

Observer Design implementation:

The way I would design my project is I first would create a Stimuli class that will represent the subject in the observer design. That class will be the parent class for all the different stimuli that this iteration will have such as light, food and whatever comes up later. This class will contain the notify function. I will also have a sensor class the all types of sensors will inherit from like the light sensor class and food sensor class. This will represent the observer object in the observer design model. The way these two class will interact is through the notify function in the stimuli class. Every stimulus will have a UpdateLocation function that constantly runs as the overall program is running and will call the notify function and that function will call the update function for all the observers for that stimuli and pass along the new POSE of the stimuli to the sensors (observers). This will allow the sensors to receive information from the stimuli which it can then pass along to the robot to influence the how the robot reacts to the stimuli.

Alternative implementation:

Another way of designing this portion of the project would be by interacting with stimulus in the same way we did with obstacles. Robots would be on the look out for the location(POSE) of stimulus. Whenever a light, for example, is near a robot then a robot would pass along the information to it sensors and each sensors would react accordingly.

Design Choice:

I will be using the Observer Design model because it insures a more reliable interaction between entities. It also works best with the design of the iteration 1 because the continuous updating of the simulation time will allow us to call the UpdateLocation everytime the function is called.