

Networked Number Guessing Game with ASCII Art Integration

Developer: Toros Kutlu

Summary

The Networked Number Guessing Game with ASCII Art Integration is a compelling project that combines the classic number guessing game with modern networking capabilities and the creative use of ASCII art to enhance the user experience. The game operates over a network, allowing a client to connect to a server and engage in a guessing game, where the objective is to guess a randomly generated number within a certain range. The inclusion of ASCII art, such as question marks and trophies, adds a unique visual aspect to the game, making it more engaging and enjoyable for users.

Development Process

- **Initial Concept:** The project began with the idea of creating an interactive game that could be played over a network, enhancing the traditional number guessing game by allowing remote gameplay.
- **Planning:** The game's basic rules were established, including the range within which the number would be generated and the feedback provided to players. The decision to incorporate ASCII art was made to add a creative artful touch to the game's interface and user interactions.

- **Implementation:** Development of server and client components, ASCII art reader, and integration of art into the game. This included handling client connections, generating the random number, and managing game logic on the server side, and establishing connections and submitting guesses on the client side.
- **Testing:** The game was tested for functionality, ensuring that the network communication was reliable, the game logic was correct, and the ASCII art was displayed properly.
- **Finalization:** Final adjustments were made to the code, comments were added for clarity, and comprehensive documentation, including a README and Javadoc, was prepared to guide users and developers.

Code Structure

- ``Ascii_Art_Reader``: Responsible for reading ASCII art from text files and displaying it within the game, adding a visual dimension to messages and responses.
- ``Guess_Number_Client``: Acts as the game's client-side interface, enabling players to connect to the server, make guesses, and receive feedback.
- ``Guess_Number_Server``: The server-side component that manages the game logic, including generating the target number, evaluating guesses, and communicating with clients.

- ``Send_Welcome_Message``: Used to send a custom welcome message to clients upon connection, which may include ASCII art for a friendly and engaging start to the game session.

Pseudo-code during initial concept planning:

Server:

```
Initialize server on a specified port  
Wait for client connection  
  
Generate a random number between 1 and 100  
  
While true  
    Wait for guess from client  
    If guess is equal to the number  
        Send "Correct!" to client  
        Break loop  
    Else if guess is lower than the number  
        Send "Higher" to client  
    Else  
        Send "Lower" to client  
  
Close client connection  
Shut down server
```

Client:

```
Connect to server on the specified port

While true
    Send guess to server
    Receive response from server
    If response is "Correct!"
        Print "You guessed the number!"
        Exit loop
    Else
        Print response ("Higher" or "Lower")

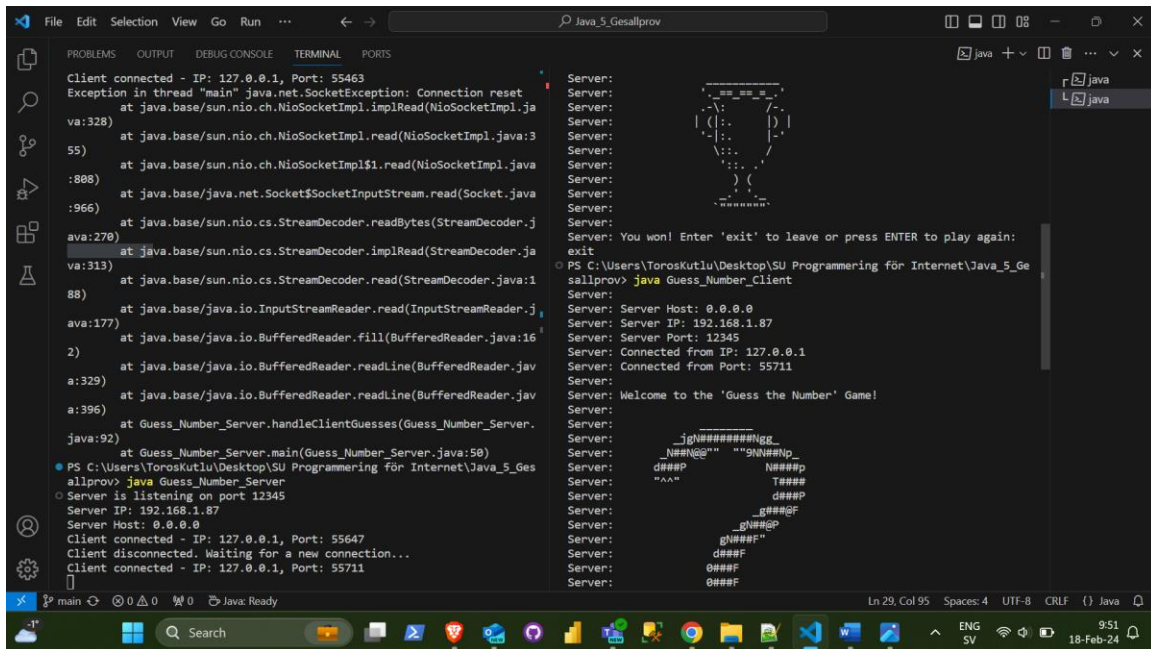
Close connection
```

Functionality

During gameplay, the server waits for client connections and initiates a new game session for each client. The client is then prompted to guess a number, with the server providing hints based on the accuracy of the guesses. ASCII art is used to enhance various game messages, such as the welcome message:

The game leverages simple yet effective networking and I/O operations to create an engaging and interactive experience.

The game is designed for Single Player use. However, multiple clients can connect and play in succession on the same server because the server waits and listens for new connections even after a client has disconnected:



```
Client connected - IP: 127.0.0.1, Port: 55463
Exception in thread "main" java.net.SocketException: Connection reset
    at java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:328)
    at java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:355)
    at java.base/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:888)
    at java.base/java.net.Socket$SocketInputStream.read(Socket.java:966)
    at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:270)
    at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:313)
    at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:188)
    at java.base/java.io.InputStreamReader.read(InputStreamReader.java:177)
    at java.base/java.io.BufferedReader.fill(BufferedReader.java:162)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:329)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:396)
    at Guess_Number_Server.handleClientGuesses(Guess_Number_Server.java:92)
    at Guess_Number_Server.main(Guess_Number_Server.java:50)
PS C:\Users\TorosKutlu\Desktop\SU Programmering för Internet\Java_5_Gesallprov> java Guess_Number_Server
Server is listening on port 12345
Server IP: 192.168.1.87
Server Host: 0.0.0.0
Client connected - IP: 127.0.0.1, Port: 55647
Client disconnected. Waiting for a new connection...
Client connected - IP: 127.0.0.1, Port: 55711

Server:
Server:
Server:
Server:
Server:
Server:
Server:
Server:
Server: You won! Enter 'exit' to leave or press ENTER to play again:
exit
PS C:\Users\TorosKutlu\Desktop\SU Programmering för Internet\Java_5_Gesallprov> java Guess_Number_Client
Server:
Server: Server Host: 0.0.0.0
Server: Server IP: 192.168.1.87
Server: Server Port: 12345
Server: Connected from IP: 127.0.0.1
Server: Connected from Port: 55711
Server:
Server: Welcome to the 'Guess the Number' Game!
Server:
Server:
Server: _jgN#####Ngg
Server: _N##N##Np
Server: d###P      T###
Server: "AA"      d###P
Server:          _gN##gP
Server:          gN##gP
Server: d###P
Server: d###F
Server: @###F
Server: @###F
```

Attachments:

- Please see README.md for instructions, details and credits.
- Pseudo-code in .txt
- ASCII art in .txt
- Screenshots from the gameplay

(Extra Bonus) Possible expansion / enhancements:

Extended Functionality and Creativity:

- **Scoring System:** Implement a comprehensive scoring system that includes high score tracking and a global leaderboard, motivating players to compete and improve their performance by showcasing top scores and player rankings.
- **Adjustable Difficulty Levels:** Allow players to choose hint details such as "much higher" or "slightly lower," to make the game more interactive, and/or the range of numbers to guess from, making the game easier or harder.
- **Multiplayer Support & Chat:** Enable multiple players to compete against each other & chat, either by taking turns or by playing simultaneously in separate sessions.

Technical Complexity:

- **Concurrency Handling:** Use multithreading to manage multiple concurrent player sessions, enhancing the scalability and responsiveness of your server.
- **Secure Communication:** Implement security features such as encryption to protect the data exchanged between the client and server, ensuring privacy and integrity.
- **Enhanced (Graphical) User Interface:** Improve the user interface for both the server and client, making the game more engaging and user-friendly.
- **Game Variants:** Introduce different game modes or variants, such as time-limited guesses or penalties for incorrect guesses, to add variety.
- **Session Reconnect:** Implement functionality that allows disconnected players to rejoin ongoing game sessions, enhancing user experience.