University of New Brunswick
Faculty of Computer Science

Course: CS2043 – Software Engineering I Deliverable #: _____7_____
Instructor: Natalie Webber Date: _____2017-04-04_____
Project group members:
Student #1: _____3413735_____ / _____Andrew Hampton_____
Student Number / Student Name
Student #2: _____35116474_____ / _____Shane Pelletier_____
Student Number / Student Name

Table of Contents

Original project schedule

Work Breakdown

Activity	Duration(hours)	Constraints
Create UML class diagram	1	-
Document UML diagram	1	Create UML class diagram finished
Create test plan	1	UML documentation finished
Code Battleship server BG game setup	3	UML documentation finished
Code Battleship server BG	5	Code Battleship server BG

game play		game setup finished
Mock up GUI	1	Document UML diagram finished
Test Battleship server and prepare report	3	Code Battleship server BG game play finished
GUI design document for customer	3	Mock up GUI finished
Text-based Battleship client	2	Test report finished
GUI design document for programmer	2	GUI design document for customer finished
Code graphical client	5	GUI design document for programmer finished
Add sound effects	2	GUI design document for programmer finished
Graphical polish	3	Code graphical client finished
Post-mortem commentary	2	Graphical polish finished

PERT

Firstly, the project team will create a UML class diagram of the Battleship server. Then, they will create a document explaining the UML diagram in more detail, and finally they will update their time sheet to account for the work done. This will culminate with the delivery of deliverable two to the customer. Next, the team will create a test plan for the Battleship server, code the Battleship server, and create a mock-up of the GUI for the battleship client. These three tasks will be performed concurrently since neither of them relies on any other task being completed before the task can be started. The coding of the Battleship server will take place in two steps: firstly, the game setup part of the BG protocol will be implemented; secondly, the game play part of the BG protocol will be implemented. Once the test plan and the coding of the Battleship server is completed, the Battleship server will be tested and a test report will be created. This culminates with the delivery of deliverables three and four. The time sheet will be updated immediately before any deliverable is delivered to the customer. While testing the Battleship server, the team will work on to writing the GUI design document for the customer, followed by writing the design document for future programmers. The time sheet will then be updated, and deliverable five will be delivered. The Battleship client will then be coded in two parts, beginning with implementing a text-based client with the user manually issuing server commands, which will be done at the same time as the GUI design for the customer, and developing into a graphical client that automatically issues server commands. At the same time, sound effects will be developed for inclusion in the game. Then, graphical polish such as explosion or splash animations will be added to finalize the Battleship game. The time sheet will be updated, and deliverable six will be delivered to the customer. Finally, a post-mortem commentary will be written that details how well the project team was able to complete their various tasks and how close their time estimates were to the actual time it took to perform their tasks.

Gantt Chart

1)UML Class diagram -> Document UML class diagram ->BG Game setup ->BG Game play-> Test plan -> Test Battleship and prepare report ->Text based battleShip client -> Code Graphical client -> sound ->commentary

1+1+3+5+1+3+2+5+2+2 = 25 hours

2)UML Class diagram -> Document UML class diagram -> BG Game setup -> BG game play -> Test plan -> Test Battleship and prepare report -> Text based battleShip client -> Code Graphical client ->Graphical polish -> commentary

1+1+3+5+1+5+1+3+2+5+3+2 = 26 hours

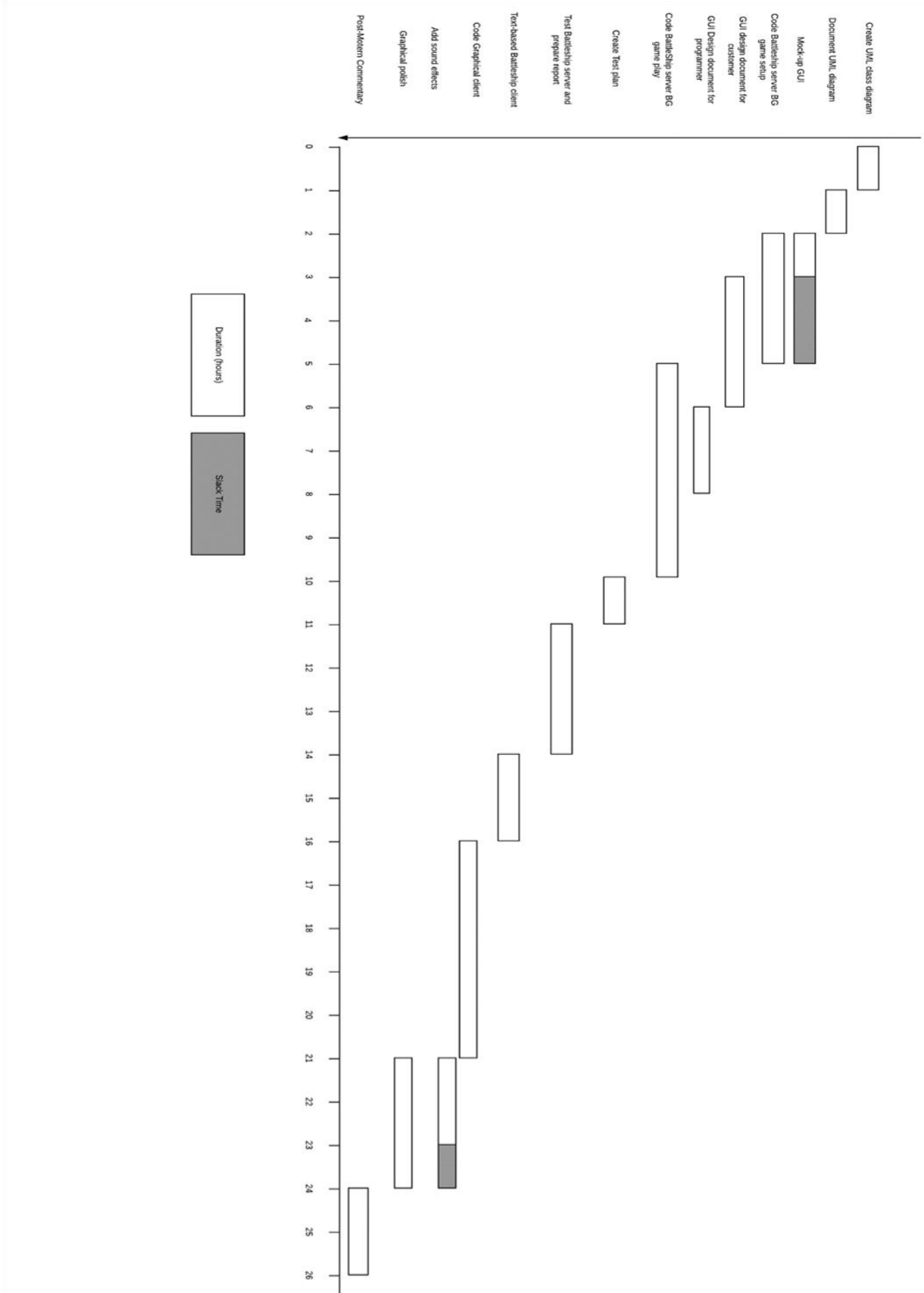
3)UML class diagram -> Document UML class diagram ->mock up GUL -> GUI design Customer ->GUI design programmer -> text based client -> Code graphical client -> graphical polish -> commentary

1+1+1+3+2+2+5+3+2= 20 hours

4)UML class diagram -> Document UML class diagram ->mock up GUI -> GUI design Customer ->GUI design programmer -> text based client ->sound ->commentary

1+1+1+3+2+2+5+2+2 = 19 hours

The second path is the critical path


Create UML class diagram
Document UML diagram
Mock-up GUI
Code Battleship server BG game setup
GUI design document for customer
GUI Design document for programmer
Code BattleShip server BG game play
Create Test plan
Test Battleship server and prepare report
Text-based Battleship client
Code Graphical client
Add sound effects
Graphical polish
Post-Mortem Commentary
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
Duration (hours)
Slack Time

Final time sheet

Activity	Start	Completion	Number hours Shane Pelletier	Number hours Andrew Hampton
Prepare project plan	2017-01-13	2017-01-24	1.5	1.5
Prepare UML diagram and documentation	2017-01-24	2017-02-02	1.5	1.5
Prepare Test plan	2017-02-02	2017-02-04	1.5	1.3
Battleship server implementation and unit testing	2017-02-14	2017-02-28	2	2
GUI documentation for customer and programmer	2017-03-09	2017-03-14	1	3
Client implementation	2017-03-14	2017-04-02	5	5
Post-mortem	2017-04-04	2017-04-04	0	3

Post-mortem

There are discrepancies between the original time sheet and the final time sheet. The notable is that on the original time sheet the server (the BG game setuo and BG gameplay) total esteemed time was 8 hours, while its actual time was 3 hours. So, it went 5 hours below the esteemed time. This can be explained by two main factors. The first is that work from other classes went up taking priority over the project. Second was that we did not meet up as much as we should have. We only met on Tuesdays and Thursdays for an hour. The effect was that not much work went into it.

The other major discrepancy was with the client. On the original time sheet, the esteemed time for the client was 5 hours. While on the final time sheet, it took about 8 hours. This discrepancy can be accounted on several factors. The first is that the server was still incomplete, so this meant that we were working on both the server and the client. The second factor was that we originally planned to use libgdx (a java framework) to do the client. But we ran into problems with the framework. This meant that we had to switch from it to coding it from scratch. Which led to time spent reviewing java gui. During the coding of the gui, there was problems with the implementation. For example, for the main

menu supposed to have a background with buttons on the top of the background. But the result was the buttons were below the background, through sometimes the connect button would appear on the background.

Furthermore, another problem that cropped up was that none of the gui components would appear, unless the frame was resized. We tried to solve these problems, but it was getting close to the deadline so we decided to move on. The next problem was there was difficulty in transition from the main menu to the setup phase. The problem was that the setup phase frame would only come up after the GameManager closed. This was solved rearranging the MainMenu code. The result is that a lot time was spent on trying to fix bugs, yet we were rushing through it because it was getting closer to the deadline.

If I should do a similar project, there are many things that could have been done differently. Two important things come to mind, first is better time management, and the second is better coordination. For the first point, we mostly got together on Tuesdays and Thursdays for about a 1 hour. Although, we sometimes got together for a longer timespan. But nevertheless, this left us without much time for things like bug testing. In additional, the bad time management led to lack in coordination, which is the next point. There was a lack of coordination between us. Meaning that we did not properly allocated different tasks between us. There were cases where one of us were working on something, while the other person was not doing anything.

Another that could have been done differently is focusing more design. We the design that we went with was the first idea we came up with. What we should've done was to spend more time on thinking through the design and coming up with different versions, because the design we used caused us a lot of problems. Furthermore, we did not focus on how structure the code, instead we just code as we went along. This means that there is a noticeable inconsistency in the code. There is inconsistency with how the jButtons were implemented. For example, in the main menu class, private classes were used to implement the event handling for the buttons. While in the gameplayGUI class another method was used instead.