

CS171- Distributed Systems Course Project

For the course project, you have to build an application which can scale using Amazon EC2. We have received the credits from Amazon (100 USD per person), Certificate/key and guidelines on using that certificate/key are given below in resources section.

Your application could either be a game or a service(ex: dictionary, etc.) where multiple players or clients can join and play. Your application should be able to launch a new game instance on a separate Amazon machine instance on request.

Your front end should have functionalities like:

- 1) **Start a New game:** This launches an Amazon Machine Instance, install your game on it and start the game. You can assign an ID/name to each game instance.
- 2) **View Games:** You should be able to display which games are running in the system. Do not worry about fault tolerance. You can keep your main application (or first instance) on a CSIL machine and there you can maintain a state, which games are running on which machines and no machine will go down.
- 3) **View all Players:** view all players currently joined to the system. You should be able to query all games and list players joined to them.
- 4) **Join a game:** Player/Client can join a game by specifying the game ID/name. You can use a CSIL machine for running a client. Once a player joins a game, then player will only interact with the game until player quits.
- 5) **Quit a game:** A player can quit from a game he was playing and can join another game.
- 6) **Terminate the game:** This should terminate the game instance and shutdown the Amazon machine on which this application was running. Only the users connected to that machine can do that. If one player terminates a game then other players should know about it and terminate themselves.

Basic Game functionalities: Apart from the above mentioned functionalities, your game should have some sort of interaction (for ex: “moves” in case of chess).

This Project is slightly different from your Programming assignment 1 because here your clients don't have any information about other clients and all information is maintained on the host game to which clients are connected. Application will store the state information for clients (for ex: requests from client1/2/3...)

Resources:

- 1) TA has shared the folder with certificate and other keys at /cs/sandbox/student/shashank/CS171. Inside “CS171” folder there is a hidden folder “.ec2” you have to copy that “.ec2” folder on your machine. Then follow the instruction given on (<http://docs.amazonwebservices.com/AWSEC2/2009-03-01/GettingStartedGuide/>). Skip “setting up an account”.
- 2) Gaming resources:
 1. Board Games: "http://www.xahlee.org/PageTwo_dir/MathPrograms_dir/game.html"
 2. The Anatomy of a Game: Program "http://home.tiscalinet.ch/t_wolf/tw/misc/reversi/html/".
 3. GNU Chess Homepage: "<http://www.gnu.org/software/chess/>"
 4. Beowulf Homepage: "<http://www.chessbrain.net/beowulf/>".

5. glChess: "<http://glchess.sourceforge.net/>"
6. Chess Programming Theory: "<http://www.chessbrain.net/beowulf/theory.html>"
7. Computer Chess: "<http://www.netlib.org/utk/lsi/pcwLSI/text/node341.html>"
8. Stragus-an open source gaming engine: <http://stratagus.sourceforge.net/>
 1. Wargus is built on Stragus: <http://wargus.sourceforge.net/>
9. Other open source games:
http://en.wikipedia.org/wiki/List_of_open_source_games

Above mentioned Gaming resources may not be distributed but, the application you will create should be distributed and should support multiple players joining from different places.