# Prog 2: Race Track

1. See code
2. See command line output. Description of characters:

O: Off track

T: On track

0-9: the track

X: If the car is outside the track

S: Start line

F: Finish line

1. The algorithm terminates when the value function doesn’t changes a lot in the last interation. We set a small threshold to 0.01. If the maximum change of one value in the value function is higher than this threshold, it’s going to execute the algorithm again. Now we have to differ two cases. With and without a smart initial value function.
   1. The value function is initialized with 0. It need’s 19 iteration to go below the threshold.
   2. With an Euclidian distance function to the nearest starting point. It needs only 14 iterations to go below the threshold.
2. See code
3. Algorithm finishes when you reach either a state outside of the boundaries or the final line
4. The Results for Sarsa are much worse than Dynamic programing. Sarsa doesn’t seem to find a good policy.
5. Use the eclipse project or execute RaceTrack.jar in your command line with the command java –jar RaceTrack.jar