American University of Beirut

Maroun Semaan Faculty of Engineering and Architecture
Department of Electrical and Computer Engineering
EECE 350 - Computer Networks - Section 1
Course Project - Spring 2019-2020
Gold-and-Silver Game Project

1 Overview

In the Gold-and-Silver game, the player has to guess a secret number of 4 *distinct* digits with as few guesses as possible. Each guess costs 5 gold coins, but after making the guess, the player earns a number of gold coins and/or silver coins based on how similar the guess is to the correct secret number. More precisely, if there is one digit in the guess which is also present in the secret number, the player gets a silver coin. If the digit appears in the same position, the player gets a gold coin. For example, if the secret number is 4813 and the guess was 1824, the player gets one gold coin for the digit 8 and two silver coins for the digits 1 and 4. The number of coins of each type gives the player partial information about the secret number, and this will allow him/her to eventually guess the entire number correctly. The player gets 50 gold coins when he/she guesses the secret number.

You have to build a client-server application which allows multiple users to play the Gold-and-Silver game. The application should including the following functions:

(a) All users need to login before being able to play the game.

(b) New users should be able to create accounts.

(c) The server maintains information about each account, including username, password, nickname, statistics, number of gold and silver coins, date and time of the last login, . . .

(d) At the first login of each day, every user gets 50 gold coins for free. If one day a user does not login, he/she will not get the 50 gold coins for that day.

(e) 10 silver coins can be exchanged for 1 gold coin.

(f) After a user logs in, he/she can play single player games or two-player games.

(g) In a two-player game, each player pays 500 gold coins to enter the game, and the winner takes 1000 gold coins. The winning criteria can be either “win with fewest guesses” or “fastest to guess the secret number”.

(h) (**Only required from groups with 4 members)** Players should be able to form tournaments with 8, 16 or 32 players. Each player pays 500 gold coins to enter the tournament, and the winner takes all.

For the above functions, you will need to develop your own protocol between the client and the server, and then implement it using a TCP client-server system, where the server must be multithreaded.

The user interface can be graphical or console-based, but must be clear and “friendly”. The above represent the minimum requirements. You are welcome to add features that you deem add value to the application, which may earn you extra credits. Additional features can include the ability to add friends, play with a friend, transfer coins between friends, chat with a friend, send a message to a friend . . .

2 Phases and Deliverables

2.1 Phase 1: Basic Components (due April 6)

In the first phase, you'll design and implement a basic client-server application with these minimal requirements:

- Working TCP client and multithreaded server that can communicate. The communication can be a request of a simple information and a corresponding reply.
- A short progress report (of at most 3 pages) explaining the design of the communication protocol (commands, sequence, error handling, etc.), in addition to tasks remaining, suggested schedule, and task division between group members.
- 20% of the project grade will be assigned to this phase.
- You will submit the code for the client and server along with the progress report (in PDF format) through Moodle.

2.2 Phase 2: Implementation of the Project (due May 2)

Here, you will complete the full project implementation. You'll be required to submit the following via Moodle:

- All source code that you developed to implement the client and server modules in addition to other accompanying files. For grading purposes, all files should be available to be able to run the code.
- If you use any open source or sample code from the Internet, it is required that you mention it explicitly in both the code documentation and the report.
- Your code should be well commented.
- A final report containing the following content:
  - For each student in the group: First Name, Last Name, AUB-ID, and AUBNET-ID.
  - A description of the functionality of your application (up to 1 page).
  - A brief overview of the structure of your code and a brief description of how to run the code (up to 1 page).
  - A description of the design decisions you made and a short justification for them.
  - A description of any additional and/or special features of your implementation. Bonus points will be given for interesting and innovative special features (up to 1 page).
  - Selected snapshots for your application, including client application design and various functionalities. Include several snapshots to highlight different functionalities.
  - An explanation of the role of each group member in the implementation of the project. That is, you have to indicate what tasks each member of the group worked on. Moreover, you have to indicate the percentage contribution of each group member to the full project development.
  - List of references with brief description how/where each reference was used.

This phase will also involve a demonstration. During the demo, each student will be asked about the design, code, implementation, application functionality, as well as his/her contribution. Any evident cheating or unusual commonality will jeopardize the project grade, and could result in disciplinary actions.

Grading

- Readability of the code: 3%
- Brief user manual: 2%
- Communication protocols design - client ⟷ server: 10%
- Functionality of client and server: 65%
- Presentation and answers during demo: 10%
- UI design and functionality: 10%

Questions or Clarifications

For any questions related to the project, contact Dr. Jad Matta on `jm97@aub.edu.lb`

Useful References

- JAVA docs:
  `http://java.sun.com/j2se/1.5.0/docs/api/`
- JAVA tutorial:
  `http://java.sun.com/docs/books/tutorial/`
- JAVA sockets:
  `http://java.sun.com/docs/books/tutorial/networking/sockets/`
- GUI design:
  `http://java.sun.com/docs/books/tutorial/uiswing/index.html`
- JDBC:
  `http://java.sun.com/docs/books/tutorial/jdbc/index.html`
- JAVA threads:
  `http://java.sun.com/docs/books/tutorial/essential/concurrency/`
- TCP Network Programming:
  `http://java.sun.com/docs/books/tutorial/networking/sockets/index.html`
- SQL Tutorial:
  `http://www.w3schools.com/sql`
- Netbeans IDE:
  `https://netbeans.org/features/java/swing.html`
- Window Builder:
  `https://eclipse.org/windowbuilder/`