Robot Wars

By Taha Khan

The task of creating the smartest fighter robot AI was a challenging but rewarding process. By the end of it, four main robots were created, each with their own level of decision making as well as pros and cons.

The initial step was to create a robot that can locate and attack the robot with the lowest health. This algorithm was incorporated in khanFighterRobot. The benefit of having a robot focused on only health is that it will most likely take down its opponent when it reaches them since they are the weakest on the battlefield. However, the limitation is that this robot does not take into account how far its target is, so the energy is not managed efficiently. This robots stats were attack-4, defence-3, and number of moves-3. The traits were fairly balanced in this robot because it did not need to focus on one particular area, as the opponents it finds will already have low health and will most likely get killed.

The second robot that was created was khanFighterRobot2. This robot made its decision on who to attack based on closest proximity. The distance each robot was from this robot was calculated using a method called findDistance() in tahaOppData, an extended oppData file. Then the distance list was sorted using insertion sort. This was effective in a way because it managed its energy very efficiently by limiting the number of moves it does. The down side to this strategy however was that it did not take any other factor into account such as health. So if a robot was 4 spots away with 5 health and one was 3 spots away with 50 health, this robot would go for the one with 50 health even though the other is a much easier target. This robot stats were attack-5, defence-3, and number of moves-2. This robot focused slightly more on attack and less on number of moves because it is already optimizing the number of moves so it does not need too many, which is why it can afford to add to the attack.

Lastly, two versions of the final robot were made. Each robot is much smarter than khanFighterRobot and khanFighterRobot2, who only had one sort each. The algorithm that these robots used incorporated health, distance, and the number of moves the enemies took. These robots were called khanFighterRobot3 and khanFleeFighterRobot. Both robots used the same formula to calculate which robot to attack. First, they gather information regarding the distance that each robot is from them. Next, it accesses the global variable called previousArray, which is an array that contains the OppData array file from the last turn. When it is my robots turn to attack, it compares the previousArray to the oppData array in the current turn. This allows it to calculate how much the other robots moved from their previous location. It always keeps track of the highest number of moves they have moved so it does not over-ride the amount they move at every turn if they moved less than before. Thus giving an accurate estimate of the number of moves and energy they have. This number of moves value, the health, and the distance was all put into one formula and then sorted using insertion sort.

The formula was:

The reason the coefficient of 0.5 was used for distance was because of the need to scale it down. Distance would normally be a value of 1-19 because that is the size of the arena. So it was multiplied by 0.5 so that it can be a value of 1-10 instead. The same reason applies to the coefficient on the health. Health would be from 0-100, this range was too large and would not be accurate when performing the calculation, and so by multiplying it by 0.1, it will be from 1-10. The value for the number of moves was already in the same range so scaling it was not needed. So these three levels of determination made khanFighterRobot3 smarter than the other two who only had one level each. The stats on this robot were, attack: 5, defence: 2, number of moves; 3. Since this robot was much more aggressive and offensive than the other robots due to its multi-level determination, it did not need much in its defence stats, so it is focused on attack and number of moves.

There was one last version of khanFighterRobot3 that had slightly different determination. This was called khanFleeFighterRobot. It used the same method to find the number of moves the opponents took, the health and the distance. However the strategy this robot used was, when its health is below 75 and above 25, run away and do not attack anyone. This robot would jump in the fight at first, but later into the game, would run away and wait for the competition to take each other out. The point was, for 50% of this robots life, it would flee, then the other 50% it would fight, which is for the first and last 25% of its health. In the end, this version of khanFighterRobot3 was not as effective as the original because it received many more penalties for not fighting in 5 rounds and constantly lost health for this. The traits this robot contained were highly speed based, attack: 3, defence: 2, number of moves: 5. This robot was very focused on number of moves since getting kills is not its priority, it is made to attack, flee, and attack again when the others are weak.

A number of trials were conducted to see how my main robot, khanFighterRobot3, compared to the other three.

In Trials 1 and 2:

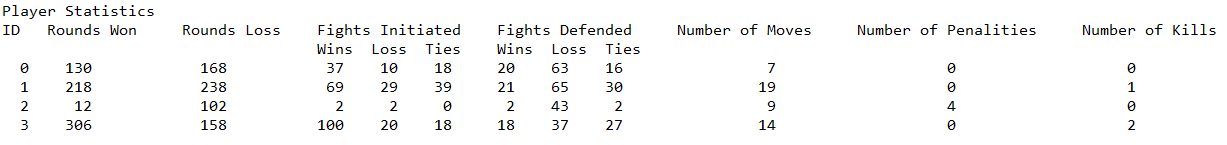
Id 0 = khanFighterRobot

Id 1 = khanFighterRobot2

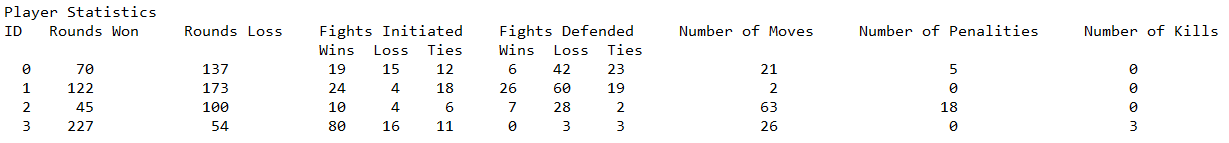
Id 2 = khanFleeFighterRobot

Id 3 = khanFighterRobot3

Trial #1:



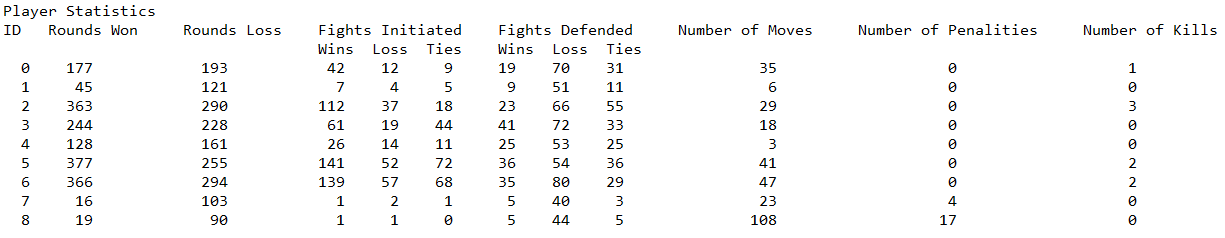
Trial #2



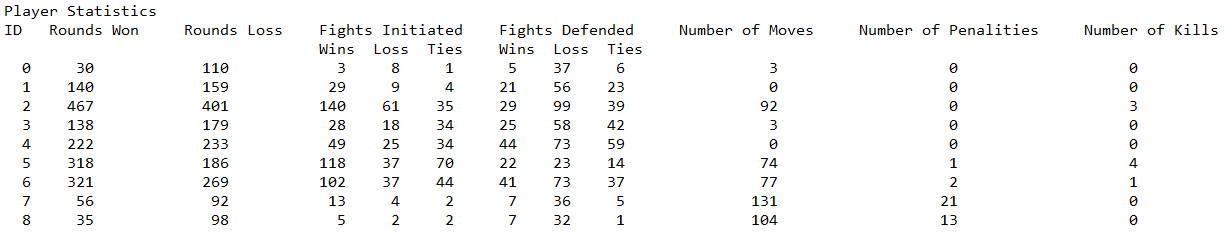
When faced against one of each other type of robot, khanFighterRobot3 is a tough competitor. It has the most number of kills in each trial and the highest win: lose ratio. As mentioned above, khanFleeFighterRobot will gain the most number of penalties out of all, making it not as effective as the original khanFighterRobot3 even though they have a similar algorithm. To further test my robots capabilities, it was tested in an arena with 9 robot total.

For the next trials, robots with ID 0-2 are my main robots (khanFighterRobot3).

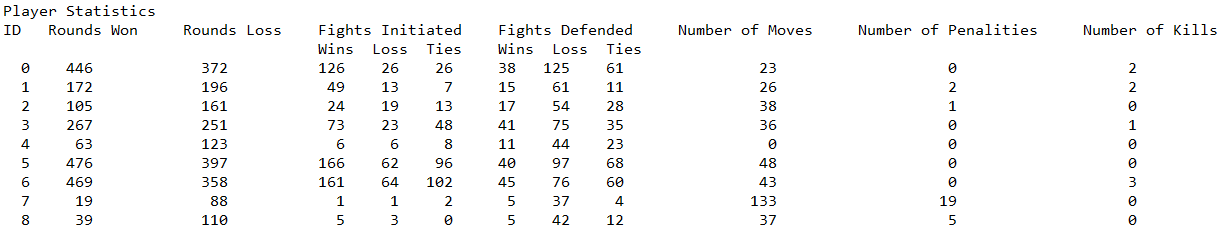
Trial #3:



Trial #4:



Trial #5:



After running these tests, khanFighterRobot3 continuously had better overall game statistics than the other robots. Looking at the number of kills, khanFighterRobot3 and khanFighterRobot (ID 6-7) consistently have the highest scores. The khanFighterRobot is designed to take out the opponent with the lowest health, so it makes sense that they have a high number of kills. But khanFighterRobot3 always has more rounds won than lost compared to all of the other robots.

Overall, these test cases show that my robot, khanFighterRobot3, is the most well rounded out of all of the other fighter robots that were made. The combination of high attack stats and a sorting algorithm that takes three aspects into consideration makes for a strong and smart robot, even though it may lack in defence.