CSci 3081W

Hailong Zeng

12/11/2017

Peer reflection of final design document and UML diagram

In Cho's design document, I think the second part (System Overview) is a perfect sum up of this iteration. However, there is a very minor flaw of the description from my point of view. He wrote: "The sensor will affect the behavior via being passed to the Motion Handler." As far as I know, the sensors and motion handlers are independent and correlated. Each sensor will have its own class with implementations and so does the motion handler. They are correlated because both are basic components of the robot. Sensors are not only affecting the behavior of the motion handler, but a place where collects the necessary information for motion handler to react.

"Collisions are created in with the Arena's function called CheckForEntityCollision. This function takes in two entities and the EventCollision object." Cho mentioned in second paragraph in 3.1. Architectural Design. It provides details that where the collisions are taken place and what parameters are passed in, which supports the "big picture" that how the robots deal with collisions. He also made an example to show how the process works. It will make the reader understood the whole process much easier and faster.

I think Cho's design is relative effective. In Cho's design, he built the all the possible event and corresponding functions inside Arena. Thus, those functions are member functions of Arena class. Whenever a new event happens, Arena could directly call the function that corresponding to the event, which makes the system much more efficient. However, this design has a small flaw. When the events created in a big scale, all the events must be dealt in Arena, which could slow down the calculation speed. This will only happen when the events are in big scale, so it will not happen in Cho's case.

In my opinion, Cho did not have enough description about how explicit the SensorEntityType and SensorTypeEmit. The way he handler these two sensors are somehow carrying a piece of information that shows which entity created the event. If it is created by player robot, the system will just ignore this event. What I think he missed are few parts. First, he did not make a class for each sensor from my reading. Then how would he make the events know that which entity it belongs to. Secondly, even he could show this information in the event class, since the entity does not have a SensorEntityType private member, when creating the event, the system will not know who will be its relating entity.

First of all, Cho did very well in the organization of whole content. Basically, each title is detailed explained with one or multiple paragraphs. I very much appreciate his manage of separation of different information. There are introduction of the whole project, system overview, system architecture and so on, which makes a beginner much easier and faster to understood the purpose and intend of this project and what is the author's thinking and changing in the code. It could be better if he focusses more on the technical information about all the sensor classes and how the procedure for each of the sensor will be called.

If I am a programmer new to this project, I would begin with the sensor touch class and collision event. Because these two parts are most easier to manipulate from the debug view. I could be adding codes or deleting code and compile to see if it works. Other classes and events are kind of abstract and ambiguous in a way that I cannot understand just by reading the content at the first time.

The first reaction for the UML diagram is what a "big picture". It looks very neat and clean, but separates with big gaps and connects with a lot of lines. The first class I saw is player robot class. Without any doubt, this class is one of the most important classes in this project. It surrounds with many classes, such as player battery, player motion handler and robot motion behavior. It gave me a feeling of these classes are sufficient and implementable.

I would take his specification of all the relations in the diagram. He basically specified all the relations of each class. From my view, I just making connection between class with other base class. I did not make connection between class with the derived class if the base class already has connection. For example, I made connection between Arena and ArenaEntity. However, ArenaEntity has multiple derived classes. I only made the connection between Arena and ArenaEntity in a sense that Arena is also connected with ArenaEntity’s derived classes.

I think Cho’s most successful part writing to guide the new programmer to the project is, again, the organization of whole content. He separates different information in different topic, and each topic has few subtitles, which makes the whole cumulative information in order. The new programmer to the project could understand the project piecewise. It could be more helpful if he could explain each small piece clearer.