

Peer Reflection

RESPONDER: Carlos Alvarenga

AUTHOR: Jeremy Herzog

1. The following opening sentence is the most effective one with respect to orienting the reader to the project: “The code concerning the robot project centers on 3 areas: Mobile Entities, the arena class, and the graphics viewer.”
2. The second sentence which reads, “In summary, the robot class contains components to help control the robot” may need reworking because it’s seems like vague statement that may even seem obvious because each class is comprised of different components that provide functionality. Elaborating on this statement from the get-go by at least listing the major components of the robot may provide more helpful insight that properly introduces the reader to the overall structure and design of the robot class.
3. A couple of sentences that provide a “big picture” while discussing low-lever details are the following: “It includes the same sensors/accept methods that the robot does. However, the main difference is in the timestep update method. If the player collides with the a robot that is a super bot, it increments a state variable that keeps the player from moving for 10 timesteps.”
4. The sentences are effective in clearly illustrating the main difference between the player and robot classes by specifically noting the timestep method and the manipulation of the state variable that controls the player’s movement within such method. This successfully communicates to the reader how the robot and player differ and how such differences were implemented within the code.

5. One main idea that can be discussed more in-depth is the superbots entity because it was a major component of the project introduced in iteration 2. It's only mentioned 3 times and it's never explicitly described how a superbots differs from a robot or the important role it plays in the simulation. Elaborating on this can make the reader more aware of an important populating arena entity.
6. I think highlighting and discussing more arena entities like superbots and homebase in a more detailed manner may help the audience gain a better understating of the different components at play in the simulation. Elaborating on such components could also provide a more detailed, complete view of simulation that has many vital moving parts that are integral to the functionality of the software.
7. The robot class is emphasized as a starting point towards engaging the code since it's the first class discussed in depth through multiple bullet-points. It's also the class of which there's the most information provided under its respective section in the document.
8. All the methods are of equal quality in that they're sufficiently documented, clear to the point and correct. Each method has a precise description that describes its role relative to overall framework and functionality of the robot class. Moreover, the parameter for each method is accurately listed to convey complete information of the method at hand.
9. Nearly all the classes have their function within the simulation properly described. Their descriptions always highlight their purpose in relation to the simulation, in general, and/or another particular class. However, the Obstacle class is the only major class that lacks such description.
10. The clean organization, flow and groupings of classes is what stood out to me the most. I could clearly identify the different sections in the UML that group different major aspects

of the simulation. This is important in that it immediately conveys information to the reader in a coherent manner so that he/she can clearly follow the software design.

11. Although I tried to structure my UML into groups of classes to make it more readable, I liked how the author clearly distinguished different components of simulation by grouping them together and labeling such component as “Events”, “Robot Parts” and “Entities”. This labeling is something that I didn’t incorporate in my UML, but find satisfying because it effectively organizes different component of the project to clearly convey how they interact with each other to make the simulation possible.
12. The most successful part was grouping and distinguishing various parts of the simulation like entities, sensors and events. This grouping helps the reader gain a better general understating of the simulation by introducing them to the primary moving parts of the code. Since there is a lot of code to delve into, breaking it up into categories may ease a programmer into working with the code since he/she knows the purpose of the code in respect to the overall simulation and, in a general sense, how different major parts of the project and code interact with one another to accomplish a successful simulation. This decision to break up components of the project into various categories could become more helpful by adding more detailed information to said categories. Specifically, entities besides robot like superbots, homebase and even player could be described in greater detail so that they’re properly distinguished from one another. Having more concrete information about these entities could be helpful in preparing the programmer towards dealing with code and simulation concepts/objectives that vary between the various arena entities.