

*** Disclaimer: Items marked with two asterisks may not actually be bugs because either they meet the requirements outlined in class or because the implementation of the stated feature was ambiguous. They are listed here for the sake of completeness, since unreported bugs are punished more heavily. ***

IMPORTANT: *To run the program, compile and Run ./build/bin/arenaviewer ./params.txt. The program will not work if you just call ./build/bin/arenaviewer with no param file argument (I believe this is expected behavior).*

Most of the bugs I couldn't get rid of are associated with the proximity sensor.

1. There is definitely some sort of avoidance behavior in all the Robots, but sometimes they get caught up on each other and end up colliding with things. **
2. Every once in awhile, it looks like a Robot isn't avoiding obstacles at all. Rebooting the program seems to fix this whenever it occurs.
3. It's very tough to test whether Robots are not avoiding frozen bots. I believe this functionality works correctly but I can't tell because sometimes they turn around; this could be due to another nearby unfrozen robot, since the avoidance behavior is somewhat wonky. **
4. SuperBots don't avoid the player, but that's because they don't avoid anything. This stems from some ambiguity in the requirements: entity type sensors can only hold one output at a time, but they may detect many objects every frame. Either Player needs to override all other entities detected by the type sensor, or SuperBots needs to get that information from their proximity sensor (breaking encapsulation). For this reason, I thought a more clear solution was to have them avoid nothing; after all, they're SuperBots. **
5. Robots don't avoid walls, but this is intentional. I believe we were told in class that this was ok. **

Other miscellaneous bugs include:

1. Player resets to a slightly different heading angle than the initial heading angle. Again, this seems like a small problem but it turns out to be quite complicated.
2. Recharge station doesn't reset battery to 100%, because the collision itself removes 10% of the battery. I tried pretty hard to fix this but I can't figure out why it's happening. Ultimately, this could almost be considered a game mechanic for added challenge (but of course it's a bug so it's listed here).

3. The angle of the text on the Player does not correspond to the actual heading angle. This does not affect the mechanics of the game, but it looks kind of strange. **
4. This isn't technically a bug, but cpplint complains about 3 errors. One of them is the infamous rand errors which we were instructed not to fix. The other two are actually the same error: cpplint is complaining about the arguments to a function being a non-constant reference. When I tried to fix this I got a segfault so I had to leave it as is. **

Finally, there are some bugs that are very rare (so rare I'm not even sure they still exist), but should be mentioned just in case:

1. This is an extremely rare bug, and I'm not even sure it still exists, but just to be safe I'll mention it here. During my testing, sometimes when HomeBase collided with a Robot (and thus turned it into a SuperBot), a second Robot in a different part of the Arena would turn into a SuperBot too.
2. Every once in a great while a Robot may do a slow "jitter" on the wall, but it's extremely hard to reproduce (users are unlikely to encounter this).

Ultimately, while this bug list contains a lot of items, you can tell just by playing around with the program that it's not super buggy (especially compared to Iteration2). I'm just trying to be thorough.