

The
MAZE

Автор: Пимкин Артем

Научный руководитель: И. Р. Дединский

2012

Цель и задачи проекта

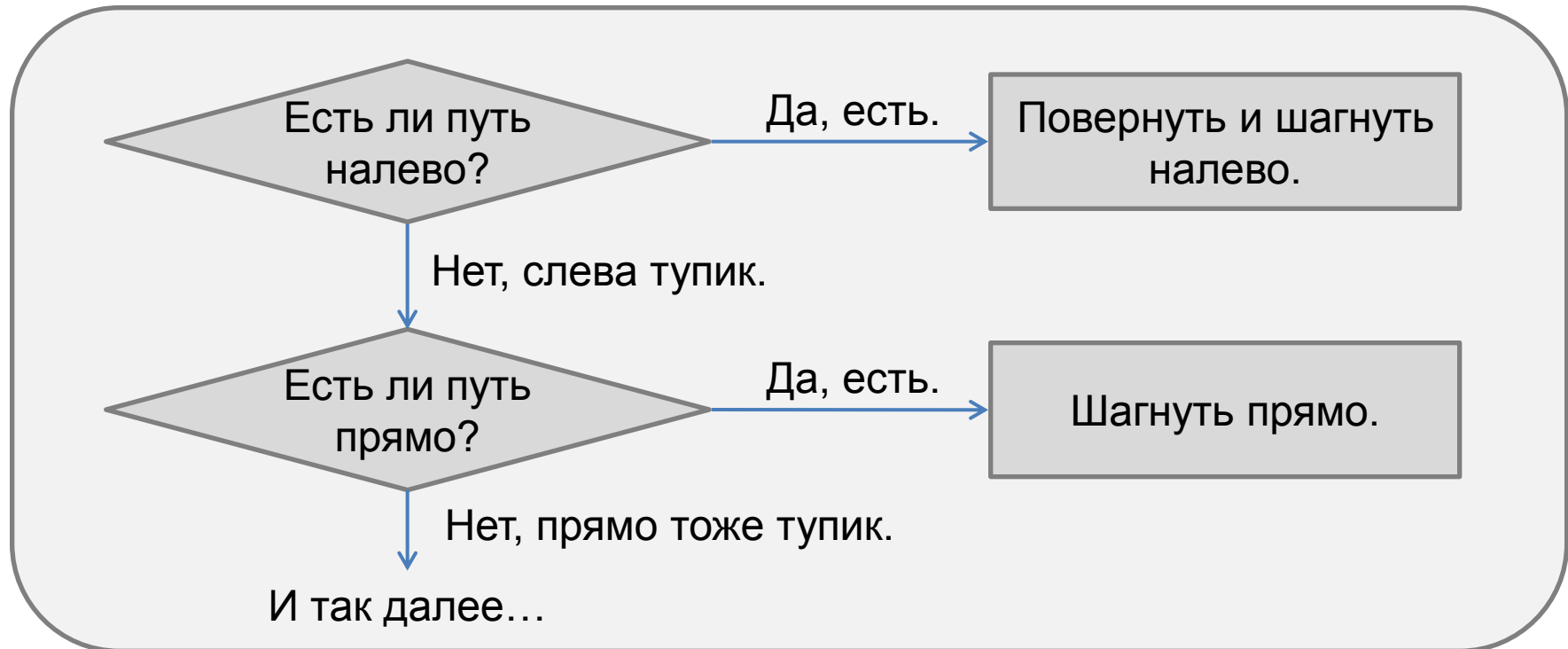
Цель работы – сравнить технологию ручного кодирования алгоритмов и автоматную технологию их реализации

Задачи проекта:

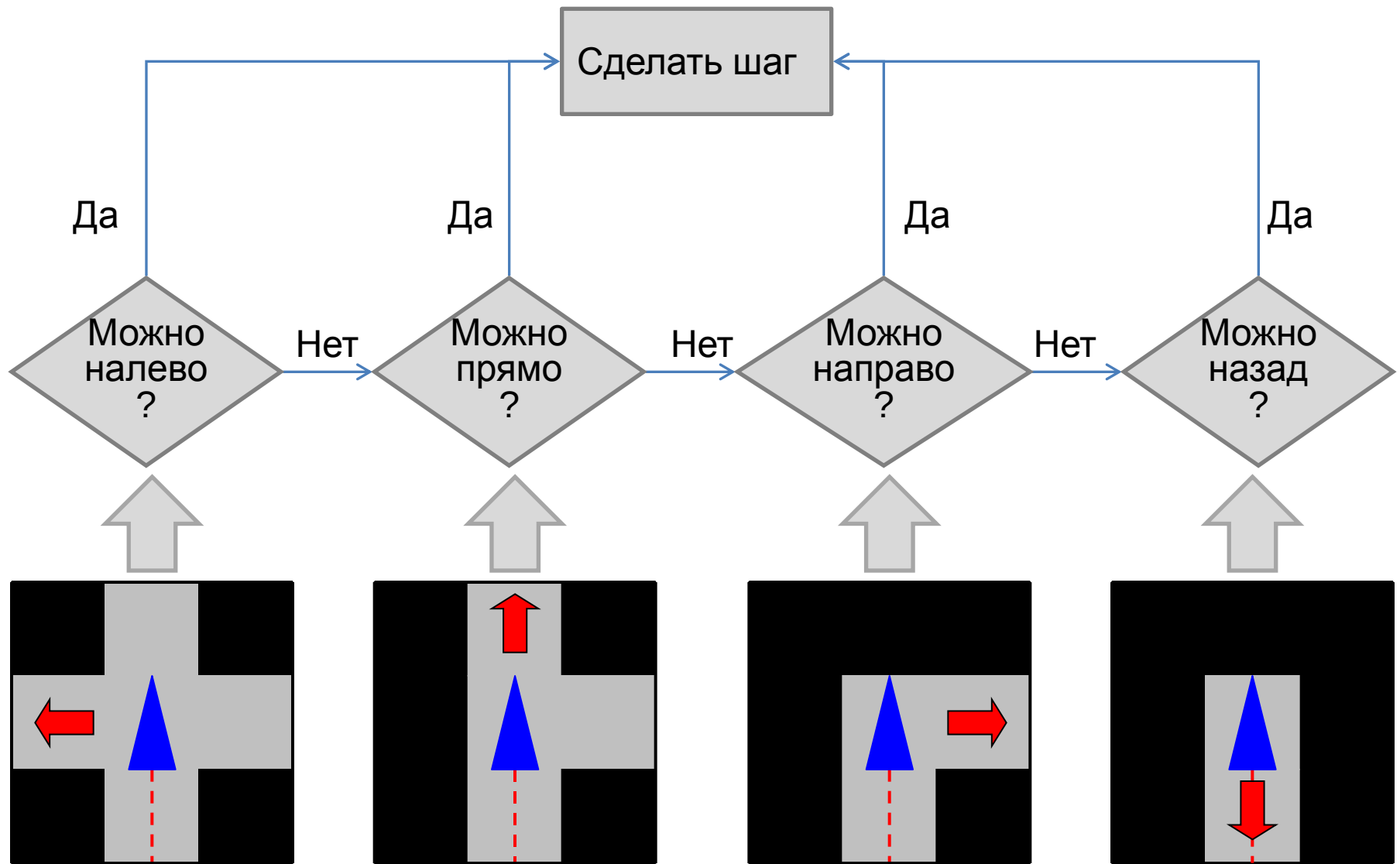
- Разработать программу с удобным пользовательским интерфейсом, работающую с агентом, находящим выход из произвольного односвязного лабиринта.
- Разработать алгоритм генерации произвольного односвязного лабиринта.
- Использовать при создании программы конечные автоматы и, в частности, switch-технология.
- Реализовать две версии программы. Одну с описанным вручную алгоритмом поведения агента, а вторую со сгенерированным из схемы конечного автомата кодом

Алгоритм поиска выхода

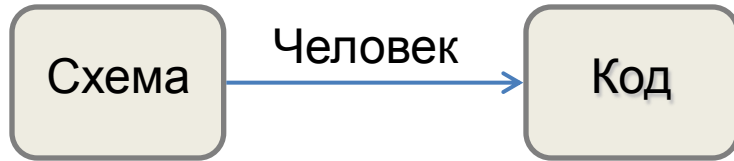
- Правило левой руки - метод выхода из односвязного (не содержащего замкнутых маршрутов) лабиринта
- Оптимальное описание алгоритма правила левой руки возможно при помощи конечного автомата.
- Правило заключается в выборе наиболее левого пути.



Алгоритм поведения агента

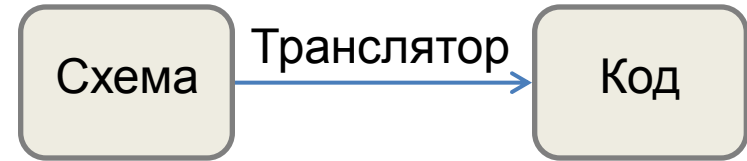


Сравнение способов разработки



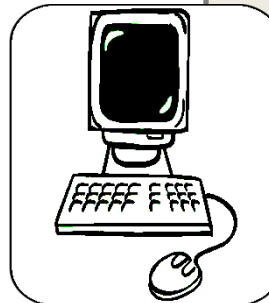
Код алгоритма написан вручную

- Значительная вероятность ошибки.
- Низкая скорость поиска и исправления алгоритмических ошибок.
- Необходимость рисовать схему для документирования алгоритма.



Код сгенерирован из схемы автомата

- Отсутствие человеческого фактора
- Быстрое исправление ошибок в алгоритме за счет внесения корректировок в схему.
- Необходимость рисовать схему. Но это - все, что необходимо сделать для результата.

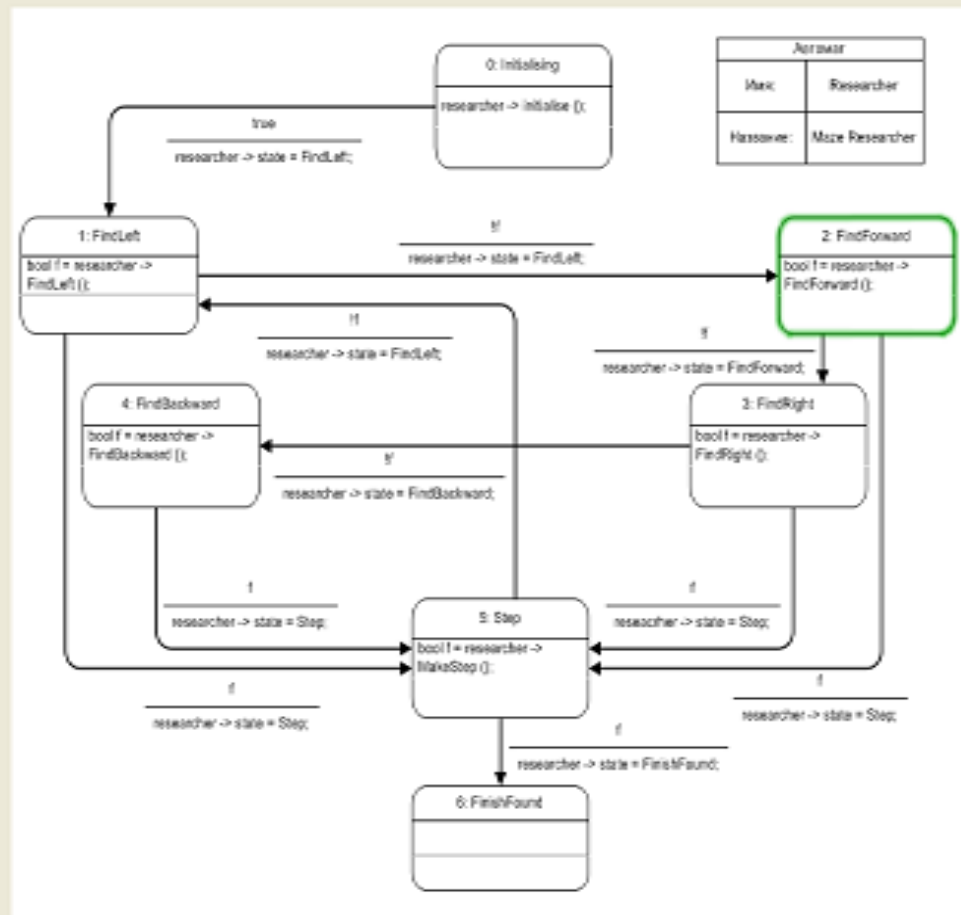


Программа Maze: интерфейс

Интерфейс программы Maze реализован для максимального удобства пользователя



State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 5:step}, Transition - Left way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 3:findRight}, Transition - Forward way was not found
State - {3:findRight -> 5:step}, Transition - Right way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 5:step}, Transition - Left way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 5:step}, Transition - Left way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 3:findRight}, Transition - Forward way was not found
State - {3:findRight -> 5:step}, Transition - Right way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found



Generate

One step

Autopass

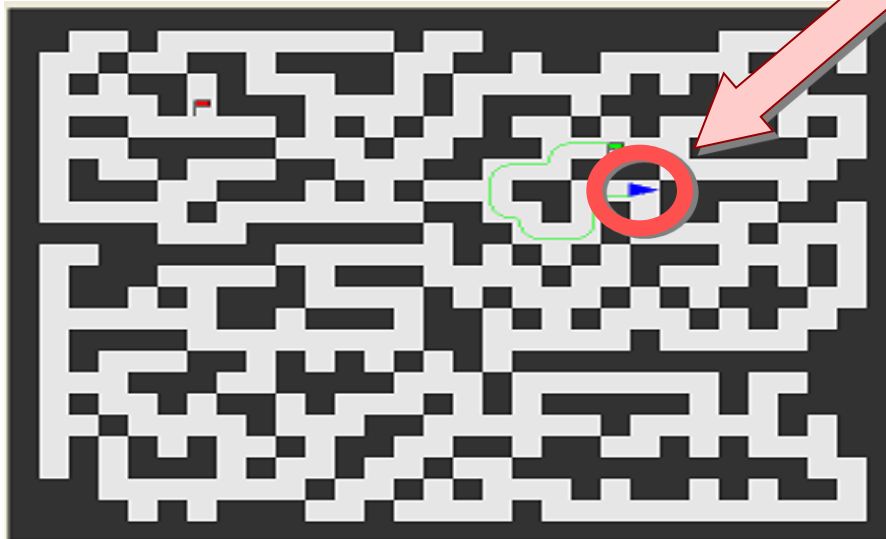
Stop

About

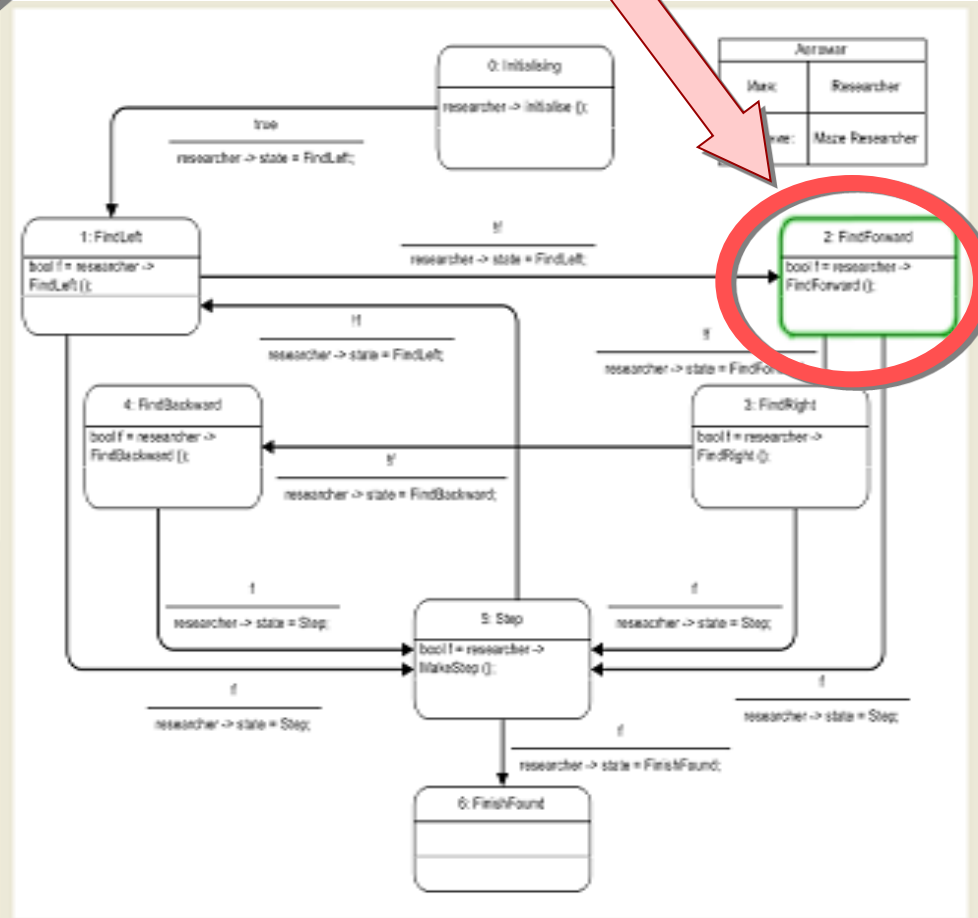
Quit

Программа Maze: интерфейс

Визуализация работы конечного автомата с помощью обновляющихся в реальном времени схемы **лабиринта** и схемы **алгоритма**.



State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found
State -> {2: findForward -> 3: step}, Transition - forward way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 3: step}, Transition - Left way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found
State -> {2: findForward -> 3: findRight}, Transition - Forward way was not found
State -> {3: findRight -> 3: step}, Transition - Right way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 3: step}, Transition - Left way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found
State -> {2: findForward -> 3: step}, Transition - forward way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 3: step}, Transition - Left way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found
State -> {2: findForward -> 3: findRight}, Transition - Forward way was not found
State -> {3: findRight -> 3: step}, Transition - Right way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found
State -> {2: findForward -> 3: step}, Transition - forward way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 3: step}, Transition - Left way was found
State -> {3: step -> 1: findLeft}, Transition - Finish was not found
State -> {1: findLeft -> 2: findForward}, Transition - Left way was not found



Generate

One step

Autopass

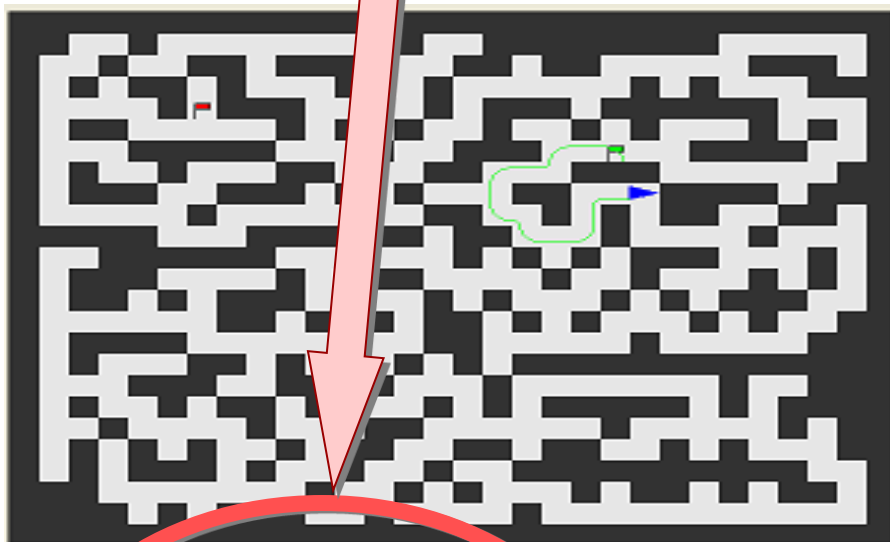
Stop

About

Quit

Программа Maze: интерфейс

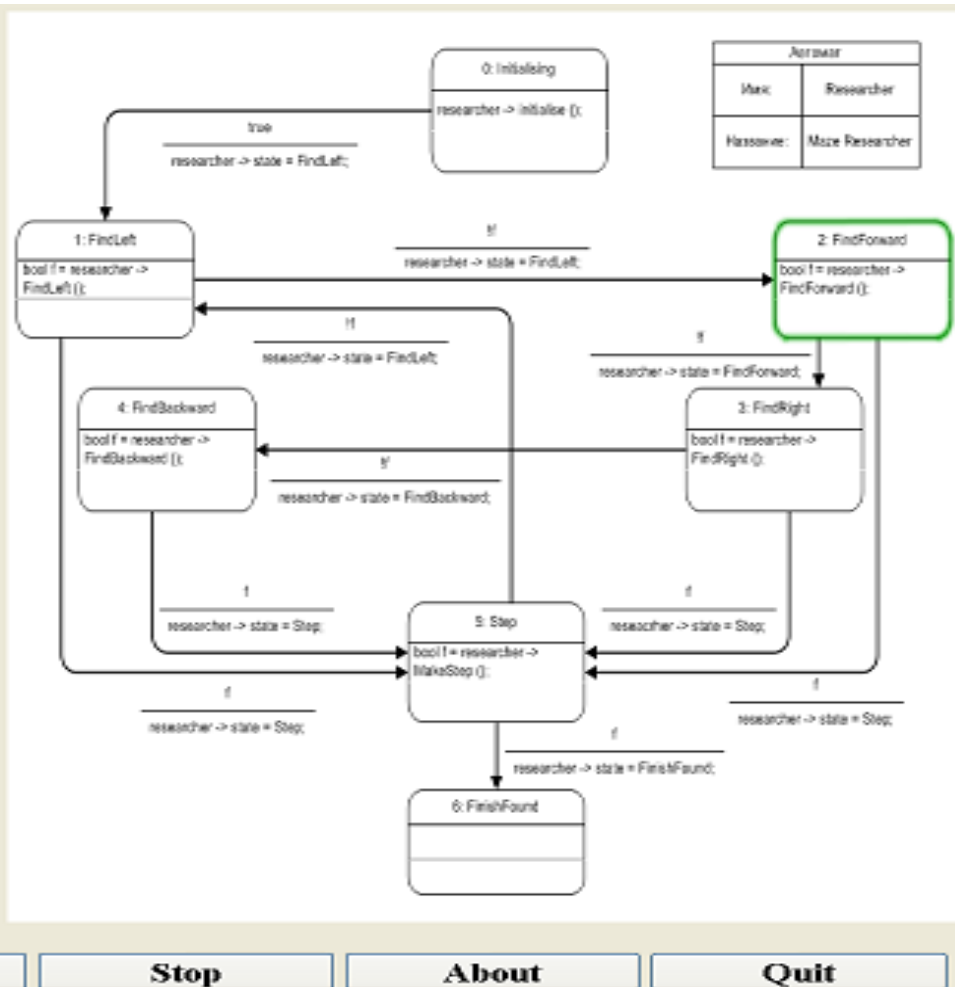
Консоль. Распечатка состояний и переходов автомата.
Возможность выполнения сценариев работы программы.



```

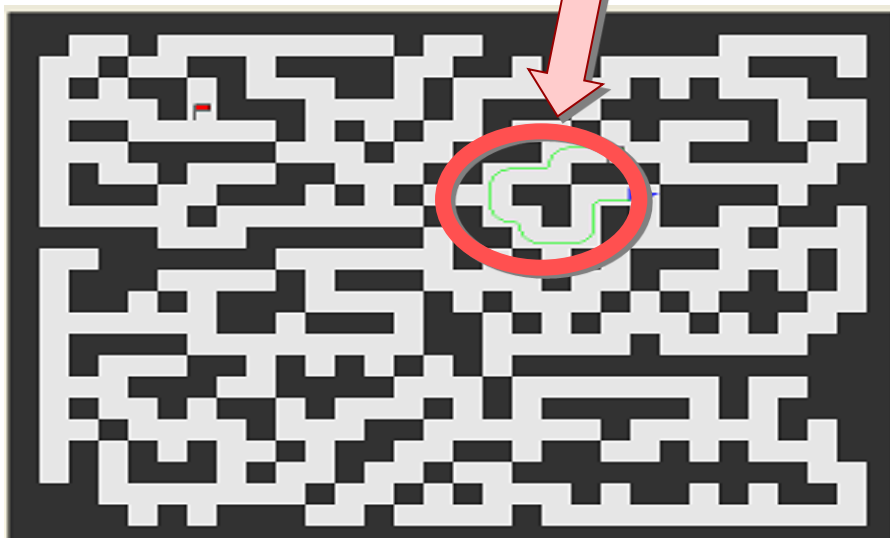
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 5: step1, Transition - forward way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 5: step1, Transition - Left way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 3: findRight, Transition - Forward way was not found
State -> 3: findRight -> 5: step1, Transition - Right way found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 5: step1, Transition - Left way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 5: step1, Transition - forward way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 5: step1, Transition - Left way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 5: step1, Transition - forward way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 3: findRight, Transition - Forward way was not found
State -> 3: findRight -> 5: step1, Transition - Right way found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found
State -> 1: findLeft -> 2: findForward, Transition - Left way was not found
State -> 2: findForward -> 5: step1, Transition - forward way was found
State -> 5: step1 -> 1: findLeft(), Transition - Finish was not found

```

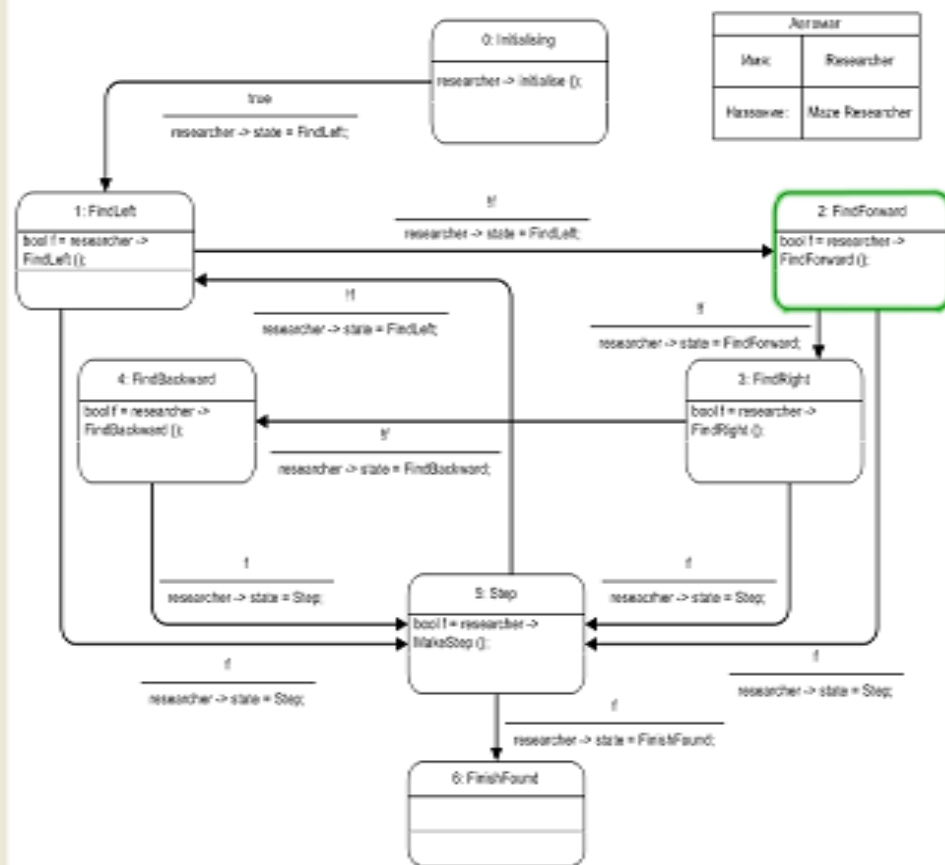


Программа Maze: интерфейс

Подсветка пути, пройденного агентом,
для простоты оценки результативности работы программы.



State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 5:step}, Transition - Left way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 3:findRight}, Transition - Forward way was not found
State - {3:findRight -> 5:step}, Transition - Right way found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 5:step}, Transition - Left way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 3:findRight}, Transition - Forward way was not found
State - {3:findRight -> 5:step}, Transition - Right way found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found
State - {2:findForward -> 5:step}, Transition - forward way was found
State - {5:step -> 1:findLeft}, Transition - Finish was not found
State - {1:findLeft -> 2:findForward}, Transition - Left way was not found



Generate

One step

Autopass

Stop

About

Quit

Выводы

- **Разработана программа** с удобным пользовательским интерфейсом, работающая с агентом, находящим выход из произвольного односвязного лабиринта.
- **Разработан алгоритм** генерации произвольного односвязного лабиринта.
- При создании программы были использованы **конечные автоматы, автоматная парадигма** и, в частности, **switch-технология**.
- Реализованы **две версии** программы. Одна с **описанным вручную** алгоритмом поведения агента, а вторая со **сгенерированным из схемы** конечного автомата кодом.
- В результате проделанной работы было показано, что **использование сгенерированного кода удобнее**, чем описание алгоритма вручную.

Благодарности

- Работа была выполнена в рамках учебного задания экспериментальной площадки по проектной деятельности И. Р. Дединского

Использованная литература

- Д. Хопкрофт. Введение в теорию автоматов, языков и вычислений.
- Н. И. Поликарпова, А. А. Шалыто. Автоматное программирование.
- М. Шлее. Qt 4: Профессиональное программирование на C++.

**СПАСИБО ЗА
ВНИМАНИЕ!**