CAP6671: A* Search Project (apetkova@cs.ucf.edu)

1. The project can be implemented in C, C++, or Java.

2. All implementations should be emailed to me at apetkova@cs.ucf.edu with CAP6671 in the subject field. It will be appreciated if the filenames also contain your name.

3. A report containing description of the algorithm should be submitted together with the solution. In addition, please specify what tool you used to compile your program. Explain what files you are submitting and how they should be compiled. In case, you decide to have flags indicating choice for forward or backward A* search or different tie-breaking techniques, please explicitly state so.

4. It is expected that each additional algorithm that you need in your solution, such as one to create and maintain the OPEN and CLOSED data structures, will be also implemented from scratch. That is, no libraries other than for input/output can be used.

5. We generated five gridworlds that you can use for input. They will be posted on Webcourses. If you want to further test you solution, you can generate more grids using the mazagenerator tool posted on http://idm-lab.org/project-m/project1.html. The format of these gridworlds is as follows:

```
Number of rows in the gridworld
Number of columns in the gridworld
Gridworld Legend:
s := start cell
g := goal cell
x := blocked cell (obstacle)
_ := unblocked cell
```

We might use a different set of gridworlds for grading but they will follow the same format.

6. Your implementation must also output the trajectory of the game character in two forms. First, it must output the gridworld and indicate with 0 the cells that correspond to the path taken by the game character from the start cell to the goal cell. Second, output the trajectory of the game character. (e.g. A5,A4,B4, etc.).

7. Make sure your source code is well commented. Poorly commented code will be penalized! Do not forget to include your name as a comment in your source files!