Assignment 2 Reflection

Wan-Ling Chang
University of Auckland
wcha305@aucklanduni.ac.nz

I. Introduction

The goal of assignment 2 is to create a simple game from scratch named Bulls and Cows. This report is to summarize the critical reflection on how I learned to accomplish this game as well as the design and implementation for my system.

Throughout the whole journey of assignment 2, I did experience lots of ups and downs. The best thing is that I put what I have learned from this course into practice as much as I could, including basic coding skills, object-oriented programming, UML, exceptions, I/O package and generics. The report aims to reveal more details about how well I went as well as the challenge I have encountered in this practice.

II. Final Design

The first step of this assignment is to design a class diagram. I built up my game based on the helpful programming experience that I gained from a lab exercise called Farm Manager 1984. In my initial class diagram, I designed the inheritance hierarchy of game roles by creating a parent class named Role along with two child classes named player and Computer. In addition, I also created several object classes such as GameManager, Game, and Result as well as few enumerations to fulfil object-oriented programming.

However, during the feedback session, the teaching staff, Melissa, provided two productive pieces of feedback on my initial design. The first one is that if it is necessary to create my own packages: game and player. For this opinion, we reached an agreement if this design wouldn't complicate my coding job too much, I could keep it. And then I decided to stick with my plan to group all related classes in the same package for better maintainable code. The second feedback is meaningful and helpful while developing the hardest part of the game. The tutor recommended that I should create three different AI classes to fulfil all tasks instead of using only Computer class to provide all services. Also, she suggested that I can take EasyAI as a general computer to perform the services that the single-player Bulls and Cows and Wordle can offer. At the beginning, I tried to build up my first prototype with the initial design. But later I realized what the problem was. I had to write lots of if-else statements to check the kind of computer before invoking the corresponding method. This made my code less readable and badly structured. Therefore, I decided to modify my design in terms of the tutor's second feedback. I built a two-tier inheritance hierarchy instead. The first tier of inheritance is same as the old design. But Computer class becomes the parent class of the second tier that present common fields and methods that computer and AIs can do. It is also an abstract class. Three AI classes are the child classes of Computer to implement guessPlayerCode. By doing so, the rest of my coding became much easier and time efficient.

III. Achievement

By complete this game development, I had few notable achievements of learning that can be listed as follows:

A. Design and implement an objected-oriented programming

I have developed the understanding of object-oriented programming by learning how to translate abstract concept into code objects. By designed a class diagram, I have learned how to define object classes and build relationships between objects.

B. Exceptions Handling

I have implemented try-catch clause to catch checked and unchecked exceptions classes to stop any immediate crashes in Game, GameManager, and Computer classes. I also created my own exception named WordleFileNotFoundException to define a checked exception in the aim to handle a specific exception.

C. Use Java I/O package

I have developed the understanding of how to handle input and output with java.io. * Package. I used PrintWriter to save the game result to a text file in Game class and Scanner to read a text file in Computer class. Meanwhile, I used try-with resources with its AutoCloseable feature to catch I/O exception.

D. *Implement Generics, Stream API and Lambda Expression*

I have implemented generics in different ways in this assignment. I used List collection to store the result of each guess and iterated through it with for-loops to print out messages on a text file. I also applied stream API and Lambda Expression to filer candidates in Line 37 of MediumAI class.

E. Reuse the existing method

I have created few common methods for the purpose of less duplicated code. These methods can be found in Game, Computer and Game classes.

IV. Conclusion and Future Work

Even with few notable achievements in this assignment, there are still two issues that I couldn't cope with well. One is the game flow control. There was no way to quit a game earlier in my first prototype. I tried to rewrite the start method of GameManager for better structured code but in vain. Next time I will change to parsing command mechanism which presented in Farm Manager 1984. The other issue is about how to debug. In this assignment, I only did limit number of tests with incomplete plan as well as few unit tests by hardcode. So, there may still be few bugs hiding in my system. Hope that I can learn more about testing tools later and have another try in my final project.

No matter the good or bad part I have had in Assignment 2, they both brought new insights of programming to me. The best lesson for me from this practice is that never be afraid of unknown bugs or issues. What you can do is to take your pen and do some practice. To find someone who can play a real game with you or write a pseudocode of your ideas is a better way to find clues that can help sort out your problems. The more you practice, the better you get.