Pathfinding in S3

The goal of this project is to learn how to implement pathfinding methods, which are the most common technique in computer games of all genres. Specifically, you will implement an A* approach to path finding.

In this project you will work with S3, a simple RTS game designed to test AI techniques. S3 looks very simple and has pretty basic graphics, but it features most of the characteristics of more complex RTS games such as Starcraft. The version of the game that you will have access to does not feature any path-finding routine. If you run it you will see that units do not move.

Specific Project Tasks:

- Download the S3 source code from here.
- Set up S3 in your favorite Java IDE (Netbeans, Eclipse, etc.).
- To test that you have it properly setup, you can try to run the file s3.base.Main. You should see S3 launch (by default it's set up for playing human vs AI). Since there is no pathfinding, no unit should move.
- Read the documentation at the bottom of this page to familiarize yourself with the code.
- Implement a pathfinding module using the A* algorithm. For doing so, just complete the ai.AStar class.
- Create a short 3 minute video demoing your project, and send it to the instructor (do not send the video file! please just send a link (e.g., DropBox link, Youtube link, etc.). You will have to present this video in class.
- Turn in the source-code of your project, plus a 1-2 page description of what you did before midnight the due date. Submissions will be done via Blackboard

Notes:

- S3 is a simple RTS game designed to test various AI techniques. The package you downloaded comes with different AIs and different maps. The different AIs will not come to life until you implement the pathfinding module though. If you want to play with a different map you can pass the following parameters to the main class: -m maps/NWTR8.xml (where you can change the name of the map to any in the maps folder)
- By default the game starts by pitting a human against a built-in AI. But you can also change it by specifying different parameters. For example using the

parameters "-p 0|player1 -p 1|player2|ai-footmen-rush" pits a human (0) controlling player1 to a built-in AI (1) controlling player2 (specifically the built-in AI being used is "ai-footmen-rush"). If you want two AIs to play each other you can do "-p 1|player1|ai-archers-rush -p 1|player2|ai-footmen-rush", etc.

- When a unit in the game wants to move, it generates a request for finding a path to the pathfinding module that you will implement. You can generate some of those requests just by trying to move any of the units you as a human player control. To do so, left click on a unit, and then right click somewhere else in the map, that should trigger the unit to move (but will only do so once you have implemented the pathfinding module).
- To check if a cell in the map is free, I recommend using the function "anyLevelCollision" of the S3 class.
- Even if coordinates are represented as doubles, you can treat them as integers (i.e., the position to the right ox (x,y) is (x+1,y))
- The path that you return should NOT include the start position of the unit, but should include the GOAL position. Also, the path should go from start to goal, and not the other way around. If there is no path, you should return "null".
- By default the game will run very, very fast (since it was created to test AIs). You can make the game run slower by changing the variable REDRAWING_PERIOD in S3App. This is the duration (in milliseconds of each game frame). Values like 10 or 20 might be more appropriate for when a human plays the game