ECON381 Fall 2024

Semester Project

In this project you are expected to do triangulation on a hexagonal grid, for an application similar to the controversial ShotSpotter system. You are welcome to watch this short news reel -- https://youtu.be/USNJ2eOme8E – to understand how triangulation works.

Our projects works for watch towers located in a large park. These watch towers could be used to watch for wildfires and other events of interest (not necessarily gunshots).

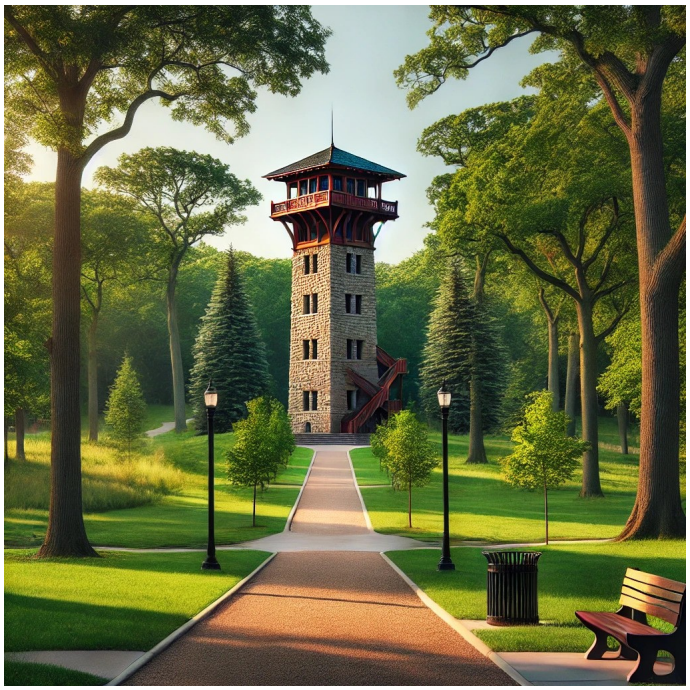

Figure 1: A beautiful watchtower

Their locations are arranged so that they are at the center of cells within a (pointy top) hexagonal grid system.

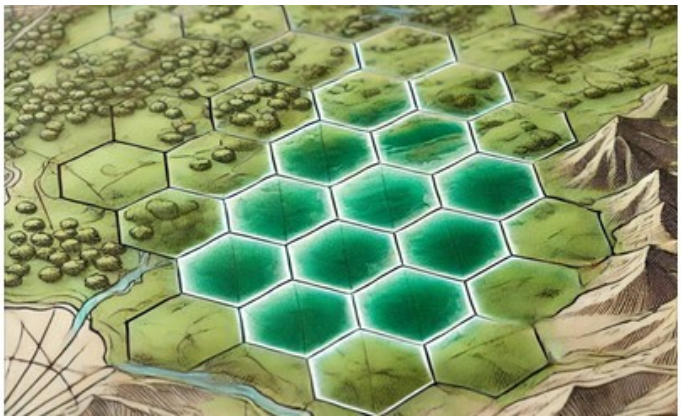

Figure 2: Hexagonal grid projected on the national park's geography

The particular type of sensor we will be using in this project cannot sense the exact location or direction of the source, but **can predict how far away the source is in terms of the number of cells**.

This also requires some sort of triangulation. For example, in the image below, two towers (denoted with the darker colors) have each reported that they have sensed something **within 3 cells** (d=1, 2, or 3). Each reported area is denoted with a lighter version of the tower. So the cells that are both within 3 cells of two towers are the intersection (denoted green).

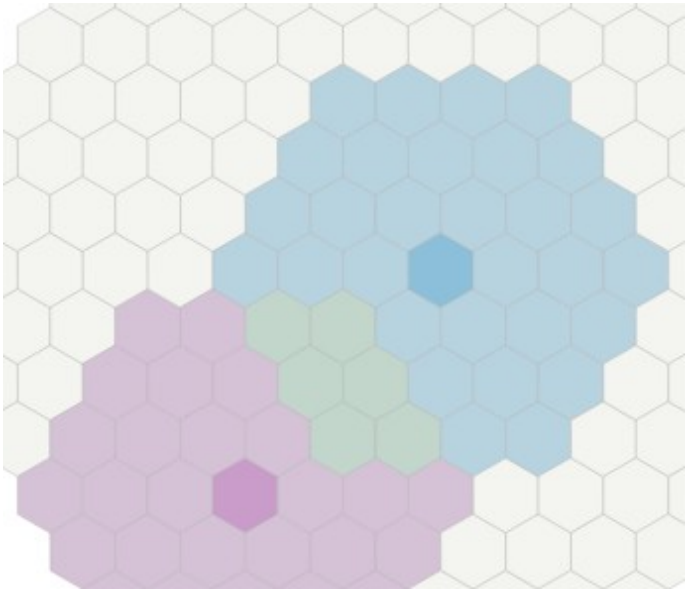

Figure 3: Intersection of two regions

If we had more than 2 towers reporting, we could do more intersections and get better at triangulating the source.

Note that:

- A distance **within** 3 cells means, the sensor actually sensed something within 3 times the horizontal space. So we are indeed **approximating a circle** with a number of hexagons.
- A distance of **exactly** 3 cells means, the sensor sensed something with 3 times the horizontal space but also not within 2 times the horizontal space. So this time we are **approximating a ring** with a number of hexagons.

At the end we could end up with either one of three results:

1. **No intersection** (means there is a false positive),
2. **One cell** (exact triangulation), or
3. **A list of more than one cell** (a region).

Questions

1. What kind of coordinate system can we use to denote the cells in a pointy top hex grid? If there are alternatives, which one provides the easiest method to compute distances, or perform intersections on ranges as depicted above?
2. Which data structure is better suited to store the entire map?
3. Which data structure is better suited to store a region defined by the sensor reading? Does it matter if the region is a circle or a ring?
4. Implement with Java the coordinate system, the map, and finding the intersection. Your program should get inputs as
    - Number of cells in map, or an indicator of its dimensions (ie. rows, columns, etc)
    - Number of cells with radar responses
    - Coordinates of cells with radar responses (repeats until the number indicated is satisfied) Then your program should output
    - The number of cells in the intersection,
    - Their coordinates.
5. Your report should also include a single test case
    - A sketch (hand made and photographed is acceptable) which marks the towers and the regions, and the intersection.
    - Screen shot of your program working, taking the inputs
    - Screen shot of your program working, showing the outputs

Submit you code and your report via Github.

ECON381 Fall 2024

Annex - Background on hex grids and distances

In order to understand the distance, you should note that the size of a hexagon in a hex grid is usually defined with respect to the radius of a circle it fits in (outer circle).

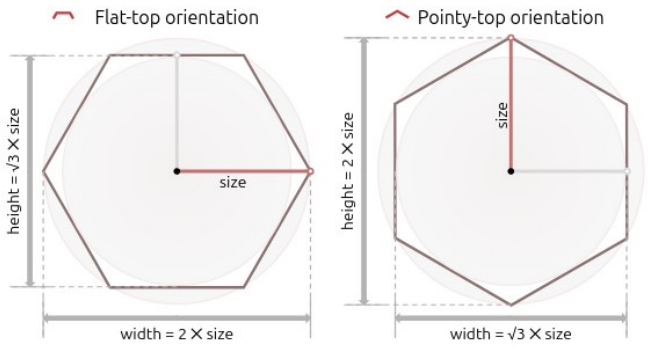

Figure 4: Size of a hexagon

And, for a pointy top orientation, the distance between the centers of two cells defines the horizontal space. For a flat top orientation, the horizontal and vertical space definitions would interchange. Also note that the horizontal space is always the same between two neighboring centers.

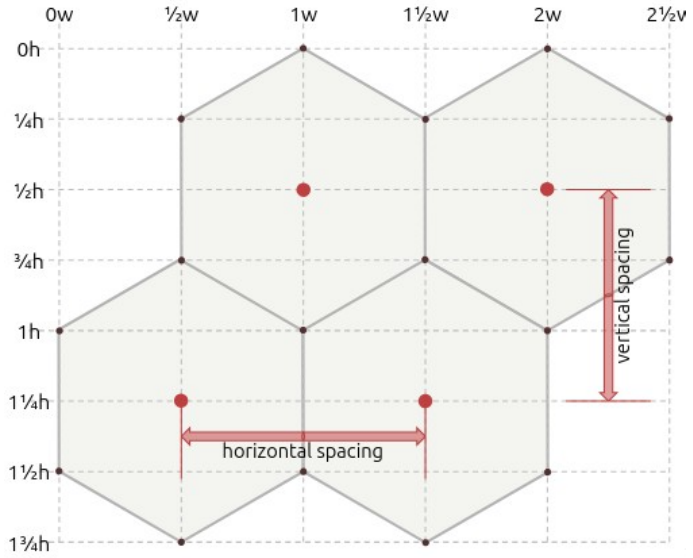

Figure 5: Horizontal space in a pointy top orientation

The figures above are from a really good tutorial on using hexagonal grids for game programming -- https://www.redblobgames.com/grids/hexagons/ You are welcome to read this tutorial, in particular the sections related to geometry, neighbors, distances, ranges, rings, and map storage. Also note that some concepts related to hexagonal grids are about presenting them in computer graphics, but you are not required to produce graphics in this semester project. So you do not need to understand everything about hexagonal grids.