Peer Reflection of Doxygen

Hanyu

Author: Xicun Shi

1. The first two sentence is the most effective with respect to orienting the reader to the project. It said “This is RobotSim v2.0, Ii is amini video game allowing the player to control a robot in an arena.” this sentence shows the method of whole functionality of this project, the general idea of how this project works. “Allow the player to control a robot in an arena”. So the method is just control the player, and maybe do whatever you want to. And following sentence is about the win and loose condition introduction. “ freeze all robot” is what users are suppose to do. So this two sentences I think is the most efficient sentence to start the mainpage.
2. In the first paragraph, there’s using many words like “ be careful” etc. I think what we need to do is just tell them what will happen when you make some actions, but not try to introducing the gaming rules, so that means we are more focus on the project functionality instead of focusing on the how to play this game.
3. In the second paragraph, it shows the purpose of this project is to control the player to freeze all of the robots that move in the arena” that shows everything will be done in the arena. Which state that the arena is the main place where all the event will take place. For me that will look more like a “big picture”. Since we actually using almost everything in the arena.
4. I think it is slightly ineffectiveness, since paragraph one and paragraph two are seems similar, then if is unneccessary to like re-explain what’s going on in the arena, and win lose condition. Just start to write entities and events in the arena will much better, and explain them with the general ideas. So the need to fill this paragraph with events and entities classes.
5. The topic is “clean code”, I think this is a underspecified, so I think for the “clean” code, the documentation is really important, when reading the code, the most comfortable things is you can clearly understanding what’s going on with the logic. You will know what is each function did even just take glance, when you look at the function name, it will be clear what’s the functionality even there’s no implement could be known in the mainpage.
6. Bug report need to be added into the main page, which will be one of the most important detail for the code, so that also means need to focus more on the details. Also need to care about the organization, the class need to be listed together to specify what is for what, since once start to explain the code, we will always mess the class together to explain.
7. I will recommend start with the main.h, and arena.h, main a the heart of the whole code, it implement everything for the software, and arena.h is like ‘virtual place’ that every events are working on there, so it will include most heading files. Those are all entities and event, once you start checking the entities or the event, automatically will need to look into those handler and behaviors header file, and gradually, will be able to check all files in our project.
8. The best is that even those classes are not able to get all ideas from the main page, but we can still by look into each single class through clicking them in the main page. For example every entities are implemented really well, but just in the main page, you can just know what it is, but you do not know what is the parameters for those class. So you can look into that for example, you can look into the “rechargestation” which will list all the parameters, like radius and pos, and color etc. but there’s some class did not using doxygen format, so they can not be recognized when compiling with doxygen. And also for some classes like “player.h” need to change some “copy and paste” stuff back to the name we want. In the parameters of player.h it said “params A robot\_params passes down from…” but what we want here is palyer\_params.
9. Did not show the bugs, since 2.0 is not the last version, need to indicate what is not done yet, the pattern is kind clean enough, the all classes’ linking and explanations are did good, it lists all the entities and sensors, but actually could make it better by reorganize, so no full credits for the effective pattern. The first two paragraph could be merged into one. And the “big picture” is showing well, I can know where I need to pay attention to. The explanation also clear and easy to understand.
10. The title catch my eyes first, but nothing really special. But the whole graph is really beautiful and clean, at least better than mine, and the note is clear too, you can easy find where’s the highlight comment is, and where is connected to. All arrows are using clear, you can know the relationship to all classes.
11. The organization! Also the comments and the arrows using are all better than mine, his uml could fit into one page, but my UML must need to zoom in for clear definition of one class.
12. First the beautiful UNL definitely the best tool to make good understand of class, then the doxygen also has a good general idea of project, from very top of this project and then slowly get into the details will help the programmer to understand better. So in my idea, the graph, is necessary and useful, then a clear explanation will be very helpful to know what’s going on the project 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?  
      
    .