Path finding

Game Engines E2013

Jakob Melnyk, jmel@itu.dk

October 22, 2013

1 Introduction

This report is the result of a smaller project to implement the A-star (A*) algorithm in a chase between two automated agents in a game world for the Game Engines, E2013 course at IT-University of Copenhagen.

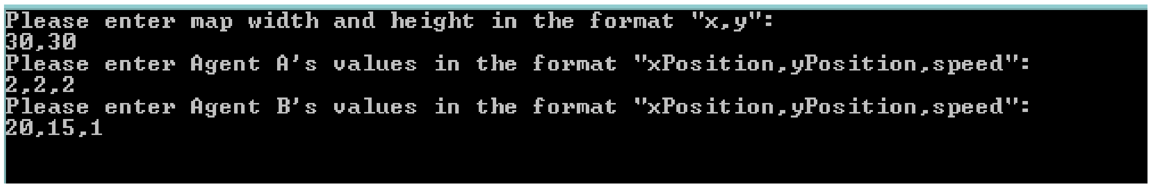
The constraints of the project were: Agent A (the chasing agent) must use the A* algorithm to calculate a path to Agent B. Agent B must escape by moving according to a finite state machine. The agents must move in a grid world where some grids are passable and others are impassable. The chase must be displayed visually.

I have implemented the project in C# based loosely on pseudo code by Rajiv Eranki[1]. The visual representation of the chase is displayed in the console.

I have worked with Jacob Grooss (jcgr@itu.dk) on some parts of this project (drawing of the map and parts of the movement for the fleeing agent).

2 How To Use

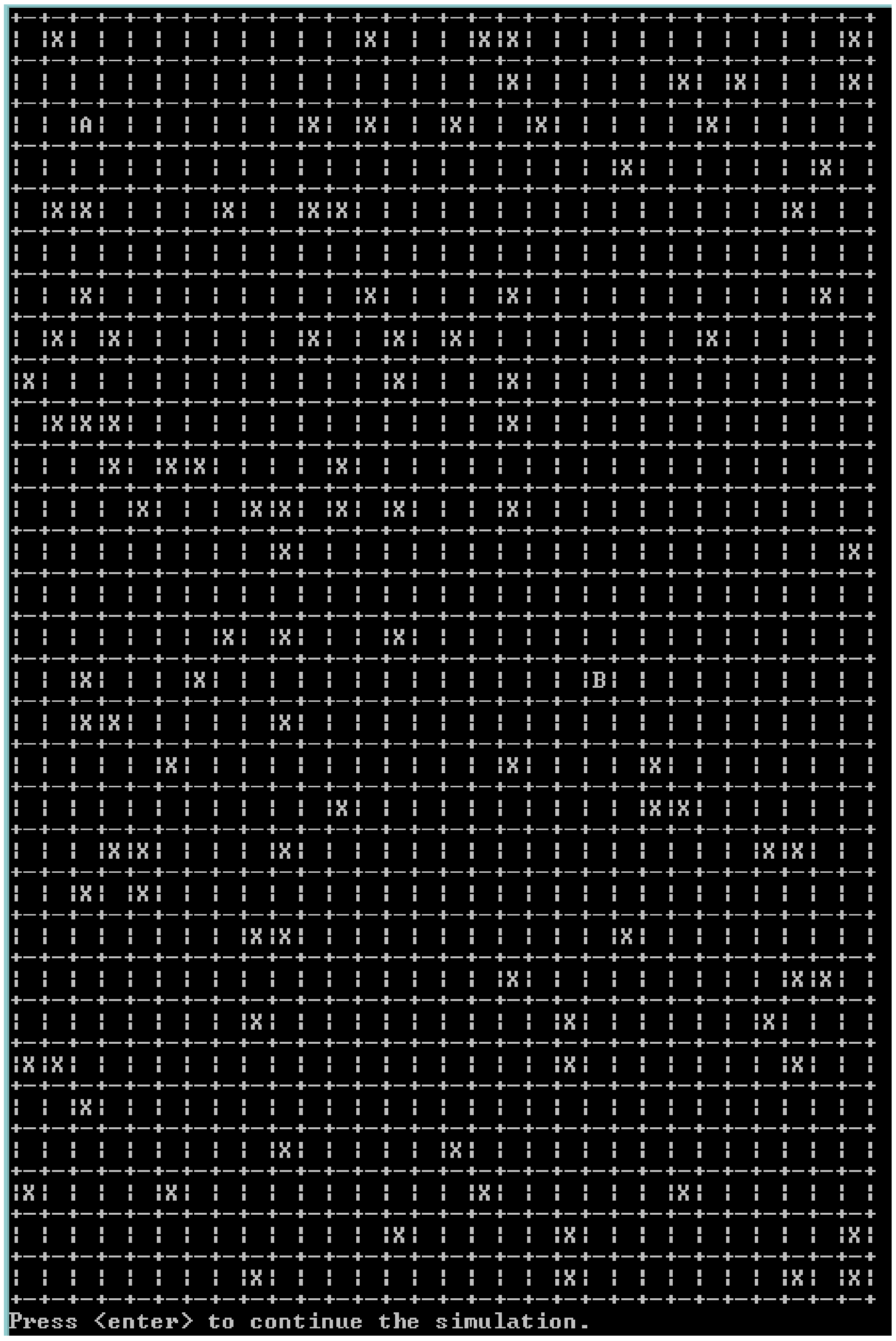
Running the Pathfinding.exe opens a console window asking for size of the game world, position of the two agents and their speed. The program automatically sets approximately 10% of the terrain to be impassable. After initialisation is done, the first map is drawn. Pressing the *Enter* key continues the simulation. If agent A catches agent B, the simulation ends.



```
Please enter map width and height in the format "x,y":
30,30
Please enter Agent A's values in the format "xPosition,yPosition,speed":
2,2,2
Please enter Agent B's values in the format "xPosition,yPosition,speed":
20,15,1
```

Figure 1: The information input screen.

[1]http://web.mit.edu/eranki/www/tutorials/search/



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |X| | | | | | | | | | |X| | | |X|X| | | | | | | | | | | |X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | |X| | | | | |X| |X| | | |X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |A| | | | | | | |X| |X| | |X| | |X| | | | | |X| | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | | | | | |X| | | | | | |X| |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |X|X| | | | |X| | |X|X| | | | | | | | | | | | | | | |X| | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |X| | | | | | | | | |X| | | | |X| | | | | | | | | | |X| |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |X| |X| | | | | | |X| | |X| |X| | | | | | | | |X| | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X| | | | | | | | | | | | |X| | | |X| | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |X|X|X| | | | | | | | | | | | | |X| | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | |X| |X|X| | | | |X| | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | |X| | | |X|X| |X| |X| | | |X| | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | |X| | | | | | | | | | | | | | | | | | | |X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | |X| |X| | | |X| | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |X| | | |X| | | | | | | | | | | | | |B| | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |X|X| | | | | |X| | | | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | |X| | | | | | | | | | | |X| | | | |X| | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | |X| | | | | | | | | | |X|X| | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | |X|X| | | | |X| | | | | | | | | | | | | | | | |X|X| | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |X| |X| | | | | | | | | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | |X|X| | | | | | | | | | | |X| | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | | | | | |X| | | | | | | | | |X|X| |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | |X| | | | | | | | | | |X| | | | | | |X| | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X|X| | | | | | | | | | | | | | | | | |X| | | | | | | |X| | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | |X| | | | | | | | | | | | | | | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | |X| | | | | |X| | | | | | | | | | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X| | | | |X| | | | | | | | | | |X| | | | | | |X| | | | | | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | | | | | | |X| | | | | |X| | | | | | | | | |X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| | | | | | | | |X| | | | | | | | | | |X| | | | | | | |X| |X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Press <enter> to continue the simulation.
```

Figure 2: Starting screen based on the given information.

3 Overview of the Program

The game world is constructed from a two-dimensional array of **Node**s. A **Node** contains information about its x/y location, its parent and its cost. The cost of the node is how many ”movement points” agent A must spend to move into the node (cost -1 means the node is impassable). Agent B can move freely around the game world.

After initialization of the game world and the agents, the map is drawn. Then agent B and agent A move until agent A occupies the same position as agent B. Agent B moves first in every iteration. If the Euclidean distance between agent A and agent B is less than twice the speed of agent A, agent B moves up to his own speed directly away from agent A.

After agent B has moved, agent A calculates a path to agent B using the A* algorithm. My implementation of the A* algorithm uses the game map as the nodes (as opposed to generating new ones as neighbours/successors) and thus is not suited for two units moving simultaneous. My implementation is also a bit slow for very, very large maps as every node must have their parent reset before a run of the algorithm.

4 Conclusion

The project successfully implements the A* algorithm and the chase between the two agents. There is impassable terrain for agent A and agent B only moves when he is alerted to agent A’s presence. The chase could be in improved primarily in two different ways.
One would be to make the fleeing agent more dynamic/smarter in his movement (not just straight away) and make the agent unable to go through impassable terrain.
Second would be to make the agents move one step of their path per simulation step (instead of their entire speed).

4.1 Future work

Other than the improvements mentioned above, I would change my A* implementation to be more flexible in its choice in distance calculation (instead of just Euclidean). Additionally I would rewrite the algorithms approach to nodes so that new ones are generated (making it more efficient for large map sizes).