Gonzales, Mark Edward M.
Lee, Hylene Jules G.
CSINTSY S13

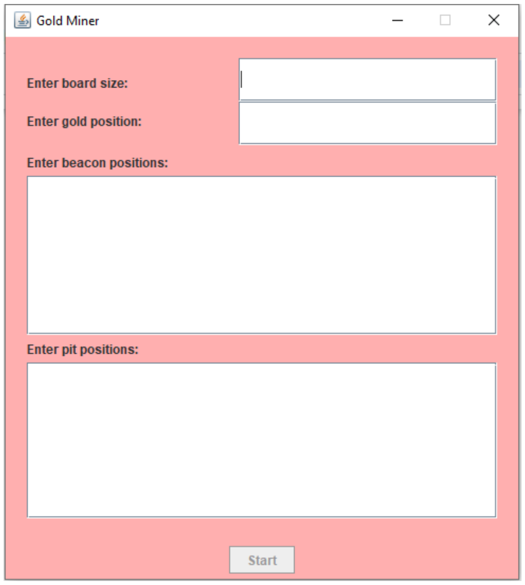
Major Course Output #1
Dr. Judith J. Azcarraga

SCREENSHOTS OF THE RUNNING INTERFACE OF THE PROGRAM

A. System Screens

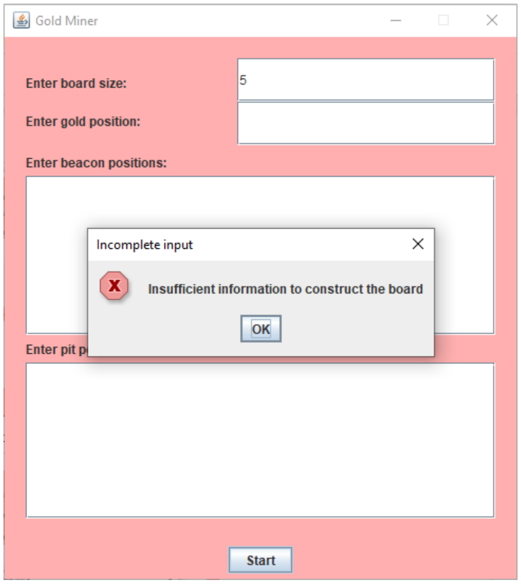
The system is composed of four screens: the initialization screen, AI behavior screen, system behavior screen, and main screen.

1. <u>Initialization Screen</u>



The initialization screen contains text boxes allowing users to input the size of the board, the position of the gold tile, and the positions of the beacon and pit tiles. The number of beacons and pits on the board is determined by the number of pairs of integers — separated by spaces — the user enters into the respective text boxes; the first integer of each pair is read as the row number of the object, and the second integer is read as its column number.

Moreover, basic error handling was also implemented for the user inputs, covering the following cases:



Enter board size: 5

Enter gold position:

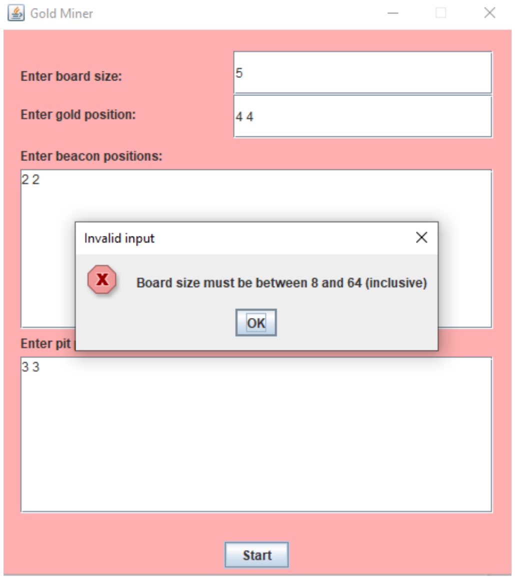
Enter beacon positions:

Enter pit p

Incomplete input

Insufficient information to construct the board

a. The board size or gold tile position has not been entered.



Enter board size: 5

Enter gold position: 4 4

Enter beacon positions:

2 2

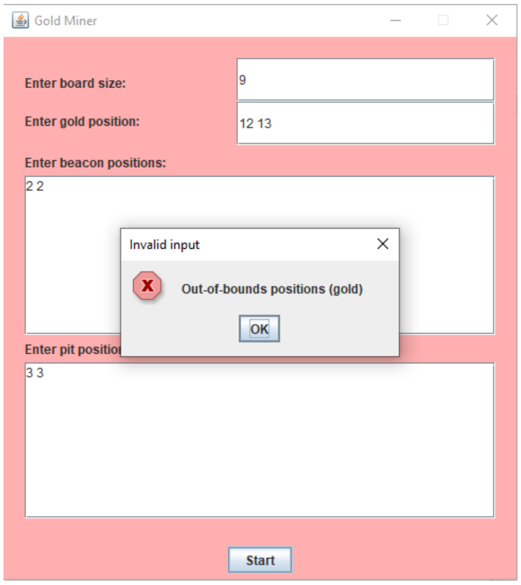
Enter pit

3 3

Invalid input

Board size must be between 8 and 64 (inclusive)

b. The entered board size is not between 8 and 64 (inclusive).



Enter board size: 9

Enter gold position: 12 13

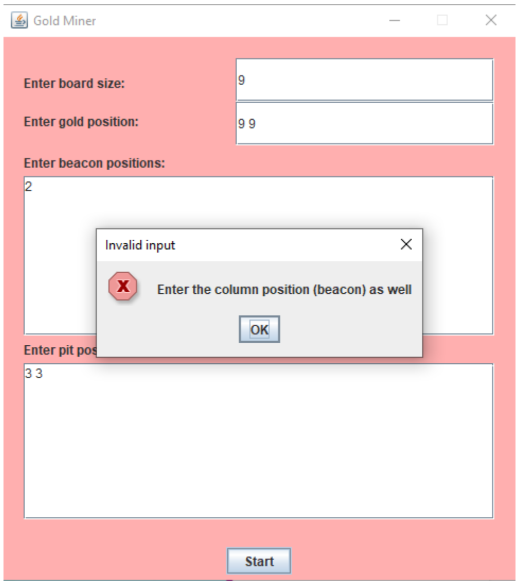
Enter beacon positions:
2 2

Invalid input

Out-of-bounds positions (gold)

Enter pit positio
3 3

c. The provided row and column numbers of the board objects are out of bounds (the valid locations on the board are computed based on the provided board size).



Enter board size: 9

Enter gold position: 9 9

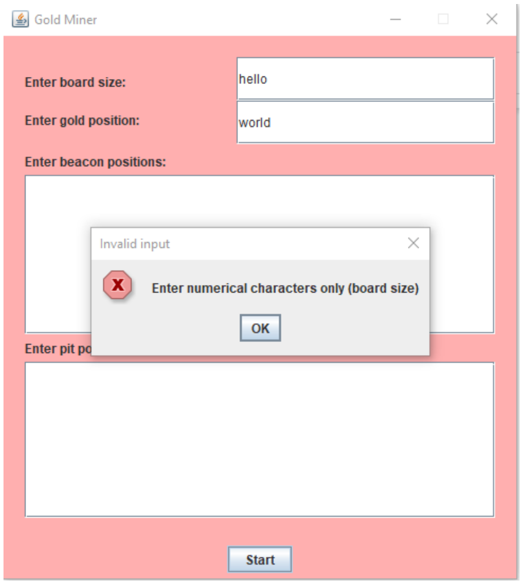
Enter beacon positions:
2

Invalid input

Enter the column position (beacon) as well

Enter pit pos
3 3

d. The integers specifying the locations of the board objects are not provided in pairs (i.e., the column number of a board object was not provided).



Gold Miner

Enter board size: hello

Enter gold position: world

Enter beacon positions:

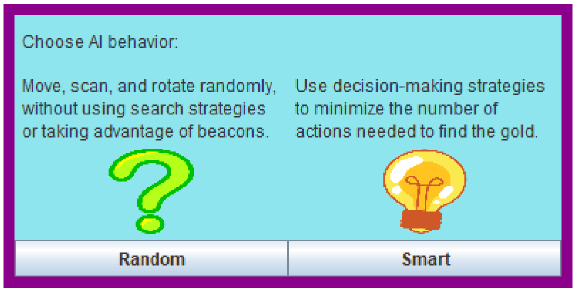
Enter pit po

Invalid input

Enter numerical characters only (board size)

e. The user provided non-numeric input.

2. <u>AI Behavior Screen</u>



Choose AI behavior:

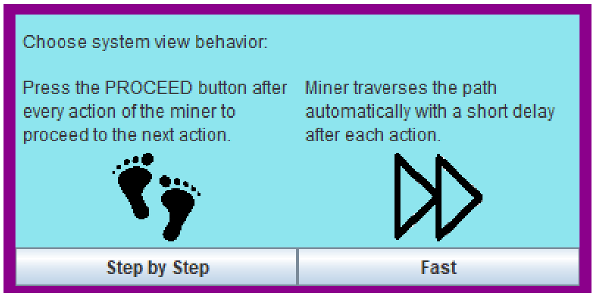
Move, scan, and rotate randomly, without using search strategies or taking advantage of beacons.

Use decision-making strategies to minimize the number of actions needed to find the gold.

The AI behavior screen allows users to specify the behavior of the miner agent. The random option runs the nonrational agent, which chooses its actions based on a random seed; moreover, the nonrational agent does not record scanned environment data — such as the positions of pits, return values of beacons, or visited tiles — making it susceptible to falling into pits or possibly entering into cycles.

The smart option runs the smart agent, which uses decision-making strategies and information stored in its memory to optimize its succeeding actions in order to arrive at the gold tile. As part of its implementation, the smart agent is also capable of determining when a board configuration is invalid — i.e., the gold tile cannot be reached — and properly terminating the search process when such a configuration is encountered.

3. System Behavior Screen



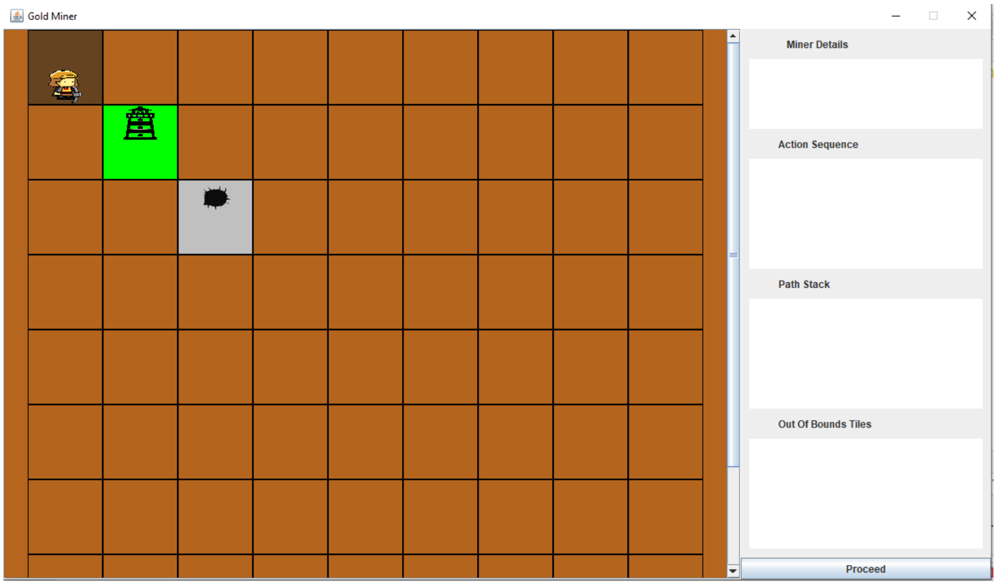
Choose system view behavior:

Press the PROCEED button after every action of the miner to proceed to the next action.

Miner traverses the path automatically with a short delay after each action.

The system behavior screen allows users to specify the type of view the system uses. The step by step option triggers the execution of the step by step view, where the system waits for the user to press the `Proceed` button on the main screen before executing the next action of the agent. The fast option triggers the execution of the fast view, where the agent traverses the board automatically with a short delay after every action.

4. Main Screen



The main screen contains a two-dimensional array of panels representing the board, stored in a scrollable pane that can be moved to accommodate larger board sizes. The miner is represented as a cartoon sprite with a distinct front, which turns to represent changes in the direction it is facing. Tiles containing beacons, pits, and the gold are similarly represented with images and respective colored panels. When the miner visits a tile (such as the initial tile in the above image), its background color is changed to dark brown.

The panel to the right of the board contains four text boxes displaying several details tracked by the system. The first text box displays the number of moves, rotations, scans, and backtracks performed by the miner, as well as the return values of beacons the miner encounters. As the nonrational agent does not make use of memory, the number of backtracks and beacon return values are not updated when the random miner is used.

The second text box displays the action sequence of the miner, which is updated after every new action the miner executes. The third text box displays the locations and contents of the tiles on the path stack of the miner, whose elements are increased as the miner visits new tiles and decreased when backtracking occurs, albeit the path stack is not decreased when the random miner is used, as backtracking is not implemented. The fourth text box displays the locations of the scanned out of bounds tiles the miner memorizes; this is not updated when the random miner is used, again owing to its lack of memory.

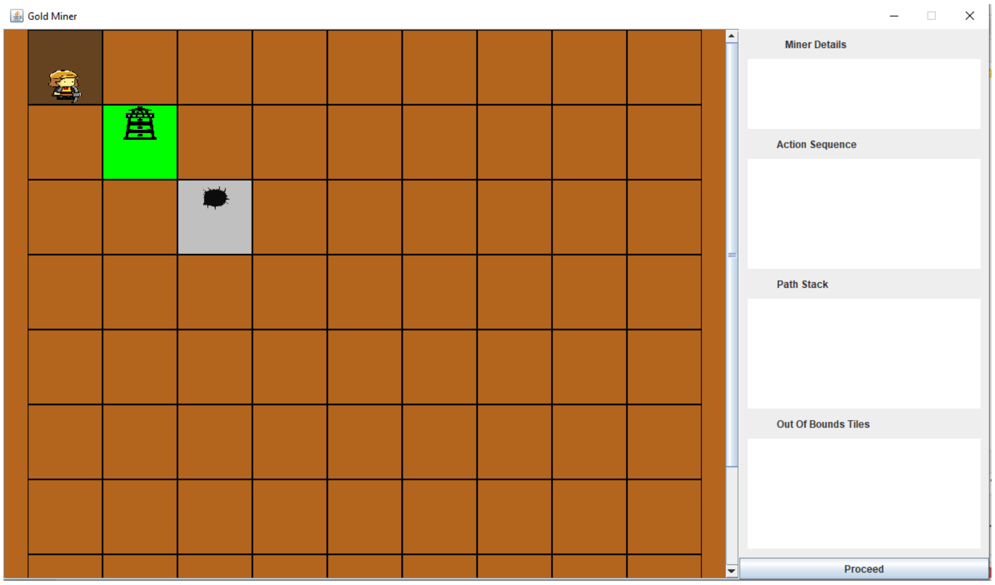
Finally, the main screen also contains a `Proceed` button; when the fast view is used, the user presses this button once to begin the automatic traversal process. When the step by step view is used, the user presses the button after every action to execute the succeeding action.

B. Features of the Running Program

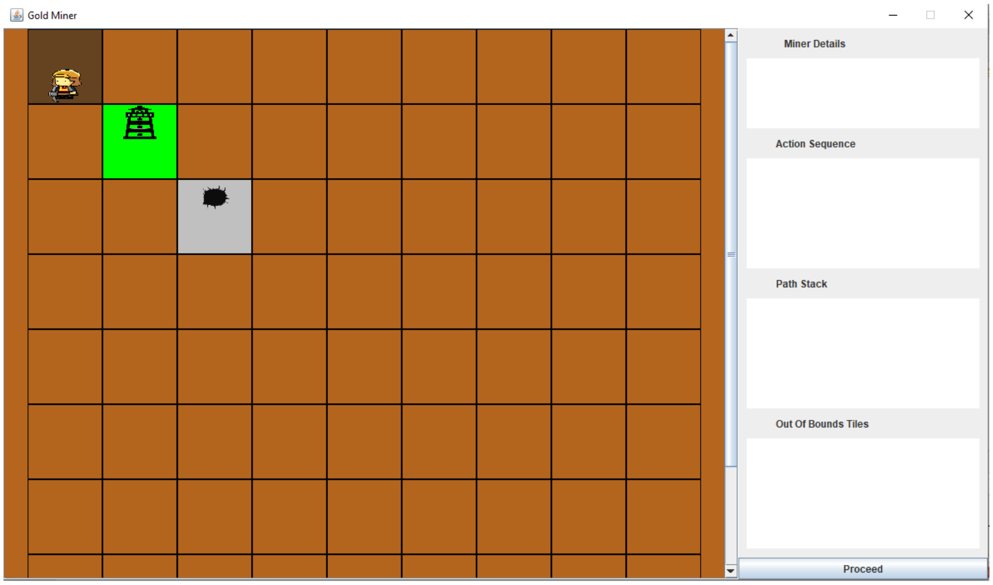
Both versions of the agent operate on three states: the initial state before the first action is executed, the active state where the miner executes one of its three possible actions (move, rotate, and scan) in order to reach the gold tile, and the terminal state, where the traversal ends with the miner either successfully reaching the gold tile or failing to arrive at the gold tile.

THE PRESENTATION OF THE FEATURES OF THE RUNNING PROGRAM START ON THE NEXT PAGE.

1. <u>Initial State</u>

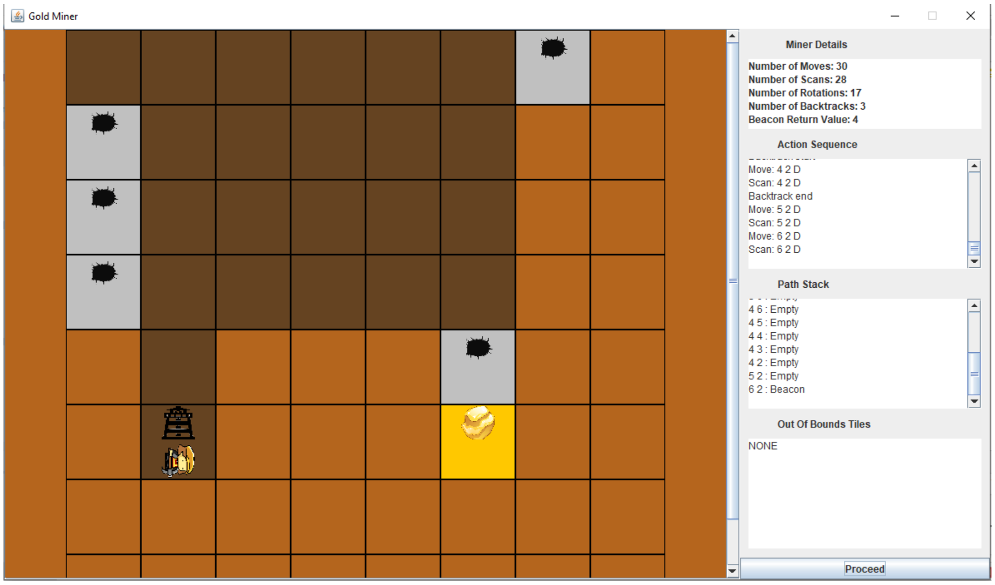


When the smart miner is used, the miner is initially located on square (1,1) (designated as the upper left-hand corner of the grid) and is facing right. Moreover, its memory, one component of which is represented in the graphical user interface as the contents of the "Out Of Bounds Tiles" text box, is not yet initialized.



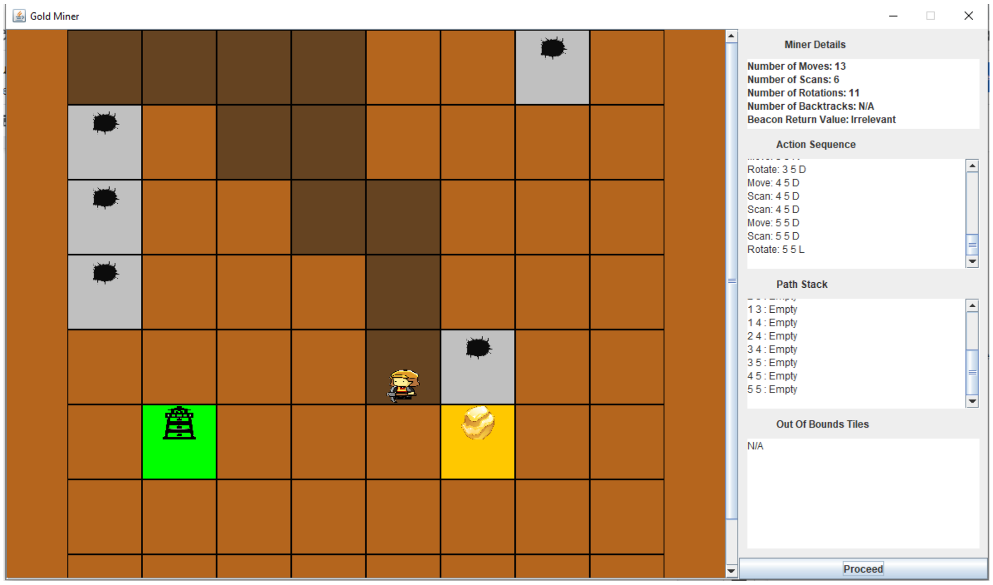
When the random miner is used, its location and initial memory is similar to those of the smart miner (though the memory of the random miner remains the same throughout its entire traversal). However, as shown in the above image, the initial direction the random miner is facing may vary, as this value is also randomized.

2. Active State



```
Miner Details
Number of Moves: 30
Number of Scans: 28
Number of Rotations: 17
Number of Backtracks: 3
Beacon Return Value: 4

Action Sequence
Move: 4 2 D
Scan: 4 2 D
Backtrack end
Move: 5 2 D
Scan: 5 2 D
Move: 6 2 D
Scan: 6 2 D

Path Stack
4 6 : Empty
4 5 : Empty
4 4 : Empty
4 3 : Empty
4 2 : Empty
5 2 : Empty
6 2 : Beacon

Out Of Bounds Tiles
NONE
```

During the active state, the smart miner uses its memory in conjunction with a search strategy to determine the next action to make. As the miner traverses the board, the system details displayed on the side panel are updated accordingly.



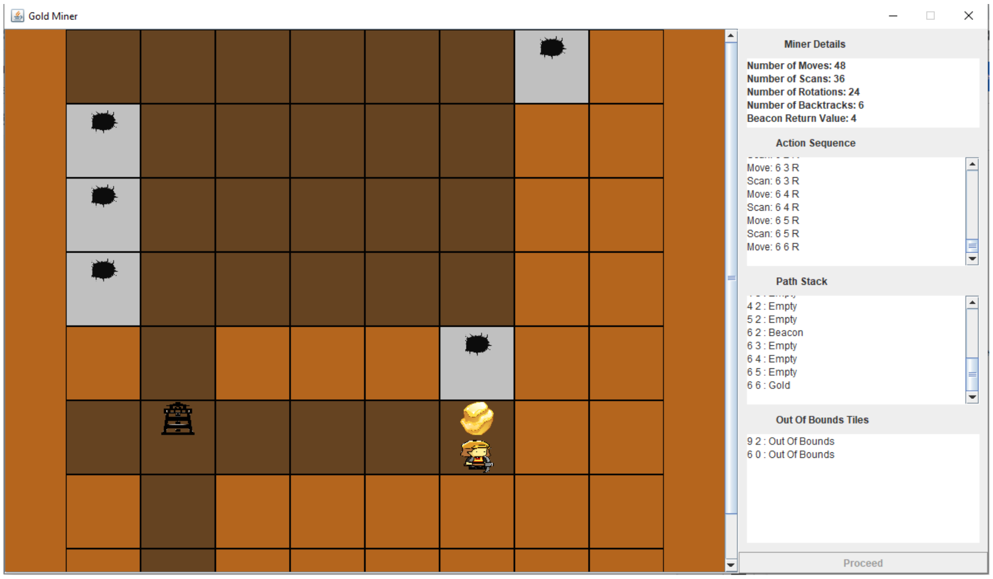
```
Miner Details
Number of Moves: 13
Number of Scans: 6
Number of Rotations: 11
Number of Backtracks: N/A
Beacon Return Value: Irrelevant

Action Sequence
Rotate: 3 5 D
Move: 4 5 D
Scan: 4 5 D
Scan: 4 5 D
Move: 5 5 D
Scan: 5 5 D
Rotate: 5 5 L

Path Stack
1 3 : Empty
1 4 : Empty
2 4 : Empty
3 4 : Empty
3 5 : Empty
4 5 : Empty
5 5 : Empty

Out Of Bounds Tiles
N/A
```

Meanwhile, the random miner chooses its next action randomly, without referring to the contents of previously scanned tiles, beacon return values, or previously visited tiles. Reflecting this behavior, the display does not keep track of the out of bounds tiles the miner encounters, and the beacon return value is listed as irrelevant to the random miner. Similarly, the random miner does not perform backtracking, making the number of backtracks inapplicable.

3. Terminal State

The terminal state can be further classified into two categories: the terminal success state, which occurs when the miner reaches the gold tile, and the terminal fail state, which occurs when the miner fails to reach the gold tile, either because it arrived at a pit tile or the gold tile cannot be reached given the board configuration.

a. Terminal Success State



Miner Details

Number of Moves: 48
Number of Scans: 36
Number of Rotations: 24
Number of Backtracks: 6
Beacon Return Value: 4

Action Sequence

```
Move: 6 3 R
Scan: 6 3 R
Move: 6 4 R
Scan: 6 4 R
Move: 6 5 R
Scan: 6 5 R
Move: 6 6 R
```

Path Stack

```
4 2 : Empty
5 2 : Empty
6 2 : Beacon
6 3 : Empty
6 4 : Empty
6 5 : Empty
6 6 : Gold
```

Out Of Bounds Tiles

```
9 2 : Out Of Bounds
6 0 : Out Of Bounds
```

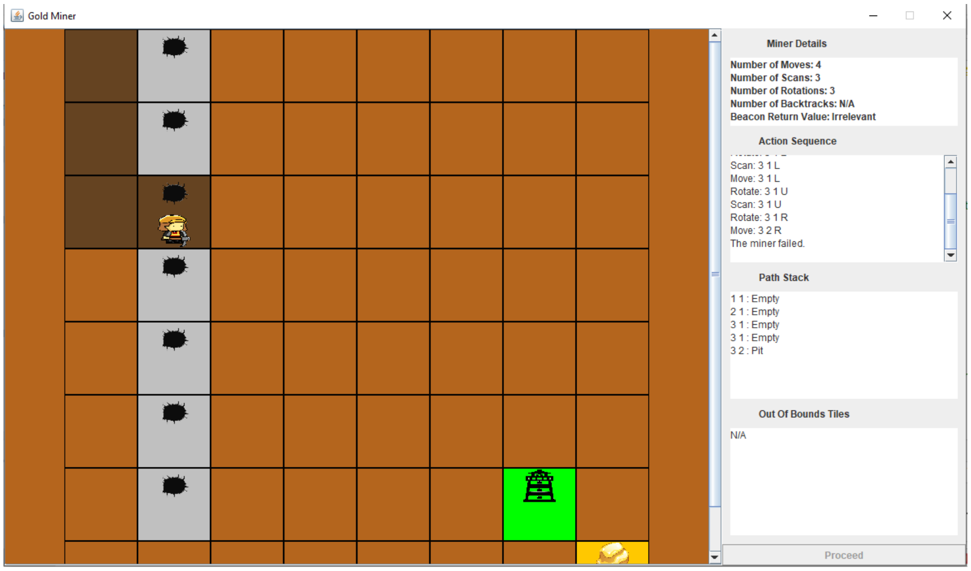
When the miner successfully reaches the gold tile, the action sequence ends and no further actions can be performed. For the fast view, the miner's movement ceases and the details on the side panel are no longer updated. For the step by step view, the `Proceed` button is disabled when the miner reaches the gold tile and further updates are similarly ceased.

b. Terminal Fail State

The terminal fail state covers two further scenarios: either the miner falls into a pit, or the board configuration is invalid.

THE PRESENTATION OF THE TWO CATEGORIES OF THE TERMINAL FAIL STATE START ON THE NEXT PAGE.

- Terminal Fail State (Pit Tile)



Miner Details

Number of Moves: 4
Number of Scans: 3
Number of Rotations: 3
Number of Backtracks: N/A
Beacon Return Value: Irrelevant

Action Sequence

```
Scan: 3 1 L
Move: 3 1 L
Rotate: 3 1 U
Scan: 3 1 U
Rotate: 3 1 R
Move: 3 2 R
The miner failed.
```

Path Stack

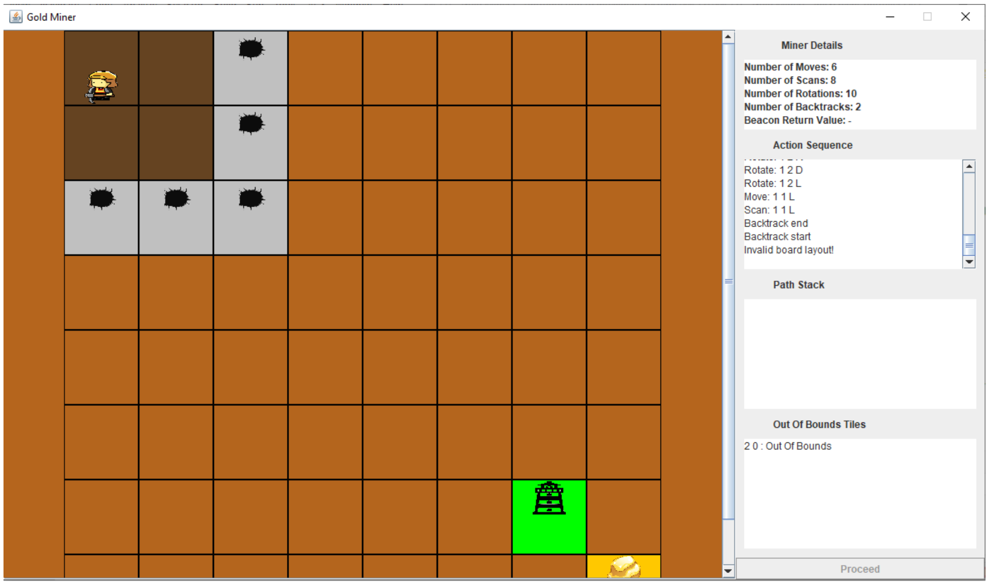
```
1 1 : Empty
2 1 : Empty
3 1 : Empty
3 1 : Empty
3 2 : Pit
```

Out Of Bounds Tiles

N/A

When the miner reaches a pit tile, further actions are ceased, similar to the system behavior for the terminal success phase. The string "`The miner failed.`" is added to the action sequence display to further indicate the terminal fail state. Due to the implementation of the search strategy for the smart agent, only the random agent can arrive at pit tiles.

- Terminal Fail State (Invalid Configuration)



Miner Details

Number of Moves: 6
Number of Scans: 8
Number of Rotations: 10
Number of Backtracks: 2
Beacon Return Value: -

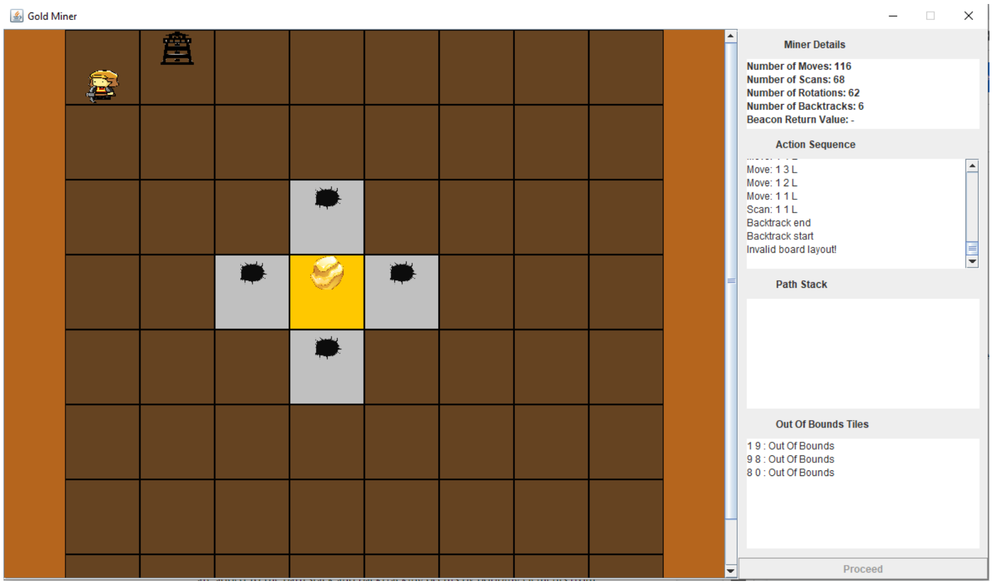
Action Sequence

```
Rotate: 1 2 D
Rotate: 1 2 L
Move: 1 1 L
Scan: 1 1 L
Backtrack end
Backtrack start
Invalid board layout!
```

Path Stack

Out Of Bounds Tiles

2 0 : Out Of Bounds



```
Number of Moves: 116
Number of Scans: 68
Number of Rotations: 62
Number of Backtracks: 6
Beacon Return Value: -
```

```
Move: 1 3 L
Move: 1 2 L
Move: 1 1 L
Scan: 1 1 L
Backtrack end
Backtrack start
Invalid board layout!
```

```
1 9 : Out Of Bounds
9 8 : Out Of Bounds
8 0 : Out Of Bounds
```

The miner determines the validity of the board configuration using its backtracking process; hence, entering the terminal fail state due to an invalid board configuration only occurs when the smart miner is used. The smart miner uses a modification of a depth-first search algorithm, where visited nodes are added to the path stack and backtracking occurs by popping elements from the stack until an element with an unvisited neighbor is found.

The emptying of the path stack during the active phase signifies that there are no longer any unvisited nodes that were reachable from the miner's initial position (as a consequence of the implementation, when the path stack is emptied, the miner returns to its initial position). Examples of invalid board configurations are those in which either the miner or the gold tile is surrounded by pits. When this occurs, the string "`Invalid board layout!`" is added to the action sequence and, similar to other scenarios of the terminal phase, further actions are ceased.