International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Implementation of Peer-To-Peer Architecture in MMORPGs

Rutvij Joshi[1], Dharmik Patel[2], Swapna Naik[3]

[1, 2, 3]Department of Computer Science Engineering, Shri Bhagubhai Mafatlal Polytechnic, Mumbai, Maharashtra, India

Abstract: ***More than a million gamers play MMORPG (Massive Multiplayer Online Role-Playing Game) daily. But most of them use Client Server Architecture. The idea of Peer-to-Peer architecture design doesn't board well for a MMORPG, so the main focus of this paper is to understand the working of Client-Server architecture for a MMORPG and create a design of a Peer-to-Peer architecture that can be implemented in a MMORPG without compromising many features which are provided by the Client Server architecture. This paper also focuses to identify and solve various problems that a Peer-to-Peer architecture can cause in a MMORPG such as cheating, network splitting during a game session and stable connection during a session.***

Keywords: Central Connection, Client Server Architecture, Massive Multiplayer Online Role Playing Game (MMORPG), Peer-to-Peer Architecture

1. Introduction

In today's world computer games are at the core of the computing industry. Over the past decade the complete meaning of games has been redefined, previously games never required hardcore functionalities, security management, fast performance like today, so game used to be generally single player, even multiplayer games used to be very simple as a 1958 game called Tennis for Two or 1972's Pong. But as the networking became more efficient, and the quality and performance of graphic cards increased, and faster 3d accelerator cards were introduced, the multiplayer games have become a core part of gaming industry.

In early 90's game become more open to exploration and considered more freedom of player rather than simply completing the given task or following a specific route. Some of these games were categorized as RPGs (Role-Play Games). During this period the simple multiplayer games that interacted with maybe a few dozen players at most were there, but as the hardware barriers and networking issues decreased Multiplayer games started to accommodate more players in a single session. Also, the freedom of a player was considered while making multiplayer games. Massive Multiplayer Online Role Playing Games feature large number of players who interact with each other generally in groups in an open game world where every player has their own goals which they need to complete in order to achieve a much larger objective.

MMORPGs have existed before 80's with great examples such as Maze-war (1974) and MUD1 (1978). In 1985 game called Island of Kesmai on CompuServe was released which was a rogue-like (pseudo-graphical) MUD. Before late 90's MMORPG term was not known to the world instead such games were referred as MUD (Multi-User Dungeon) games. They operated very much similar to MMORPGs but had less freedom when compared to the new titles in the early 21st century. The first fully graphical MURPG was Neverwinter Nights it was delivered through America Online in 1991.Over the decade we have watched the birth and growth of this genre of gaming and watched many changes over time. Unlike the older MUD games MMORPGs offer a much more freedom, pace and user interaction. Today some of the biggest games leading this genre are The Elder Scrolls Online, Dark Souls series, World of Warcraft, Eve online etc. But all of this games use Client-Server architecture.

This led us to wondering despite some of the advantages of Peer-to-Peer network over client server network why is it not used as an architecture in MMORPGs. In this paper, we are going to work on designing a Peer-to-Peer network for an MMORPG.

This paper is organized as follows: section 2 consists of Literature Survey of Client server architecture for MMORPG and Peer-to-Peer Ideas for MMORPG. Section 3 discusses the Client Server Architecture in MMORPG. Working of Client Server in MMORPG is explained in Section 4. Section 5 contains a general Peer-to-Peer design and its disadvantages for MMORPG. Section 6 presents a design for MMORPG based on peer-to-peer Network. Section 7 discusses the working of design of peer-to-Peer architecture on MMORPG. Section 8 consists of conclusion followed by the references.

2. Literature Survey

'Distributed Architecture for Massive Multiplayer Online Role Playing Games' by Marios Assiotis and Velin Tzanovpresents a centralized distributed architecture for MMORPGs and explains how client server architecture works and the shortcomings of Peer-to-Peer architecture. They have considered the architecture based on various elements like player, virtual world, non-playable characters etc.

'Hybrid Peer-to-Peer solution for MMORPGs' is a thesis written by Frode Voll Aasen and Tom-Christian Bjorolo Johannessen in which they worked on a hybrid peer-to-peer concept that reduce cost of operating an MMORPG, so that it allows smaller game developers to compete against major titles. They have also considered various Game examples such as World of Warcraft, EVE online, etc. to understand the concepts of an MMORPG.

'Distributing game instances in a hybrid client-server/Peer-to-Peer system to support MMORPG playability' discusses

gaming aware distribution mechanisms needed to distribute game instances among servers and avoiding load imbalances that affect performance negatively. In this work, they work on tackling the problem of distribution and scalability of a game by use of a hybrid Client-Server/peer-to-peer architecture that scales dynamically as per the demand.

We also referred to an article on the website howstuffworks.com by Tracy Wilson for understanding the client server interaction works in MMORPG and what roles do a client and server have.

Other documents that we referred were Eric Cronin, Burton Filstrup, Anthony R. Kurc and Sugih Jamin's 'distributed multiplayer game server system'. We also referred various Wikipedia pages related to MMOs and MMORPGs games and several other sites to study various concepts about MMO games and to learn the history of MMORPGs.

3. Client Server Architecture

As we mentioned earlier Peer-to-Peer network is mostly obsolete in MMORPG's and Client server architecture is widely used to design MMORPG architecture. Client is a combination of playing area in a window which display's game's world. Every client has almost everything about the Game world such as maps of world, model files, data files, etc. And they are stored in client's computers. Every client has all these files stored separately in their computers. Algorithms and other rules are used to link these files and put together a meaningful game world which a character in game can travel.

In a MMORPG, a server typically isn't used in singular form, it generally refers to a group of servers which are divided into various regions across the real world. Games in general divide servers as per player population to reduce traffic and load on a single server. The divisions are generally done as per zones or continents or for really big games harbouring millions of players per week by countries. They are generally known as world or realms. These realms/ worlds are physically, and virtually similar identical, but each have different players than other realms who don't communicate with other players of different realms.
Server has various roles to perform. Some of these are as follows:

- Check and compare the location of a player character with another NPC's (Non-Playable Character) or other players.
- Calculating character's range or if a player is attacked.
- Notifying a client if he/she is being attacked.
- Calculating the success rate of your attacks, and notifying the results of it to the client.
- Notify the player when damage is taken of dealt, or healing is done to self or other players.
- Determining when loot is picked up by client and notifying the client or other clients about it.
- Detecting any cheating behaviour within players and taking necessary actions against it.
  There are various types of servers used in an MMORPG, as one server can't bear the load of all the operations itself and it is more organized to operate and update.

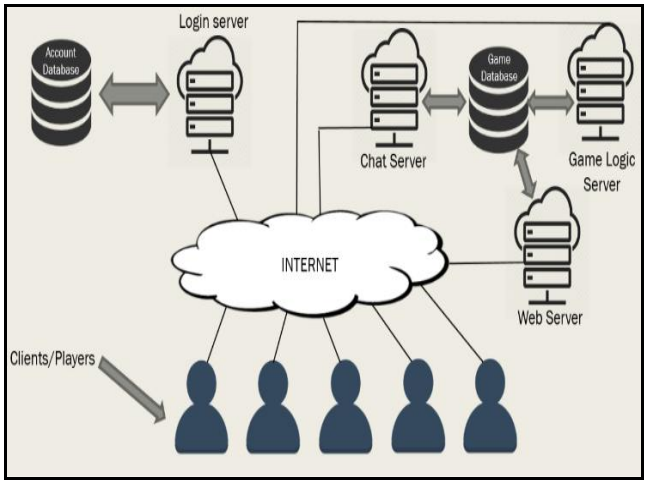

Figure 1: Basic Client Server Architecture for MMORPG

- **Login Server**
  It allows players to login to the account and access the game's world. They can also allow player to join a specific realm.
- **Chat Server:**
  It receives and transmits all the texts/audio players use to communicate with one another.
- **Web Server:**
  This server allows players to access their account information and other web services (if any used by game). They are also used to access and display clients inventory items. To do this game data-base have to be accessed by the web server.
- **Game Logic Server**
  Games have this server to perform necessary calculations related to gameplay and physics of the game. Some games also use this server to monitor cheating [6].

4. Working of Basic Client Server Architecture in MMORPG

When player approaches an object, the players' client informs the server about his/her location, and the server tells the clients which object is nearby. The client accesses files relating to the objects design and physics, which are stored in his/her computer's hard drive.

1) When the player interacts with the object, the player's client sends a message to the server, informing it of the interaction. The server transmits that information to the rest of the connected clients.
2) The clients inform the server of what actions they take. The server calculates result of the interaction and informs each client.
3) The objects artificial intelligence (AI) protocols determine how it behaves during the interaction, and the server transmits that information to the clients. If the player was successful, the server informs each client to display the success message (animation, text message, etc.), which is stored on every clients' computer's hard drive.
4) The voice or text chat (whichever is available by the game) happens continuously. It doesn't affect the gameplay nor does the gameplay affect the chat as chats are generally maintained using servers dedicated for chatting through which a message is transmitted[7].

5. Peer-To-Peer Network

When we talk about peer-to-peer networks the first thing that comes to mind is that we get good data distribution among all connected peers and there is no need of a central server which makes it cheaper to implement. But when it comes to multiplayer games and especially MMORPGs these advantages of Peer-to-Peer become its cons. Because without a central authority to monitor the overall game; it will be very difficult to prevent cheating. Also in Peer-to-Peer the internet connection of one pc can affect others as well, which is not the case in client server architecture.Peer-to-Peer architecture can be divided in two categories pure and hybrid. Pure Peer-to-Peer networks have no central server; all the files are stored by the clients in the network.

Using pure peer-to-peer solution in an MMORPG would cause three major problems:

- The game network could risk getting divided into smaller networks, which can cause loss of game data and game state.
- There would be no main connection point for PCs and hence haveno stable way of connecting to the game.
- The game would have no central authority, which can lead to cheating in a game.

In a hybrid Peer-to-Peer network generally there is a client who acts as a server or one sever which connects a group of peers to another group of peers. But the goal is to avoid the use of servers completely, so we aregoing to propose mechanisms which will be available within game data of all the clients. These mechanisms will help monitor and reduce various issues such as cheating, connection maintenance, network splitting,etc.[1] [2].

6. Designing A Peer-To-Peer Architecture for MMORPGs

While designing a peer-to-peer architecture for an MMORPG we are going to focus on three main aspects.

a) Avoid network splitting.
b) Reducing cheating.
c) Maintaining a Stable Connection.

Before we work on reducingcheating,we have to make sure that the network in which the players are connected doesn't break/split. In a peer-to-peer network it will be very hard to keep a count of all the connected peers unless we configure a peer in a network to act as a server. And without the track of all the connected computers in a network, a time lag or disturbance in the network can cause break in the link between two peers.Network splitting can be defined as disconnection of a node from a network. A split between two nodes is capable of splitting the network into two parts. Now in a MMORPG which has a peer-to-peer architecture network, splitting can cause problems such as loss of game data or disruption of the current game state.

6.1. Central Connection

Central connection (CC) is a group of twoalgorithms which resides along with other game data in each client's computer. The following two algorithms make a central connection:

- CC Assigning Algorithm
- Cheating Reduction Algorithm

These algorithms in a peer's PC will become active making them the central connection in the network.

We are going to use this Central Connection concept to keep track of all the connected peers in the session and reduce cheating. Also, the Central Connection will have the ability to appoint the new central connection in-case the current Central Connection has to leave the session. Initially when a gaming session will start the host will become the Central Connection of the game.

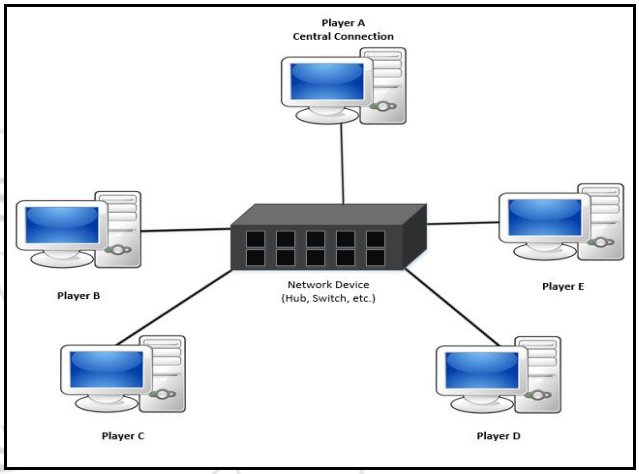

Figure 2: Peer-to-Peer Architecture for MMORPG

6.2. Connection to A Game Session:

Now before discussing more about how the Central Connection will work on cheating reduction, we need to first establish a stable connection in a network where in a particular session every peer will know about all the connected peers in that session. Doing this will also help inreducing the threat of network splitting. In-order to do this, every peer will maintain a Player-tracking table which will contain the list of player ID, IP address and other required details of all the connected peers in the session.

Maintenance of the Player-Tracking Table:
To make sure that all the peers contain the same data in their tables, after particular intervals of time a broadcasting process can be repeated in order to keep the Player-tracking tables up-to-date.

Broadcasting Process
The peers will send a broadcast of their game details along with their IP address and Player ID in the network and update their own table using these details.All peers in that session will accept them. A Tracking Algorithm will be used by the peers to handle the data received from the broadcast.

```
Algorithm 1. Tracking Algorithm
1:  Store the Player ID, IP address and other details received
    from the broadcast at a temporary location.
2:  Traverse the Player-tracking table.
3:  if Player ID at temporary location is already there in the
    table;then
4:  if the peer's ID, IP address and details are already in the
    table; then
5:      No action will be taken
6:  elseif the Player details are there but they are outdated;
    then
7:      Update the Table with the details from the Tempo-
        rary storage.
8:  elseif Player ID is not available in the table;then
9:      Add at Data from temporary storage to Player-
        tracking table.
10: endif
11: Return
```

Algorithm 1 will also help when any peer wants to check the list of all the connected players in their game. When a player leaves the game session, before leaving the session they will broadcast their ID. The other peers will accept that information and will delete all details linked with that player ID from their Player-tracking table. After broadcasting the information, the peer will remove its own details from their own Player-tracking tablethe peers won't be considered disconnected from the network. This can reduce the risk of network spitting.

Table 1: Player-tracking table maintained by a peer

Player ID	IP Address	Player Level	CC Rank
Player A	192.168.0.0	56	0
Player B	192.168.0.10	34	0
Player C	192.168.0.25	28	0

The Player-tracking table will also maintain a CC Rank attribute. Whenever a Peer is added to the table, the CC rank of that peer table will be initialized to zero. This attribute will be used by the CC Assigning Algorithm while determining a new Central Connection.The data fields in the Player-Tracking table may vary as required by game developers.

6.3. Central Connection Processes:

A. CC Assigning Algorithm

Now after a stable connection is established we need to make sure that a Central Connection is always present in a games session.So, if a peer who is the current Central Connection decides to leave the session or is voted to be kicked out of the session, before leaving the session it is essential that a new Central Connection is appointed. The CC Assigning Algorithm will help in doing so.The CC Assigning Algorithm of the current Central Connection will select and appoint a player from the localPlayer-tracking table as the new Central Connection; that player's computer will activate their Central Connection Algorithms and become the new central connection of the session.

This Algorithm will use the player's details the Player-tracking table to select the new Central connection. Let's consider that Table 1 is the Player-tracking table in the peer's computer. So, from Table 1 the CC Assigning Algorithm can select the new central Connection for player with highest level. Any other game data present in the table can be used to appoint new Central Connection. The CC Assigning Algorithm will select the player who has the highest value of a specific predicated column as the new Central Connection. But this method has one flaw. It is common notion that peer who is Central Connection should have a high-end pc because the Central Connection will have extra algorithms working for Peer tracking and Cheat reduction. If any game data, such as player level or rank is used to select the Central Connection there is no way to make sure that the peer appointed as Central Connection will have a high-end pc to handle all the extra work load. So, in order to make sure that appointed pc will have a more performance than other pcs, instead of using any game detail (player rank, experience points, level, etc.) to select the new Central Connection, we can use their device details such as RAM, GRAM, processor speed, etc. But any single detail can't be used we need to consider multiple details to find a peer with an optimum PC. So, we need to change the attributes of the Player-tracking table accordingly.

Table 2: Player-tracking table with Peers PC details, convenient for CC Assigning Algorithm

Player ID	IP Address	RAM (in GB)	GPU Speed	GRAM (in GB)	CC RANK
Player A	192.168.0.0	8	1.3 GHz	1.5	0
Player B	192.168.0.4	16	2.7GHz	2	0
Player C	192.168.0.2	6	2.3GHz	2	0
Player D	192.168.0.3	4	1.8GHz	1	0

Since multiple attributes are to be considered the CC Assigning Algorithm will help in deciding which peer is the most eligible for becoming the Central Connection.

```
Algorithm2. CC Assigning Algorithm:
1:  Set N = Total no. of connected peers.
2:  A = Name of attribute in table
3:  set I = 0
4:  while (I ≠ N)
5:      Set K = 0
6:      while (K ≠ N)
7:          If (A[K] ≤ A[I])
8:              CC_Rank = CC_Rank + 1
9:          endif
10:         K = K + 1
11:     endwhile
12:     I = I + 1
13: endwhile
14: return
```

Algorithm 2 will be used on all the attributes containing peers- device details. So, at the end a CC Rank column will be available. This column will contain numbers. Every peer will have a certain CC Rank number. The peer with the highest CC Rank number is the most eligible for becoming the Central Connection. A Selection Algorithm will be used to find the peer with the highest CC Rank number. The current Central connection will send a request to that peer using their IP address which is available in the Player-tracking

table. The peer will then accept the request by activating their CC Algorithms. The new Central Connection will send an acknowledgment to the peer who was the previous Central Connection allowing that peer to leave the game session.

B. Cheating Reduction Process

Cheating is a major problem when considering designing a peer-to-peer architecture in a MMORPG. In a client server architecture either the game logic server or dedicated check servers are used to monitor cheating. Since a peer-to-peer doesn't use a server, it becomes difficult to avoid cheating. But still the Central Connection that we used to solve network splitting and establish a stable connection can be used to monitor cheating during the game session. Though the problem with this will be that the Central Connection is a peer who itself is playing the game, so this isn't an efficient way to overcome cheating. So instead of using a one peer to monitor the cheating we can use multiple or all the peers connected in a session and establish a voting system. Here all the peers can send their calculation about a particular action made by a peer to the Central Connection and the Central Connection will compare the data sent by all the peers and estimate whether a specific player cheated or otherwise.

Cheat Reduction Process will have two parts. The first part will work on all the peers for checking if a particular player is cheating. When we say 'all the peers' it means huge data load in the network, this can make a game session slow. So instead of all the peers checking for cheating, there can be only a particular number of peers in a session like a panel who will monitor cheating. This panel can be selected using the CC Rank from the Player-tracking table. This panel will consist of peers with the highest CC rank, the number of peers in a panel during a session can be set using the following equation:

$$P = I * \frac{N}{100} \quad (1)$$

Where P = Number of peers in a Panel, I = Percentage of peers required in the panel and N = Total number of Peers in a session.

If the peers in the panel will check each and every action made by all the peers in the session, it will cost a huge data load on their own game. So instead of checking all actions for cheating we can tag certain crucial action in the game which can be tampered with, as cheating prone actions. Whenever a player in a session does any such cheating prone action, then only the panel will check for cheating. When any such action occurs in a session every peer in the panel will compare the result of that action with the possible results available in their own local game files and after that they will set a value for a cheat point. This will be a binary value, either '1' or '0'. '1' indicates cheating is detected, '0' indicates otherwise. All peer in the panel will send their cheat point to the central connection for the next stage of evaluation. Now even the central connection is a part of the panel so the central connection won't send their cheat point in the network. Also, the central connection will store the suspected player's IP on a temporary location.

The second part will be the Algorithm that will work only on the Central Connection. The Central connection will analyse all the Cheat Points sent by the peers in the panel and add them to get a Cheat Point Total which it will store at a temporary location and compare it with a predefinedestimated value to select whether the given peer is cheating. The Central Connection will also store suspectedvariable; it will be used to store the final decision of the cheating algorithm. The predefined value for comparison can be calculated using the following formula:

$$PV = \left(\frac{P}{2}\right) + 1 \quad (2)$$

Where PV = predefined value, P = Number of peers in the panel. Floating point value will not be considered.

Algorithm 3. Cheat Reduction Algorithm

```
1:  if Cheat Point Total ≥ PV
2:      Set Suspected Player = cheater
3:  else
4:      Set Suspected Player = innocent
5:  endif
6:  if Suspected Player = Cheater
7:      Kick out the player using the stored IP
8:      & Reset all the cheating variable.
9:  else
10:     Reset all the cheating variables.
11: endif
12: return
```

After a complete cheat process is done all the panel members will reset their cheat variables. This method of cheat reduction can be useful but the main problem will be its impact on data load. Even though all the peers in the panel will have high end PC's then the normal peers, still when searching for action results in their local game files will create some lag. Also, the transmission of all the cheat checking data in the network can cause decrease in the bandwidth. Even though this method will reduce the chances of cheating to a certain amount, it won't be possible to completely remove cheating or take steps against it. So here a hybrid peer-to-peer architecture can be used which can include one server dedicated for cheating reduction.

7. Working of Peer-To-Peer Architecture in MMORPGs

When a player approaches an object in the game, since there isn't a server the peer will broadcast the object and its interaction details to the other peers in the session. No server will be there to do the calculations. All the calculations will be done by the players themselves.

1) When the player interacts with an object, he/she will send the object details along with the interaction details to all the peers in the session using a broadcast or multicast whichever the game uses as the transmission method.
2) Using these object and interaction details all the players will locate the required files in their own PC's.
3) Once the interaction is complete, the player will send a completion message to all the peers in the session indicating the interaction is complete.
4) Since there are no check servers to avoid cheating, certain objects and interactions will be tagged as cheating prone.

5) Whenever a player accesses the objects or indulges in the interactions which are tagged as cheating prone, the cheating reduction process will simultaneously work along with the regular processes and calculations to avoid cheating.
6) If cheating is detected appropriate actions will be taken as specified in the cheating reduction process.

This is our idea on how peer-to-peer architecture will work on a MMORPG. But one problem that this design will face is that the data load can cause some amount of lag on the peers'PC's. So, some data load scheme will have to be design to overcome this problem.

8. Conclusion and Future Work

This paper offers a peer-to-peer architecture for an MMORPG. A peer-to-peer architecture will help in reducing the restrictions imposed by the servers to the players. We introduced the concept of Central Connection which will be the most essential part of peer-to-peer network. This paper works on reducing network splitting, cheating reduction and maintaining a stable connection in a game session. We proposed three algorithms i.e. Tracking Algorithm, CC Assigning Algorithm and Cheating Reduction Algorithm. The Tracking algorithm helps in maintenance of a stable connection in a game and reduces the risk of network splitting. The CC Assigning algorithm ensures that a Central Connection is always present in a session. The Cheating Reduction algorithm uses a panel mechanism to check any suspicious activity in a gaming session.

Future work can focus towards improving the cheat reduction to enhance the overall game experience. Furthermore, work can be done on reducing the overall data load in the network and on individual computers to make the game play smoother and flexible. It would be interesting to use a combination of peer-to-peer and client server to design an architecture for a MMORPG.

References

[1] Ignasi Barri, Francesc Gin´e and Concepci´o Roig, "A Hybrid P2P System to Support MMORPG Playability", 2011 IEEE DOI 10.1109/HPCC.2011.80.
[2] Hu. Liu and Y. Lo., "DaCAP - A Distributed Anti-Cheating P2P Architecture for Massive Multiplayer On-line Role Playing Game", InCCGRID, 2008.
[3] Marios Assiotis Velin Tzanov, "A Distributed Architecture for Massive Multiplayer Online Role-Playing Games", NetGames '06 Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, Article No. 4, ACM New York, NY, USA ©2006.
[4] Ignasi Barri, Concepció Roig, Francesc Giné, "Distributing game instances in a hybrid client-server/P2P system to support MMORPG playability", DOI 10.1007/s11042-014-2389-0, Publisher Springer US, 28 December 2014.
[5] Frode Voll Aasen, Tom-Christian Bjørlo Johannessen, "Hybrid Peer-to-Peer Solution for MMORPGs", Submission June 2009.
[6] How Stuff Works, Tracy V. Wilson, How MMORPGs Work,http://electronics.howstuffworks.com/mmorpg6.htm,
[7] How Stuff Works, Tracy V. Wilson, How MMORPGs Work, http://electronics.howstuffworks.com/mmorpg7.htm,
[8] Massively multiplayer online role-playing game – Wikipedia, https://en.wikipedia.org/wiki/Massively_multiplayer_online_role-playing_game