

Project Title: AI Cooperative Path Finding  
Project Members: Brett Henne  
Date: March 9, 2014

## Functionality Implemented in Prototype

The prototype is written in C# and was designed using the MVC design pattern. Each agent navigates the map on a separate background thread. All of the AI algorithms have been implemented using an abstract class that provides implementation for common methods and abstract definitions for required methods. A screenshot of the UI is shown below in figure 1. The UI can be divided into the following sections:

- AI Parameters
- Map Generator
- Simulation Preferences
- Control
- Help
- Simulation Results.
- Map
- Map Edit

Each section is described below. The software is able to generate random and empty maps, edit walkable/non walkable nodes, edit agent start/finish locations, and draw multiple agents path finding through the map simultaneously. Agents also have the ability to cooperative by sharing map information when they are within the field of view of one another. The agents navigate through the map using 1 of 2 path finding algorithms, A\* or Dijkstra.

**Dijkstra's Algorithm:** This algorithm maintains a visited and unvisited list to keep track of the nodes in the agent's field of view. All node values are set to infinity except the starting node. The distances of the unvisited neighbors are calculated and added to the current nodes cost value. During this step the parent node is updated for use in back tracking when the algorithm has completed. If the calculated value is less than the value stored in the node the value is updated. Once all nodes have been evaluated the algorithm selects the node with the lowest value from the unvisited list and places that node in the visited list. The algorithm is repeated until the target node is found or if the next node to be evaluated has a node value of infinity, in which case the algorithm terminates.

**A\* Algorithm:** This algorithm is an extension of Dijkstra's algorithm with a heuristic approach. The algorithm maintains an open and closes list to keep track of the nodes. The open list maintains a list of nodes to be evaluated that may or may not be on the shortest path. After all adjacent nodes to a given node have been evaluated the node is moved to the closed list. The next node is chosen by calculating a path score. The path score is a combination of the movement cost to move from the starting node to the selected node plus the estimated cost to move from the selected node to the destination node. The second part of this formula is the heuristic.

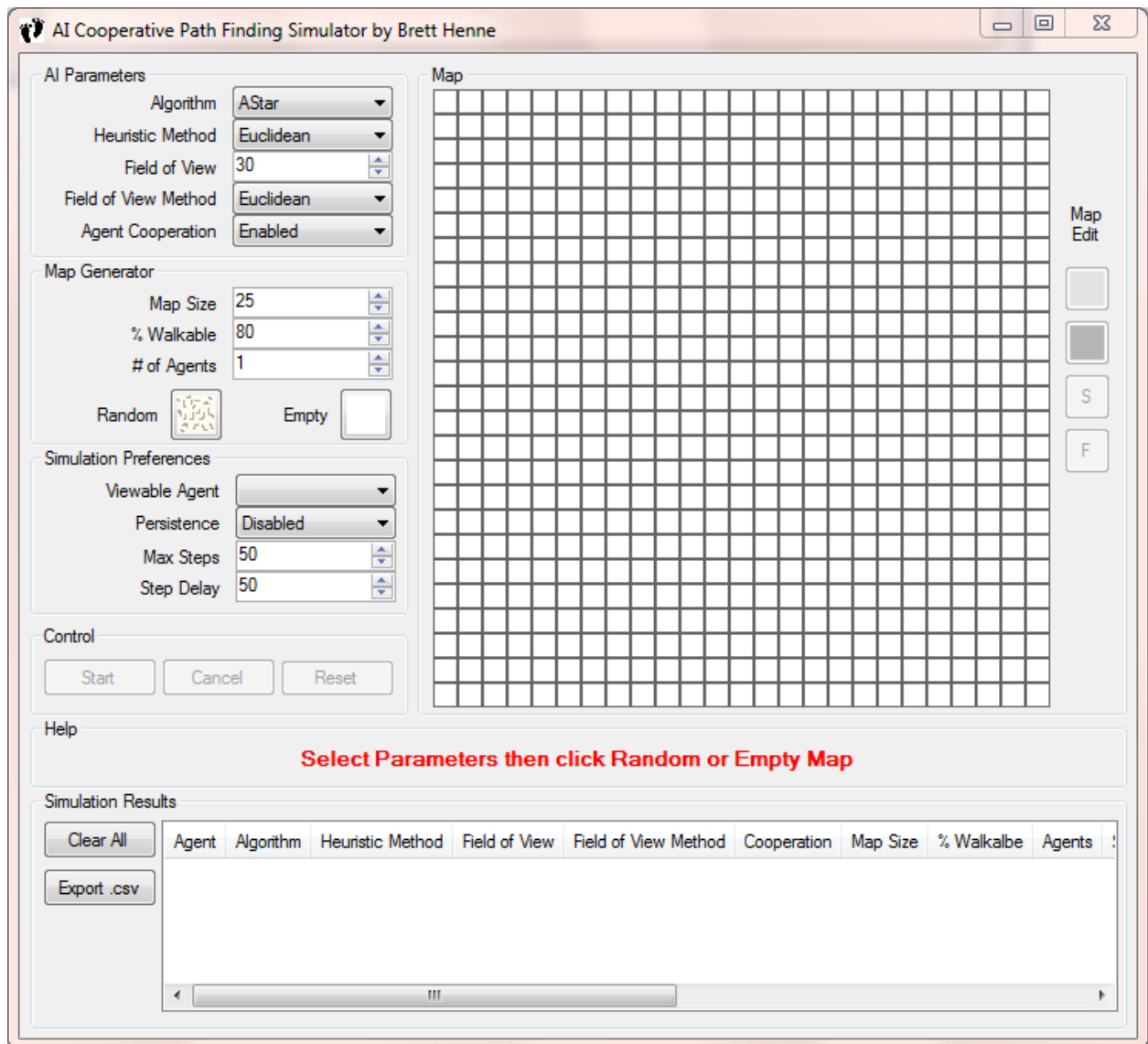
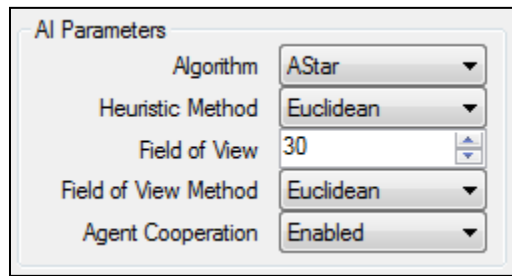


Figure 1 - Main UI Screenshot

## Section: AI Parameters

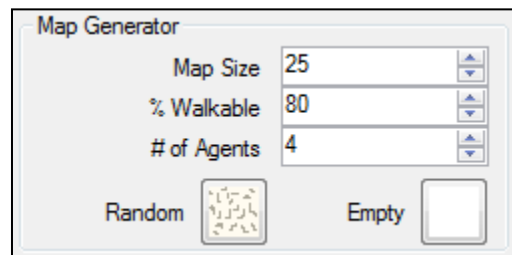


**Figure 2 - AI Parameters Screenshot**

This section gives the user the ability to select which path finding algorithm to use, the Heuristic method used (if applicable), the field of view of the agent, the field of view method used, and the ability to enable/disable agent cooperation. The field of view allows the user to define how far the agent can see. This affects how many nodes are in the agents map and how many nodes the algorithms use in determining which steps to take. To date, I have included A Star and Dijkstra's algorithm, of which the latter does not use the heuristic. The A\* Algorithm has the ability to select between the following methods for calculating distance between nodes:

- Diagonal Distance
- Euclidean Distance
- Manhattan Distance

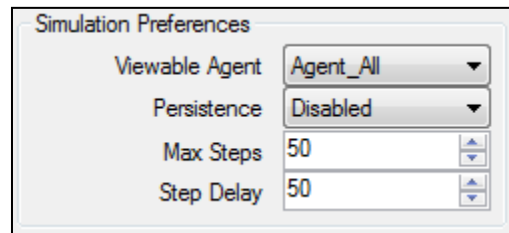
## Section : Map Generator



**Figure 3 - Map Generator Screenshot**

This sections gives the user the ability to generate random or empty maps of any size within reason. The agent(s) start and finish locations are randomly chosen among the set for both map types. The size of the map, the percentage of the map that is walkable, and the number of agents are all variables that can be modified. The software limits the number of agents currently to 10.

## Section : Simulation Preferences

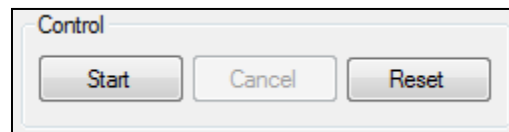


**Figure 4 - Simulation Preferences Screenshot**

This section is where various preferences can be set. To date the following preferences have been added to the software:

- Viewable Agent - this allows all agents to be viewed or just one.
- Persistence - this allows the path to remain colored or cleared as the agents walk through the map.
- Max Steps - this sets the maximum steps the agent can take before termination. This is needed due to scenarios of limited agent visibility. In these case agents can sometimes get trapped into areas on the map in which the algorithm selects between 2 alternating nodes indefinitely.
- Step Delay - this allows the user to modify the pace of the simulation, allowing for slower or faster runs based on needs.

## Section: Control



**Figure 5 - Control Screenshot**

This section is used to control the start and stop of the simulation and to reset the fog of war visibility of the map. It contains three buttons:

- Start - this will start the current simulation based on the current AI preferences and latest generated map. If Map Generator settings have been modified without clicking the Random or Empty button, these settings will be discarded.
- Cancel - this will asynchronously stop all agent threads.
- Reset - this will reset the map to the initial condition.

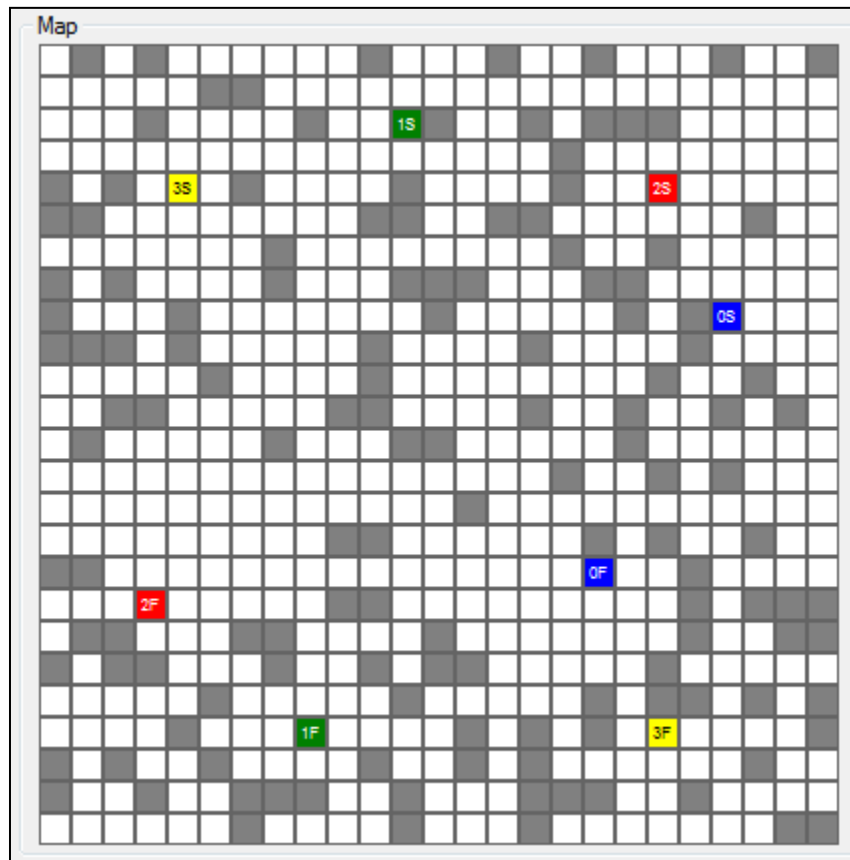
## Section: Help



**Figure 6 - Help Section Screenshot**

After having my wife try to use the software I quickly realized that I needed to add a Help section that guides the user. This section will display help information during various phases of run-time.

## Section: Map



**Figure 7 - Map Screenshot**

This is the map that all agents modify in order to show their start, finish, & current locations. The walkable nodes are white, and the non walkable nodes are gray. The colored nodes are agent start and finish locations, post-fixed with S and F respectively. It has a hidden feature that will display a tooltip of the nodes coordinates when the mouse is hovered over a node. This feature has been beneficial during various debugging phases of the prototype.

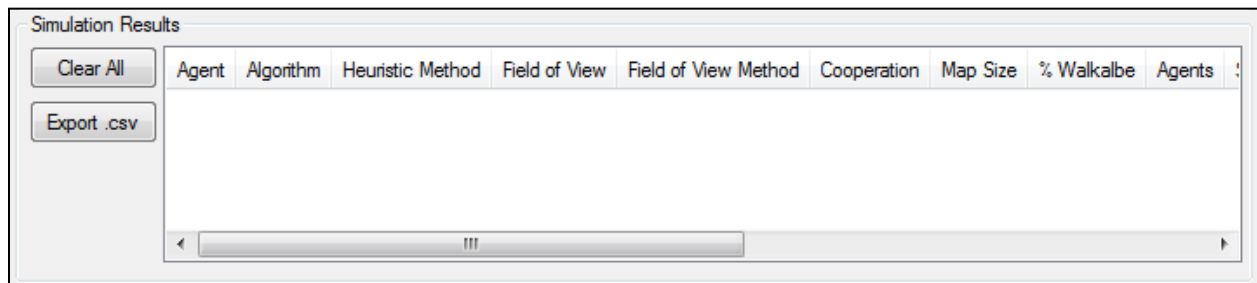
## Section : Map Edit



**Figure 8 - Map Node Editor**

This section gives the user the ability to modify the walkable/non-walkable nodes and the start and finish nodes for each agent. The walkable node is the white button and the non-walkable node is the gray button. The agent start and finish nodes are selected by clicking the button until the desired agent ID is displayed on the button followed by "S" for start node or "F" for finish node. The state of the edit feature is based on the last button clicked, therefore if the last button was the yellow "3S" above then every click in the map will move Agent 3's start node every click. If the last button clicked was the white walkable button then every click in the map will modify the node to walkable. Nodes that are currently selected to be an agent's start or finish node cannot be modified to walkable/non-walkable.

## Section: Simulation Results



This section is where all the simulation results will be viewed by the user. This section contains two buttons.

- Clear All - This will clear all the contents
- Export .csv - This will open a dialog allowing the user to save the current table data to a CSV file.

To date the following results are being displayed in the table per agent:

- Agent
- Algorithm
- Heuristic Method

- Field of View
- Field of View Method
- Cooperation
- Map Size
- % Walkable
- Agents
- Start - Finish Distance
- Steps
- Nodes Evaluated
- Original Nodes
- Shared Nodes
- Total Nodes
- Success



## Functionality Planned for Final System

My goal that I set for myself was to get all the major parts of the final system coded by the prototype deadline, therefore there is very little functionality difference between this prototype and the final system. This aggressive schedule has put me ahead schedule, however I plan on adding more analytics, fixing bugs, verifying algorithm correctness, and possible adding more path finding algorithms if time permits. As of right now having both a heuristic and non-heuristic path finding algorithm to compare and contrast analytically is where I plan to spend the majority of my time on this project.

## Evaluation plan for AI methods in final system

Each AI method will be evaluated over a variety of random maps to include the following variations:

- With and without agent sharing.
- Various field of views
- Various heuristic methods

Each test run will record the following data points, which will be analyzed:

- Algorithm used
- Heuristic used (if any)
- Agent Field of view
- Number of steps taken
- Destination reached successfully
- Number of nodes evaluated to reach destination
- Number of agent shared nodes evaluated
- Number of self discovered nodes in agents map
- Number of agent shared nodes in agents map

## Updated Task Schedule

Plan Submitted	2/9/2014
GUI Framework	2/17/2014
A* Algorithm	2/17/2014
Agent Multi-Threading	2/17/2014
Agent Visibility	2/24/2014
Dijkstra's Algorithm	3/3/2014
CSV File Output	3/6/2014
Agent Sharing	3/8/2014
Prototype	3/9/2014
Analytics	3/16/2014
Testing and Analysis	3/24/2014
Presentation	3/30/2014
Final System	4/18/2014