

A Web Service Framework for Multi-User Gaming Environment

Aanthoniraj Amalanathan¹ Ravi Theja Y²

¹Assistant Professor ²M.Tech

^{1,2}School of Computing Science and Engineering, VIT University, Vellore, India

Abstract--A multiple client gaming service environment consists of a hosting server that is capable of allowing users interacting with others players who are running that game. Server provides the communication between host servers gaming server. Game play options are decided with type of game they wanted to play. The player management manages the players between the game roles. In this paper proposed an administration arranged construction modelling for enormous multiplayer online diversion and web administrations. This expands the adaptability and security of the framework.

I. INTRODUCTION

Streaming videos, shows in devices from YouTube, streaming websites become predominant way of enjoying. Like that only games are going to be a part of playing as a stream of video. The stream generators generate each and every event accordingly as game progresses to any device is a gaming service [4]. This is to provide freedom from console devices. So that a player can play anywhere without any consoles.

Any device gaming, click and play, less hassle (no setup process, no downloads etc.). By this service any one can play any game with all features and high end games. Gaming is provided as a service than fixed to pc's and console devices though there are many conflicts to be solved to provide this feature it can be attained in a smaller level at present.

The game is developed by a developer and his team and hosted in a server that is being used by a user with the medium internet as a service. If multiple users are involved then all the players are using that service and being hosted by a host server.

II. DESCRIPTION

An interactive gaming system consists mainly hosting server, capability to run that application, management programme.

A hosting server is controlled by a client i.e. client can host a game [6] and it is managed by management protocols.

A software is maintained for example sdk to run a particular application that interacts with clients.

Management programmes [6] are stored in a database (at the location of game). They provide different options like game mode or levels etc. These manage the operation of that level, initialized when any client hosts game mode respective programmes are initialized.

The system communicates with at least two clients through internet.

Games have some levels and their respective missions. A player will be upgraded according to his statistics and scores.

The players can upgrade by forming teams and sharing the items between them. For example wood for

constructing building houses in age of empires. Generally teams are formed in tournament type of gaming.

If a player playing in online then he must upgrade his level of expertise in every factor to defeat a player. Even though a player is excessively upgraded he cannot advance to next level until all finishes that level and if some games are hosted means only that level can be playable all other clients. But here a problem is there that if the hosted server crashes then all the players connected to it also lose the game in normal servers. So proxy servers link those nodes via other route in network then game can be continued if host is dead also others can play it.

In real time a request is sent from client and received by server. If sever has slots to enter into that game then sends a response to the client. Many clients are requesting for that service and they are accessed by server continuously.

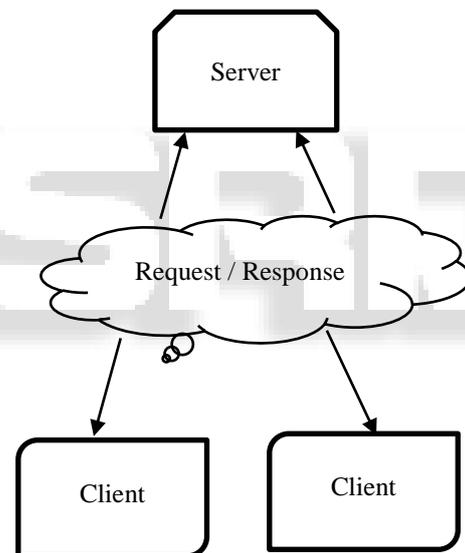


Fig. 1: Communication of System

III. LITERATURE SURVEY

A. Client Server Communication:

Some first individual shooter web diversions like Quake and counter strike regularly utilize the client server design. In this building a design a solo server is answerable for all clients.

The game environment is a collection of objects within that virtual world which can be controlled by the machine. At each point server is responsible for function invocation.

All copies of game states to be identical and there comes the synchronisation of servers. Clients are distributed to number of servers so burden is decreased on a server.

B. Peer to peer games:

A multiplayer game played over the network can be implemented using several different approaches, which can

be categorized into two groups: authoritative and non-authoritative.

In the authoritative group, the most common approach is the client-server architecture, where a central entity (the authoritative server) controls the whole game. Every client connected to the server constantly receives data, locally creating a representation of the game state. It's a bit like watching TV.

If a client performs an action, such as moving from one point to another, that information is sent to the server. The server checks whether the information is correct, then updates its game state. After that it propagates the information to all clients, so they can update their game state accordingly.

In the non-authoritative group, there is no central entity and every peer (game) controls its game state. In a peer-to-peer (P2P) approach, a peer sends data to all other peers and receives data from them, assuming that information is reliable and correct (cheating-free):

IV. ARCHITECTURE:

This diagram determines the process of game initialization with server to service [4].

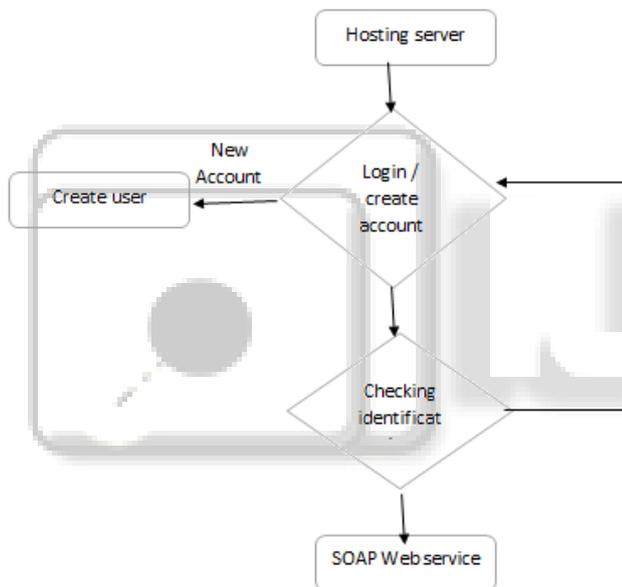


Fig. 2: Process of Initialization of Service

When any user want to play an online game then he connects to hosting server then it directs for a site or some medium to login that has two options login or create an account

If user has an account he can login with the account details or else create an account. Account is mandatory to play an online game. The login details are verified and then web service is initialized to provide the service i.e. gaming environment.

The web service gets the environment from server and user plays there. Here SOAP web service is considered though it is not effective as rest in terms of speed and access different data formats. It has point to point SSL for security and can maintain the data integrity which is crucial in some part of a service.

V. ALGORITHM:

Start

Step. 1 : Create a class to represent the Deck of cards.

Create card types as "Clubs", "Spades", "Diamonds", "Hearts" And CardValue as {"Ace", "King", "Queen", "Jack", "10", "9", "8", "7", "6", "5", "4", "3", "2"};

Step. 2 : Create a Shuffle method that takes all cards in the deck.

Create a temp card value for (inti=0; i< cardsInDeck; i++) to shuffle the cards randomly use function rand Index. Next Int(cardsInDeck) swap the cards

Step. 3 : Create a Draw method that takes the top card from the queue (which would be random).

Draw a random card Random seed = new Random ();
List of cards drawn is maintained to evaluate
List<Card>cardsDrawn = new ArrayList<Card>();
Draw card only once per set

Step. 4 : Validate the cards and calculate the points of all opponents.

Step. 5 : Decide the winner and end that set GO TO Step 2.

Step. 6 : End the game .

A. Algorithm Description:

A class card is created to store the card values and types of cards. Different types of sets are available so they are differentiated by colours.

A class shuffle is defined to shuffle the e sequence of sets. This is initialized after each and every game so that all cards are shuffled without any sequences.

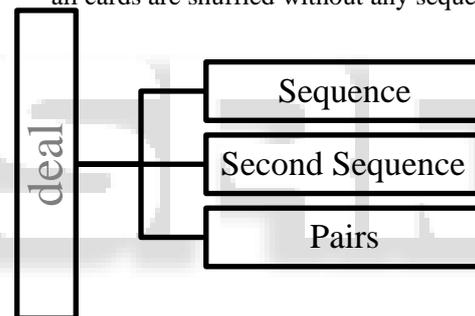


Fig. 3: deal rules to satisfy

Draw method is defined to draw a card from the deck. The sequence of drawing cards is done next to the person who distributed the cards. This is initialized for each and every person and has two types of taking the card from the deck or dropped card from opponent.

If a user kept a deal then his cards are validated according to the rules of the game. Whether he had sequence, supporting sequence, pairs or sequence. It is validated by comparing many probabilities after submission of cards. If he kept valid deal then he will get zero points and opponents get points based on their cards availability extra cards availability after sequences, pairs etc.

VI. DATA MANAGEMENT

The detail of every client has to be maintained in a server to provide security. Each and every algorithm[6] and its AI scripts are to be maintained so the concepts of artificial intelligence has scripts in such a way if anything goes wrong by some means these scripts take care of that.

The data is maintained up to a point so if any player disconnects because of some issue he can continue from that point but it is not possible in all conditions.

Load balancing:

Although AI provides many possibilities for a player it is too difficult to maintain in huge gaming environment because of load on those scripts. As players go on joining then nodes taking load continuously by their actions. If load increased on server then service providers cannot get the data on time then the game lags. The actions in the game differ at that time. So to avoid those conditions the client nodes are split into many parts and connected to different servers to main server.

VII. NETWORK APPROACH

To play any game through a network client and server system I common for all for any device. Most of them are connected to proxy server that acts as a server that communicates with the main server. The game is maintained in a database and it is being accessed through application service providers and these send the data to proxy servers. A server has a limit of clients so on basis of clients the proxy servers re maintained if limit is crossed then the server crashes.

A dedicated server simulates the virtual gaming world without any input or output. Players are to be connected to server with client programs.

Listening servers run as client system. They interact with remote players (other players).

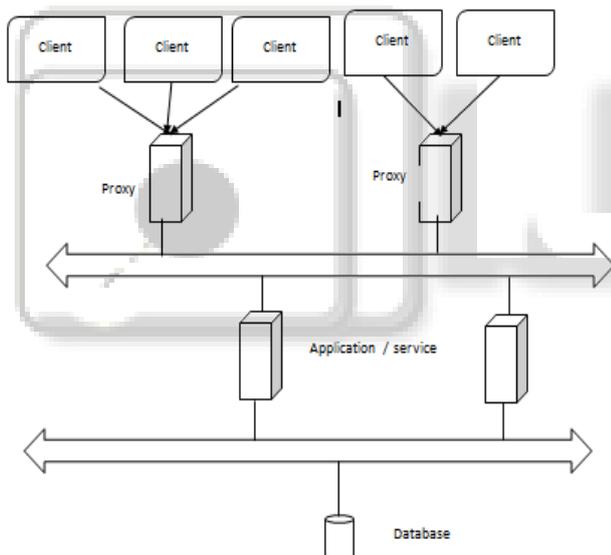


Fig. 4: network approach for gaming environment

VIII. CONNECTION LIMITATIONS

A. Hardware

These problems arise with the router or network adapter. If you are playing in your device without connecting to internet [9] then not many issues arise but for online gaming network medium is mandatory so connection problems with network addresses or port address etc. may arise.

B. Software Issues

The compatibility issues or firewall issues come under this category. If hosting client has older version of operating system then compatibility of both not matches so higher end one has to degrade it to lower compatibility.

In most systems firewall asks to allow the connection or not for security in network traffic. If it's ok for a player then he can allow that.

IX. CONCLUSION

In this paper we said that how gaming can be used as a service with device independency. The problems that may arise and how they are to be controlled with proxy server approach in mass gaming environment. The service example that helps in gaming environment is taken as an online card gaming system and how they implement it with artificial intelligence.

REFERENCES

- [1] <http://www.virtualizationpractice.com/gaming-as-a-service-21246/>
- [2] <http://math.hws.edu/javanotes/c5/s4.html>
- [3] "Massively Multiplayer Online Strategic Multipurpose Game" by VITALY BELKOVSKY
- [4] "Discovery service in a service-oriented gaming network environment" by Christopher W. Blackburn, Rory L. Block, Thomas A. Gentles, VikramSwamy, Terry D. Warkentin.
- [5] "Systems and methods for providing an interactive multiplayer story" by Henk B Rogers
- [6] "RCAT: A Scalable Architecture for Massively Multiuser Online Environments" by Thomas DE Beauvais, Arthur Valadares, Cristina V. Lopes Institute for Software Research and Department of Informatics, UC Irvine ftdebeauv, avaladar, lopesg@uci.edu
- [7] "Methods, devices, and systems for video gaming" by James Charles Vago, Mark Anthony Charles
- [8] "Internet Gambling: An Emerging Field of Research" by Christine reilly and Nathan smith.
- [9] "Internet Gambling: Past, Present and Future" By Robert T. Wood Department of Sociology, University of Lethbridge. Gambling Statistics and Research Paper – Information for Territorial Authorities by jarrod true
- [10] Self-efficacy in pathological gambling treatment outcome: development of a gambling abstinence self-efficacy scale (GASS) by Hodgins, David, Peden, Nicole, Makarchuk, Karyn