

Distributed Multiuser Texas Hold Em Poker Application

Issue Date: 20th October 2009
35%

Weighting:

Due Date: 4th December 2009
1/2 students per group

Groups:

Deliverables:

Each group must submit a zip file containing all source code, clearly commented. Each zip file should contain a readme file detailing the working of each part of the application as well as execution and compilation instructions.

Texas Hold- Em is the latest gaming craze sweeping the globe. This poker variant begins by dealing two hold cards to each player. Each player then decided to bet or fold their hand based on how good they believe their chance of winning the hand is. After this the dealer shows 3 common cards to all users, another round of betting begins, before a 4th card is shown to all remaining users. Another round of betting occurs before a final 5th card (called the river) is dealt and then a final round of betting.

You must design and implement a fully working distributed multiuser poker application. The Application will essentially consist of a Poker Server with a connected backend database, and a poker client.

If you're not sure of how to play the game of texas hold-em, you will find a wealth of information online, and huge number of free applications that you can download and play online for free with.

Client Application:

Users running the client application will connect to the server and be given the options to register a new user account or to login to an existing account. Once logged in users should have the option to add cash to their accounts or view their transaction history. They should also have the option to view available games and join any open game.

Server Application:

The server should maintain user accounts, host active games, deal cards and calculate winners.

Both applications should be console based; you may add GUI features if you managed to complete all of the required components.

A sample application design is outlined below, this is a suggested design and students are free to modify this design, to suit themselves.

Grading:

- Code design, structure, commenting and documentation
- Functionality

10%

- A basic Server/client that authenticates user logins
20%
- Multiple players viewing and joining open games
15%
- Play a full multi player game
20%
- Allow the server to manage multiple games at once
15%
- Extras
20%
 - Any additional features beyond the basic functionality
 - Detail any extra features in your documentation
 - Some advanced extra features are listed, any of these would be excellent

Poker Application Sample design

Client:

1. Once started should try to connect to main poker server
2. Once connected, the client application should give users the options to
 - a. Register a new user account (username and password)
 - b. Login to an existing account
3. Once a user has registered an account and entered their username and password to log on, the user should see their current balance and have their following options
 - a. View available games
 - b. Cashier
4. The cashier should give the user the options to add more cash and view their transaction history
5. Once a user can see the list of available games they should be able to join any open game
6. The fee for the game should be removed from the users account
7. The user should then join the game (ideally open a separate window) and receive their cards and bet till the game is over.
8. If they win they account should be updated with their winnings

Server:

1. Should accept connection from client programs
2. Check usernames and passwords before displaying user account details
3. The server should maintain a list of active games
4. Perhaps create a user account with admin access to create new games.
5. The server should run separate threads for each game.

Extras:

1. A more advanced version would allow a player to play several games at once. Each in separate window ideally

2. The server should check that the clients connecting are valid signed applications, ie. Add software signing and security
3. Add any GUI elements to make the software more useable.

Network Distributed Computing
M

Mark Cummins (ITB)