CMPT 276 Phase 2: Report

Group 14: Darrick Gunawan, Mahyar Sharafi Laleh, Hanxi Chen, Boyu Zhang

During phase 2 of our project, which takes place between Febuary 14 and March 17, we implemented our game that we designed in phase 1 of the project.

Overall Approach:

With all our team members being full time students with an ever changing schedule, we decided to use a modified version of the scrum method during the implementation phase. This was decided in our initial meeting for this phase, and was the best approach for this project as our vastly different schedules allows us to work on this project separately while still communicating with each other thoroughly. This method also allows us to adapt to change frequently if anything ever happens in our uncertain lives.

Just like the scrum method, we had frequent meetings, usually every 2 to 3 days to discuss what we had worked on, pick up new things to implement and share our knowledge regarding issues that may come up during implementation. We also have a backlog with a list of requirements for our project that is sorted by a priority value, depending on what was explicitly stated in the assignment requirements and what would make the project better.

One thing we did not do from the scrum method was have a scrum master, as each member of the team has varying skills and abilities that aids in the project differently. We also did not have demos, as we do not have a customer to show software increments to.

Adjustments and Modifications from Phase 1:

Overall, we tried to follow both the use cases and the UML we had from phase 1. While the use cases we wrote on phase 1 stayed the same, we had to change a couple minor things in the UML that makes more sense from an implementation standpoint.

One such minor change is removing weapons. We realised that the weapons are not actually used in the gameplay and implementing them would only bloat the code. As such, we have decided to remove that class entirely, along with the different types of weapons that was originally planned in the UML.

We also had some asthetic modifications from what was originally planned in phase 1 to make things more consistent. Such changes includes UI changes and how things look in the actual game compared to what we originally drew in phase 1. While the functionalities stay the same, we just changed a couple of the namings, such as trap/bomb into another type of enemy.

Management Process and Division of Roles:

Since our team consists of beginners with varying sets of skills, we do not have a strict responsibility that was assigned to each person. Due to our frequent meetings, we often pick up whatever was available in the backlog and work on that, along with often switching tasks if someone runs into an issue that another person can solve. This ensures that everyone gets the flexibility needed to do things efficiently.

Even though we do not have a strict responsibility, we did each originally assign a part of the UML diagram to implement in our first meeting for this phase. This makes it easier to section of initially, since we do not know what problems we might come accross in the future. Our initial responsibilities at the beginning are:

Darrick: Items and Barriers
Mahyar: Player and Enemies
Hanxi: Board/Cells and UI
Boyu: Game class

This original responsibility was chosen firstly by what we each worked on in phase 1 and secondly by what the game needed in the beginning. Of course, once we were initially done with the original implementation, we cross-checked and worked on each others' sections frequently, such as some minor and major bug fixes where it would only happen on certain devices, and sound, which was added in all of the classes.

External Libraries:

During the span of our projects, we used many external libraries that are available to us. For the base of our project, we used teh Abstract Window Toolkit (AWT) for the GUI of our project, being the base of every game states that we have. That goes hand in hand with the Swing Library, which we also used.

On top of that, we used io and imageio packages to read and write files. This is mainly used for our sprites, which are png files, along with our map, which is a txt file. We also used this as a background for our menu screens, as we needed to read a png for the background.

The two libraries mentioned above combined with the sound library was used for sound effects, reading wav files for all the different sound effects and music that played in the background.

Measures to Enhance Code Quality:

We took a decent amount of measures in order to have good code quality. This was important to us as we had to understand each others' works on top of our own due to our division of roles. As such, we first and foremost made sure that we followed the UML design as closely as possible. That way, each of our group members will always have a good and easy

reference to look at. On top of that, we made sure that all the code are organised, both the actual code and the files containing the code. We made sure that each classes follow a strict order, such as grouping all the setters and getters together. We also made sure that each file is in correct folders, following as closely to the UML as possible. Of course, we also have the staple code quality enhancement in programming, where we removed all unused variables and imports, along with commenting all our code.

Challenges:

The biggest challenge we faced was the learning process. Not only was this the first game all of us has created, this is also the first project we worked on using the required technologies. This means that we had to absorb alot of information in short notice, such as learning how to use MAVEN, AWT and even Java, as most of us were more comfortable in other programming languages prior to this project. While all of us have used git before, learning how to use gitlab as a group instead of individually was also a big challenge, as we had to solve merge conflicts and adhere to strict branching rules to make things more manageable, which was all very new to us.

Another problem we ran into was time constraints. Being full time students, each of us had trouble following strict timelines for each progress, as frequent and spontaneous issues would pop up regarding this project and other things in our lives. On top of that, arranging meetings that works for all of us are difficult, as we had varying class schedules on top of other things outside of school life that makes us only available at select times of day.