**DKIT Fruit Salad Design Doc by Sarah, Daniel and Minh Thu Le.**

***General Description:***

To preserve and try not to waste power, our tank will run away from the sentry bot at the start of any game and it will have random movement with the relative fire power to the distance between the robots.

***Implementation***

**1.Position Set up:**

At the very beginning of the game, the bot will turn back and run away from Sentry, but just to the angle of 160 degree so that it won’t stay on the same line for Sentry to aim and shoot. Our turning angle from current heading is the difference between 160degree and the current heading: turnAngle = 160-getHeading() The sign of this result of subtraction will decide whether the robot will turn left or right.

if (e.isSentryRobot() && !sentryInit) {

sentryInit = true;

if (e.getBearing() > 0) {

turnLeft(160 - e.getBearing());

} else {

turnRight(160 - Math.abs(e.getBearing())); // avoid 180, flip the heading totally

}

ahead(400);

} //

**2.Random Movement:**

We randomised our movements by using the Math.random() function. Which generated a random number between a range and stored it as a variable so it could give us a random turning angle and we used another random number generator to decide whether to go right or left. We placed this inside of our while loop so new numbers would be generated every time the loop runs.

RandomAngle = (int) Math.floor(Math.random() \* Range + MIN);

LeftOrRight = (int) Math.floor(Math.random() \* 2 + 1);

**3.Firing Bullets:**

We decided to fire different bullets depending on how close you are to the other robot. The distances were decided through testing the robot over and over again we also added a scan at the end to check if the tank is still there and if it is we can then fire again.

if (!e.isSentryRobot()) {

if (e.getDistance() <= 60) {

fire(3);

fire(3);

} else if (e.getDistance() <= 150) {

fire(3);

} else if (e.getDistance() <= 400) {

fire(2);

} else {

fire(1);

}

scan();

}