

Documentation

AI Elements

I have an instance of a Finite State Machine for each component of the robot (body, gun, radar).

The gun uses linear targeting to try to predict where the enemy is going to be, so we know where to shoot to hit the enemy.

I have used the Arrival and Flee steering behavior to make sure my robot tries to be a certain distance away from the enemy. Since it might try to flee backwards into a wall, I have implemented a safeguard that changes the body state if we are stuck for too long.

Since I implemented several states that the body can use (different types of movement) when in the preferred range of the enemy, I made a kind of dumb-learning system to select the best state. It simply takes the best scoring (calculated by taking our energy - enemy energy at the end of the round) state after a set amount of "sample iterations". This required file IO, so I could save the results between rounds.

Satisfaction factor

I like the way I made the somewhat-learning system, although it could be improved, both by adding more states it can try, and by increasing the factors it uses to calculate the score. I also like the way I divided up the different parts of the whole project, so that all the elements work fairly independently. This made it much easier to make and test, as problems were easy to track down.

More time

With more time I could have used steering behavior better. I have some movement that isn't really a steering behavior, and I couldn't get the Wander behavior to work correctly.

I would like to make a more robust system for the State handling. Currently you can enqueue states that control the gun to the body state machine and so on. This wasn't a problem for me, since I was alone, but on a larger project this would obviously be a bigger issue.