**IMPORTANT**:

* A precondition for all these movement functions is that a function has been implemented to handle wall collisions.
* Each robot would have its velocity set during initialization.

**Still()**

Narrative: Robot doesn’t move. Not even sure we will need a function call.

Pre-conditions: None

Post-conditions: Robot hasn’t moved from its spawn location.

**BulletAvoid(minDist, maxDist, rotateDir, moveDir, HitByBulletEvent)**

Narrative: Robot only moves when hit by a bullet, functions similar to the “Fire” robot in sample.

Pre-conditions: rotateDir & moveDir are either +/- 1. HitbyBulletEvent object is still defined. This is called when hit by a bullet.

Post-conditions: Robot moves away perpendicular to the bullet.

Rotate to be perpendicular to the direction the bullet came from

//which way it rotates, clockwise vs. counterclockwise, is determined by rotateDir

Distance = random value between min- and maxDistance \* moveDir

Flip “directions” of rotateDir and moveDir

Move distance

Examine surroundings

**Random(minDistance, maxDistance, minRotation, maxRotation, moveDelay)**

Narrative: Randomly moves the robot about the map.

Pre-condition: Potentially none (depends on how moveDelay is implemented).

Post-condition: Robot moves to a new location.

If not moving

Delay by a random value between 0 and moveDelay

//Not sure if we would want to use a java function for this or look at one of Robocode’s

// methods (i.e. waitFor).

Distance = random value between min- and maxDistance

Distance = distance \* Randomly selected direction (forward, +, or backward, -)

Move distance

Rotation = random value between min- and maxRotation

Rotate by rotation

**Linear(minDistance, maxDistance, moveDir)**

Narrative: Moves the robot back and forth.

Pre-condition: None

Post-condition: The robot moves.

If not moving

Distance = random # between min- and maxDistance \* moveDir

//It might be better to have the distance set during initialization and use the same

// amount each time.

Move distance

moveDir = moveDir \* -1

**RandomLinear(minDistance, maxDistance, moveDir, moveDelay)**

Narrative: The robot moves back and forth at random intervals.

Pre-conditions: None

Post-condition: Robot moves

If not moving

Delay by a random value between 0 and moveDelay

//Not sure if we would want to use a java function for this or look at one of Robocode’s

// methods (i.e. waitFor).

Distance = random value between min- and maxDistance

Distance = distance \* Randomly selected direction (forward, +, or backward, -)

Move distance

**Circular(minDistance, maxDistance, minRotation, maxRotation)**

Narrative: Robot moves around in a circular path.

Pre-condition: May want to assign rotation about during initialization rather than calc each time movement is called.

Post-condition: Robot moves.