# Week 3 Documentation

# Project Group#08

**Index**

1. Final Design Mechanism
2. Attacker/Defender Strategy
3. Testing Plan (included separately)
4. Increasing Component Efficiency
5. Physical and Dynamic Errors
6. System Model Summary

**Final Design Mechanism**

The mechanical design consists of the following:

1. 2 light sensors: One for localization and one to locate the flag
2. 3 ultrasonic sensors for wall detection
3. 4 motors: 2 motors for the wheels and 2 motors to move the mechanical arm
4. 2 NXT bricks for controlling the robot components

The design relies on the movement of a mechanical arm, comprising of pivot points, and controlled by 2 motors. The mechanical arm is able to extend forward, grab an object, and then elevate it to more than 20cm. One brick will control the mechanical arm while the other brick controls the wheels. The arm is comprised of three parts. One part, comprising of claws, will be used to grab the object. One will be used to provide a pivot for the base and the other part of the arm. The second part of our design comprises of the second brick and the movement components (wheels and apparatus). 2 motors will be used to rotate the wheels. A small wheel at the back rotates on its own and provides stability for the whole structure.

At the base of our design is the NXT brick that will be used to control the arm and some of the sensors. A total of 2 light sensors and 3 ultrasonic sensors are being used. One light sensor will be placed at the back, for the purpose of localization. The other light sensor will be mounted at the front of the robot, to detect the light source. The three ultrasonic sensors will be used for wall following and obstacle avoidance. One ultrasonic sensor is mounted at the front, with the other 2 mounted at the sides.

**Attacker/Defender Strategy**

The competition will be between 2 teams. Each team will be assigned attacker/ defender roles.

Our strategy for the attacker has been defined. We already have coordinates of light source for attacker. We only need to find shortest path (as we did for navigation), while avoiding the obstacles. If no obstacle is present, the robot will calculate shortest path. If an obstacle is present in the robot’s path, the robot will navigate around the obstacle before continuing its movement towards the object. For the attacker, there are 2 possibilities to consider. Either the beacon has been placed on top of the obstacle or it is lying on the ground. The attacker will start with the corners, gradually closing in towards the centre (in a rectangular path), until it detects the flag. The robot will then return to its original corner.

Our defender will first moves towards the light source grab the beacon and hide it. As the defender, our robot will be aware of the coordinates of the place where the beacon is initially located. The defender will move towards the light source, grab the flag and then hide it in front of an obstacle, close to the obstacle wall. The other possibility would be for the defender to place the beacon on top of an obstacle. With regards to the defender’s navigation, once it has the beacon, it will go to farthest point, find an obstacle and drop the object there.

We are still in the process of finalizing our strategy and refining the key points.

**Increasing Component Efficiency**

Component efficiency can be increased through a series of steps. Increasing distance between motors and wheel centre will reduce drag between wheels and the motor. The ability to rotate the ultrasonic sensors will greatly augment their detection and wall following ability. Placing the light sensor on top of the claw, will permit an elevated scan of possible light sources, thereby improving the line of sight.

**Possible Physical And Dynamic Errors**

Errors occur, particularly when the ultrasonic sensor comes within close proximity of the walls. These errors are indicated by a distance of 255 units on the NXT display. With regards to physical errors like wheel slip, an error in the orientation originating from a one off wheel slip will grow linearly as the robot travels further. A linearly growing error such as this will sum up with other linearly growing errors. If all the errors were to grow linearly then the sum of the errors (total error) would also grow linearly. The errors present also originate from both the navigator and the odometer. The inaccuracy in the odometry comes from the errors which the odometry is unable to detect, such as contact friction with the floor and wheel radius asymmetries. Therefore, when the odometry deviates from the correct coordinates and bearing, the error filters through to the navigator, causing it to calculate the wrong angle of bearing.

**System Model Summary**

