Darie Picu **s0935756**

SDP individual report 4

I worked in the strategy team improving the current methods so that we take into account the obstacles on the pitch and finished the ball predicting methods. I also worked in the control team trying to do a simple design for milestone 4 without using the arcs that we had trouble implementing.

Strategy

All the methods in strategy such as the ones checking whether we had a clear shot or an angular shot possible, were changed in order to take into account any of the obstacles on the pitch. Also we were only shooting for the middle point of the opposition goal, now obstacles and our position on the pitch are considered in order to decide the best point to shoot at in the goal. The most important work that I did in strategy for this milestone was improving my ball prediction method and writing a ball intercept method, that was returning the best interception point, given the current worldstate. In order for the interception point to be computed, I had to know how much time would take our robot to get from one point on the pitch to another. Experiments were done to find a good way of approximating this. I also did about 15 JUnit tests to test my ball predicting method.

Control

Our control was changed such that the plans generated from strategy would be received as soon as ready. So control was not getting plan on demand, but when strategy sent one. This was done about 30 times per second. This might be a good idea if one wants to implement the arcs properly, but for the simple solution I tried to implement for milestone 4, this control proved very difficult to deal with. Two simple commands such as a rotate followed by a move forward for a distance, were impossible to execute. Unfortunately, I spent a lot of time trying to work out a solution. At one point for example I had 2 boolean variables declared, one for moving forward and one for rotation and was setting them true and false for different amounts of time just to ensure that only once command was executed at a time, not a queue of 2. I was allowing 1 second for rotations and 4 for moving forwards.

The simple design I tried implementing was just to get the best interception point using my methods, then turn towards it and move half the distance to that point, then check again what the intercept point was and do the same thing until I am close enough to the ball. Because control was flooded with new plans, sometimes the intercept point was wrong and this solution proved inefficient. Furthermore, the robot is not accurate when a turn by an angle command is sent, even if the speed of turning is set low. This coupled with the fact that the intercept methods were assuming that the ball would go in a straight line (which it does not) and that it goes at constant speed, meant that the success rate of the methods was quite low.

I will continue working in the strategy team. My goal is to improve the plans, so that we would do well in the final tournament.