BETA DEMO ANALYSIS

Capture the Flag Autonomous Robot

**Project:** ECSE 211 DPM ‘Capture the Flag’ Robot

**Document Version Number:** 1.0

**Date:** 2015/11/26

**Document Author:** Asher Wright

**TABLE OF CONTENTS**

Overview: 3

Software Status Overview: 3

Expectations: 4

Setup: 4

Performance: 5

Consequential Improvements: 5

Conclusion: 6

# Overview:

The beta demo entailed a subset of the competition on a reduced grid of 8’ x 8’. The demo started once a message was received from the WiFi server.

After the message is received, the objective is for the robot to:

1. Localize to the grid
2. Navigate to the “Red Zone”
3. Avoid collisions with the wooden blocks
4. Search for the indicated flag
5. Grasp the flag

Grasping the flag marked the successful completion of the beta demo.

# Software Status Overview:

Below is a summary of each software task that was considered up until the beta demo. There is also a brief qualitative analysis of how well these tasks were working:

1. **Ultrasonic Localization**

USL was working for every initial orientation. The robot localized and traveled to an appropriate light localization position and orientation every time.

1. **Light Localization**

LSL was working without problems (within 1° orientation and 0.5cm position).

1. **Navigation**

Navigation was working well, but without odometry correction. It traveled to the correct location, but was off by ±10cm upon arriving. This was not a huge concern for the beta demo, since it still got the robot to the block zone (roughly).

1. **Obstacle Avoidance**

The obstacle avoidance was working very well. The only issue was, since the obstacle avoidance caused some slipping, and there was no odometry correction, there was some more positional error.

1. **Flag Searching**

Flag searching worked well, given a good starting position. The robot had some troubles finding the flag if there were many blocks (more than 3) in the zone.

1. **Flag Detection**

Flag detection was actually incorrectly coded at this point, and it always assumed that, if it recognized the block, it found the flag.

1. **Odometry Correction**

Odometry correction was not implemented for the beta demo.

1. **Wifi Communication**

Wifi communication was working without any problems. All of the information could be passed in through the server successfully.

# Expectations:

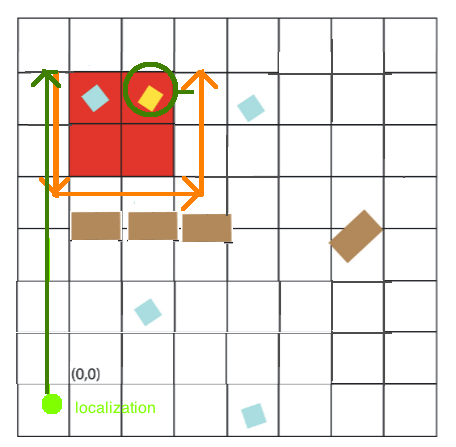
We expected the robot to perform quite well at the beta demo.

The main potential issue was the lack of odometry correction. Without correction, the robot had some positional error upon reaching block-zone. However, this was found to not be that much of an issue. This is because the block-searcher would still be able to search for the blocks, even if the position was off a bit (+-10cm). Thus, the team allocated time to other tasks, rather than implementing odometry correction.

# Setup:

We were one of the last groups to perform at the beta-demo test, and as such, were only given one try (as opposed to two).

The arena was setup as shown:



**This setup was unexpected.** One can see our robot’s expected navigation path (green line), as well as its expected searching path (orange line). First of all, the path has no object avoidance, which was unfortunate as it did not allow us to demonstrate its ability to avoid objects. More problematic, however, was how close the navigation path was to the wall (this will be discussed in the next section).

# Performance:

The robot did not perform as well as we had hoped. We assumed that the robot’s small position error after traveling to the block-zone was negligible. This was not valid for the given setup (and many other setups that should have been considered).

In this setup, the robot was to travel right next to the wall. The robot was trying to travel 10cm to the left of the block-zone (see algorithms document). In this setup, that was dangerously close to the wall. This means that the error of +-10cm was relevant, as it meant our robot ran right into the wall.

Analyzing any part of the test after the robot ran into the wall is tough to do, since much of it was dependent on not running into the wall. However, the block searcher appeared to work (although obviously it did not search the right section). Additionally, the block investigator worked (what approached the blocks). However, the block detector did not work. It picked up the dark blue block, even though it was supposed to pick up the red block. We later found out that this was due to a bug in the code that caused the robot to pick up any block it recognized. This bug was caused when we added the Wifi block parameters.

# Consequential Improvements:

The first obvious improvement was to fix the bug resulting in picking up the wrong blocks. We did this by ensuring that the robot only picks up a block if it recognizes it AND if the block is the correct colour.

The next improvements were also pretty clear. We clearly needed some sort of odometry correction. We needed its position to be especially accurate right before we were searching the block, if we wanted it to travel in a perfect rectangle. Otherwise, it would collide with the blocks (which is not a problem for the beta test, but would be a problem for the final competition).

We added odometry correction so that it’s position was always pretty good, even if there was some slippage.

We then added a “relocalization” method to the light-sensor localization class. This allows the robot to perform light-localization at any corner of the tile (any intersection of two lines). It then updates the robot’s position and orientation accurately (1deg, 0.5cm). This is time consuming, and we decided to do this just once, before we search the block area.

# Conclusion:

The beta demo, although not totally successful, resulted in many positive changes to our robot’s design. These changes make for a more predictable robot that can perform better under more conditions.

Although our robot design would have worked much better in a different beta demo situation (with the block-zone not next to the wall), it was helpful to see a situation in which it did not perform at all as expected.

The improvements will be tested thoroughly both by themselves and integrated into the entire system.