CS7056 – Lab 4 – Sensing

The purpose of this lab is to learn about sensing.

As you have probably noticed, our West World agents (at least the Outlaw and Sheriff) rarely encounter each other if you run the simulation with a map of any significant size. The reason is of course that they cannot sense each other from afar; they only register each other's presence by exact co-location. The purpose of this lab is to add rudimentary sensory capabilities to the agents such that they can sense events in the game world. We will implement sight first, but your code should be sufficiently general to also support other senses.

As you solve the tasks, take notes about the decisions you made. You will need them for documenting your AI toolkit at the end of the semester.

Task 1: Add a Sense Events and Sense Handlers

Give a declaration of a Sense Event type to be added to your AI toolkit and a method declaration for a handler that would be invoked when an agent received such an event. Assume your toolkit will have support for three sensory modalities (sight, hearing, smell) but that we won't be concerned with the differences in propagation speed.

Task 2: Adding Sensory Attenuation Data to Your Map

See 'Sensing' course notes. In those notes, we assume a linear reduction in the intensity of sense events as they propagate. We also had special checks (view cone and line of sight) for some senses. In West World, we have a simple world representation (e.g., our agents don't have orientation) and can therefore simplify things a little. First, let's assume that our agents look around as they walk and that the Sight Sense therefore is omnidirectional.

Furthermore, let's assume that the different types of terrain in your game world will affect your agents' senses in different ways, i.e., with different Attenuation Values. For example, it will likely be easy to see across a river or lake, but not into a mine or building or through a mountain. For the other senses, the principles would be the same but the propagation values different. For example, sound might propagate very poorly across a rushing river, somewhat poorly in a noisy saloon but very well in a quiet graveyard.

Add a Sight Propagation value (probably of type integer) to your world representation and assign a meaningful numeric value to each type of location. Also give an outline of how more sensory modalities could be added. How many Sensory Attenuation values would you need for each location?

Task 3: Implement Sight

Since sight is linear, you can first implement sight using Unity's *Physics2D.RaycastAll* to detect whether another agent is in line of sight. With this you can then check if a specific object of interest is in sight and call a sense event.

Task 4: Other Senses

Once you have implemented sight, repurpose your existing A* implementation from Lab 3 to add another sense, such as hearing. The idea is the same as for seeing, but this time we use pathfinding instead of a raycast to detect if an object of interest is "sensable". Since Sense Events propagate according to the Attenuation Data from Task 2 (rather than the Movement Cost from Lab 2), this requires A* to operate on a Sensor Propagation Graph for each sense implemented. If you implemented the suggested grid structure from Lab 3, you can now create alternative cost grids for each sense. You will need to decide which heuristic (h-function) to use and how to invoke A* to find the relevant agents.

Add logic to generate a sense propagation graph based on the Attenuation Data described in Task 2.

Task 5: Sensory Perceptiveness

In the 'Sensing' course notes, each sensor has a threshold for Sense Events that it can receive. In West World too, we might like some agents to be more perceptive than others and we want to be able to change that perceptiveness over time. We could also have some aspects of the sensory system change of time, e.g., sight could be more costly at night.

If you repurposed your A* implementation from the Animation course, please implement the below. If you implemented A* from scratch for this project, you only need to outline (i.e., give a 1-paragraph description, but not an implementation of) the following:

1. How would you add Sensory Perceptiveness values for your agents. Would your approach allow the agents to change their Sensory Perceptiveness values dynamically? (For example, Miner Bob could reduce his values after visiting the saloon.)
2. How would you extend the approach to allow the agents to change their values based on external factors? (For example, all agents could have reduced Sensory Perceptiveness at night or during particular weather.)