**Спецификация**

**Назначение программы**

Разработка программного приложения на языке Java, предназначенного для визуализации алгоритма А\*.

**Описание алгоритма**

A\* пошагово просматривает все пути, ведущие от начальной точки в

конечную, пока не найдёт минимальный. В начале работы просматриваются точки, соседние с начальным. На каждом этапе алгоритм оперирует с множеством путей из начальной точки до всех ещё не раскрытых соседей клетки — множеством частных решений, которое размещается в очереди с приоритетом. Приоритет пути определяется по значению f(x) = g(x) + h(x), где f(x) – предполагаемая стоимость пути от текущей точки до конца, g(x) – стоимость кратчайшего пути от начала до текущей точки, h(x) – эвристическая функция. Если текущее частное решение дает лучше значение g(x), то меняем значение g(x) для текущей клетки и выписываем новый кратчайший путь до точки. Алгоритм продолжает свою работу до тех пор, пока значение f(x) конечной точки не окажется меньшим, чем любое значение в очереди, либо пока все клетки не будут просмотрены. В итоге получается путь от начальной точки до конечной с наименьшей стоимостью.

**Входные данные:**

Будет доступно 2 способа ввода:

1. Ввод из файла.
2. Выбор ключевых точек с помощью мыши.

В случае ввода из файла:

Данные представлены в файле “input.txt”. Из файла считываются 2 числа по которым устанавливается размер поля, затем само поле, где символ “.” означает что сектор свободен, а “w” что в секторе находится препятствие.

После поля в файле находятся 4 целых числа, первые 2 обозначают координаты начальной точки, остальные – конечной.

В случае выбора ключевых точек с помощью мыши нужно будет отметить стартовую и конечную точки, а также заблокированные сектора.

Также будет доступна опция выбора эвристической функции. Предполагаемые эвристичекие функции для реализации:

1. H(x) = |goal.x – current.x| + |goal.y – current.y|
2. H(x) = sqrt((goal.x – current.x)^2+(goal.y - current.y)^2)

В процессе программы можно будет выбрать эвристическую функцию, с помощью которой будут происходить расчеты.

**Выходные данные:**

На экран выводится поле соответствующего размера. Сектора поля будут раскрашены в различные цвета в зависимости от их состояния (просмотренные, активные, свободные и занятые сектора будут раскрашены различным цветом). В процессе работы программы сектора будут перекрашиваться. Также будет добавлено дополнительное поле, которое будет содержать комментарии.

Возможные комментарии:

1. Добавляем клетку (координаты клетки) в OpenSet
2. Добавляем клетку (координаты клетки) в ClosedSet
3. Удаляем клетку (координаты клетки) из OpenSet
4. Удаляем клетку (координаты клетки) из ClosedSet
5. Выбираем клетку (координаты клетки) в качестве продолжения пути, так как значение f(x) для нее минимально

Итоговый путь будет выделен отдельным цветом. Также будет выводиться его стоимость.

**Требования к программе:**

Программа реализует алгоритм A\* с визуализацией в лабиринте, и пошаговой визуализации работы алгоритма.

**Требования к пользователю**

При запуске проекта будет создаваться отдельное окно (рисунок 1). Данная

программа будет иметь 1 окно. Окно представляет собой меню и карту.

Сверху располагаются кнопки, внизу находится окно вывода лабиринта.

Будут доступны следующие кнопки:

1)Restart - начать программу сначала.

2)Start - начать выполнение программы.

3)Next step - следующий шаг алгоритма (при работе в пошаговом режиме).

4)Stop - остановить программу на текущем шаге.

5)Continue - продолжить работу программы.

6) Load from file – загрузить данные из файл

Рисунок￼

Рисунок 1

**Структуры данных**

Class Model - содержит в себе состояние лабиринта.

Class Controller - согласовывает работу между View и Model.

Class View - отображает состояние модели и ведет диалог с пользователем.

Class Field – представление лабиринта.

Class Point – отвечает за координаты.

**План разработки**

02.07.18 Согласование спецификации и плана разработки.

04.07.18 Прототип

06.07.18 Конечный проект. Первая версия. Предоставление проекта с полной

функциональностью. Проект пояснительной записки

08.07.18 Конечный проект. 2-я версия. Будет добавлена возможность пошаговой реализации алгоритма и поле с комментариями. Приемочные тесты

10.07.18 Конечный проект. Готовая для сдачи версия и пояснительная

записка с исправлениями по замечаниям. Будет добавлена возможность выбора эвристической функции и выбора ключевых точек с помощью мыши.

**Распределение обязанностей**

**Воропаев А.О.** Реализация алгоритма A\* на языке Java.

**Вайгачев А.О** Разработка пользовательского интерфейса и визуализация.

**Жук К.А** Оформление пояснительной записки и разработка приемочных тестов и тестирование. Разработка структур данных