Bryce Vanderlaan and Kenneth Sullivan

Prof. Herve

CSC 412

Due December 21st, 2022

Final Project: Report

**Implementation Decisions**

Right from the start of the project, we felt it would be beneficial to structure the code in such a way that it was always able to accept and create multiple travelers depending on command line arguments. Instead of hardcoding the number of travelers to 1 for Version 1, we simply allocated our array of threads with only size 1. This made the creation of Version 2 quite easy, as all it really entailed was passing a different command line argument in the build script. Another important decision made very early on in the development process was that some kind of pathfinding algorithm should be used. From our perspective, this decision made the most sense since it had the potential to expedite later debugging sessions at the upfront cost of writing a potentially tricky algorithm. Considering we also wanted each traveler to calculate its path to the exit quickly, the algorithm turned out to be tricky indeed. A\*, while generally known to be the best of the pathfinding algorithms, is quite difficult to implement, and took a considerable amount of time both to write and debug. One particularly stubborn bug had to do with attempting to both allocate and initialize the 2D array of Cell structs used for A\* at the same time. We were unaware of this, but evidently the memory address of a struct in an array is different before and after its initialization. Our lack of knowledge on this front

**Version 1:**

For Version 1 we needed to create only 1 thread but for the purposes of future versions we decided to start with a Global ThreadList variable that would store the threads used. Then, inside the initializeApplication() function we initialize the thread to use the Traveler\_Thread() function, which creates the traveler and then creates a path from the starting point of the traveler to the EXIT and then makes the traveler move. We made the threads join the main thread\_\_\_\_\_\_\_\_\_....