **Код работы**

Код работы находится на Gitlab по ссылке <https://gitlab.com/KraynovAlexander/spbspu-cxx-term-works/tree/master/kraynov-alexander>

## **Необходимые зависимости**

Программа создана с использованием библиотеки SFML, в частности модулей –lsfml-graphics, -lsfml-system, -lsfml-window, которые должны быть переданы линковщику. В работе был использован заголовочный файл SFML/Graphics.h, местоположение которого тоже должно быть известно для успешной компиляции проекта.

## **Скриншоты**



1. Заключение

В результате выполнения данной работы была разработана объектно-ориентированная программа с графическим интерфейсом, разработана архитектура классов программы, проведена отладка и тестированиe. Были приобретены навыки разработки программного продукта с графическим интерфейсом.

1. Список использованной литературы

1. Галовиц Я. С++ 17 STL – стандартная библиотека шаблонов: отдельное издание, изд. Питер, 2018 г.

2. ИТМО, Викикоспект - Алгоритм А\*, [https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_A\*](https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_A*), 2019 г.

2. Статья cppreference об std::unordered\_set <https://en.cppreference.com/w/cpp/container/unordered_set>

3. СПбПУ – Краткое описание библиотеки SFML, 2018г.

4. Атчисон Л. – Масштабирование приложений. Выращивание сложных систем. Изд. Питер, 2018г.