Peer Response

Project Author: Lei Huang

Da Song

5330740

1. In the first paragraph (or first section) of your design documentation, what is the most effective sentence with respect to orienting the reader to the project?

Answer: “This project is a robot simulation program acting like a video game. There are many different components (player, robot, superbot, homebase, recharge station, obstacle) in a visible graphic window. Player, robot, superbot and homebase are moving in this visible field while other objects keep stationary. ”

2. Identify a sentence in the first paragraph that needs to be reworked and state what you think is problematic about that sentence. (Do not edit it.)

Answer: “The previous version of robot turns into the player in this iteration that is controlled by the user. ” This sentence is not so clear while it wanted to express the relationship between old version robot and new version player. I would say “The new version player can be controlled by user, which is more likely to the old version robot.”

3. Identify a sentence or two in any of the paragraphs that provides the "big picture" with respect to the software, design, or class structure, AND is accompanied by low-level details that help the reader better understand the "big picture."

Answer: “This viewer class controls everything in general including creating the graphic window and the GUI, visiting the Arena object to update the time step, visiting the Arena object to draw each component respectively on the graphic window and accepting the keyboard, mouse command to manipulate the components in Arena object. In this iteration, we need to refactor the code and use observer pattern to implement the relationship between arena (subject) and robots (observer). My choice is to make robot to be the observer in this pattern instead of the sensor because it will be easier and more reasonable to update them directly in the arena. ”

4. Comment on the effectiveness of this technique in the example from (3). If it is effective, analyze why you think it works here. If you think there are other details that would be more elucidating, state those.

Answer: I think it is effective because Lei used only a short sentence illustrated the design pattern (observer pattern). Plus, he said the reason why he would not like to choose sensors to be the observer. Actually I met the same problem with him. I spent lots of time to decide using robots to be the observer. And he explained the graphic viewer of robot simulator clearly. Readers can understand some interface and necessary software in this project. Thus I think it is effective.

5. Identify a topic in the writing that is either underspecified or is discussed too in-depth. If underspecified, what is the most important idea that is missing? If too in-depth, what can be removed?

Answer: I think Topic 5 is too in-depth. He already gave us enough information in Topic 3. What in topic 5 told readers looked for code to find some specific algorithms. However, I think gave readers a general description is a better idea.

6. What do you think would be the single most impactful change to this document - in other words, what would you recommend to the author as the one area on which to focus? It could be related to the content (e.g. level of detail, more or less technical information, highlight more or fewer classes, etc.) or to the writing (e.g. reorganize paragraph or sentence order, condense text, improve sentence structure, etc.).

Answer: From my perspective, I think Lei needs to focus on the structure of the whole documentation. Firstly, he wrote some redundant topics. For example, he describes observer pattern twice. One is in Topic 3.1.2 and another one is in Topic 3.3.2. Secondly, some content which described similar things should be together but separated in his documentation. For example, the converted of robot and superbot should be put in Topic 3.2.4 but separated in Topic 3.2.4 and 3.3.1. Fianlly, I think Lei needs to pay attention on the splitting his paragraph. A giant paragraph is a little hard to read and understand.

7. As a programmer new to this project, which class do you think the document is emphasizing as the place to begin to engage the code? This might be explicit or implicit. What part of the writing made you think you should start with that class?

Answer: I think arena_entity is a good place to start engaging the code. In the System Overview, Lei said “This project is a robot simulation program acting like a video game. There are many different components (player, robot, superbot, homebase, recharge station, obstacle) in a visible graphic window. Player, robot, superbot and homebase are moving in this visible field while other objects keep stationary. ” I consider that the most important thing at the beginning point is to know the element in the robot simulator and understanding how to play this simulator. Thus I think I should start with that class.

8. What do you consider to be the best and worst documented method in that class and why. OR, if you think they are all of equal quality, comment on the level of detail provided in the documentation. Is it sufficient, clear, and correct? If it is excellent, state what makes it excellent.

Answer:The best is the constructor of ArenaEntity. The documentation described each argument specifically and clearly. Readers can easily go next step. The worst documentation is some getters and setters documentation. It is not clearly enough to understand why we need to get or set these variables.

9.  Skim through all the brief comments on the main classes page design document. What strikes you as you look at the collection? Is there an effective pattern in the comments? Is there something consistently lacking?

Answer: Those event classes comments are logical and specified. Readers can understand how the event work after reading the comments. But I think arena.h file lacking some comments about the important functions.

10. Where did your eye go? What jumps out at you on the page? Is this an important element, thus warrants the attention? If not, offer a suggestion on how to make it less visually prominent.

Answer: I first saw Arena class. I think this is an important element because arena contains the registerObserver, removeObserver and notifyObserver methods which shows the observer patter applied in this project. And arena class can update all events and let entities accept the event.

11.  What did the author do in her/his UML diagram that you would like to incorporate into your UML? Why do you like that part of the UML and how does it differ from what you did?

Answer: I like the Robot class’s position. Lei put it at the bottom of page. Robot contains a lot of compositions, which means we need draw so many lines on it. Put it to bottom means he had large space to draw those lines and avoid some interaction between lines. I put robot class at the center of page and gained lots of pain while I draw the lines.

12. Try to recall your sense of your first attempt to engage the base code, and think of how it is even more complex now. Keeping that in mind, what do you think was the most successful part of the author's writing (in doxygen and UML) with respect to helping a programmer get acclimated to the code? What do you think could be very helpful but needs some rework?

Answer: The most successful part is his documentation. He wrote a clear description about the events, sensors and the logic of for loop in arena class. Although there are some redundant stuff, readers can understand the project after reading his documentation. As far as he achieved this, I think this is a successful documentation. The comment of some main classes should be clearly but not. Those comments can give a intuitive feeling for the project. It could be helpful but not in Lei’s code. I think it needs some rework.