FINAL REPORT

Project: LIBERTY
Task: *Final Report*

TEAM 1

Andi-Camille Bakti: 260677964

Bill Zhang: 260604186

Edward Son: 260332932

Gabriel Negash: 260679520

Xijun Liu: 260654285

The undersigned members of **team 01** agree 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.

TABLE OF CONTENT

TABLE OF CONTENT **2**

1. Introduction **3**

2. Team Organization **3**

3. Issues Encountered **5**

4. Budget **6**

5. Process Contribution to Success of Project **7**

6. Validation **8**

7. Conclusion **8**

1. Introduction

The objective that we designed our robot to achieve on demonstration day was to make the robot autonomous and capable of localizing, and traversing over a 12 by 12 gridded playfield via a path and a zipline, in order to detect and recognize a block of a specified color within a zone. In order to achieve this objective a set of functional and nonfunctional requirements were generated, as can be seen in our constraints (***CON-GEN***) and requirements (***REQ-GEN***) documentation.

What was the main reason(s) for doing the project?

However the main purpose of the project was not necessarily the result the robot achieved, but rather the design process and build methodologies behind it. As a design process course the project provided us an opportunity to attempt to implement all of the design principles and methods we learnt during the preliminary section of the course.

What was the project intended to achieve?

The project was intended to teach us the skills required to approach a design project in-industry. As such we needed to learn communication skills, documentation skills, as well as time-management skills. In order to instill these lessons to us the project was designed so as to provide a less-than ideal time frame and nebulous requirements.

2. Team Organization

How were tasks allocated?

Initially project roles were decided via a democratic voting method in which we listed out the roles we would like to fulfill as both our primary role. Secondary roles complemented the primary ones, both the software and hardware leads were secondary testing engineers as testing was an essential aspect of the development process and test results were dependent on both the hardware and software requiring testers to have an intimate knowledge of the system at hand. Due to the sheer amount of development work required in the software design, as well as the large amount of software and computer engineers in our team both the testing manager and documentation manager acted as secondary software developers. In order for the project manager to keep track on the project as a whole the project manager aided all other subteams.

How was the initial Gantt chart designed?

The initial Gantt chart was designed in order to accommodate two main design decisions. The first requirement that went into the Gantt chart design was to leave two weeks for full integration testing prior to the competition. The second requirement integrated into the chart was to ensure that for the first week all members of the team worked on the initial software and hardware design concurrently and together in-person. This second requirement was to ensure that further along the project unnecessary rework would not be needed as well as serving to aid in team communication, and design simplification. The initial Gantt chart was also designed to provide slack in terms of budget.

What information was used to estimate the initial task breakdown?

A key source of reference for time estimation was our experience in prior labs. Development times required for prior labs were taken to be a good estimator for the project as the labs were themselves designed to mimic subsystems of the final project. Other tasks were decided in order to keep on track with the weekly deliverables. On-top of lab experience each person's individual skills were taken into account into account as it was the responsibility of the person completing the task to provide an estimate of task duration.

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

A guideline utilized in developing the chart was an equation for estimating the time necessary to complete a task. While evaluating time necessary to complete a task we considered the optimal time we could potentially finish it in, while also considering the worst-case time required.

3. Issues Encountered

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

In the beginning of the project, we identified all dependencies which match our final dependencies. For the majority of the project, the progress matched our initial plan, especially for documentation and testing. However there were some minor drifts in the software in terms of class arrangement from initial dependencies to final.

What dependencies contributed to the critical path of the project?

As visible in the gantt chart (Version 5) the main set of dependencies that contribute to the critical path are those of the interplay of software and and testing. As hardware remained static post week 2, the dependency of software refactoring on component testing & integration testing contributed the most to the critical path.

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

As the project was in development, the design naturally changed along the way. Fortunately, most of our initial ideas were correct. Although, in the software structure, there was a functionProvider class which a superclass for all functional classes such as Capturing, Localization and Odometry Correction. The superclass contains the abstract methods to receive data. The goal of this design was to create all sensor objects in one class. This turned out to be over-engineered as most sensors were utilized by one class at all. Thus, the functionProvider superclass was unnecessary and was deleted.

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

Besides the integrating issue mentioned before, other minor factors caused some delays. For example, the miscommunication among the team member caused the overwriting of work at the early stage of the development. The infrequent code update with GitHub caused overwrite of the code. A series of Standard Operating Procedures (SOP) were developed to solve this issue.

How did these affect the project progress?

With the major challenge from the integrating software design and all minor issues regarding miscommunication and standard operating procedures, the project progress was not greatly impacted in terms of scheduling and budgeting. However, these factors together impacted the performance of our design during the final demonstration. The system did not behave like a fully integrated system initially at the demonstration.

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

There are some other variations from our original idea like switching from polar coordination system to Cartesian system to accommodate the correction algorithm. Although there are some variations in our final design comparing to the initial idea, the final project closely followed the plan we had initially.

4. Budget

What constraints did the budget place on your team?

As a team of five the extra constraints on our budget forced our team to work as efficiently as possible. In order to minimize time wasted communication was key, as it was stressed that if a team member is stuck he/she should bring the issue to the attention of the team so as to resolve it as quickly as possible.

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

Our initial planning for resources in the first two weeks were relatively lax in order to leave spare budget for the software build and testing, however the build acted as a bottleneck on development, delaying testing as it took slightly longer than anticipated. While our initial design was robust and had multiple processes in parallel, software development was a necessary bottleneck.

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.

In the preliminary version of the **Gantt chart (version 1.0)** resources were not allocated to all project tasks as the capabilities of each member was not fully known, as well as the difficulty of each task. As such it was impossible to accurately predict the time required to complete a task. Tasks for the upcoming week were allocated, to aid scheduling. Another reason for not initially assigning tasks for the entire project was that due to the inevitable delays to be encountered in some subsystems different people will be available at different times and on different days to complete tasks, requiring reshuffling of resources.

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?

With extra budget I would have liked to put more time into testing. While we had predicted two weeks of budget for testing, we hadn't realized the amount of budget required for the actual build of the software system. That build depleted a portion of the extra budget we had been saving for slack time and extra testing.

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?

We were weak in resources during our preliminary software build as since the software required 4/5 developers we only had one primary tester for unit testing at this time. As such, given a larger budget I would have liked to have more team members working on initial unit testing during development, to minimize the time between development and improvements.

5. Process Contribution to Success of Project

Was the process useful in achieving the goals? How would you modify it?

By identifying the goals early, we were able to work towards something, and have an objective in mind at all times. Doing it again, we would focus on each problem equally. One of the reasons we did not succeed is because we thought some aspects of the design would be harder to implement than others, when in reality it was the opposite.

Which parts of the process were the most difficult to implement and why?

The software build part was the most difficult to implement, as unforeseen complications arose in integration. This is due to the fact that each team member was not equally adept at programming and also had different programming idiosyncrasies, so the adjustment to follow a standard was challenging.

How much time was devoted to testing? Was this at the subcomponent level?

22% of our team budget was devoted to testing, which is 58 hours of testing. There were tests for subcomponents of the system as well as the integration of the system. The subcomponent testing was key to getting the basic functionality to meet our design requirements.

Were the tests you designed sufficient?

With the allocated budget and a team comprising of only five members, the tests designed were sufficient for our process, since they covered every component of our system.

How much time did you estimate full prototype (i.e. integration) testing would take?

We estimated that full prototype testing would take a total of two weeks. In fact it did take two weeks, however ideally we would have had an extra week to test the full system. This is due to the fact that fixing failed tests took longer due to our constrained team budget, therefore further testing on the same integration system that failed was not possible.

How would you change your test design process to make it more effective?

I would organize a short meeting after each test with the software and hardware team in order to properly relay the outcome of each test, and what improvements could be made.

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

The beta demo demonstrated the ineffectiveness of our navigation. It proved that this component was not tested thoroughly enough. The impact of this on our design process was to focus on integration test involving navigation. As we were already in the validation and presentation phase, regression testing was executed, on navigation (bridge crossing) and integration tests that involved navigation for both green and red team.

6. Validation

What is your impression of how the robot performed?

On the last round of the demonstration, the robot performed exactly as expected, succeeding in localizing, navigating, mounting the zipline and arriving at the search area. The subsystems after that had a high failure rate (as seen in ***Test - Bridge Navigation*** and ***Test - Green Team Integration)***. The previous rounds proved to be affected by a bounding issue explained below.

Did the robot perform as you expected – i.e. if you wrote down what you thought it would do before the demonstration, did it meet or exceed these expectations?

The robot on its first 3 tries failed to execute a proper navigation which was highly unexpected. We had already faced this problem during the previous Beta Demo but the issues in our navigation were addressed and resolved after doing a regression test on the system. We had expected according to previous tests before the competition to fail other subsystems such as mounting the zipline, returning from capture and traversing the river. However before the last round of the final demonstration we had discovered the bug and the robot performed as expected.

If the robot failed (i.e. did not meet all the performance requirements), why did it fail?

After further investigation before the final round we found that we had set the dimensions of the playing field to be one tile larger on each side (13 by 13) than in reality (12 by 12) which explains why the robot missed the zipline by exactly one tile. The performance after capturing had been previously tested and was expected to fail (as seen in ***Test - Bridge Navigation*** and ***Test - Green Team Integration)***.

7. Conclusion

What did you learn from this course?

Over the course of this semester we learned the importance of formulating and following a formal design process. Moreover this meant we had to manage our time effectively and even moreso as we were a team of five with a smaller allocated budget. Furthermore, we understood the value of maintaining clear and unhampered communications which limited the number of miscommunications we had. However unavoidably we had not eliminated them completely.

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

The process helps to avoid confusion and miscommunication between team members as well as between developers and clients regarding the requirements. We have come to realize that this sort of confusion can be very costly as it often required to go back to the previous phase of the process. Furthermore, as mentioned earlier, the reduced budget means that we needed to leverage our division of labor to increase our overall productivity. In doing so all team members knew the set of tasks they needed to accomplish before a certain date and where held accountable for them.

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

Not only were the formal design skills applicable to any course with a design project but also any group work.

If so, what and why?(name the courses)

ECSE 456 & ECSE 457 would greatly benefit from our experience in 211 as the design project itself provides a more rigorous and technical design, hence its requirement of 211 as a prerequisite. As would ECSE 321, ECSE 223 as they both require group design work.

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

If given a second chance to the challenge we would define earlier on SOP (or standard operating procedures) and focus on the Navigation subsystem as it provided the greatest challenge as well as the most important link in the dependency chain. Furthermore, we would seek out further help from previous years. Very often advice and tips that we received could have helped us early on during the R&D phase (labs 1 through 5).