CS425 Game Programming 1
(Fall 2016)
Homework Assignment 4
(Avoidin')

Due Thursday, October 13th 2016 at 3pm

Submission:

- This is an individual programming assignment. Please use your code from assignment 3 as a base for this assignment.
- When you have completed the assignment, delete the Debug and Release directories and the .sdf file. Then zip together the rest of the directory. Be sure to include your .vcxproj and .sln files along with all of your source code files. If the project cannot be loaded and run properly, you will get a zero. Name the zipped file in the following way: **LastName_FirstName_HW4.zip** and submit it through Blackboard by the due date.

The goal of this assignment is to use A* to plan paths for the character to move from one grid node location to another.

For all parts of this assignment, use the Sinbad character and an environment with obstacles loaded using your level loader. Be sure to test your implementation with several different obstacle configurations including boundary cases. Make sure that your implementations can generalize to different layouts and environments.

1. Use the Grid class implemented in assignment 3 as a foundation for your A* implementation.
2. Implement the A* algorithm presented in class to enable your characters to avoid static obstacles as they move around the grid (Other characters should NOT be considered obstacles).
    a. Load a level with a few static obstacles (e.g. knots) on the plane floor. They should be centered and scaled such that they fit within a grid square or automatically associated with multiple blocked squares. Sinbad characters should also be created through the level loader.
    b. Using the GridNode and Grid classes, implement the A* algorithm as presented in class.
        i. Your implementation must generalize to any reasonable grid size and dimensions.
        ii. Be very clear about how to change the start and goal nodes. Perhaps, allow a pointer to a grid to be passed as a parameter.
    c. For the open and closed lists, chose an appropriate data structure from the C++ standard library.
    d. Add a walkTo method to the Agent class that enables the agent to walk from their current position (i.e. node location) to a specified goal position (i.e. node location) while avoiding obstacles.
        i. The walking behavior from previous assignments should be used.

ii. When the A* algorithm has determined a collision-free path, use the Grid class's printToFile method to print out the path. For example:

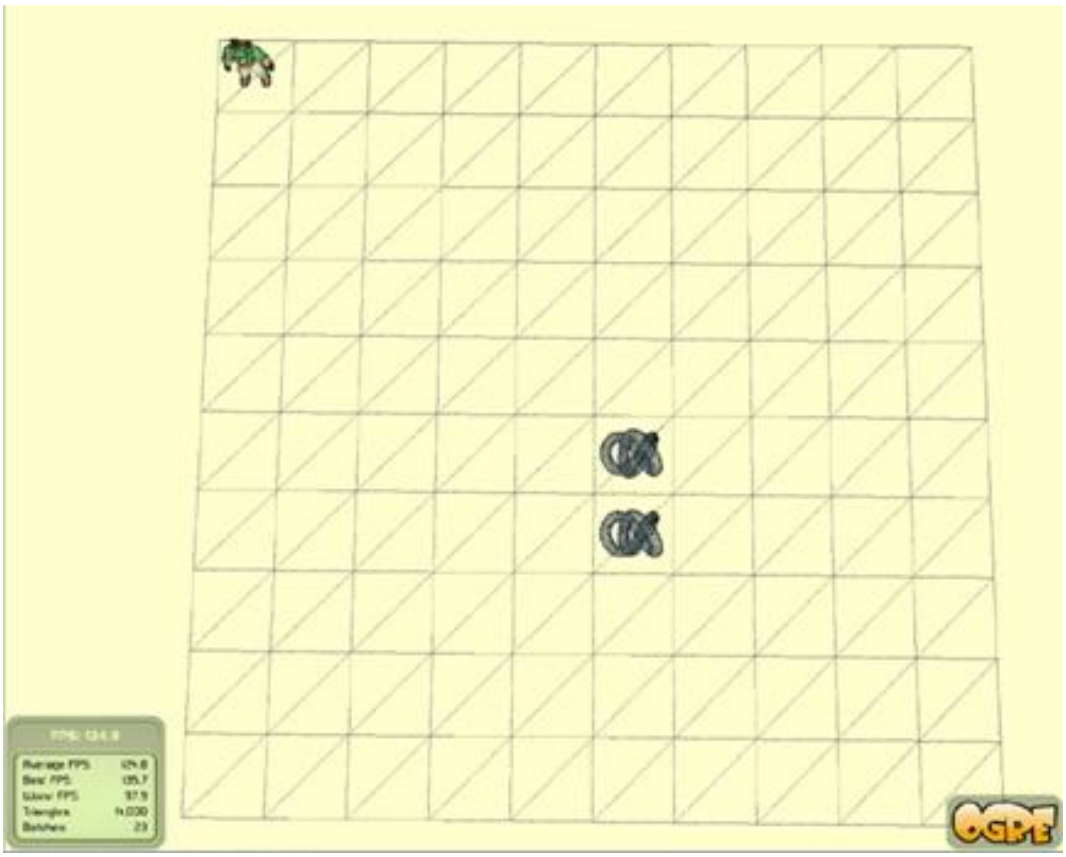
```
S . . . . . . . . .
. 0 . . . . . . . .
. . 1 . . . . . . .
. . . 2 . . . . . .
. . . . 3 4 5 . . .
. . . . . B B 6 . .
. . . . . . . . 7 .
. . . . . . . . . 8
. . . . . . . . . 9
. . . . . . . . . G
```

1. Where S is the start node, the B's are blocked locations, and nodes along the path go from 0 through 9 and then start at 0 again.
2. Updating the contains field of the GridNodes will be required.
3. This file can also help you to double check and debug your implementation.

3. Testing: Test your implementation with different environment sizes and layouts. Test boundary conditions. Test multiple agents finding paths and traveling at the same time. Test what happens when there is no possible path to be found. Test different length paths, including 0. With your submission include at least 5 different environments (i.e. levels) and their corresponding printed grid files (with the paths).

A few tips/hints/things to think about:

- Start NOW. Work hard. Think about what you are doing and why.
- You may want to pass a pointer to the grid into the agent when or after it is constructed.
- Remember to set the agent's current position within the grid
- A* will be created and run from within the agent's walkTo method
- Grid coordinates are different from grid positions. The getPosition method of GridNode takes this into account.
- I will be looking closely at your code, so use good practices, including deallocating memory and reasonable error checking.


FPS: 124.8
Average FPS 124.8
Best FPS 125.7
Worst FPS 97.5
Triangles 14,030
Batches 23
OGRE