COMPSCI718 (Assignment 2)

Stephanie Morton 4643222
Computer Science
University of Auckland
Auckland, New Zealand
stephaniem0202@gmail.com

I. Reflection

This assignment has gone well. I have gained a deeper understanding of classes and how to use instances of classes vs. static classes to store information. I have also learnt how to group different methods and variables into classes of similar methods or data types. Module 6 to module 10 have been very helpful for my understanding of the methods I have used in this game.

The design is well integrated and utilizes flexible code that can change depending on user input. The classes have been structured based on what type of data are being used and how it is being used, with the exception of the static Utilities class. In this class a few different types of methods have been amalgamated for the sake of having a simpler class structure. I have also chosen to break down the guessing method for the Hard computer level. For this level the computer will make an ArrayList of possible guesses to choose from. The method for trimming this ArrayList has been separated into 5 methods based on different response results. This gives more options for developing a balanced difficulty system in future as the current criteria specified by the assignment for the medium level difficulty, does not offer a challenge that is, practically speaking, any harder than the easy level.

A few changes have been made to the initial class structure diagram.

1. The Computer classes (easy, medium, and hard) all extend the User abstract class. This is because the code was significantly different between the three classes that it did not warrant inheritance from one Computer class to another.
2. The Keyboard class has been expanded to include more methods involved in taking in user input, and an IO and Utilities class has been made. These helper classes are for extra static methods required for game play.

Testing was completed by running the program through the console.

One improvement that could be made is to utilize more libraries to reduce the amount of code, as these methods have been hand written the long way. However, given the time restrictions and lack of experience with the possible options available, there was not enough time to research more efficient methods. As such, more manual methods for this assignment were chosen.

II. Advanced configuration

I have implemented a maxNumberRounds and a codelength property in the GameEngine class that allow the user to change the number of rounds they may play and the length of the secret code they must guess. These can be changed in two ways:

1. During game play the player is asked to select their difficulty level. If the player selects ‘easy’ difficulty they are then asked if they would like to change any settings. The player may reply yes and follow the console prompts to change settings before they start.
2. Open the GameEngine class and change the two hardcoded properties maxNumberRounds and codeLength. Reply “no” when asked during game play if you want to change settings.

References

[1] https://www.baeldung.com/java-decimalformat
[2] https://docs.oracle.com/javase/7/docs/api/java/io/BufferedWriter.html
[3] https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/TimeUnit.html