# Bug Report

- The main bug is that a segmentation fault may randomly occur whenever one decides to play the game. I'm not sure about the origin of the bug, but I think that it has something to do with the dynamic casting that I'm doing in arena when checking and sendings events. This may be solved by using the entity type sensor to appropriately identify types of the entities being sensed. As of right now, I'm just using getters for entities to return their respective type ids instead of utilizing the entity type sensor. This could be improved by making full use of the entity type sensor and events to appropriately distinguish between different entities.
- At times, a mobile entity may get stuck at the wall. I've noticed that it occurs most frequently in the bottom and top walls and this may be caused by how I'm handling the bouncing-off aspect of robots when they detect an entity.
- My robots are accurately frozen by the player but they don't always decrease their speed when their proximity sensor detects another entity. I think this is since I'm using one sole motion handler for both robots and superbots. One way that I can try fixing this is by creating two separate motion handlers for robots and superbots to handle their respective speeds and heading angles accurately. This method seems better than what I'm doing right now, which is setting a flag in robot whenever it collides with homebase to implement a different functionality within a motion handler that's used by the robots when it's a regular robot and superbot.
- Whenever a robot is frozen, its heading angle may still change depending on whether or not it's proximity sensor detects another entity within range. I'm not sure if this is an issue that needs to be addressed because it's not hindering the functionality of the program but addressing it may lead to a more robust program.
- There are still times when a robot doesn't accurately or completely avoid an obstacle (most likely another robot) within its range. This is probably due to how I'm handling the distress and proximity events because the robot won't avoid an entity that has its distress sensor signal emitting.
- Whenever the player freezes all the robots, the game may not immediately restart until you freeze another robot. Also, if the game does restart successfully, then the last robot that was frozen may stay frozen at the start of the next game and not resume to its normal speed.