1. **Main Ideas**

Programming language: Python3 supported by Google Cloud platform.

Framework: Flask micro web framework written in Python.

Test framework: pytest and unittest.

IDE: Visual Studio Code.

VCS: GIT powered by GitHUB.

1. **Class diagram**

A screenshot of a cell phone

Description automatically generated

1. **Algorithm:** A\* pathfinding algorithm

A\* assigns **start node** to the current position of the bot and **goal node** to the next position of the player.

A\* selects the path that minimizes:

**F = G + H**

***F*** *is the total cost of the node.*

***G*** *is the distance between the current node and the start node. Calculated as a number of the step to current node.*

***H*** *is the heuristic — estimated distance from the current node to the end node. Calculated as cartesian distance between current node and goal node*

A\* terminates when the path it chooses to extend is a path from start to goal or if there are no paths eligible to be extended.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G function distribution | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 9 | 8 | 7 | 8 | 8 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | | 8 | 7 | 6 | 7 | 8 | 8 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | | 7 | 6 | 5 | 6 | 7 | 8 | 8 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | | 6 | 5 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | | 5 | 4 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | 4 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |  |  |  | 14 | 15 | 16 | | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  | 15 |  | 15 | 16 | 17 | | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |  | 14 | 15 | 16 | 17 | 18 | | 3 | 2 |  | 2 | 3 | 4 | 5 | 6 | 7 |  | 13 | 14 | 15 | 16 | 17 | | 4 | 3 |  | 3 | 4 | 5 | 6 | 7 | 8 |  | 12 | 13 | 14 | 15 | 16 | | 5 | 4 |  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | 6 | 5 |  | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | | 7 | 6 |  | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | | 8 |  |  | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | | 9 | 10 | 9 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |
| H function distribution | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 8 | 9 | 10 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 7 | 8 | 9 | | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 7 | 8 | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 5 | 6 | 7 | | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 4 | 5 | 6 | | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 3 | 4 | 5 | | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 2 | 3 | | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 3 | 4 | 5 | | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 4 | 5 | 6 | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 5 | 6 | 7 | | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 6 | 7 | 8 | | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 7 | 8 | 9 | | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 8 | 9 | 10 | |
| F function distribution | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 27 | 25 | 23 | 23 | 22 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 25 | 27 | 29 | | 25 | 23 | 21 | 21 | 21 | 20 | 21 | 21 | 21 | 21 | 21 | 21 | 23 | 25 | 27 | | 23 | 21 | 19 | 19 | 19 | 19 | 18 | 19 | 19 | 19 | 19 | 19 | 21 | 23 | 25 | | 21 | 19 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 19 | 21 | 23 | | 19 | 17 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 17 | 19 | 21 | | 17 | 15 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |  |  |  | 17 | 19 | 21 | | 15 | 13 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |  | 17 |  | 17 | 19 | 21 | | 13 | 11 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |  | 15 | 15 | 17 | 19 | 21 | | 15 | 13 |  | 11 | 11 | 11 | 11 | 11 | 11 |  | 15 | 15 | 17 | 19 | 21 | | 17 | 15 |  | 13 | 13 | 13 | 13 | 13 | 13 |  | 15 | 15 | 17 | 19 | 21 | | 19 | 17 |  | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 17 | 19 | 21 | | 21 | 19 |  | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 19 | 21 | 23 | | 23 | 21 |  | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 19 | 21 | 23 | 25 | | 25 |  |  | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 21 | 23 | 25 | 27 | | 27 | 27 | 25 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 25 | 27 | 29 | |
| *Blue: bot, Purple: player, Yellow: start node, Red: goal node, Green: path* | |

1. **Algorithm:** Path comparison algorithm

After generation of paths to the players, algorithm chooses shortest.

Algorithms selects the path that minimizes:

**C = L - T - 3\*N**

***C*** *is the total cost of the path.*

***L*** *is the length of the path*

***T*** *is the number of Turbo recharge laying on the path.*

**N** *is the number of possible turbo steps (3).*

*Turbo steps are available for the bot if:*

1. *The bot has Turbo recharge;*
2. *The given path contains 3 consequent blocks in one row or column.*
3. **Algorithm:** Survival Algorithm

If A\* terminates when there are no paths from start to goal, the bot will move towards the part of the map with less density of obstacles.

Algorithms selects the direction that minimizes:

**D = O / B**

***D*** *is the density of the direction.*

***O*** *is the number of obstacle block of the directions.*

***B*** *is the number blocks of the directions.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D function distribution | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | 10 |  |  |  |  |  |  |  |  |  |  |  |  | |  | 7 |  | 20 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| *Blue: bot, Purple: player, Yellow: start node, Red: goal node* | |

1. **Possible algorithm optimizations**

a) **Path slicing**: instead of recalculating the entire path, we can recalculate the following n-steps of the path.

b) **Map slicing**: instead of recalculating the entire path, we can recalculate the part laying in the modified region.

c) **Prediction**: calculate path to the region of all possible players' movements based on prediction.

d) **Neural network**: design neural network and train it based on game rules.