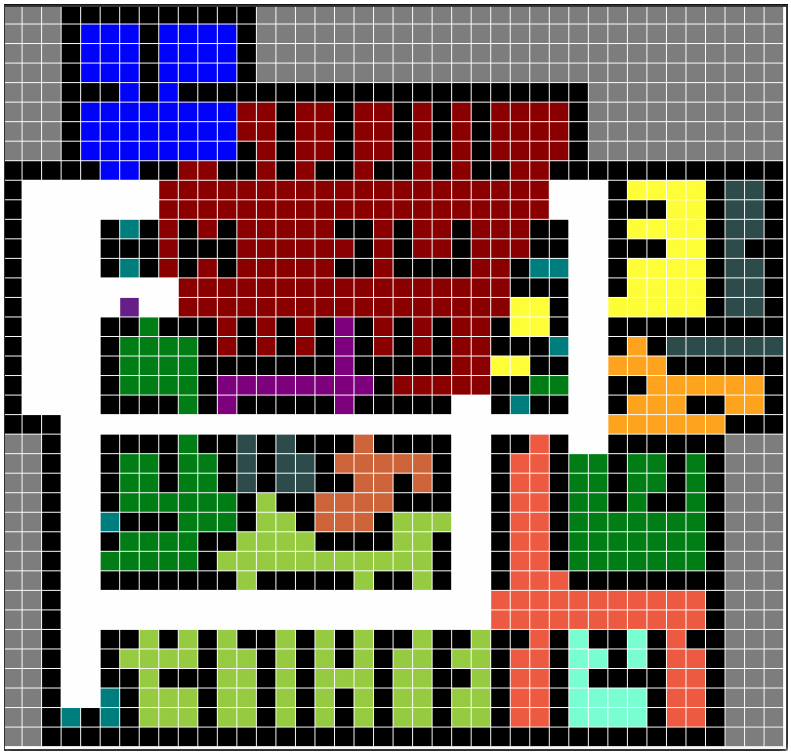
Sean Rock and Chris Nolan
CSC 362
Dr. Forouraghi
May 7, 2024

Final Project Report

Our final project required us to program a robot nurse to make deliveries on a hospital floor. We did this in AI_Final.py using AStarMaze_V1.py as a starting point. The first step of creating the program was defining the hospital floor layout. Based on the provided floor plan image, we worked together to encode a matrix with numbers representing the wing of each location. We then edited the source code to make each ward represent a different color on the canvas.

Our Hospital Representation



The next step was allowing for the starting point and destination list to be set. We prompted the user from the console for input. This was handled with regex and other input checking for illegal characters and length requirements. If the input is proper, it is then appropriately separated into arrays of tuples for different maze coordinates. If the coordinates mark an illegal space (wall, out of the hospital, or out of bounds), the user is forced to correct the mistake to continue to priority ordering.

Some Homebrewed Regex Used

- \(([^\)]+)\) Identify values between parentheses
- (?:[-.][0-9]+)+ Identify any negative or decimal numbers
- [a-zA-Z] Generic letter identification

Once the destinations are correctly input, we programmed a method to order the traversal of location by ward priority. After separating the coordinates by priority, they are then grouped with other destinations in the same ward so that all the deliveries to a specific ward are done consecutively.

Priority List

- **Priority 5**: ICU, ER, Oncology, Burn Ward
- **Priority 4**: Surgical Ward, Maternity Ward
- **Priority 3**: Hematology, Pediatric Ward
- **Priority 2**: Medical Ward, General Ward
- **Priority 1**: Admissions, Isolation Ward

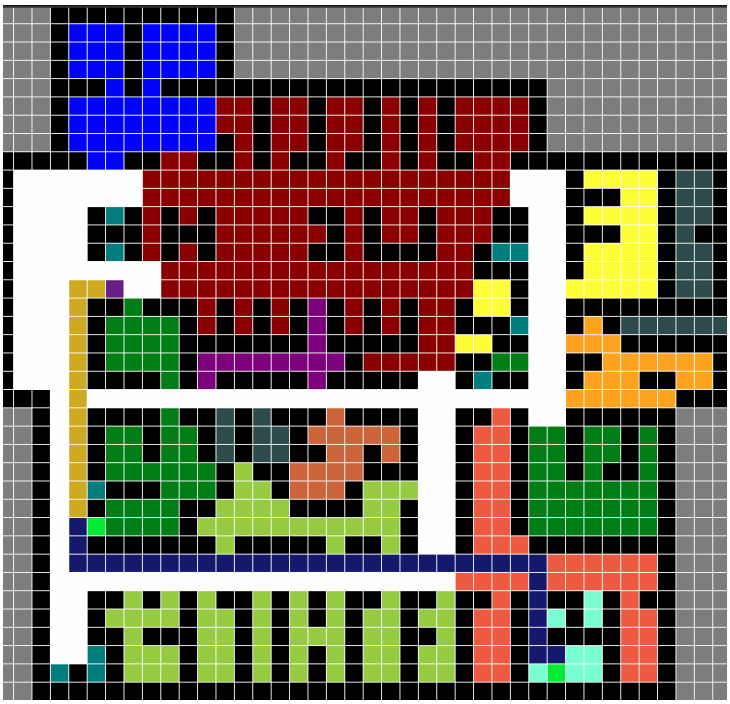
With the start position and destination list now set properly, we move onto the A* algorithm. The final code is altered from the source code in order to find the path to several destinations in a row going from one endpoint to another. It returns “Success” if at least one location is able to be reached and “Failure” otherwise. It also returns a list of all points that cannot be reached because they are blocked by walls.

The code uses the recursive function do_one_frame in order to print each tile one by one. It saves the locations of each path to a destination to list l2, reverses it so that the start position is printed first, and then adds each coordinate of the path to array l1 that has the complete, combined path. After a destination coordinate is reached, the color variable is incremented or reset in order to cycle through colors to make it clear when a new delivery is being made. The function is called recursively with a short delay until the path is completely drawn on the graph.

Our project was a success due to our effective collaboration. We always worked on the project together, in the same location, which enabled us to ask questions or assist with any bugs or problems we encountered. We also divided the project into sections, assigning each section to the person who could best handle the task. We played to each other's strengths and supported each other in areas where we struggled. Working together like this allowed us to both have a deep understanding of the project since we were involved in both parts of its creation. This effective collaboration was a result of working together on multiple projects, and understanding how each of us works.

Sample Run

(Start position = Dark purple, Gold = 1st Path, Blue = 2nd Path, Green = destinations)



Member 1: Chris Nolan

My teammate and I agree that I handled **50%** of the overall project. My specific tasks included:

Task 1: I worked with Sean in order to design the hospital representation that we would use in our program. The matrix is based on the image provided in the canvas page but with a 40x38 grid overlayed on top of it. We scanned the image manually row by row and I would tell Sean what value to fill into the matrix based on the grid on the picture.

Task 2: The first part of the project that I worked on was sorting the list of destinations given to the user by priority and then grouping the same wards together in that priority.

Task 3: My second task was to handle all user input and error checking that was performed for the project. This included creating multiple regex statements to handle grabbing the information from the command line and making sure what the user entered was valid and prompting them to retype any incorrect information. This was important since there were no checks on the A* side of the project, so any errors in input had to be caught here.

Task 4: Grammar and spelling check on the final report and add some additional information where required.

Member 2: Sean Rock
My teammate and I agree that I handled **50%** of the overall project. My specific tasks included:

Task 1: I worked with Chris in order to design the hospital matrix based upon the floorplan.JPG image. We went row by row, and he'd tell me which ward and then I'd number it in the matrix.

Task 2: I focused on the functionality of the path finding algorithm in our AI_Final.py program. I ensured that the A* algorithm could find multiple paths, print them one tile at a time, change colors after a destination is reached for visual clarity, and return whether the program succeeded or failed to find a path to at least one location.

Task 3: I put together most of our project report. I provided informative screenshots and detailed the approach of our code.