Ben Grass, Cameron Russell, Andrey Yao

Design

In order to implement the smell food sensing I used Dijkstra's shortest path algorithm, but instead of having access to all the graph nodes at the beginning of the algorithm, the algorithm creates nodes each time it encounters a new tile. In other words the algorithm starts with only a single node in the frontier queue, the one that represents the critters current tile. That node is then expanded, which uses a static method from World to get all of a given tiles neighbours, before converting those tiles into GraphNodes with the proper distance and relative direction and finally adding those GraphNodes to the priority queue (sorted by distance). If the node to be expanded is a food node, the process ends and the output is returned in the format dist * 1,000 + relDir.

This was made much easier by the fact that I returned the array of neighboring tiles in order of their absolute directions from 0-5, so I could then just iterate over that array and treat them as being relative to the critters direction.

Most other aspects of this assignment will be bug fixes from our A4 through A6 code. We will develop the design of the server and client aspects of this assignment by attending OH and conducting thorough research on how this should best be implemented.

Division of labor

- Go through previous assignment and correct all current bugs which include:
  - Mutation errors, **Cameron**
  - GUI refresh bug, **Andrey**
- Implementing a thread-safe web server that runs the simulation and responds to HTTP requests from multiple clients according to the given API, **Cam + Andrey (+ Ben) (+Third Party Mercenary for Hire)**
- Updating your GUI to be an HTTP client that talks to your server, **Cam + Andrey**
- Implementing a thread-safe ring buffer, **Ben**
- Smell/Food Sensing **Ben**
- Solving the written problems, **completed by Ben**
- Preparing your presentation for the course staff, **all**