CSCI 4511 Writing 4

Sid Lin (linx1052), Noah Park (park1623)

March 21, 2020

Our project will be an analysis on common AI search algorithms, as discussed in the first third of the class. We will be using at least three methods of search: BFS, a variation of DFS, and A*. In class, we learned that these algorithms can be used to solve some very common AI problem scenarios. The particular focus of our project will be the slide puzzle problem. Both of us are currently taking this course while also enrolled in a game programming course, CSCI 4611; the slide puzzle scenario can be used to research the different methods of search while having the possibility of being published into an actual mobile game in the future.

The first step to completing this project is creating the search space and implementing the algorithms. The data structures used to model the slide puzzle can be done in a similar way to the AIMA code in any object-oriented language. Implementation of the search algorithms is also a matter of translating pseudocode into our respective language. After the implementation step, we could start testing and recording various benchmarks, such as number of moves, actual time, or size of the search space.

We have a few languages in mind when implementing the slide puzzle problem. A common game engine, Unity, uses C# code for scripting. If we use C#, we would be able to construct a game using the Unity engine on top of the project code. Swift is also another option because it be ported to an iOS app, but it is restricted to OSX users only. XCode, Unity, and Visual studio are all free IDEs, and this will be the only software we need.

To evaluate our solutions, we can use benchmarks for each of the algorithms along with different presets of the slide puzzle. For example, we can run the three algorithms on the same random shuffle of the board. In terms of statistics, appropriate ones would be total time, number of moves, or storage used. Expanding the board to a 4x4 or 5x5 could possibly highlight the differences in our selected algorithms. Finally, multiple heuristics can be tested out with the A* search.