**Final Project Write-up**

*ThankGod Ofurum*

*Teammates: Franck Dosso, Yohannes Seifu*

*Computer Science Major*

*Bemidji State University*

*Data Structures*

INTRODUCTION

To design a program with efficient, bug-free execution, capable of withstanding of test-cases in the multitude and with enough functionality to meet users’ expectation in a given period of time, a group of individuals, with adept knowledge concerning the mechanics and the language of the program being designed and with sufficient dedication to making the program incomparable, is necessary. This was the very scenario depicted when the final project was awarded to team empire; this was a band consisting of night-owls, code-blooded control-freaks, and soda-drinkers, and with such attributes combined nothing was impossible to accomplish. However, even the greatest of empire crumble under the weight of distractions, poor communication, lack of motivation and, perhaps inevitably, lack of time, and unfortunately, this empire was not too different. From the greatest highs of meeting up early to make plans concerning the base project for the delivery program, to the bitter lows of failure to correctly understand, integrate and debug extensions, team empire pushed on till the end and made a decent attempt at presenting the program designed thus far. In all this, the most significant lesson to be learned is that whether you win or lose, you do this as a team, no scapegoats.

THE EARLY DAYS

To begin, the mastermind of team empire, and leader in my mind, Yohannes suggested that we met just a couple of days after the final project and groups were assigned: it was his understanding that it’s best to make hay while the sunshine, and was the sun shining ever brightly at that time. Our first meeting was in a back room at the library, where we began to troubleshoot the situation before us over a snack of Erbert & Gerberts. The first task was to figure out the functions that are to be integrated to form the overall structure of the delivery program. I knew that a sorting and selection function was mandatory to make the program as efficient as possible, and Franck further suggested that a find time function was needed in the selection process. We then reached a consensus that packages, clients, city, truck, and muscles functions were also to play a role in the overall structure of the entire program. With this accomplished our next struggle was with the structure of the city: we were unsure as to the way in which the streets and avenues actually operate, and so we decided to leave this as a question for professor Wolf upon our next appointment. Our focus then turned to the issue of mapping client objects for sender and receiver to the correct package, and at first, Yohannes suggested that we used the clients text file to hold package ids, but I pointed out that it’s far more efficient to use packages text file to hold both sender and receiver ids as packages share a one-to-one relationship with both sender and receiver. Franck then pointed out that to do this we must somehow integrate the client functions of getting receiver and sender into the packages class, a task Yohannes was all too prepared for. Following a few more comments we departed for the day and went on developing individual plans for enhancing the program.

THE GREAT CODE-REVOLUTION

Following our meeting with professor Wolf, in which I was unceremoniously teased for being the uninformed attendant aka that guy, the empire decided to reassemble, this time at a location beneath the dormitory building known as “Birch hall”, to plan the next course of action in tackling the ever-present threat of the final program. Once settled we shifted our attention from discussing the dynamics of the program to actually considering how the mechanics of the program would be made and who would be responsible for what part. I had always wanted an excuse to code an algorithm from the Algorithm’s textbook and, having the select package's function in my sight, I immediately volunteered to tackle the challenge of correctly implementing the function. I was also given the task of correctly implementing the packages and sort function whose algorithm, as Franck suggested, is the greedy algorithm in the Algorithm textbook, and Franck was appointed the task of subduing the find time, update data and parse packages functions, while Yohannes took on the read and write client and load in and deliver truck functions. We agreed to put forth a most earnest attempt to get these functions complete and bug-free before the due date for the base, 5th Dec 2016, and in the words of Yohannes, “If you get stuck, don’t hesitate to ask”. Thus began the great code-revolution.

PRESENTATION ONE: BY WATER

After days of programming and making unnecessary and time-consuming errors, such as designing a merge sort function for the program, I finally completed my part of the program, and I received feedback that fellow groupmates have also completed their part of the program, with some days to spare before the first presentation. However, then came the task of integrating each individual pieces into one fully functional and bug-free program. With neither I nor Franck being too familiar with pointers Yohannes stepped in to convert all objects to appropriate pointers and to integrate them into the one delivery program. While I was relieved to have a member of the team so knowledgeable about pointers, I was quite dismayed to find out that most of Yohannes’ coding and debugging was done with make files and sublime text, and this is because I had always used visual studio to debug my programs due to the convenience of seeing the variable modified before my eyes. Despite how much longer, in my mind, this process took to debug the program both I and Franck knew that there had to be acceptance for the team to be effective. Upon completing the integration and debugging of the base program, we had it presented to Marty the following day, and for all intents and purposes, the program seemed to be a success, despite the absence of Franck on that day. We then discussed with Marty concerning possible extensions to make the program more robust and user-friendly, and we reached an agreement of obtaining full points for correctly and robustly integrating multiple trucks to multiple parts of the city function, GUI for clients’ package entries and monitoring function, and package aging function.

THE DOWNTURN

Having accepted the terms and conditions set by Marty we immediately communicated these conditions to Franck and received a positive response, and I decided to implement the truck delivery system and the random clients and package generator. Our base program was quite flexible and all we had to do was implement three new functionalities to the base program and we were golden; how much easier could it get? However, it’s at this time that other classes began placing enormous demands on us: each member of the empire focused on completing final projects for their own respective subjects, and thus could not focus on implementing the final features of the delivery program. As stated, “a house divided cannot stand”, and this division kept us from finding time to integrate our features correctly and efficiently to the delivery program, though we did complete these feature.

PRESENTATION TWO: BY FIRE

After a frantic period of trying to find time to meet to integrate our features correctly to the final program, it was finally presentation day, and although the base project and parts of the extensions function as required, there was still room for much improvement given more time and less workload. However, the final program was complete and presented under the ever-observant eyes of Marty and our grade will be allocated as seen fit.

CONCLUSION

In conclusion, designing an efficient and bug-free program, capable of meeting the expectations of users, does require a group of adept and dedicated individuals. However, it also requires a high level of consistency until the program is fully functional and bug-free, and this equates to little to no distractions, high quality of communication and high levels of motivation, so that the empire remains alive and fully capable.