Homework 1: Games B03902124黃信元

Usage:

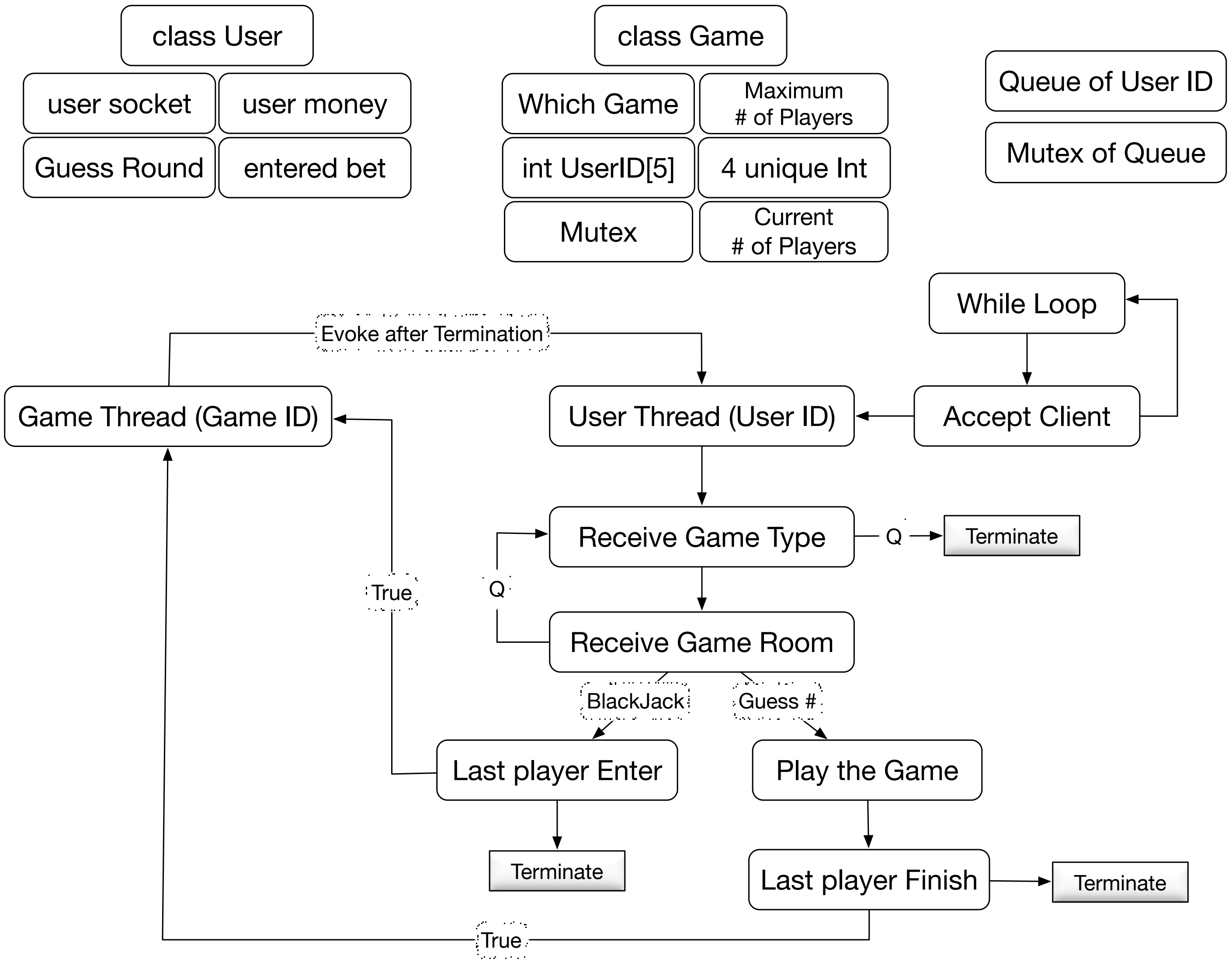
In the current directory,

1. type ***g++ -std=c++0x server.cpp -o server -lpthread*** to create **server**.
2. type ***g++ -std=c++0x client.cpp -o client -lpthread*** to create **client**.
3. type ***g++ -std=c++0x clientX.cpp -o clientX -lpthread*** to create **clientX**.
4. type ***./server <PORT_NUMBER>*** to run the server on a host.
5. type ***./client <HOSTNAME> <PORT_NUMBER>*** to connect to the server.
6. type ***./clientX <HOSTNAME> <PORT_NUMBER>*** to connect to the server.

client is the normal playing mode, while **clientX** is the AI version.
The AI will only help you during the game, and will not decide how many bets you enter or which game you want to play. (how many bet entered can change the gameplay strategy, for example, entering too many bet, you might not be able to double down in BlackJack)

Design:

A rough work flow for the server is shown in the figure below. (Workflow for client follows similar form as to coordinate with server, thus will not be discussed)



Some details on the design of the system and AI:

Design of server:

- When server is serving multiple clients, I use threads to handle multiplexing.
- I uses c++11 which includes a new feature of **<thread>**, it makes thread creating very easy. Creating thread is now much like calling a function, but runs in parallel.
- c++11 also includes another easy-to-use **<mutex>**, which is pretty much the same old mutex, but is very convenient. Mutex is very important in dealing with multiple clients, so the calculation for number of players in a room will not messed up.
- There are 2 main threads, User threads and Game threads. User threads terminate itself after calling Game thread. And Game thread calls several User threads before termination. This can cleanly separate the job of game menu selection and gameplay.
- Based on different gaming rule, the workflow for GuessThe# and BlackJack are very different. For GuessThe#, game is played in User thread since everyone plays independently. For BlackJack, all the players played together in one Game thread. Game thread for GuessThe# is for deciding who is the winner and the calculation of gaining.

Game Design:

- The gambling system for GuessThe# is designed so that each player may still earn some money when winning the game (Less than winning in a single-player room). But the one (may be more than one) that guessed in the least round can gain a lot more (depends on how many players participate in this game play, the more the more).
- Although BlackJack is somehow independent for each players, I design in a way such that player 1 plays first and then player 2, player 3, … etc. Playing in series creates more interaction, e.g. people can do cards counting (although I am using 4 decks together as a normal casino would do, so card counting is quite hard). The BlackJack rule follows US style, i.e. there is one hole card (one of dealer's cards is not shown), and all the player can see both cards of all other players (written on English Wikipedia).

AI Design:

- AI player for GuessThe#: I use brute force to eliminate impossible solution. And choose randomly from the possible ones. This simple AI player is actually very effective. It can guess the correct answer in about 6 rounds in average .
- AI player for BlackJack: I hand-coded the movement it should take when it encounters different situations. The move is based on the cheatsheet of BlackJack on the Internet. They stated that they have run millions of computer simulation to come up with this optimal strategy.