

Исходные требования к программе

Начало работы.

Пользователю доступен начальный экран (до запуска алгоритма), на начальном экране можно:

- Нарисовать граф (добавление вершин (двойной клик ЛКМ), протягивание рёбер между ними ("Shift"+перетаскивание курсора на другую вершину, удерживая ЛКМ), указание весов рёбер (клик ЛКМ по ребру, указание в меню на правой панели веса ребра), удаление вершин(клик ЛКМ по вершине, нажатие кнопки "Delete" в правом меню), удаление рёбер(клик ЛКМ по ребру, нажатие кнопки "Delete" в правом меню))
- Удалить граф (полностью стирается нарисованный граф)
- Сохранить нарисованный граф (число вершин, затем идут разделённые специальным символом "/" наборы пар вида <вершина:вес_ребра> (может быть несколько пар в одной последовательности), являющиеся списками смежности вершин, номера которых соответствуют порядковым номерам наборов в файле)
- Загрузить граф из файла, отрисовать его (загрузить из сохранённого ранее файла, к примеру)
- Начать выполнение программы (кнопка "Begin")

Подготовка к выполнению алгоритма.

После нажатия кнопки "Begin" пользователю предлагается ввести некоторые начальные данные для работы алгоритма:

- Временной интервал, который определяет частоту выполнения шагов алгоритма и отрисовки в автоматическом режиме (см. далее)
- Начальную вершину. Выбрать её можно, кликнув по какой-либо вершине (можно выбрать несколько раз). Затем необходимо подтвердить выбор нажатием кнопки "Approve"

Выполнение алгоритма.

Алгоритм начинает работу, рассматривая начальную вершину. Пользователю доступен выбор режима: автоматический или пошаговый (изначально по умолчанию включён пошаговый).

- Автоматический: включается кнопкой "Auto", каждый шаг выполняется через указанный временной интервал, кнопка перехода к след. шагу "Step" неактивна, можно приостановить автоматический режим, нажав на кнопку "Auto" во время выполнения
- Пошаговый: при условии, что автоматический режим отключён, кнопка перехода к следующему шагу "Step" активна и приводит к его выполнению

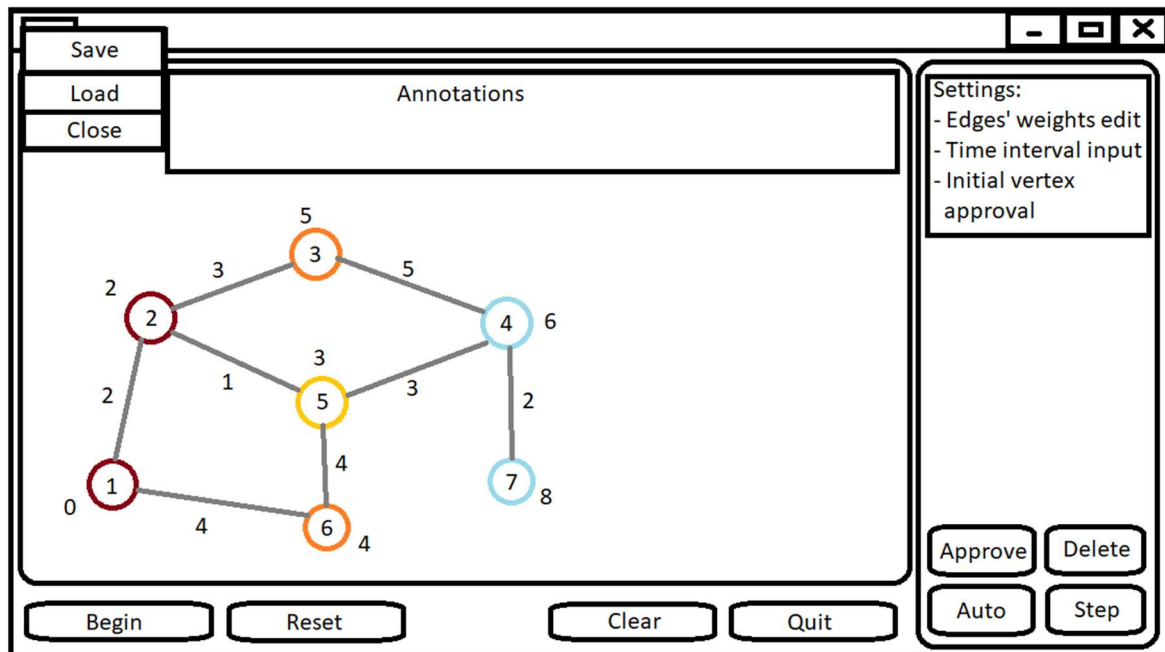
В процессе работы алгоритма можно включать/отключать автоматический режим, если отключён - последовательно выполнять шаги в пошаговом режиме. Также можно нажать кнопку "Reset", которая остановит выполнение алгоритма, удалит все промежуточные данные и вернёт программу в режим "Начало работы", сохранив при этом нарисованный граф и вернув начальную раскраску (все вершины помечены одним цветом 3, см. далее)

На каждом шаге алгоритма:

- Отработанные вершины (до которых путь уже найден) помечаются цветом 1
- Вершины в очереди с приоритетом, которые подлежат рассмотрению, помечаются цветом 2
- Вершины, до которых ещё не найден хоть какой-либо (пусть не самый короткий) путь, помечаются цветом 3
- Текущая рассматриваемая вершина помечается цветом 4. Путь до этой вершины найден на данном шаге и указывается рядом с ней в виде подписи, о данном событии сообщается пользователю, сообщение имеет форму краткой заметки в специальном поле.

По завершении работы алгоритма программа выводит итоги работы алгоритма (например, вершина, длина пути до неё, последовательность вершин, являющихся этим путём - так для каждой вершины) и возвращается в режим "Начало работы".

Предполагаемый вид интерфейса.



План разработки

1. Разработка алгоритма Дейкстры, реализация классов "Vertex" и "Solver". "Vertex" – класс вершины, содержащий необходимые данные и методы для работы с каждой вершиной, "Solver" – класс, содержащий граф и методы для взаимодействия с ним, а также реализацию алгоритма Дейкстры.
2. Разработка GUI, позволяющего пользователю создавать объекты на поле без привязки к структуре данных (графу) и реализующего функции взаимодействия с элементами окна.
3. Привязка графического интерфейса к методам взаимодействия со структурой данных (графом).
4. Тестирование программы выполнением алгоритма на различных входных данных, а также использованием доступного функционала GUI. Исправление ошибок.

Распределение ролей в бригаде

- Павлов Роман: создание структур данных и их методов, необходимых для реализации алгоритма Дейкстры; привязка GUI к методам взаимодействия со структурами данных.

- Рыжих Роман: разработка GUI, дизайн интерфейса, привязка GUI к методам взаимодействия со структурами данных.
- Сорочина Мария: тестирование алгоритма и GUI, оформление и структурирование документации.