Project #2: A\* Pathfinding (due week 7).

 **Absolute requirement**: Draw the open list of quads as one color, the closed list of quads as another color, draw the final path, and have Tiny follow the path.

 (20%) Draw one step of A\* each frame / Draw entire path at once (use toggle Single Step)

1. When single step is on. Each frame it draws one step of A\*.
2. When single step is off. Draw entire path at once.
3. Color grids on openlist. Blue is recommended.
4. Color grids on closelist. Can't be the same color as above. Yellow is recommended.
5. Arrows shows path from tiny to goal.

 (15%) A path is always found if it exists (completeness)

1. If there is path from tiny to goal, must find the path.
2. If there is no path from tiny to goal, must not find the path. A\* search must end when no path is found.
3. When movement is on. If there is a path, tiny must walk to path.
4. When movement is on. If there is no path, tiny must stay at her own grid.
5. Path must be reset if tiny starts a new search.

 (15%) The path is optimal with a heuristic weight of 1.0 (1.01 for Octile heuristic)

 (15%) The path is optimal with a heuristic weight of 0.0, and the search grows in a **circular** pattern

 (5%) The path uses diagonals and doesn't run into corners

 (5%) Calculate heuristic using both Euclidean, Octile, Chebyshev, and Manhattan methods (use toggle)

 (5%) Straightline optimization (use toggle)

1. When straightline is on and there is straight line from tiny to goal, path must be that straight line. A\* should not be executed.

 (10%) Rubberband final path where possible (use toggle)

1. Whole path must be rubberbanded, including beginning and end.

 (10%) Smooth using a Catmull-Rom spline (use toggle)

1. Should do smoothing even if the path only contains 2 or 3 nodes.
2. Smoothing must be done for every 2 adjacent points.
3. If rubberbanding and smoothing are both on:
   1. First run rubberbanding, then add points back evenly, then do smoothing.
   2. When adding points back. Distance between any two points should not be larger than 1.5 times grid width.

 (-10%) If it crashes at any time

 (NOTE: Project 3 will regrade all of these items, so there is enormous motivation to get everything done correctly on this project)

 (extra credit) See slides

**Turnin Instructions (-10% for missing directions )**

 Remove "Project1" directory and homework1!

 Submit everything in "Project2" directory.

 Submit all code, resource files, and exe (-5% if I can't double click exe and run) in exactly same order as the framework. (-5% for large unnecessary files: ".ncb" file, hidden subversion directories/files, a ipch/Debug/Release dir)

 Must be able to build project by running Batch.bat.

 Submit readme.txt with any special directions or extra credit explained.

**Note**

1. The framework supports VS2013 and VS2015.
2. By default the framework uses VS2013. You can change the setting via:

Project->Property->Congiguration Properties->General->Platform Toolset

1. After you finish the project. Please change “vs\_version” variable in Build.bat file for the version of Visual Studio you use. (2013 for VS2013, 2015 for VS2015)
2. Double click Build.bat to build the project and place executable in “Pathfinding” folder. Then test your executable to make sure it runs and does not crash.
3. Change your executable file name to your user name. (So when TA build your project the executable does not get overwritten)
4. If you attempt speed challenge, mention in readme.txt
5. Before submission, double click Clean.bat to remove all unnecessary files.
6. Do not change “Pathfinding” folder name.

**Checklist**

1. Executable is in **release** mode.
2. Required: Draw the open list of quads as one color, the closed list of quads as another color.
3. Required: Draw the final path, and have Tiny follows the path.
4. Feature: Draw one step of A\* each frame if “Single Step” is on. Draw all steps if “Single Step” is off.
5. Feature: A path is always found if it exists.
6. Feature: The path is optimal with a heuristic weight of 0.0. And the search grows in a circular pattern.
7. Feature: The path is optimal with a heuristic weight of 1.0 for Euclidean (and 1.01 for Octile).
8. Feature: The path uses diagonals and does not run into corners.
9. Feature: Implement Euclidean, Octile, Chebyshev, and Manhattan methods.
10. Feature: Straight line optimization. Do not do A\* search if there is straight line.
11. Feature: Rubberband final path where possible.
12. Feature: Smooth using Catmill-Rom spline.
13. The program does not crash.
14. Wrote special directions, experience, and extra credit in the readme.txt.
15. Submitted on the network (“Submit”) folder BEFORE 6:00PM.
16. **DO NOT ZIP THE PROJECT**.
17. **DO NOT BUILD IN x64.**
18. All code and resource files were in “Project2” folder.
19. Removed all other files and folders in your submission folder.
20. Changed “vs\_version” variable in Build.bat file for the version of Visual Studio you used.
21. Did you test the exe runnable without the debugger? Test it by double clicking the exe to run (NOT IN THE DEBUGGER).
22. Did you remove all unnecessary files? Include but not limit to: the debug folder, release folder, ipch folder, ncb file, sdf file, and all subversion files. (Use Clean.bat)