Retrospective Write-up

Meeting Log:

The first official meeting for project 2 took place on Wednesday the 6th of March directly after EECS 448 lecture from 9:00 to 9:50 am in a LEEP2 study alcove on floor 2 with all members present. We began the meeting by discussing the minesweeper project we received (Team Rocket’s project). Specifically, we dove into obvious things that were missing from the requirements of project 1 and minimal requirements to finish the current project. We wrote out a list that consisted of: adding adjustable rows and columns, adding a help button that would give instructions on how to play, making flags not return when revealing recursively, making the win conditions that all mines must be flagged and spaces revealed or when all non-mine spaces are revealed, implementing “cheat” mode that would reveal the board and be able to return to the game on a whim, and adding a feature that would be decided at a later date. We then discussed a few of the features we might want to add: adventure mode, RPG elements, a leveling system, and power-ups. Finally, we each agreed to open the project and look over the code. This proved more difficult than anticipated as we had no idea of how to open the .vb files included in the project. Upon leaving, each of us said our priority would be opening those files and figuring out how to edit them.

Later on the evening of the initial meeting all members took part in a video chat via Discord from 8:30 pm to about 10:30 pm. Note that this was not a constant meeting with all members, more of a flow of members coming and going as they saw fit. The goals of this meeting were to discuss working on the project from home, how to open the .vb files, and which parts each of us would work on. We ensured each team member had the same IDE (Visual Studio 2017 version 15.9.8) and we attempted to find the minimally needed installed add-ons in order to open the .vb files from visual studio. Due to the diligence of our team, we discovered from the official Microsoft website that the type of files discovered in the minesweeper project were Windows Forms. This discovery was made after noticing the files had a common “designer” file associated with them. This meant, according to Microsoft, we needed to use Visual Studio. With this discovery we added the recommended .NET development package and Visual Basic pacakge to Visual Studio via the installer, created a new Windows Forms App, and added all of the .vb files to the project. With this, we were able to open the .vb files and edit them freely. We concluded the meeting here with much relief.

On 3/5/19 from 9:20 to 10:45 am in Spahr, a meeting of Jeff, Jon, and Thomas this time with a focus on planning what would be needed to communicate between the front end and back end of the project. The obvious elements like rows, columns, and mines were listed, however we didn’t want to alter the code heavily outside of refactoring unless our changes made sense so we decided to look at the code given to us. Promptly, we decided to move many more elements to the back end. This change on our part came after realizing that they had a random mix of elements scattered in the back end and front end that both accomplish the same task. In some cases, we noticed they had elements in a front end implementation when the structure for expansion was already in the back end. Thus, we decided to add cheat mode and guess position to the back end. The meeting then moved onto GUI interface and elements to be added. Upon finishing some white-boarded designs and playing with Visual Studio features, we concluded the meeting.

A smaller fourth meeting took place on 3/6/19 from 9:00 – 9:50 am with team members Jeff, Jon, and very briefly Benjamin. The primary topics of this meeting were finalizing communication between the front end and back end, as well as our feature. We agreed to discuss two different features: adventure mode and power-ups. Adventure mode would see difficulty increases as the player won the respective boards and power-ups would see the user gain special power-ups to use following wins on a respective board. These power-ups were to be:

1. Reveal adjacent mine cell, that hasn’t been revealed
2. Reveal not adjacent mine cell, that hasn't been revealed
3. Flag a mine cell, that hasn't been already flagged (any that hasn't been flagged)

In the end, we decided on power-ups as they seemed both more challenging to implement and more fun. After getting approval for our feature, we concluded the meeting.

Later that evening, 3/6/19, from 8:40 pm to about 12:30 am Jeff and Jon held a coop coding session. For the first few hours of the meeting we used Discord chat in addition to a Visual Studio add-on called Live Share - Preview. The feature is currently unfinished hence the preview at the end of the name. With this Microsoft created feature we were able to start coding together on the same project. This was very neat and made coding as a team very functional as we would be editing the identical file. After some time we decided to work separately but remain in voice chat. Due to internet issues and the work progressing in different directions the separate approach became much easier. After fixing, adding onto, and generally making the quality much nicer for the front end of the minesweeper project in Visual Basic Jeff called the meeting due to needing rest.

Workload Split:

The workload split for this project was decided early and was followed through. Benjamin and Thomas would focus on the back end, Jeff and Jon would focus on the front end. The workload split was hardly debated at all but did revolve around individual members’ strengths. From influencing design of the code, UI, features to add, direction to go in, general debugging, and moral support; each member contributed and made the final product a success. Large portions of code were done in groups but with spring break upon us, the team adapted very well to dividing the work. The break was utilized to nearly complete the project with regular commits on github and regular reviews of the changes by other members.

Benjamin primarily focused his efforts on the back end of the code, the C++. This included refactoring the code, implementing board resizing, cheat mode, flags not moving on recursive reveal, as well as many quality of life changes. These changes made things much easier for the front end and made the project seemingly run much smoother overall. While that would be on going throughout the project other contributions included: bug fixing, testing, adding images, revealing the board upon a win or loss, quality of life additions, code refactoring, code clean up, and documentation.

Jeff, as previously, began by voluntarily logging all meetings and being the primary writer of the retrospective write up. Upon opening the GUI/front end elements of the project in Visual Basic, Jeff’s interests were piqued. Thus, his efforts were focused on this portion of the code. Specifically, changing around variables to work in the front end communication with the back end, implementation of the help button/GUI of the help screen for how to play, cheat mode through the front end, a reset function for starting a new game, changing the win and loss screens, and a play again button. Other duties included refactoring code, finding bugs or quirks in the game, testing the game, aid with design elements, quality of life additions, clean up, and even more documentation.

Jon got going from the outright by figuring out how to open the visual basic .vb files (this was a large effort that will be discussed further). After outlining the project needs in a fantastic readme.md, he succeeded in refactoring a 6000 line file in the project (from Team Rocket) into an under 150 line file by making clever use of a for loop. With these discoveries Jon’s interest in the front end was also piqued and he decided to focus his efforts there. Once again, Jon showed his strength with coding on the front end by refactoring much of what was in the code already in much simpler ways. In addition, he added the control box (GUI element that would take user input for rows, columns, mines, etc), changed the look of the board, put text instead of icons for the surrounding mine numbers, implemented power-ups, and implemented a way to communicate between the front end and back end. Of course, he aided in bug fixes, testing the code, and quality of life changes as well.

Thomas, at home with C++, decided to take on the back end immediately. This included bug fixes, code refactoring, and all around ease of use changes.

Challenges:

Features that didn’t make the final product:

What we would have done differently:

Works Cited: