



EECE 350 – Computer Networks

Section 3 – Dr. Rajai Nasser

Team 3:

Christian Wakim, 201901797

Majd Harake, 201902525

Paul Frangieh, 201902684

Application functionality:

To start, our application asks the user to enter his username and password to login, all new users are required to create an account so that they can login later on with it to save their progress in the game. This account would be saved in the server and different info about this account would be saved on it, such as the password, number of gold and silver coins, and the time and date of the last login of the user. After logging into the account, the user can now play the game in single player mode!

Concerning the game itself, at the first login of the day the user would earn 50 gold coins. The game itself is based on guessing a 4-digit secret number (distinct numbers) with fewest shots possible, where each guess costs 5 gold coins. However, the user can earn coins along the way where if his/her guess contains a number that is already present in the secret number, then the user would earn a silver coin, if the guessed number is also in the correct position in the secret number then the user would earn a gold coin. Each 10 silver coins can be exchanged for 1 gold coin. In that way of earning, the user could get hints that would help him guess the number. Guessing the secret number would earn the user 50 additional gold coins!

Code Structure

Our project was divided into 2 categories, both executing the same tasks. On one hand, we created a purely console-based game with login form. When executed, the terminal asks the user to enter a unique username and a password. The code will then search for the username in the database. If it is already there, the program checks if the password and username perfectly match. If not, the already entered credentials will save in the database based on a text file. Another built-in feature in the console-based program, if the user enters a valid username but with a wrong password, the program will ask for the correct password to be inputted.

Once the login/registration is done, the game can then run. We have also built the same program but on a GUI. The program is basically two classes. The first one is the login/sign up class which will read from the text file available in the same directory, or write in case the username isn't there. The second class is the one for the game. It contains a random number generator that will generate a 4-digit number (each digit can be any number from 0 to 9). It also contains a function called *checkGuess*() that will iterate over both the user's guess and the randomly-generated number to check whether we have correct numbers in correct or incorrect position. The program was built in a way to keep asking for guesses until the number of gold coins is less than 5 (which is the cost of one guess). Another

option was also included. If for any reason the user wants to stop playing, he can input “STOP” (in the console-based) or click on the STOP button to terminate the program and display the correct number.

```
Please login or register if you want to play 4-Digit Mania.

Enter your username: christian
Enter your password: 123
You logged in on 04/05/2020 18:33:45

-----
Welcome to 4-Digit Mania!
-----
The computer has generated a unique 4 digit number.
-----


Quick instructions: - You will start with 50 gold coins.
                   - Each guess will cost you 5 gold coins.
                   - You will need to input one 4-digit number, each digit being unique in each guess.
Each time you guess a number in its correct position, you will be awarded 1 gold coin.
If you guess a number but it is not in its correct position, you will be awarded 1 silver coin.
You will be awarded 50 gold coins if you guess the full number correctly. However, try not to run out of gold coins to play!
You have: 50 left to use

Please enter your guess:
1234
You have guessed 0 correct number(s) in correct position, and 1 correct number(s) in incorrect position.
You have received: Gold = 0 and Silver = 1
Gold = 45
Silver = 1
You have: 45 left to use

Please enter your guess:
```

This is basically what happens when the program is executed and the user inputs one guess.

Whereas on the other hand, for the GUI program, the two classes happen on two different windows (as shown below).



Login Form

4-DIGIT mANIA

USERNAME

PASSWORD

Login

Register

This is the login form we have created for this project. It contains the game's name as well as 2 JText Fields to input both username and password. If both username and password match, the game UI will appear.

Your guess

Gold 50

Silver

Correct position

Incorrect position

GUESS

Correct number

STOP

When the stop button is clicked, the correct number will be displayed and the game will end with a Message Dialog “You have stopped the game!”

Task Division:

Due to the current circumstances (not meeting up in person and the jammed schedule of quizzes and finals we had), we conducted several zoom meetings and worked hand in hand to finish the implementation of phase 2. Christian and Majd were concentrating more on the logic of the code of the game and its functionality and database, facing a lot of problems while debugging and setting the output to the console to observe the possible errors that we might face while running the code, whereas Paul was concentrating more on the GUI and the new syntax we had to deal with which is swing. Dr. Matta was of great help, were he pointed out to different errors we had and complications in our code and program and helped us in the debugging process during a zoom meeting we had.

Some references used

- Java Swing tutorial (Posted on Moodle)
- <http://java.sun.com/docs/books/tutorial/uiswing/index.html>
- <https://netbeans.org/features/java/swing.html>
- <https://stackoverflow.com/questions/2885173/how-do-i-create-a-file-and-write-to-it-in-java>
- <https://stackoverflow.com/questions/41325105/create-simple-login-based-on-txt-file>