


System Design Project - Individual Report 3

Biser Hong – s0840625 – Group 12


Introduction



This report describes my personal achievements since milestone 2 and the goals set for the next milestone and friendly match. My contributions **are** detecting collisions and providing the strategy with easy access to the drawing board of the simulator. **On the strategy part**, I have implemented a circular buffer to store the last positions of the ball and the robots.

Simulator


For collision detection, I first tried using an external physics library. However, the team's usage of the Simulator **is** very specific and I wrote my own solution. The different objects **handle** their own specific collisions: the ball **bounces** off the walls of the pitch, the robots **collide** with each other and the walls, and the kicker **kicks** the ball at the angle of the robot. The current implementation **is** sufficient for this milestone but the collisions between the robots and the ball **are** not properly handled. This is the first of the goals for the next milestone.



Improving strategy debugging efficiency, I implemented a class that **makes** access to the drawing capabilities of the simulator easy. This class **abstracts** all of the work needed to draw on the simulator's screen and **provides** a better alternative to the previously scattered console printing statements that clogged the code. We can now easily visualize the state of execution of our algorithm, any calculation we doubt or need, and the navigation points used in the strategy.

Mouse and keyboard shortcuts **provide** further functionality like repositioning the objects on the pitch and enabling and disabling the robot in use. The first addition **allows** us to immediately test the behavior of the strategy in any possible case. The second one **allows** us to stop the robot from executing commands. This way, we **can** easily examine the current state and parameters of the strategy. Adding real-time parameter tuning is the next step.

Strategy



Looking ahead, I implemented a flexible class that encapsulates the functionality of a circular buffer. It keeps a history of any number of the previous positions of the objects on the pitch or their orientations. Orientations can be averaged out to provide a better estimation. The previous positions are used to extrapolate the angle, by fitting a line through them, at which the ball and the opponent are moving. This will be used to predict the position at a future moment. This is made very easy to visualize with the drawing class described earlier.