

Problems and Solutions

Centring while moving through the maze

Eventually we need to expect our robot to come off course due to in-precision of motors, setup of the 90-degree-turn and other factors like friction effects.

Problem

To centre the robot we can use the IR and US sensors mounted to the motor. For that, we need to determine the offset of the robot to the centre of the squares of the maze it is currently in.

Solution proposal

1. Horizontal Centring

First we need to precisely measure the relative distance to a wall.

```
final int NUMBER_OF_MEASURES = 5;
final int GRIDSIZE = 40;

public float measureRelativeWallDistance()
{
    float preciseDistance = 0;
    for (i = 0; i < NUMBER_OF_MEASURES; i++)
    {
        float USValue = measureUS();
        if (USValue >= 60)
        {
            USValue mod GRIDSIZE;
            preciseDistance += USValue;
        }
        if (IRValue < 60)
        {
            float IRValue = measureIR();
            IRValue mod GRIDSIZE;
            preciseDistance += IRValue;
        }
        // Using mod enables us to centre even if walls are not right next to us
        // Will be less precise with increasing distance nevertheless
    }
    return preciseDistance / NUMBER_OF_MEASURES;
}
```

Then we can use the method easily to centre the robot.

```
float offset = 0;
rotationMotor.rotateTo(90); // Look to one side of the robot
float distanceL = measureRelativeWallDistance();
rotationMotor.rotateTo(-90); // Look to the other side
float distanceR = measureRelativeWallDistance();
offset = (distanceR - distanceL) / 2;
// if relevant enough to bother centring the robot
if (offset > 3) // positive = adjust to the right
{
    pilot.rotate(-90);
    pilot.travel(offset);
    pilot.rotate(90);
}
if (offset < -3) // negative = adjust to the left
{
    pilot.rotate(90);
    pilot.travel(offset);
    pilot.rotate(-90);
}
```

2. Vertical Centring

The exact same thing but no turning of the robot involved and moving sensors to 0 and 180 instead of 90 and -90.