Group 9

Project: Maze Solver using Artificial Intelligence

Project Description
- Self-governing and self-reliant.

- Human intelligence.

- Shortest path finder.

Further Work
- Modification of algorithm making it easier.

- Calculate weightage by its own for every individual node.

Possible enhancement
- Further path sensing capacity, before selecting direction at node.

- Detecting obstacle and making way to destination.

Implementation details
- Used Recursion to find path

- Path selection process by implementing it by checking the conditions and

- Used backtracking to find

Design Issues

Initially we clubbed in one goTest.pl file. Then after fragmenting the project code into small files we come up with easy way to check and proceed for debugging the project program.
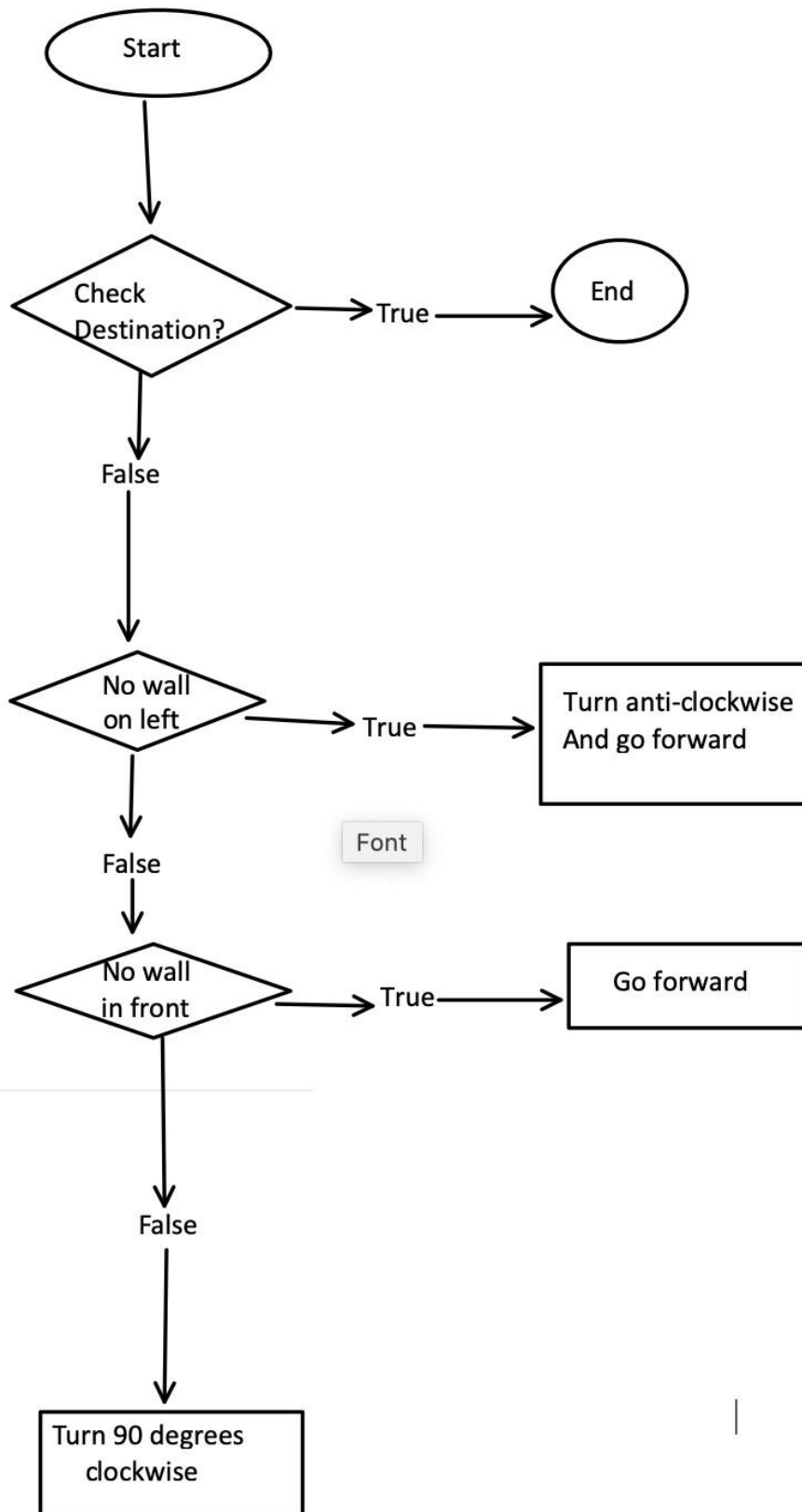
Method Used
Prod([A], 1), to calculate product of entries in list.
Fill (a, X,[]), To fill empty place on maze.
Stats(U,D,L,R), number of cells opening in up, down, left, right.
StatsHelp([], [], [], []), to find the total of up, down, left and right moves.
FindDistance(L), find distance D from start.

```
                    ┌─────────┐
                    │  Start  │
                    └────┬────┘
                         │
                         ▼
                      ╱Check╲
                     ╱Destination?╲──── True ────►  ( End )
                      ╲       ╱
                         │
                       False
                         │
                         ▼
                      ╱No wall╲
                     ╱on left  ╲──── True ────►  ┌──────────────────────┐
                      ╲        ╱                  │ Turn anti-clockwise  │
                         │                        │ And go forward       │
                       False                      └──────────────────────┘
                         │
                         ▼                Font
                      ╱No wall ╲
                     ╱in front  ╲──── True ────►  ┌──────────────────────┐
                      ╲         ╱                  │    Go forward        │
                         │                        └──────────────────────┘
                       False
                         │
                         ▼
                ┌──────────────────┐
                │ Turn 90 degrees  │
                │ clockwise        │
                └──────────────────┘
```

Start

Check Destination? → True → End

False

No wall on left → True → Turn anti-clockwise And go forward

False

Font

No wall in front → True → Go forward

False

Turn 90 degrees clockwise

Difficulties Encountered

- Calculating weight for deciding path

- Finding sequence of paths to reach the goal state.

Performance analysis

Nodes can be defined as a point on maze from where robot decides direction to be taken in order to reach goal state. Some points on maze have more than one choice for path to be taken which leads to destination. Best first search algorithm can be used to choose the best shortest path to be taken which will save time and reduce distance to be covered in order to reach goal.

Result performance

For 4 * 4 matrix path taken by maze solver to reach destination.
Possible paths starting from (0,3) to (3,2)
1st move (0,3) [start state]
2nd move (0,2)
3rd move (1,2)
4th move (1,1), (2,2)
5th move (0,1), (1,0), (2,1), (2,3)
6th move (0,0), (2,0), (3,1), (3,3)
7th move (3,2) [destination state]

	0	1	2	3
3	Start	■		
2				
1				
0				■

Contribution chart:

Nov 3, 2019 – Dec 4, 2019

Contributions: **Commits** ▾

Contributions to master, excluding merge commits

| Dhruval10 | #1 |
| 11 commits 425 ++ 250 -- | |

Nov 03 · Sat 09 · Fri 15 · Thu 21 · Wed 27 · Tue 03

| riyapatel888 | #2 |
| 8 commits 233 ++ 73 -- | |

Nov 03 · Sat 09 · Fri 15 · Thu 21 · Wed 27 · Tue 03