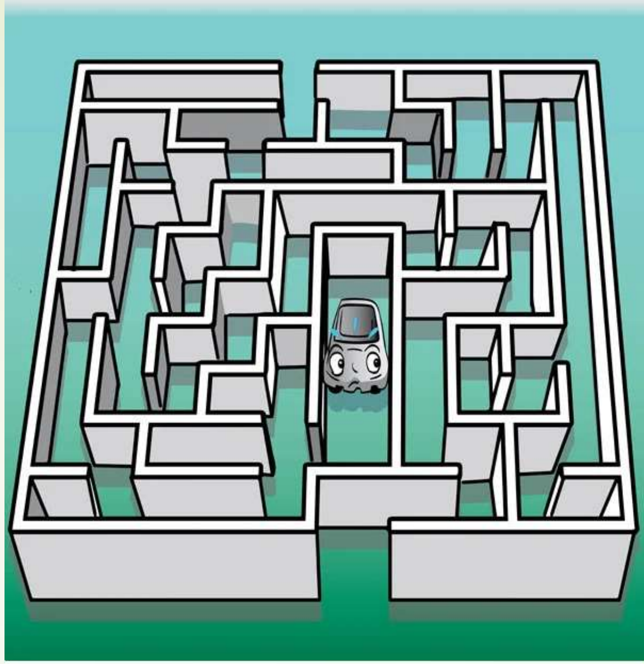
Generating Visualizing and Navigating a maze in Webots



Under the guidance of
Primary Supervisor
Terry Payne
Secondary Supervisor
Valentina Tamma

Final Presentation By:
Rishabh
Jaiswal201540604

Project Description

- How does autonomous cars navigate in an unknown area? Or how does robot vacuum cleaner move around the room?
- Real world unknown location can be mapped as a maze with obstacles, walls and allowable moving area in the form of Worlds in webots
- Webots – Robot simulation software that allows user to create 3D virtual worlds with physical properties and programmable robots to exhibit desired behavior.
- World - A world, in Webots, is a 3D description of the properties of robots and of their environment
- Robot – Adept pioneer 2 is a three-wheeled robot (2 motorized wheels and one passive caster wheel) mounted with several sensors including 16 ultrasonic sensors
- To find the shortest path to a location we need to read this world file and convert it to a programmable form.
- After successfully reading this file all the location of walls and obstacles along with their volumes can be abstracted and mapped to a 2D binary array where enclosed area by 1s denotes the area enclosed by obstacles and 0 denotes the free area
- Now we have the allowable paths for the robot but to solve the problem of finding the shortest path this areas are converted into a graph
- Now on this graph shortest path finding algorithm A* is applied and the shortest path is calculated
- A Wall following robot is not simulated to navigate on this path

Aims and Objectives

Aims

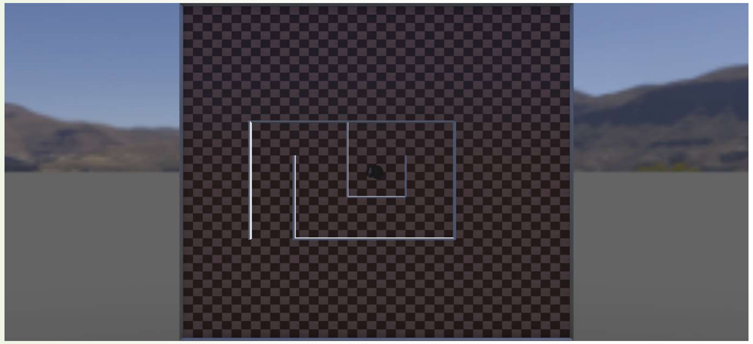
- Create a maze and generate both a volumetric and a feature map by parsing a configuration file. The user should then be able to specify a destination location and a two-wheeled differential drive robot should then navigate from its current position to the destination using a planning algorithm.

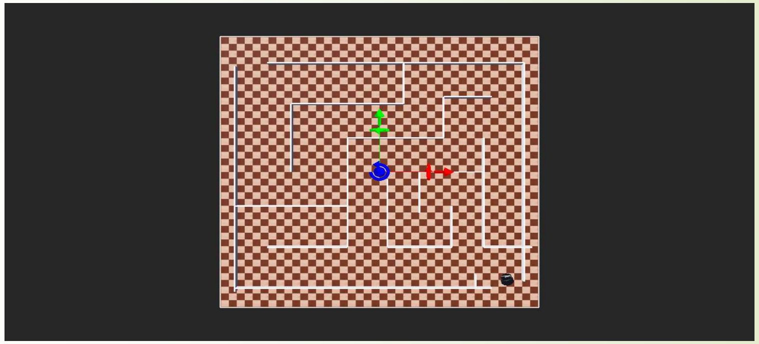
Objectives

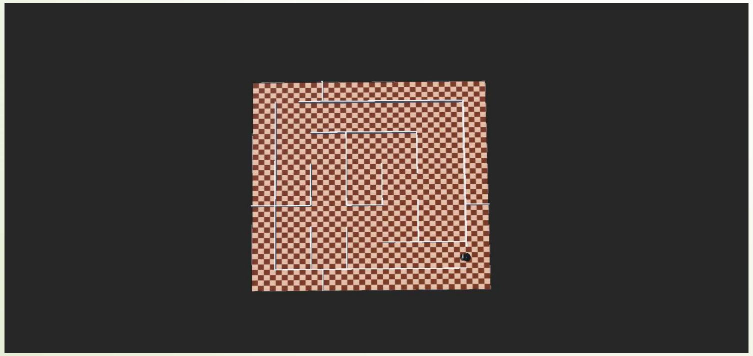
- **Build a test maze and two evaluation mazes of increasing complexity** ✓
- **Create a volumetric (grid-based binary) map to aid in maze visualisation** ✓
- **Create a feature (graph-based) map with nodes at intersections** ✓
- **Implement a route planning algorithm (A*) that uses the feature map to discover a route to a certain location.** ✓
- **Create a user-friendly interface for Webots so that users may choose a destination and provide telemetry.** ✓
- **a differential drive robot should use a wall following (or other) algorithm to navigate the map.** ✗

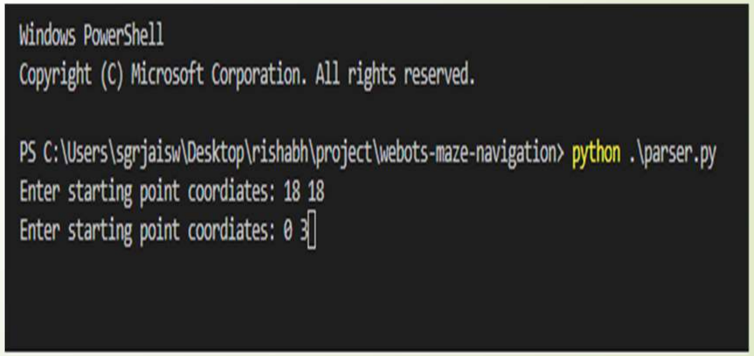
Output

- A python program was developed in which the user can input the entry and destination location from the users and gives the shortest path using the terminal.

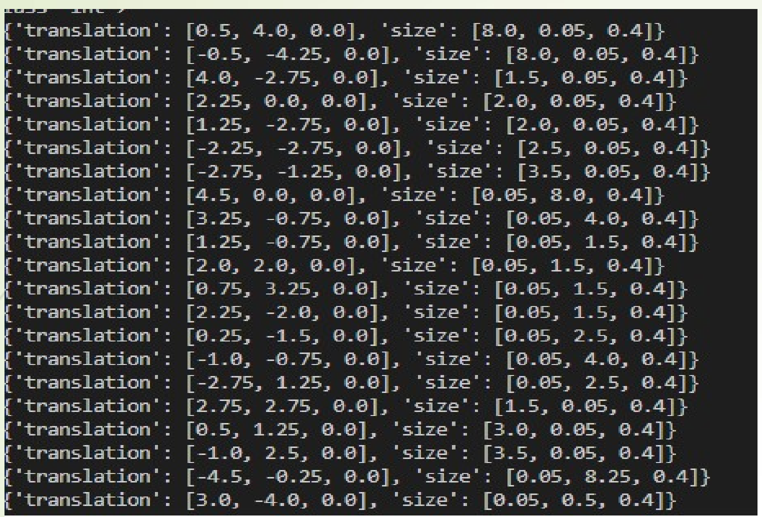




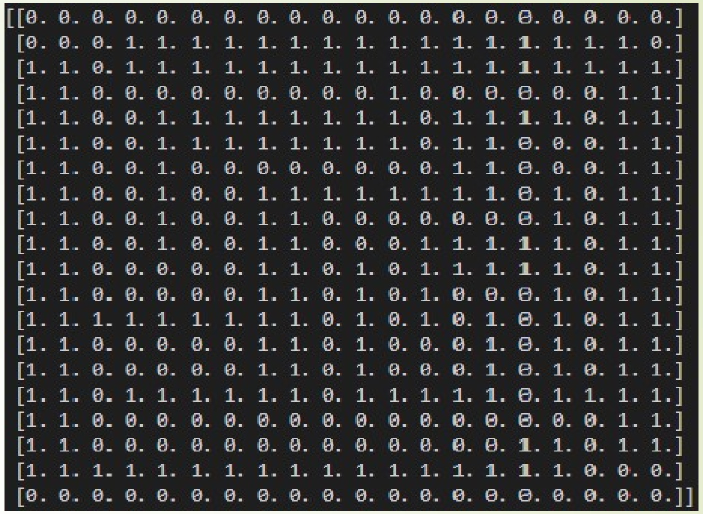




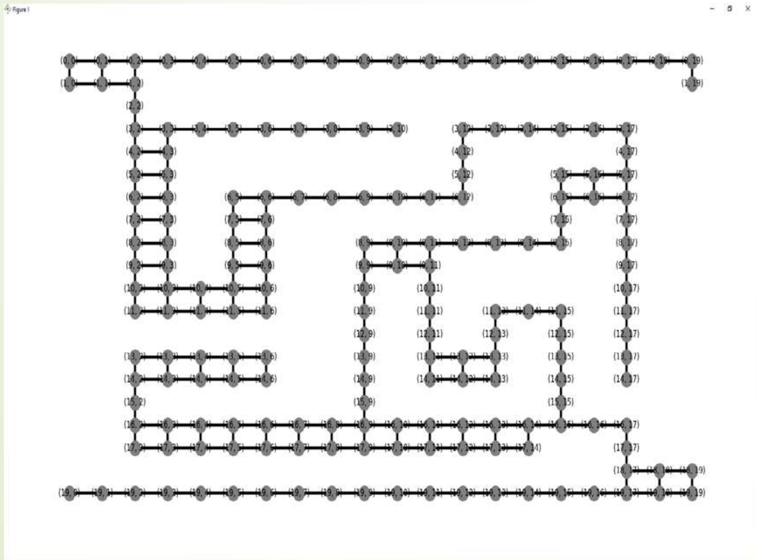
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\sgrjaisw\Desktop\rishabh\project\webots-maze-navigation> python .\parser.py
Enter starting point coordiates: 18 18
Enter starting point coordiates: 0 3
```

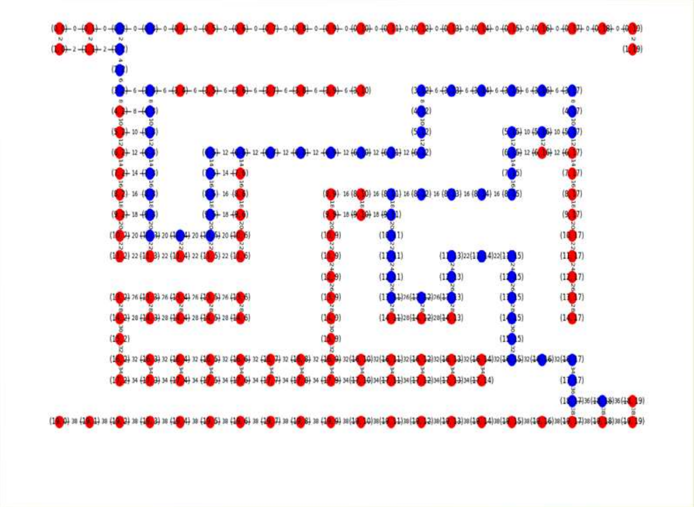


```
{'translation': [0.5, 4.0, 0.0], 'size': [8.0, 0.05, 0.4]}
{'translation': [-0.5, -4.25, 0.0], 'size': [8.0, 0.05, 0.4]}
{'translation': [4.0, -2.75, 0.0], 'size': [1.5, 0.05, 0.4]}
{'translation': [2.25, 0.0, 0.0], 'size': [2.0, 0.05, 0.4]}
{'translation': [1.25, -2.75, 0.0], 'size': [2.0, 0.05, 0.4]}
{'translation': [-2.25, -2.75, 0.0], 'size': [2.5, 0.05, 0.4]}
{'translation': [-2.75, -1.25, 0.0], 'size': [3.5, 0.05, 0.4]}
{'translation': [4.5, 0.0, 0.0], 'size': [0.05, 8.0, 0.4]}
{'translation': [3.25, -0.75, 0.0], 'size': [0.05, 4.0, 0.4]}
{'translation': [1.25, -0.75, 0.0], 'size': [0.05, 1.5, 0.4]}
{'translation': [2.0, 2.0, 0.0], 'size': [0.05, 1.5, 0.4]}
{'translation': [0.75, 3.25, 0.0], 'size': [0.05, 1.5, 0.4]}
{'translation': [2.25, -2.0, 0.0], 'size': [0.05, 1.5, 0.4]}
{'translation': [0.25, -1.5, 0.0], 'size': [0.05, 2.5, 0.4]}
{'translation': [-1.0, -0.75, 0.0], 'size': [0.05, 4.0, 0.4]}
{'translation': [-2.75, 1.25, 0.0], 'size': [0.05, 2.5, 0.4]}
{'translation': [2.75, 2.75, 0.0], 'size': [1.5, 0.05, 0.4]}
{'translation': [0.5, 1.25, 0.0], 'size': [3.0, 0.05, 0.4]}
{'translation': [-1.0, 2.5, 0.0], 'size': [3.5, 0.05, 0.4]}
{'translation': [-4.5, -0.25, 0.0], 'size': [0.05, 8.25, 0.4]}
{'translation': [3.0, -4.0, 0.0], 'size': [0.05, 0.5, 0.4]}
```



```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0.]
 [1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1.]
 [1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1.]
 [1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 1.]
 [1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1.]
 [1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1.]
 [1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1.]
 [1. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1. 0. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 0. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 1. 0. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 1. 0. 1. 1.]
 [1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
 [1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```





Evaluation

- Two evaluation mazes was developed with increasing complexity in the number of intersection of walls called nodes
- Both the mazes contains two paths between starting and ending locations
- The software presents the shortest path which was further verified by calculating the distances of all the path
- Changes made from original proposal includes:
- The proposal was to use both configuration file and sensors to make grid map but sensor based approach was not implemented as it was a much harder problem
- The proposal was to convert grid map to graph map using Voronoi diagrams. But it was implemented using networkx library of python for faster development of the software

Software Demonstration

- A python program that takes starting and end point from users was developed
- A test maze for development and two mazes with increasing complexity for evaluation purposes was developed(contains multiple routes)
- The maze is converted into a grid based Volumetric binary map in the form of 2D array
- This volumetric map is converted to feature based graph map
- A* route planning algorithm was implemented to produce the shortest in the grid

THANK YOU!!!