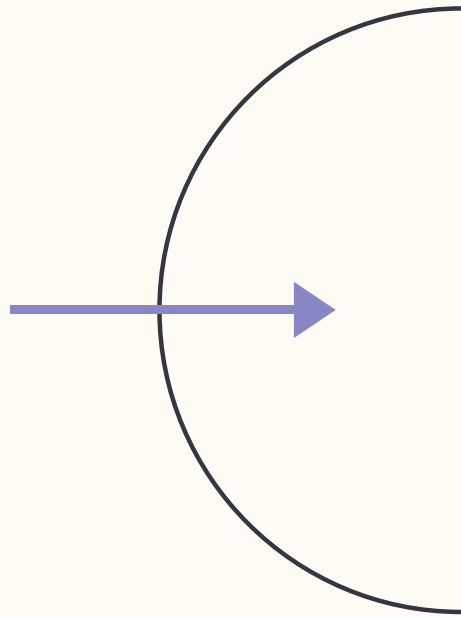


Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»

Курсовая работа на тему : Система  
управления движения  
беспилотного транспорта для решения  
задачи нахождения оптимального  
маршрута в  
неизвестной местности

Выполнил студент : Коробейников Кирилл Владимирович гр.09-732  
Научный руководитель : Ассистент Багавеев В.А.



# Цели и задачи

Целью курсовой работы была разработка системы и разбор алгоритмов способных генерировать лабиринты и решить задачу нахождения оптимального пути в неизвестной местности.

Достижение указанной цели осуществлялось путем решения следующих основных задач:

- 1) Раскрытие понятие беспилотного транспорта и рассмотр основных подходов решения задач
- 2) Выбор наилучшего алгоритма и углубленное изучение теории
- 3) Построение симуляции(лабиринта) для решения поставленной задачи
- 4) Реализация алгоритма и вывод

# Основной подход - классический

## Локализация

Модуль  
локализации  
отвечает за то, чтобы  
машина понимала,  
где она находится

## Распознавание

Модуль  
распознавания — за  
то, что находится  
вокруг  
машины.

## Планирование

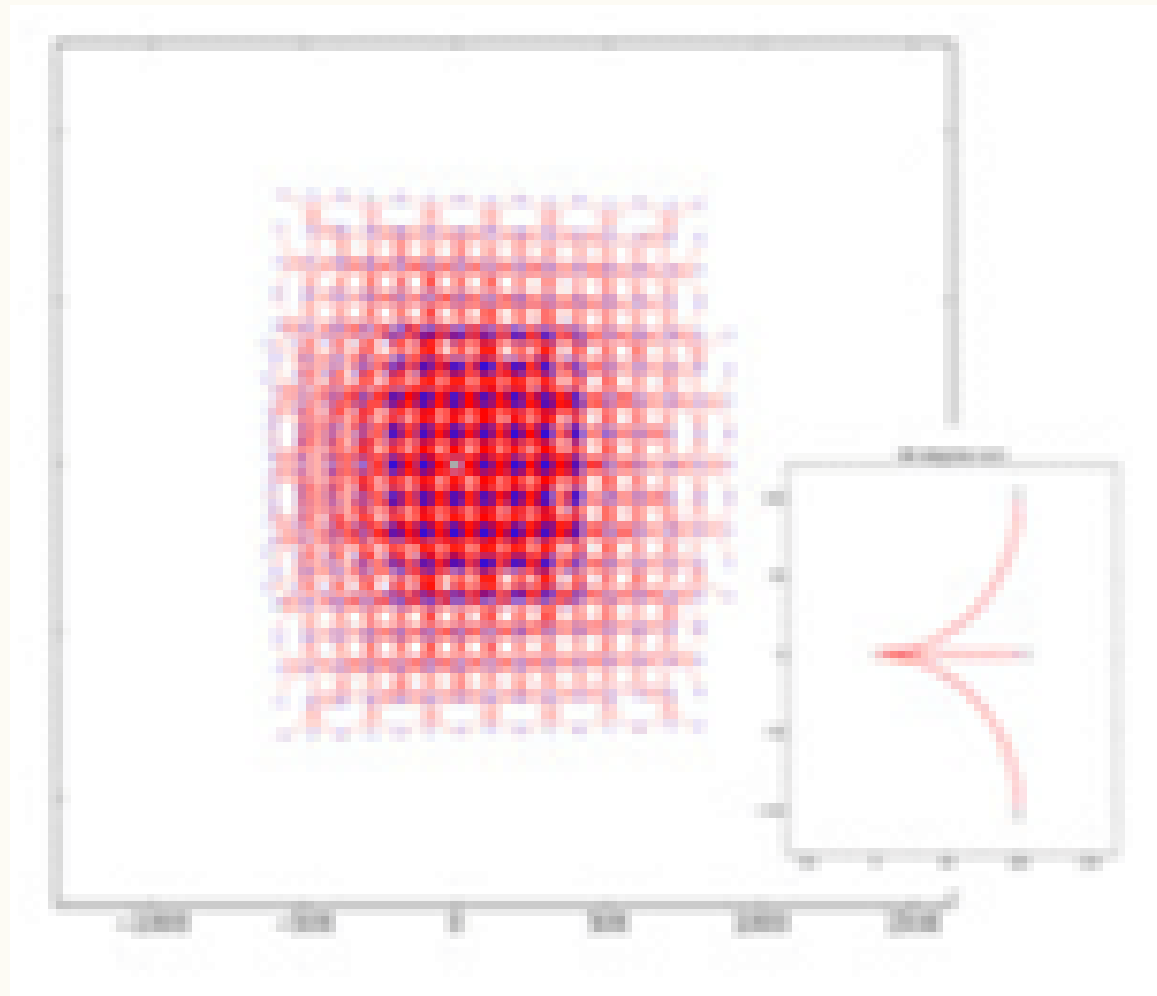
Модуль  
планирования  
обладает  
информацией о том,  
что  
находится вокруг, и  
зная, куда хочется  
приехать, строит  
маршрут

## Управление

Модуль управления  
говорит, как же ехать  
по  
маршруту, чтобы  
приехать, выполнить  
эту траекторию

## АЛГОРИТМЫ НА ГРАФАХ

Вот стоит машина, мы можем понять, где она стоит, но в реальности графа никакого нет, вершин ребер на дороге не нарисовано. Нам этот граф нужно как-то придумать самим. Первое, что делаем: разбиваем все пространство на клетки, рассмотрим маленькие клеточки и скажем, что есть вся наша поверхность земли, разбитая на клетки 25 на 25 или 50 на 50 см. Потом соединим ближайшие клетки ребрами и будем искать на них путь. Это будет довольно далеко от того пути, который БАС может проехать, но какое-то приближение это даст. И у нас будут такие вершины в двумерном пространстве.



## ПОСТРОЕНИЕ ОКРУЖАЮЩЕЙ СРЕДЫ БАС

Помимо деления пространства на клетки, мы можем сказать, что мы можем двигаться небольшими шагами в форме примитивов. Например, переместитесь на 50 см вперед или на 50 см вперед и поверните налево или на 50 см вперед и поверните направо. И с такими примитивами мы можем соединить все наше пространство. У нас нет явных ячеек, но если эти маленькие шаги достаточно хорошо скоординированы друг с другом, мы получаем регулярную сетку, которая хорошо и аккуратно покрывает пространство.

# Алгоритм A\*



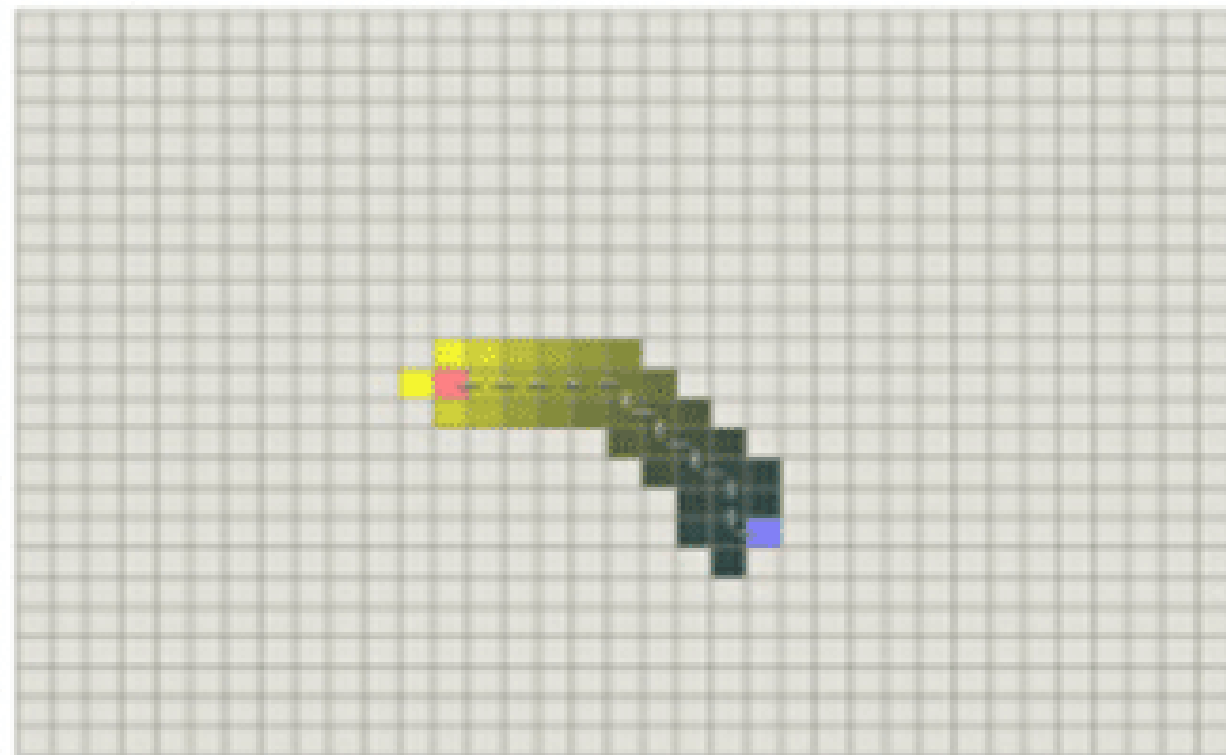
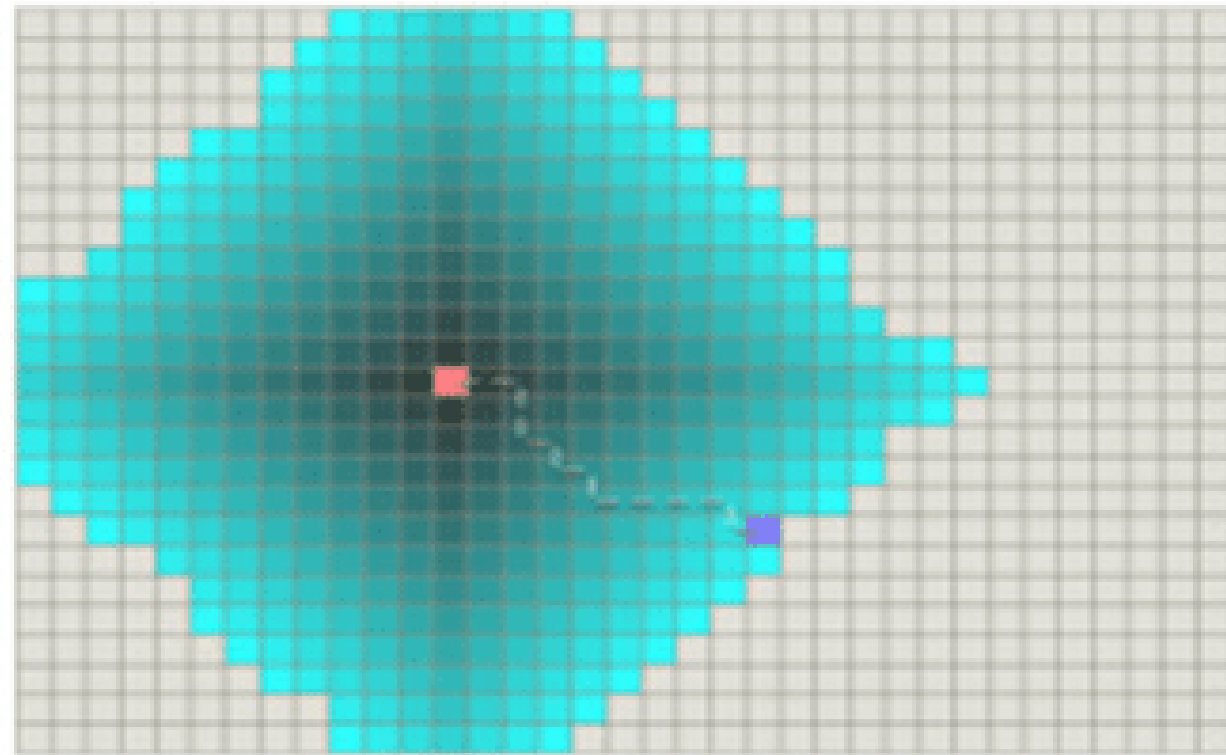
Основан на алгоритме Дейкстры

Дейкстры обходит вершины графа, доставая их из очереди с приоритетом по расстоянию от стартовой вершины –  $f(x)$ ;

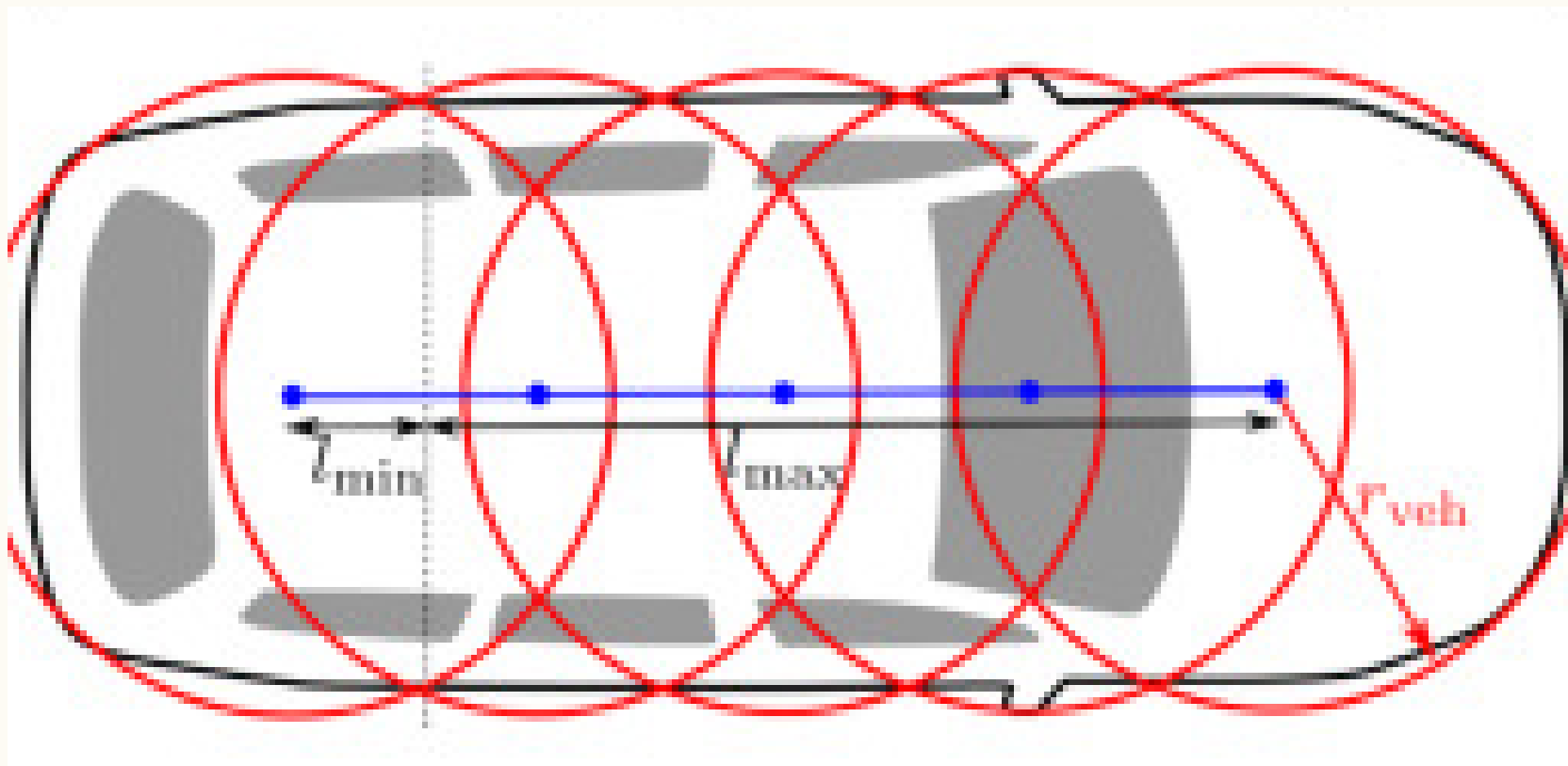
Алгоритм

A\* достает из очереди вершины по возрастанию  $f(x) + h(x)$ , где  $h(x)$  – эвристическая функция оценки расстояния от вершины до цели;

Необходимое условие для  $h(x)$ : значение функции оценки не должно превышать действительное расстояние до цели.



# Другие алгоритмы



## Оптимизационные методы

Наш БАС— объект довольно сложной формы, и препятствия, которые мы объезжаем, это тоже объекты сложных форм. Следовательно тут нужно производить какое-то упрощение. Например, машина — просто пять окружностей, которые ездят в разные стороны.

## Стохастические алгоритмы

Будем строить дерево итеративным образом. На каждой итерации:

- Выберем случайную точку в пространства;
- Найдём в текущем дереве ближайший к ней узел;
- Построим ребро в сторону симуляции проезда нескольких метров.

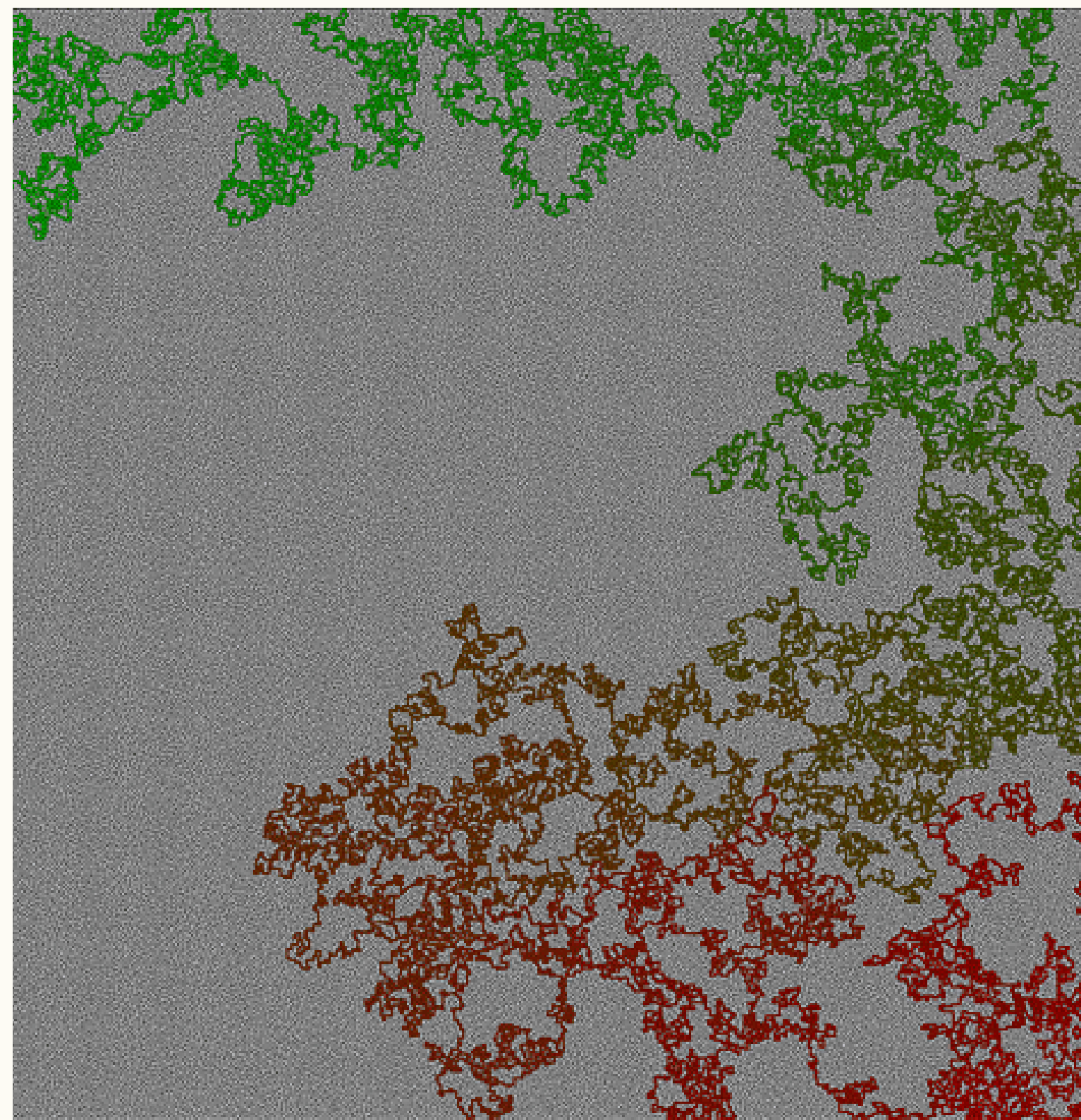




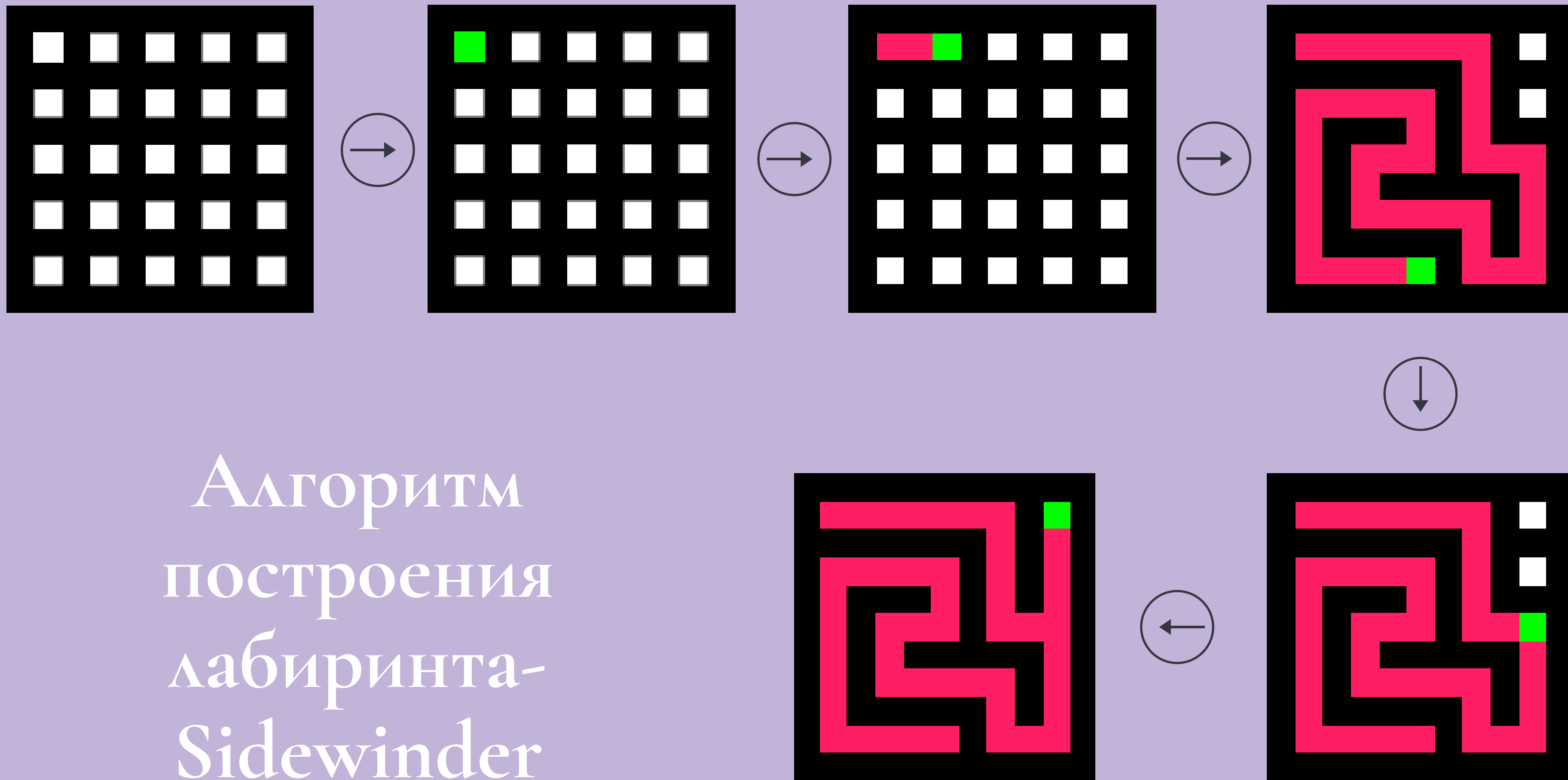
# Нейросетевой подход

Нейросетевой подход в последнее время показывает неплохие результаты относительно классического подхода. Давайте просто картинку с камер подавать в нейросеть, предварительно ее обучив на каких-нибудь человеческих перемещениях. Обучим, в каких ситуациях куда нужно крутить руль, тормозить, газовать, возьмем много-много данных, сделаем большую нейросеть, и по задумке она должна хорошо себя вести после этого.

# Задача построения лабиринта и его определение







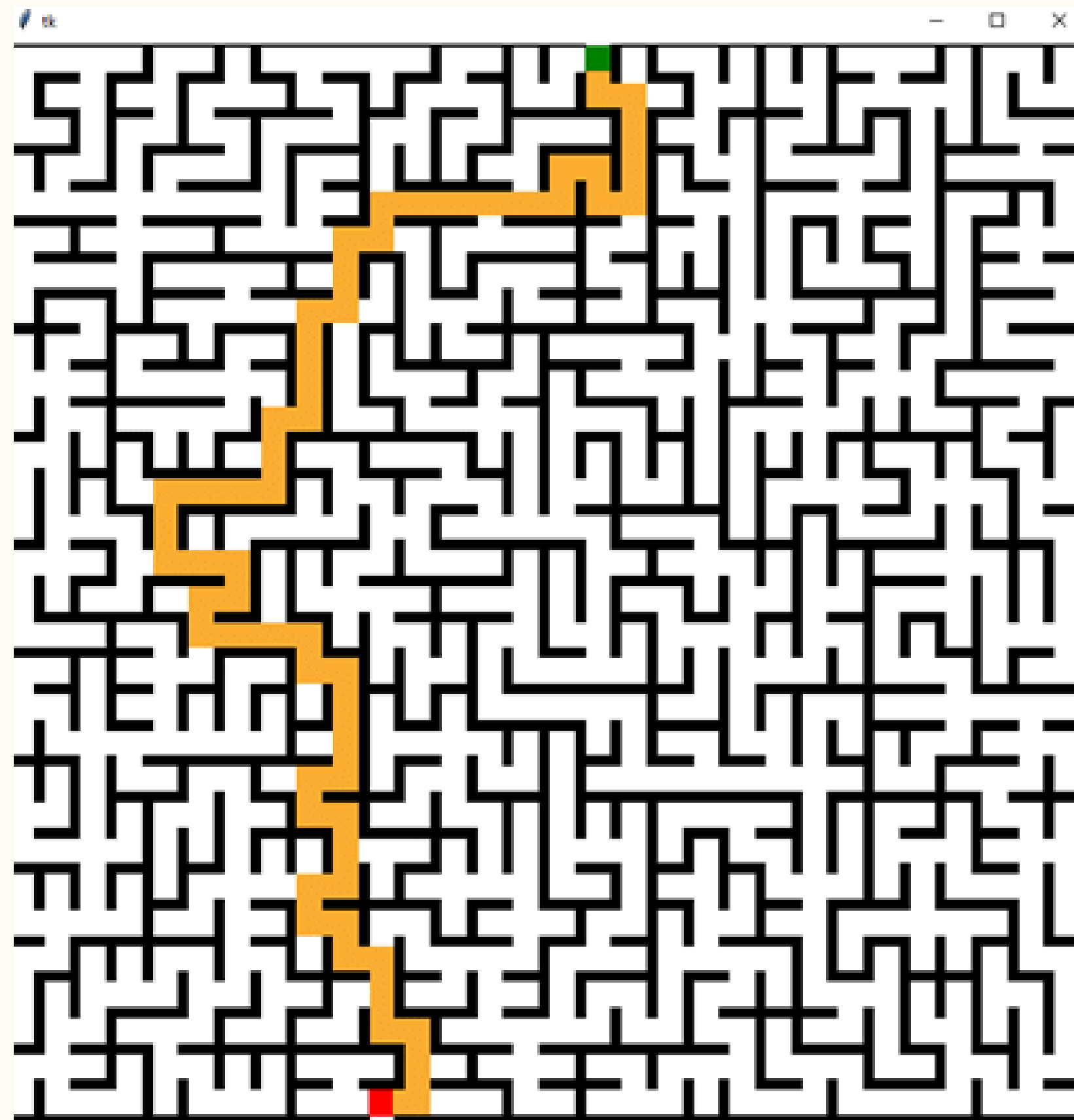
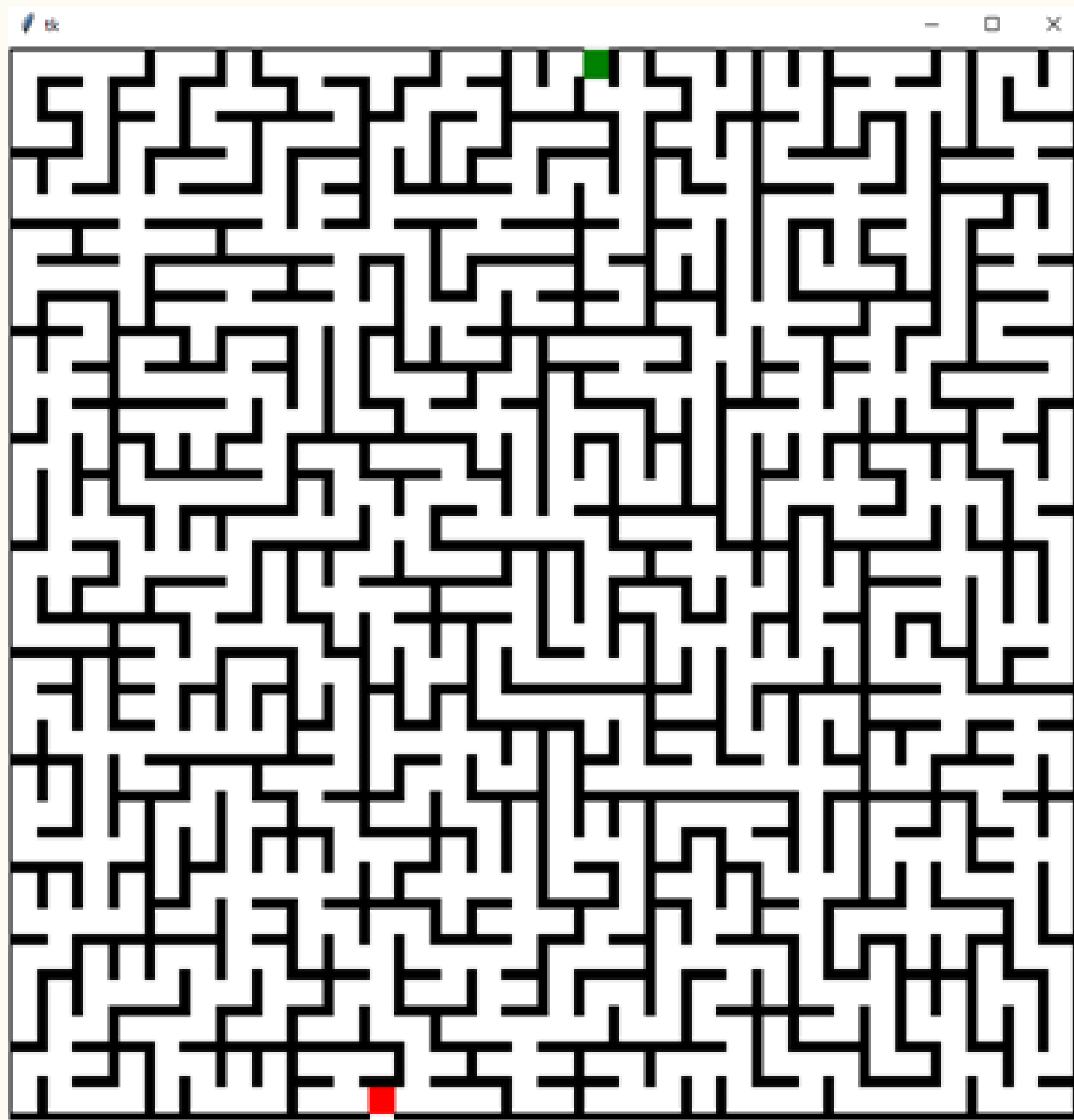
# Алгоритм построения лабиринта- Sidewinder

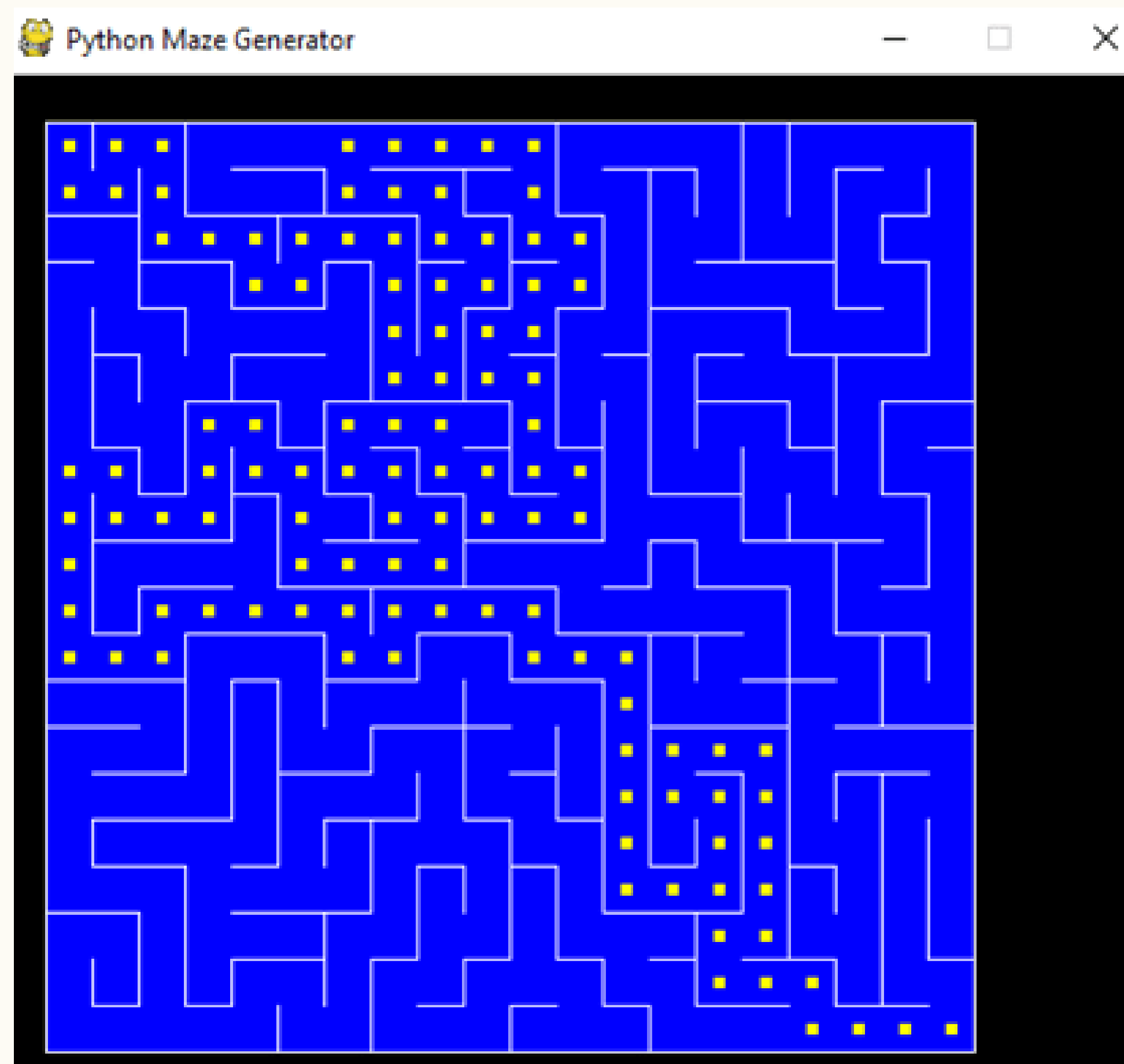
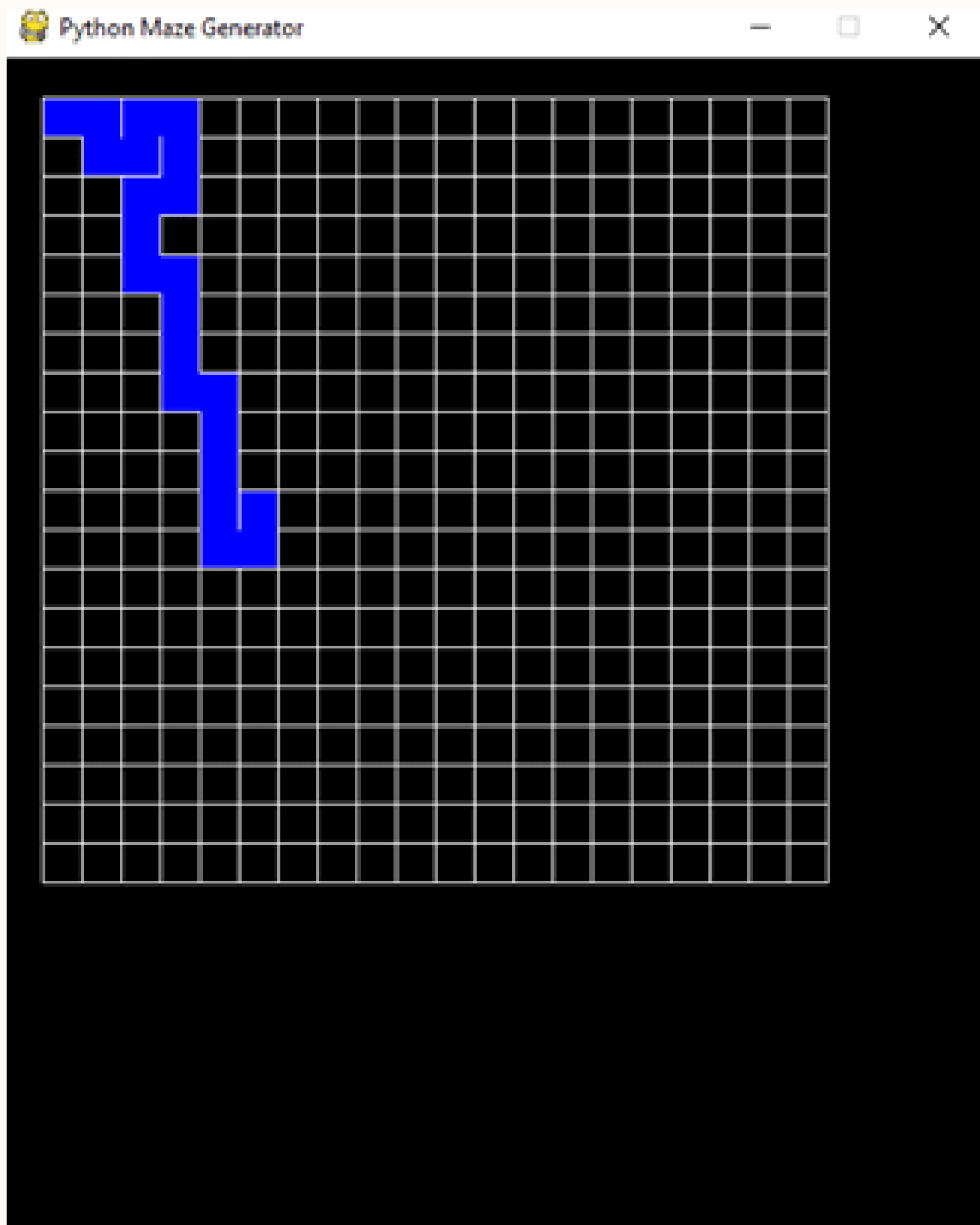
# Планирование пути – Эвристики для $A^*$

Расстояние  
до цели напрямик

Расстояние  
с учетом  
кинематических  
ограничений, но без  
учета препятствий

Максимум  
допустимых эвристик  
есть допустимая  
эвристика.





# Заключение



В данной работе был реализован алгоритм  $A^*$ , а также рассмотрена возможность его применения для задачи поиска оптимального пути в неизвестной местности . После построения и генерации лабиринта, алгоритм  $A^*$  решает задачу ,при этом имеется возможность управлять генерацией лабиринта с помощью указания дополнительных условий генератора.