Credit Name: Project D
Assignment Name: Phidget

REFLECTION ON CODE

How has your program changed from planning to coding to now? Please Explain?

The Phidget rover kit was pretty challenging for me because I needed to program the code that can easily output in the rover. The rover challenge was extremely challenging because the code was not outputting my code perfectly. It took some examples and testing to perfect it. The only problem I struggled with was the chances I had for the rover to work every time(perfected the first time, other tries were not perfected).

```java
package phidgets;

import com.phidget22.*;

public class RoverSquareChallenge
{
    public static void main(String[] args) throws Exception
    {
        //Connect to wireless rover
        Net.addServer("", "192.168.100.1", 5661, "", 0);

        //Create objects
        DCMotor leftMotors = new DCMotor();
        DCMotor rightMotors = new DCMotor();
        DistanceSensor sonar = new DistanceSensor();

        //Address
        leftMotors.setChannel(0);
        rightMotors.setChannel(1);

        //Open connections
        leftMotors.open(5000);
        rightMotors.open(5000);
        sonar.open(5000);

        //Set acceleration
        leftMotors.setAcceleration(leftMotors.getMaxAcceleration());
        rightMotors.setAcceleration(rightMotors.getMaxAcceleration());

        //Move in a square pattern 4 times
        for (int side = 0; side < 4; side++)
        {
            //Move forward
            leftMotors.setTargetVelocity(0.5);
            rightMotors.setTargetVelocity(0.5);
            Thread.sleep(2500);

            //Stop
            leftMotors.setTargetVelocity(0);
            rightMotors.setTargetVelocity(0);
            Thread.sleep(500);

            //Turn 90 degrees - adjusted timing for precise turn
            leftMotors.setTargetVelocity(0.5);
            rightMotors.setTargetVelocity(-0.5);
            Thread.sleep(650); // Reduced from 825ms to 650ms for exact 90-degree turn

            //Stop
            leftMotors.setTargetVelocity(0);
            rightMotors.setTargetVelocity(0);
            Thread.sleep(500);
        }

        //Final stop
        leftMotors.setTargetVelocity(0);
        rightMotors.setTargetVelocity(0);
    }
}
```