

Design Document Draft

In my observer pattern, I will try to make robots as my observer patter and set arena as the subject. Since in iteration 2, we will start to handle the case of multiple robot, so the observer pattern is necessary to be applied now.

In my implementation, arena will hold the access to all robots with an vector of robot. With providing addRobot and removeRobot function, as subject arena could add or remove robot from the observer list which enables robot to receive update information. In the arena update function, at any time if some event which would affect the state of robot as observer happens, like collision or approaching with light or food, corresponding notify function will be called inside arena so that robots as observer would receive notification and react as requirement. Since the event will be diverse, the notify function inside arena will be diverse as well. And in the implementation of notify function, corresponding react function belonging to robot will be called directly so that appropriate reaction will be activated. Since the relationship between arena and robots is a typical one to many relationship, it is naturally to set them as subject and observer which will offer flexibility for our functionality implementation and refactoring.

For the alternative approach to implement observer pattern in our project, the one I figured out is to set the sensor_touch entity inside robot to be the observer. Definitely it is not a setting as logic as setting robot as observer. If let arena as subject directly hold the access to the sensor_touch of each robot, the hierarchy of our oop design will be broken since sensor_touch could be accessed

simultaneously by robot and arena, which causes over coupling for the whole structure. That's why I choose to set robot as observer.