SDP Individual Report

Jakub Kaszyk, s1108328

April 24, 2014

The System Design Project (SDP) is a group-oriented practical, with the task of using Lego Mindstorms to play two-a-side football similar to table football (foosball). Our team, the Mindstorm Troopers, approached this task by dividing the work among four different teams Engineering, Strategy, Robot Programming, and Vision. At the first meeting, I was appointed as team leader of the group. My main task in the team was organization, however throughout the semester I was involved in all parts of the project. Organization mainly involved scheduling meetings, allocating tasks, reviewing design decisions based on time constraints, aiding in communication between team members, and at times solving disputes between members.

Even though Milestone 1 involved only a marginal part of our overall project building a robot and programming basic movement we decided that we needed to get started on building blocks for later work as soon as possible. In the first week of the project, I split the group into individual teams Engineering, Vision, Strategy, and Robot Programming, and allocated tasks to each person.

After a week, I appointed a team leader for each of the teams, based on initial involvement in the project. We then moved to a system in which I, jointly with each team, decided on the next deadlines. It was then the team leaders' job to assign tasks within their team.

At first, the team met once a week officially to discuss progress and tasks. In addition to our weekly meeting, we quickly expanded to multiple meetings as often as possible, so that team members could work on their assignments together, and progress more quickly.

Initially, our organizational system worked well, however with less guidance from the course organizer after Milestone 3, I quickly discovered that we needed our own methods of keeping track of progress. I implemented specific, short deadlines, which not only allowed for tasks to be finished as quickly as possible, but allowed for feedback from the rest of the team in case any more changes were needed.

At various times during the semester, task reallocation was necessary due to problems we encountered such as illness, delays, or imbalanced workloads. Tasks were reallocated as needed to ensure that work progressed as smoothly as possible. Naturally, this also involved changes in leadership two of the four teams.

Aside from organizational tasks, I was directly involved in all parts of the project. For Milestone 1, I debugged our robot, which was having problems up until the hour before the Milestone. It was at this time that I discovered that sensor readings are kept in a buffer even after the robot is turned off, and due to this, every time the robot started, it read readings from the previous cycle.

I also calibrated the kickers for Milestone 2, so that the defender could kick into the attacker zone without kicking further than that. Throughout this time, I was also working on the vision system. Setting up the correct libraries was the most time consuming part of SDP in the first three weeks. Once I was able to finally do that, I wrote the basic tracking algorithm for any object, which simply finds the mean value of the pixels.

At Milestone 3, some of the team problems were starting to become more obvious. While Milestones 1 and 2 could be done fairly quickly, Milestone 3 required a lot more work from everyone. It was then that I discovered that not everyone in our team has the same motivations, and is working for the same marks. Due to some technical difficulties as well as decreasing motivation

from certain people in our team, we had a lot of delays in all aspects of the project, which caused us to perform poorly on Milestone 3. Seeing the problems we were still having the night before Milestone 3, I implemented a nave version of the defending robot, which simply moved to the same vertical position as the ball using pixel locations as coordinates. This design was able to successfully defend two out of three shots directed at our goal.

Following Milestone 3, we had a few weeks without assessments, and could focus on solving the technical problems we encountered, as well as improving our design without the pressure of assessments. Our plan was build holonomic robots, which would increase our chances for interception, by removing the necessity of having to turn before moving. However, due to the long break before the first assessed friendly, again quite a few people in the team lost perspective of the objective of the course, and stopped working. I dealt with this by reassigning tasks to make sure the most critical work was done a moving and kicking robot. While this was completed before the friendly, it didn't leave us with enough time to test our system. We discovered too late that the three wheel triangular design that we were using wasn't stable enough at higher speeds, and again this inhibited our performance. This unfortunately became a pattern for the remainder of the semester large delays in most areas of the project, and only myself and one other person working on the project full time.

Throughout this second half of the semester, I continued scheduling meetings, assigning tasks, and reaching out to team members, but received very little response. In effect, I took a lot of the project into my own hands, completing parts of the strategy code, and testing the performance of the system. A few times we noticed that after we saw progress start, it for some reason reverted to something completely unexpected. It was at this point that I implemented formal test cases. These test cases provided a methodical approach, which allowed for sanity checks every time we changed something in the system.

When two weeks before the final day we still didn't have anything to show, and most of the team hadn't been in touch, I decided that building a whole system wasn't really an option any more. From this point on, we just tried to have anything working at all. Along with one other person, I was able to program all of the basic strategy reactions —interception, moving towards the ball, catching, turning to the goal, and kicking. Unfortunately we weren't able to connect all four states using simple conditions, but it did show basic movement and showed some form of intelligent behaviour.

While I did everything I possibly could to save our project, there are things that I would have done differently if I did SDP a second time. In particular, I would prioritize work at the beginning differently. I evenly split up our group into four teams at the initial meeting, and for the most part, people stayed in those teams. However, I should have initially put more of a focus on the engineering, since until the very end of the course we didn't have a stable robot to test on. Had more people worked on the robot as the primary task, we would have had less delays in later stages, and wouldn't have had to rebuild the robot so many times.

A lesson learned the hard way is that we should have tested our system a lot more. As team leader, I focused on distribution of the workload, but a more formalized method of development would also have been beneficial.

In retrospect, SDP has been an eye-opening experience for me. Throughout the course I refused to believe that situations like these could happen in real life, but I recently have come across similar working environments, and I am grateful that I have experienced SDP.