

**Final Report**

ECSE 211 – Design Principles and Methods

GROUP 03

Garret Holt, Sung Hong, Nawras Rabbani,

Matthew Rodin, Tiffany Wang & Troy Yang

APRIL 15TH, 2016

1. **INTRODUCTION**

***This should review the goals of the project***

The goal of the DPM project was to build a robot capable of playing a simple game similar to soccer. In other words, the robot must be capable of: localizing itself on a platform, avoiding wooden block obstacles while moving, identifying the color of the ball, picking the ball up to shoot it on the offense zone, defending at the goal and implementing the robot with a WI-FI class. However, one can say that this was simply a secondary goal and the main focus was to teach engineering students about design process. Throughout the semester, we had various lectures on design principles and methodologies. This taught the students proper problem solving skills, the proper design process and documentation. The objective was to provide an experience, which would force us to apply these techniques. As well, it taught students how to work coherently with a fairly large group. The purpose was to demonstrate the effectiveness of good communication in a group setting.

***What was the main reason(s) for doing it?***

The main reasons for doing this project was to give engineering students a taste of the future. In many cases, we will be working in design-based groups during our employment. The skills learned during the project will no doubt be applicable in the future.

***What was the project intended to achieve?***

Although at the beginning, the project seemed to be all about having a perfectly working robot, the main purpose of the project was more about learning the process and steps made towards building the final robot as mentioned above. The intention of the project was to learn about the basic design principles and methodologies, involving planning and dividing the project among group members, and improving the project through constant editing, revising and making adjustments under the given budget.

1. **TEAM ORGANIZATION – THE START UP OF THE PROJECT**

***How were tasks allocated?***

The tasks were delegated depending on each person’s role in the team, yet it is important to mention that their role’s description and responsibilities does not delimit their duties. Everyone crossed over their labeled job and helped with other project components. Member’s availabilities were also taken into consideration when allocating tasks. In this case, the compatibility of everyone’s schedule and capability is important for efficiency purposes. If someone is unavailable to accomplish work before the deadline, it will be given to someone else. As well, everyone’s position varies along with the progression of the project.

***How was the initial Gantt chart designed?***

The initial GANTT chart was designed based off the preliminary one provided by Professors D. Lowther and F. Ferrie and the team member’s availabilities. Not knowing much about how the project would turn out, the details of each task and time needed for each member to accomplish them, the GANTT was planned out according to the deadline, the meeting dates and the milestones. In general, things were updated mostly up to three weeks following the kick-off. In addition, the ideas were pretty broad, and members were preliminarily assigned to tasks. An analysis of a critical path is crucial for the development of the GANTT chart. For this project, the mechanical design and the sensor characterization are events that occur in the beginning, as opposed to software development and testing, which happen mostly near the end. However, as we advance in the project, the GANTT was tailored to our progress and to our robot; the tasks were also broken down into details.

***What information was used to estimate the initial task breakdown?***

At our very first meeting, we had a brief discussion about each team member’s strength, weakness and interests. As well, everyone’s previous experience in each disciplinary, mainly documentation, project managing and coding, is exposed for the analysis of the initial task breakdown. From there, we voted on the positions each member would hold for the final project. Every member is given responsibilities that match their experiences, their interests and their strengths.

***Were any guidelines followed in developing the first version of the chart?***

We followed the GANTT chart rules from lectures Design VII and Design VIII (12 and 15 February 2016 respectively). Dependencies between tasks are really important and ought to be analyzed out. Hence, the milestones were set as reference points, from where we can build up our testing plans. In addition, Prof. Ferrie and Prof. Lowther emphasized on the importance of sensor characterization during the lectures, testing and the flexibility of the task delegation. Leaving space for potential changes is important, for we do not want to be taken by surprised and fall behind the schedule.

1. **ISSUES ENCOUNTERED IN THE PROGRESS OF THE PROJECT**

***Were all the dependencies correctly identified at the start of the project?***

No, not all the dependencies were clear at the very beginning of the project. We knew that the software design would depend on the hardware design so we focused on the hardware design in the beginning. We also started the software development after that. Later on, we found out that the placement of the balls was changed so we had to change the hardware design twice, then we also modified the software code and conducted more testing.

***What dependencies contributed to the critical path of the project?***

As we found out during the process, the mutual dependency between the hardware and the software is essential to the success of the project. Every single component we changed hardware-wise would affect the software design and testing needed to be redone in order for the robot to work properly. Even a small component that we add to the robot would change the weight distribution and thus the overall performance of the robot. And due to the software restraints, some of the hardware ideas we had could not be implemented. Different threads in software also play a vital role. During the demo, we failed as the threads were working individually but they failed to work together.

***What initial ideas turned out either not to work or be based on wrong assumptions?***

Many of our initial design ideas are based on wrong assumptions, especially about how the balls would be placed. Initially we thought that the balls would be placed randomly in a certain area and we designed the “ball-scooper” in a fashion that it would rotate outwards and pick up the ball, which will then fall into the ball storage area. Later, we were told that the balls would be dispensed horizontally on a wooden structure, which forced us to change the scooper so that it swipes horizontally.

***What other issues/factors had an impact on the project?***

We had to deal with the inaccuracy of various sensors. For instance, we utilized the light sensor to identify if the right ball is being picked up. But since the range that the sensor can detect is limited, so we had to place the sensor in a way that it is very closed to the ball. The restraint created difficulties in hardware design since we need the balls to fall into the right spot every time and make sure the color sensor does not block the balls while being very close to them. Another factor is the size of the robot, since it is a complex project we had to make the robot relatively big, and it turns out that it would affect the mobility of the robot. Finally, as we carried on with the testing, we found out that the balance of the robot as well as the bumps in the tiles can create issues too.

***How did these affect the project progress?***

These affected the project process in many ways. We found out that due to the limitations of the sensors we don’t have much freedom in terms of their placements. They need to be placed really precisely in order to work properly. For example, the light sensor needs to be placed really close to the floor and to the ball otherwise there would be too much error in its reading. As for the big size of the robot, it was hard to turn in corners and avoid obstacles. Therefore, we had to spend a lot of time testing the localization and refine the code.

***In particular, did the project run to the plan you had initially created?***

Mostly the robot ran as we expected. For the individual tasks such as localization, ball pick-up and ball launching, the robot performed very well. The only problem that we found difficult to solve is for the robot to integrate these different parts into one successful run.

1. **THE BUDGET**

***What constraints did the budget place on your team?***

The budget was 324 hours for the entire project, which comes out to a weekly total of 54 hours (9 hours per person). Although there was less constraint at the beginning of the project, there began to be some pressure during the final week to finish the project in the allocated time. In the end, we used ~92% of the allocated budget and therefore it was not the project’s biggest constraint.

***How did initial planning for available resources and budget spending affect the development of the timeline?***

Knowing the budget before the project began was a big help when it came to planning for available resources. We knew that we would not all work exactly 9 hours a week for the entire duration. Although testing and software was constant throughout the project, hardware hours peaked towards the start while documentation put in more hours towards the end. Proper planning allowed us to constantly monitor our inputted hours in order to stay within budget.

***Did you allocate resources to all the project tasks, i.e. all the way to 15 April, at the start of the project and use this to estimate the budget. If not, explain why not.***

During the first week of the project, we allocated resources all the way to April 15th (visible on the Gantt chart). Although, this inevitably changed throughout the course of the 6 weeks, we were constantly updating our budget table in order to never exceed the 324 total hours.

***What would you have spent if there had been no limits on the budget and when in the process would extra budget have been useful?***

Seeing that we used ~92% of the allocated budget, there was sufficient hours for our team to finish the project. We must thank proper planning and extreme productivity for allowing us to finish on time. Of course, if there was more allocated budget, it would have mostly all gone to integration testing during the final week. We spent most of our testing budget on unit testing which left little to none for testing the integration aspect of the project

***Where were you weak in resources and what would you have done to resolve this issue if you had fewer budgetary constraints? At what point in the project could these extra resources have been brought in?***

The main point of resource weakness was the software development. Nawras and Garret were mostly in charge of coding throughout the process. However, a project of this magnitude, especially for a first or second year student, is extremely complex for only two developers to complete. If more resources could be allocated, we would have definitely used them in the last two weeks to help the software team fix the multithreading issues with the code.

1. **HOW THE PROCESS CONTRIBUTED TO THE SUCCESS (OR FAILURE) OF THE PROJECT**

***How would you modify the process to increase your probability of success? & Which parts of the process were the most difficult to implement and why?***

Despite not successfully completing the project, our process was effective at coordinating members and getting work done. Through our various outlets of communication, we were always able to get ahold of each other. This was important because we were able to coordinate meetings and work sessions effectively. Almost all of the work done for the project was done in pairs. For example, we always had the two coders working together, or a coder with a tester to maximize efficiency. A difficult part of the process was the planning. We often ran into unforeseen problems or problems that we thought to be minor but ending up taking a long time to solve. This caused us run out of time towards the end, which meant we did not have enough time to comprehensively test the full run

***How much time was devoted to testing? & Was this at the subcomponent level or did you leave it all to the end? & Were the tests you designed sufficient?***

A possible modification to our process would have been a greater focus on testing; more specifically, we should have spent more time testing the full run opposed to subcomponents. Despite allocating a lot of our time to testing and having around 40 testing documents, our testing was insufficient. We simply ran out of time to run more elaborate tests that would be comprehensive in covering every situation possible that could cause potential error. Most of our tests were basic and simply tested subcomponents. For example, we ran many tests to assess the accuracy of navigation and localization. Our tests should have had more layers to them for example navigation with obstacle avoidance and with ball pick up. Multi layered tests such as this would have provided a lot more information.

***How much time did you estimate full prototype (i.e. integration) testing would take? & How much time did it actually take? - If there was a difference, why? & How would you change your test design process to make it more effective?***

Our group did not allocate enough time for integration testing simply because we were behind schedule. We were still working on perfecting subcomponents the weekend before the demo, which left very little time for the testing of the full run. Obviously this was not the case, and we ran into many unexpected problems that took up the few remaining days. Looking back at the project I predict that ideally no less than a solid full week of testing would be sufficient to get the project running consistently. However, simply due to time constraints this was not the case for us. The changes to our test plan seem fairly obvious, putting less emphasis of the testing of the subcomponents and more on the success of the full run. For example, we spent a lot of time on our localization and got it working so well that it was almost always within 1-2 degrees of perfect. We could have instead used this time for testing the full run.

***What was the impact of the beta demo on your design process?***

The beta demo was a big wake up call for our group because it looked like it went horribly. Our robot localized and then drove into the wall, which made it seem like we were far from completing the task however, the problems were actually quite minor. It was a caused by the threading and odometer, but the demo was the push that we needed to realize and fix these problems.

1. **THE SUCCESS OF THE DESIGN (ROBOT) IN MEETING THE ORIGINAL SPECIFICATIONS AND THE PERFORMANCE REQUIREMENTS**

***What is your impression of how the robot performed?***

The robot did not complete the full run during the competition today. It was able to localize precisely, however, it was not able to navigate to the right offending zone and perform the other tasks. The same is also true when we were playing the defending role.

***Did the robot perform as you expected?***

We were expecting the result before the final competition. Due to the limitation of the budget we were not able to run more testing and modify the code accordingly. Thus although we were very close to being able to complete the competition, we failed.

***If the robot failed (i.e. did not meet all the performance requirements), why did it fail? Can you point to the sections of the documents that describe the decisions that led to the failure (provide the references to those decisions)?***

The main reason the robot failed during the final demo was due to its inability to navigate to the proper location on the board. This was due to a disconnect between the flow chart (see “Full Documentation” – page 17) and the robot’s actual performance. Although the robot localized properly, the robot was unable to navigate to the ball area.

1. **CONCLUSIONS**

***What did you learn from this course?***

Each of our team members learnt a lot from this course and the project. We acquired many valuable practical lab skills, how to communicate effectively with partners and how to allocate the tasks accordingly depending on each person’s strength and weakness. We also learnt the importance of documentation and how essential it is to record the process in a clear and concise manner for others to understand.

***Explain why a clear, effective and controlled process is necessary when working in a team and what it helped you achieve.***

An effective process allows everyone in the team to know exactly what should be done on time. With the Gantt chart as well as weekly deadlines, the team kept its momentum and made progress consistently throughout the process. Breaking down the tasks allows us to utilize the time more efficiently so that we can avoid time conflicts as much as possible and maximize the efficiency, which is crucial for the budget.

***Is any of it applicable to other courses you might take?***

Yes. What we learnt from this course can be used in courses such as ECSE 456/457 - ECSE Design Project (1 & 2): Engineering practices (research, experimentation) and documentation

***What would you change in what you did if you were doing it over?***

If we were to do it over again we would have settled on the mechanical design earlier. Throughout the whole process we have changed the mechanical design three times and each time we had to redo the testing, which took a lot of time. Moreover, we would have spent more time doing the testing for the full run. We failed during the beta demo because there were problems with the threading in the code and we believe that this could have been prevented if we did the full run testing earlier and corrected the problems. Lastly, we would have done the testing and the coding in parallel because as it turns out that would be really effective. Many of the big changes we made was because of the lack of testing. Doing testing right after coding would help spotting the problems and solve them early.

“The undersigned members of team 03agree that the contents of both this report and the information handed in on CD, DVD or memory key, provide an accurate representation of the work done on this course and the contributions of each team member.”

Garret Holt

Sung Hong

Nawras Rabbani

Matthew Rodin

Tiffany Wang

Troy Yang