CMPT276 Phase2 - Report

During the phase2 part of this team project, each team member worked individually on 2-3 classes. We held a regular meeting every Wednesday to stay up to date on all the changes that were made and discussed what was working and what was not. This allowed us to keep better track of who was doing what. In addition to these regular meetings, we stayed in touch on discord where we would ask for help or share what was working well for us.

Continual Redesign

Given that none of our team members had ever worked with game design and programming before in addition to most of the team members being fairly new to using Java a lot of the design choices that we made in phase 1 had to be redesigned. The use of different modules also led to a continual redesigning of classes as the project went on.

Most of the classes that were designed in phase 1 were completely redone. This was because as we were programming we would realize that our first conception did not work practically. During our regular meetings, we would discuss how each of our classes would interact which meant that we would need to add methods or remove methods to interface properly with the classes that other team members were working on.

Management Process

The team management was done as a group effort. During our regular meetings, one of the team members would take the lead to steer the group meeting and ensure that everyone had a chance to talk about their code and the process. The way that we decided who did what just came up organically as we talked about what needed to be implemented. We divided up all of the classes into 4 groups so that every member had a few classes to work on. We decided to group the classes based on their connection so that one person would work on a superclass and its subclasses to ensure that everything worked well together.

This way of dividing up the work worked well for the most part but we should have had a little bit better communication between meetings to ensure that our classes would interface well together. It may have also been better to pick one person as team lead throughout this process so that they could check up on each member to ensure that the work was getting done on time.

External Libraries

For this project, no external libraries were used. For the GUI we used the JPanel classes which proved to be sufficient as we did not need a very feature-rich GUI. Additionally, for the drawing of the graphics, we used the built-in graphics classes for drawing all of the items such as the map and prizes. One big problem with this was that the graphics classes are quite difficult to work with and they are not very intuitive.

Code Quality

To ensure good code quality we added comments wherever needed including javadoc comments for the classes and methods. This allowed us to quickly understand the code written by other team members. In addition to this, we looked through code at our team meetings and discussed how to better implement various features.

Challenges

For Chenkun, the biggest challenge is how Character class to get others attribution appropriately. As Character class, it has to get some object’s position. Because of the separate class, it is hard to find an appropriate way to access that.

The other challenge for us in this project is designing the game. Due to inexperience in game design, the first version of the game design does not provide sufficient functionality for the game engine. For instance, the first design does not address the class for managing objects in the game. This class serves the purpose of checking the current status of all types of game objects. Also, the team needs to redesign current classes, such as adding extra attributes or methods.

Additionally, I was not able to get the character to update and redraw the map. This is a serious issue as the game is not playable at this stage. Even though most of the classes have been well written it has been very challenging to integrate them and get the game running correctly. I hope to rectify this during phase3 to ensure that the game will be playable without any errors.

As for Shenyu’s challenge in this program, the challenges are coming from 3 main parts. The multi-threads programming for the enemy. For the first thing, multi-thread programming is a different way of running the program with multiple threads running concurrently, but it also needs some kinds of communication with different threads. Even though the tutorial is helpful, it is still not enough to make me get this skill expertly. So I have to spend a lot of time doing that, while the time is too tight in this implementing phase.