Report 1 - Juozas Kaziukėnas - s0820151

I was assigned to robot construction team where 3 of us worked on an initial design for Milestone 1. My main job was testing and troubleshooting potential ideas for future improvements. Stability of the robot and performance of the kicker were two issues I spent most time thinking about and trying potential designs. At the end we ended up sticking to our initial design of having a kicker directly attached to a motor, because other designs didn‘t worked at all or the speed they kicked at was just not good enough.

In a second meeting my fellow team mates realized that I'm actually much more useful in Movement team as they have been struggling a bit with Java code for both server side (code running in controlling PC) and client side (robot), and I'm very proficient in Java. One of the first things I did while in Movement team is restructuring code base to follow a clean structure allowing for feature developments. This was then for the first time committed to our teams SVN repository. I'm aiming to have a modular design and follow a clean structure throughout whole project.

At the same time I was discussing our interaction with Vision team leader and after a brief discussion we decided to use Standard I/O as our communication protocol. We didn't had time to test it before first Milestone, but I think that this is going to be the most efficient and fastest communication protocol. Other protocols we considered are HTTP Sockets and other socket based communications, but having worked with them in a past, I suggested that overhead would be too big for communications needed. Initial design for a protocol is:

```
Position1X Position1Y Direction1 Position2X Position2Y Direction2 Position3X Position3Y
```

Getting Bluetooth working both for uploading code to the robot and communication between the robot and a PC was crucial point for not only first Milestone, but also for whole project. Thus I worked on getting that to work and I managed to make Milestone 1 demo controllable from PC by issuing a command “1” which is then sent to a robot and robot executes it.

In parallel to coding work I was busy planning how my team is going to test our code library. One of the solutions was suggested by Vision team is to build a simulator, but, at least for our side of code, it's much more practical to work with unit tests and have a tests suit to automatically test our applications. This is not decided yet, but I'm about to start implementing it just after finish of Milestone 1.

Overall I'm quite happy with both my team and personal progress in this milestone. We have successfully demonstrated working robot and because I was able to get Bluetooth working for transmitting commands our robot didn't required any direct controls rather it was executed using our server application.