A picture containing text, outdoor, sign

Description automatically generated

Software Engineering Department

Braude College of Engineering

Capstone Project Phase B

**User Manual**

**B-23-2-R-5**

Students:

Daniel Balaish  
Liron Lavi  
  
Supervisor:

**Dr. Sarai Sheinvald**

Link to the project’s GitHub repository:

[GitHub - BMC Path Planning](https://github.com/danielbal21/BMC_Path_Planning)

# User Manual

When the software is first run, four options are available to the user

1. Importing a .sys file.
2. Generating a system randomly given specific parameters.
3. Designing a system manually.
4. Exiting the software

A screenshot of a computer

Description automatically generated

If the option to design a system manually was made the following window will open

A screenshot of a computer

Description automatically generated

Users can enter the grid size in the top textbox, with any changes clearing current input. To add bad robots (counteragents), users select the initial cell and click "Set Initial." Next, they choose the cell for movements and check relevant checkboxes for each required movement. All paths must form closed loops to prevent undefined behavior, per the designer's requirement. After setting a counteragent, users click "Add Robot" before designing another or proceeding to the next stage by clicking "Finish."

Invalid robot behaviors, such as lacking an initial state or containing open loops, will prompt an error message upon addition.

The Designer can be zoomed in and out with mouse wheel, it could be panned using the mouse right click and auto adjusted using the mouse wheel button.

If the option to randomly generate a system was clicked, then the following window will show.

A screenshot of a computer

Description automatically generated

Users should fill the form as they see fit, elaborate details on each field are presented in the book Software Development section.

Once the form is filled and valid (constraint are details in each field’s label parenthesis) the “Generate” button should be clicked.

No matter which input type was selected in the initial window, eventually you will reach the Solver Configuration window.

A screenshot of a computer login

Description automatically generated

The form should be filled as the user see fit, configuring timeout for the solution time or the max steps allowed for the agent to make.

The “View M2” button can be pressed to open a figure representing the counter agent systems Kripke structure.

A screenshot of a computer

Description automatically generated

Pressing a state in the Kripke structure open a text pane showing the positive occupancy indicator for the grid at that time step, the time step itself is labeled on the states, depicting a timeline.

The figure can be manipulated, panned, and dragged using the controls at the top of the window.

The “Save System” button opens a file save dialog, giving the option of saving the system into a Kripke structure for future use.

The “Solve” buttons initiates the algorithm and solver.

A green and white progress bar

Description automatically generated

After clicking “Solve” the solver window will pop up and show the time elapsed in seconds and iteration (current path length attempted).

If a solution is found it will be presented on a grid, showing how occupancy indicators change along with the agent route.

A screenshot of a graph

Description automatically generated

The agent is the blue square while the red squares are occupied cells or potentially unsafe cells, as the animation rolls the “T” field will increase, the animation runs in a loop. Statistics of the total time taken, and the average iteration time are also shown to the user.

In the event of a timeout or no solution a window will pop up with the relevant prompt.