**IMPORTANT**:

**Total bullet strategy behaviors**: 5

1. DistanceBased – adjusts bullet power based on how far the target robot is.
2. LightFast – focuses on low-power, quick shots.
3. PowerfulSlow – focuses on high-power, slow shots.
4. Medium – Average power, and therefore speed shots.
5. HitRateBased – adjust the power based on hit rate.

**Additional genetic parameters**:

1. farDistance – the cutoff for high-power vs. low-power shots. Used in distance-based calculations. Should be between 0.0 and 1.0 as it will be a percentage of the farthest possible distance two robots can be from one another.

**Additional non-genetic parameters**:

1. hits – how many shots have hit another robot.
2. totalShots – the total number of shots fired by the robot.
3. firepower – how strong a bullet should be. This is used in methods that need to adjust the power of shots. Default value is 1.
4. maxDist – the farthest possible distance two robots can be from one another. Would be the diagonal/hypotenuse of the battlefield. My thought is that it would be calculated during the initialization of the robot and then passed around when needed.

**BulletStrategy(ScannedRobotEvent e)**

Narrative:

Precondition:

Postcondition:

If bulletStrategy is distance

Call DistanceBased(e)

Else if bulletStrategy is light

Call LightFast(e)

Else if bulletStrategy is powerful

Call PowerfulSlow(e)

Else if bulletStrategy is medium

Call Medium(e)

Else if bulletStrategy is hit

Call HitRateBased(e)

Else

Report an invalid bullet strategy

**DistanceBased(ScannedRobotEvent e, double maxDist)**

Narrative: Adjusts the power of the shot based on how far the robot is from e.

Precondition: maxDist should be calculated beforehand.

Postcondition:

Set distPerc equal to e.distance / maxDist

Set firePower equal to 3.0 \* (1 – distPerc)

Clamp firePower between 0.1 and 3.0

Fire(firePower)

**LightFast(ScannedRobotEvent e)**

Narrative: Focuses on delivering quick, low power shots.

Precondition: None

Postcondition:

Fire(1)

**PowerfulSlow(ScannedRobotEvent e)**

Narrative: Focuses on high power shots at the expense of speed.

Precondition: None

Postcondition:

Fire(3)

**Medium(ScannedRobotEvent e)**

Narrative: Takes the middle road in terms of fire power and speed.

Precondition: None

Postcondition:

Fire(2)

**HitRateBased(ScannedRobotEvent e)**

Narrative: Tries to adjust the speed based on how successful the robot is at hitting e. It will decrease the power of its shots if its hit rate falls below the threshold of 0.8 as weaker shots travel faster and therefore have a greater chance of hitting e.

Precondition: A BulletHitEvent needs to be set up to increment the hit counter of the robot.

Postcondition:

If hits / totalShots is greater than or equal to 0.8

Increase firePower by 0.1

Else

Decrease firePower by 0.1

Clamp firePower between 0.1 and 3.0

Fire(firePower)

Increment totalShots

**onBulletHit(BulletHitEvent e)**

Narrative: Called whenever a bullet hits another

Precondition:

Postcondition: Increases the number of hits the robot has successfully made.

Increment hits by 1