# 1. Introduction

For this project, many different modules are combined to create a functional robot that can perform the desired routine as per the specifications; therefore a bottom-up strategy will be employed to help target the source of the error, which is especially important when the robot is running multiple threads at the same time. The test cases designed will try to minimize the chances of an error happening on D-Day, but also try to make sure it would not cause the testers to overspend its time budget on it.

# 2. Test Items

The following items will be the focus of the robot:

* Basic cases
  + Odometry
  + Navigation
  + Bluetooth
  + Ball launcher
  + Localization
* Intermediate cases
  + Obstacle avoidance (ultrasonic)
  + Obstacle avoidance (touch sensor)
* Advanced cases
  + Defender role
  + Attacker role

# 3. Test cases

## Basic cases:

### Odometry

|  |  |
| --- | --- |
| Test case ID | 1.1.1 |
| Case name | Rotate |
| Purpose | To determine if the robot can determine where it is pointing at after rotation is performed |
| Description | The robot will turn to various angles and see if it can turn back to its origin |
| Prerequisites |  |
| Steps | 1. Make sure the robot is pointed to the 0 degree for this test 2. Turn to 60 3. Turn to 180 4. Turn to 350 5. Turn to 80 6. Turn to 0 |
| Actual results |  |
| Expected results | Robot can point back to 0 degree after rotation routine is completed. |