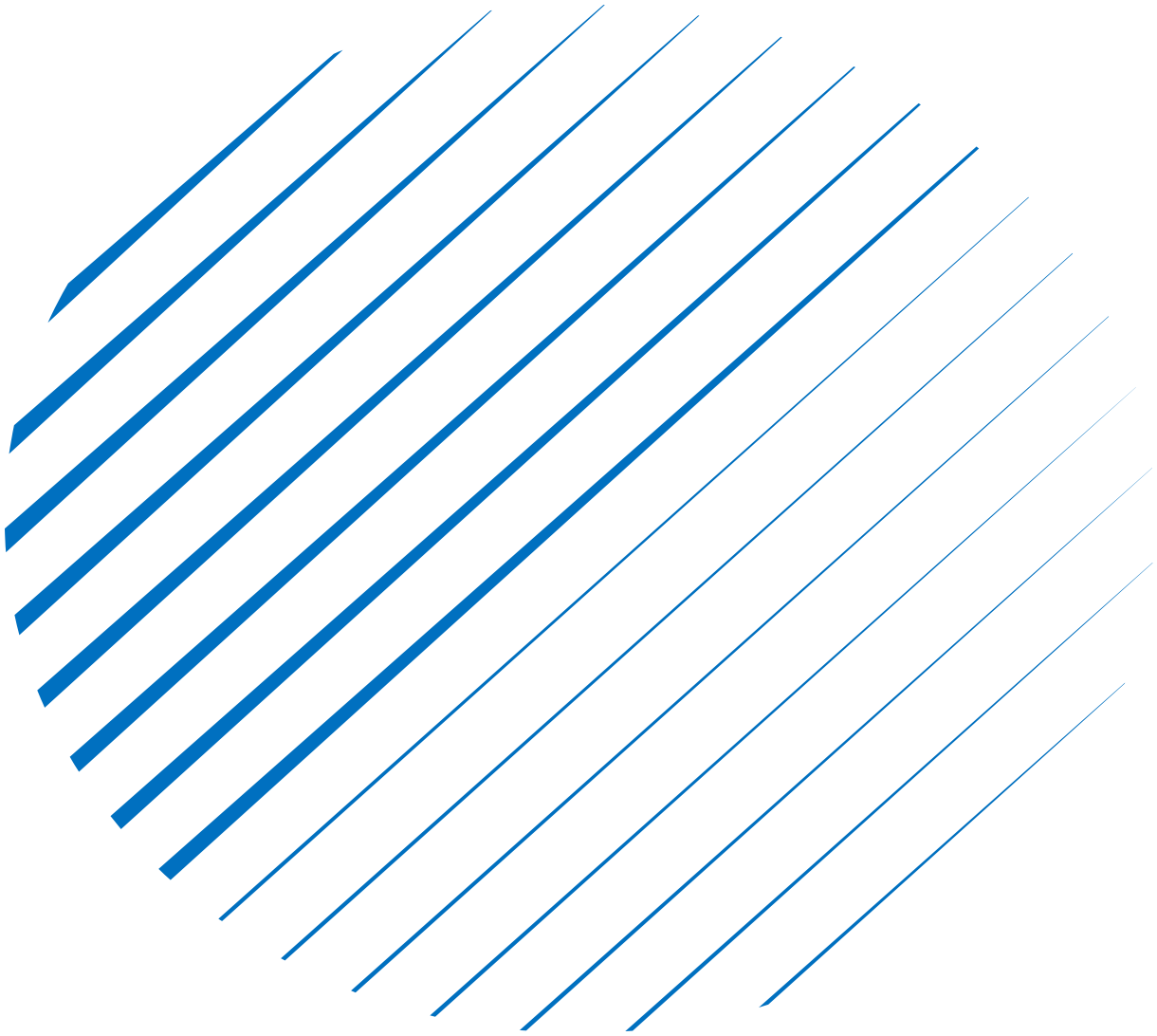
Haskell Project

Team 111



George Elhamy T26

Youhanna Mentias T25

Teodor Maged T6

Youssef Magdy T26

- **Description :**

  The project is an AI agent returning a sequence of steps to be done to collect mines in a 4X4 board by knowing the starting position and the position of each state. It works by searching blindly through the next states of each position until it reaaches the goal which is basiclly collecting all the mines in the board.

- **Implemented functions :**

1. **up:** This function takes a state and returns a state of going up if possible and if not it returns null.
2. **down:** This function takes a state and returns a state of going down if possible and if not it returns null.
3. **left:** This function takes a state and returns a state of going left if possible and if not it returns null.
4. **right:** This function takes a state and returns a state of going right if possible and if not it returns null.
5. **collect:** This function takes a state and returns a state after collecting the mine in the same position if there is a mine in that position and if not it returns null.
6. **nextMyStates:** This function takes as an input a state and returns a list of all possible states from the current position (Up-Down-Left-Right).]

7. **clear**: This function takes as an input a list of states and returns also a list of states but removes all the states with a value of null, and it is used to clear the nextMyStates list when it is called in the search method.
8. **isGoal:** This function takes as an input a state an checks whether this is the final state that has the solution by looking at the list of mines and checking whether it is empty or not, if the list of mines is empty it returns true and if not it returns false.
9. **search:** This function takes as an input a list of states and returns a state if the goal is found the function returns this state and if not the functions calls itself recursively and adds that next states of the head state to the end of the list until the goal state is found.
10. **constructSolution:** This function takes a state and returns a list of strings representing the sequence of actions that should be done to reach the goal state.
11. **solve:** This function takes a cell and a list of mines and returns the solution in terms of a list of strings representing the sequence of actions that should be done to solve the board.

- **Examples of the code :**

```
Main> solve (0,1) [(2,2),(3,1)]
["down","down","down","collect","up","right","collect"]
```

```
Main> solve (3,2) [(1,1),(0,3)]
["up","up","left","collect","up","right","right","collect"]
```

```
Main> solve (3,2) [(2,1),(1,0)]
["up","left","collect","up","left","collect"]
```

- **Examples of the BONUS code :**

```
Main> solve (2,2) [(3,5),(8,6)]
["down","right","right","right","collect","down","down","down","down","down
","right","collect"]
```

```
Main> solve (3,3) [(4,2),(6,5),(7,4)]
["down","left","collect","down","down","down","right","right","collect","up
","right","collect"]
```

```
Main> solve (2,1) [(3,5),(6,7),(8,10),(12,11),(14,9)]
["down","right","right","right","right","collect","down","down","down","rig
ht","right","collect","down","down","right","right","right","collect","down
","down","down","down","right","collect","down","down","left","left","colle
ct"]
```

```
Main> solve (2,1) [(3,5),(6,7),(8,10),(12,11),(14,9),(17,5),(20,12)]
["down","right","right","right","right","collect","down","down","down","rig
ht","right","collect","down","down","right","right","right","collect","down
","down","down","down","right","collect","down","down","left","left","colle
ct","down","down","down","left","left","left","left","collect","down","down
","down","right","right","right","right","right","right","right","collect"]
```

```
Main> solve (3,4) [(9,18),(13,23),(3,4),(31,27),(1,5),(45,29),(6,7)]
["collect","up","up","right","collect","down","down","down","down","down","right","right
","collect","down","down","down","right","right","right","right","right","right","right"
,"right","right","right","right","collect","down","down","down","down","right","right","
right","right","right","collect","down","down","down","down","down","down","down","down"
,"down","down","down","down","down","down","down","down","down","down","right","right","
right","right","collect","down","down","down","down","down","down","down","down","down",
"down","down","down","down","down","right","right","collect"]
```