AIR HOCKEY

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF

REQUIREMENTS FOR THE

**COMPUTER NETWORK** **PROJECT**

Submitted by:

**PIYUSH SHANDILYA (2K18/IT/082)**

**PRIYBHANU YADAV (2K18/IT/090)**



**IT DEPARTMENT**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

**ACKNOWLEDGEMENT**

The real spirit of achieving a goal is through the way of excellence and ascetic disciple. I would have never succeeded in completing my task without encouragement, cooperation, and help provided to me by various personalities. I want to thank the Information Technology Department, Delhi Technological University for providing me the necessary platform to deliver my project work. I would like to express my sincere gratitude to our esteemed supervisor, Ms. Anamika Chauhan and Mr. Jasraj Meena, Delhi Technological University, Delhi for his valuable guidance in carrying out this work.

**CONTENTS**

**ACKNOWLEDGEMENT………………………………………………………………...**2

**CONTENTS.……………………………………………………………………………….**3

**1. INTRODUCTION………………………………………………………………………**4

**2. PROBLEM DESCRIPTION…………………………………………………………...**4

**3. PROPOSED APPROACH**

**3.1 DESCRIPTION…………………………………………………………………….**5

**4. IMPLEMENTATION DETAILS**

**4.1 TECHNOLOGIES USED AND THEIR PURPOSE………….………………….**7

**5. RESULTS………………………………………………………………………………**10

**6. REFERENCES………………………………………………………………………….**12

**APPENDIX………………………………………………………………………………...**13

1. Introduction

Air hockey is an arcade game, played between two players, where both of the players are at either side of the board and try to score maximum goals in opponents corner in order to win.

In this project, we have tried to model the same arcade game, built solely in python where two players play against each other on either side of the end-to-end network. The game mechanics are implemented using the pygame library. The communication between the players is handled with socket programming, which allows the two players to play in real-time against each other. Players get particular points after they score goals and the first player to score a particular no of goals wins the game.

1. Project Description

In a multiplayer game, to get maximum real-time experience , the communication between the players should be instant and lag free. To achieve this, we have used socket programming which allows us to transfer the data to a common server and since the players can easily send and receive objects(in this case, player and ball movements, etc.) over the server so the in-game communication between the players can be instantaneous.

In Air Hockey, the players are on the client-side while they send their corresponding game-states to the common server, this game-state has multiple attributes such as the speed and direction of both ball and opponent player. A server is a common socket which will be connecting both of the players. Since, Both of the players have access to the other player’s state through the server. We have used Multiple threads for accepting new connections of clients and processing the connections concurrently on the server. Hence, any movement of either player will be displayed on other player's game which provides real-time experience to both the players.

1. Proposed Approach

* Air Hockey is based on a client-server model where both the players are at the client-sides, here the server provides the necessary resources and services to the client/players.
* Both players communicate in-game with the help of socket programming
* Both players should be implemented as clients on a network.
* The Clients will be passing the state of their games over a common server to achieve End-to-End Communications between the players.
* A server will be implemented which is a common socket connecting both the players to each other.
* The ball and its mechanics will be maintained on the server.
* The server will communicate with both the players at the same time without any delay and relay the overall game state to both of them.
* The clients will create an interface for the game and allow the players to play against each other, using the pygame library.
* When the server starts, it waits for the both players to join, when both the clients/players joins, then the server initiates the game for both the players.
* When the game starts, both players play simultaneously without having to wait for the other player to make its move, each player have to move vertically in order to hit a moving ball, once there is a collision, the new speed and direction of the ball is calculated simultaneously and send to both the clients.
* A scorecard is maintained for each player alone as well as the overall scorecard is also maintained on the server, so whenever any player reach the minimum score, i.e. 5 or 10 goals, the scores are checked and the player with maximum score is declared as a winner.

1. IMPLEMENTATION

# Game Mechanics

* The Players and the ball in the game, are objects of the pygame rect class.
* The players have a vertical motion which is controlled by the arrow keys on the client’s side.
* The ball has 2 components of its velocity which controls its motion all around the screen.
* The Collisions between the ball and the walls are perfectly elastic collisions and the walls will reflect the ball after collisions.
* The motion of the ball is controlled by the players, using the position the ball hits the players.
* The implementation of this feature is achieved by using a gradient function, over the body of the players which is used to vary the y component of the ball’s velocity.
* When a point is scored, the ball is warped back at the center of the screen in order to start a new round.

# Client-Side

* Both of the players are on the client-side which is a socket that establishes an end-to-end connection with the other player through a common server.
* Each player will share the state of the game with the server, which will include the player’s position and speed, the ball’s position and speed as well as the direction of the ball.
* Each player will request the data of the other player through the server along with the current score, to give a real-time gaming experience.
* The client will draw the gaming interface with the received game state, using pygame, and give us a beautiful gaming environment.

# 

# Server Side

* The Server is a socket that will be receiving the game states of both the players i.e. their positions, and their speeds, etc.
* The server is responsible for the motion of the ball.
* It manages the relative positions of the objects on the screen, and checks for collisions among them.
* It manages the in-game sounds of the collisions and the goals.
* The server will also maintain the score of both the players.
* It relays the current state of the game to both the clients ( players ).

**4.1 Technologies Used**

## Socket Programming

Sockets are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer or one that’s physically connected to an external network, with its own connections to other networks. The obvious example is the Internet, which you connect to via your ISP. TCP sockets are used for making this project, and therefore we use AF\_INET and SOCK\_STREAM flags. We use them over UDP sockets because they’re more telephonic, where the recipient has to approve the incoming connection before communication begins, and UDP sockets are more post-mail sort of thing (anyone can send a mail to any recipient whose address s/he knows), so they don’t really require an establishment of connection before communication can happen. Clearly, TCP suits more to our purpose than UDP sockets, therefore we use them.

**Purpose**: In this project, we have used sockets to transfer the game-states of

both the players, our server socket keeps the positions of both players, their

scores , current position and direction of the ball, maintains the physics for

Collisions, etc.

## Python

Python is an interpreted, high-level, and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

**Purpose**: Python is easy to use, versatile and fast to develop. Python provides the pygame library which is a multimedia library for making games and multimedia applications, it helps to develop the game mechanics and to develop GUI for the game. Python also provides socket and thread library which are used to develop this project.

## Multi-Threading

A **thread** is an entity within a process that can be scheduled for execution. Also, it is the smallest unit of processing that can be performed in an OS (Operating System).

Running several threads is similar to running several different programs concurrently, but with the following benefits −

* Multiple threads within a process share the same data space with the main thread and can therefore share information or communicate with each other more easily than if they were separate processes.
* Threads are sometimes called light-weight processes and they do not require much memory overhead; they are cheaper than processes.

**Purpose**: We have used Multiple threads for accepting new connections of clients and processing the connections concurrently on the server. On the client-side multiple threads are used to receive and send data at the same time. So, both the players function at the same time, giving the real-life experience of multi-player gaming.

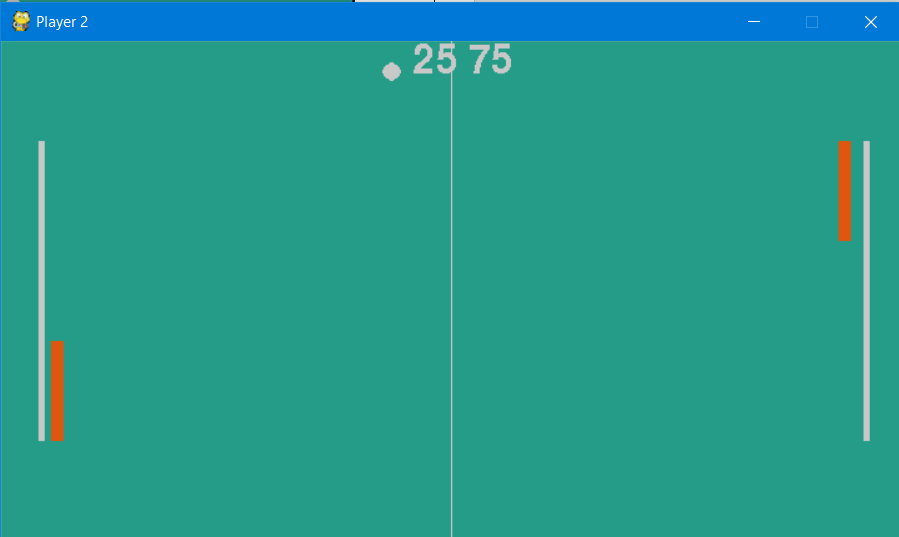
## Pickle

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dictionary, etc.) into a character stream.

**Purpose**: We have used the pickle library to send the game state across our network of clients and a server.

5. Results





6. References

### <https://pythonprogramming.net/sockets-tutorial-python-3/>

* <https://docs.python.org/3/>
* <https://www.pygame.org/wiki/tutorials>
* <https://medium.com/iothincvit/pygame-for-beginners-234da7d3c56f>
* <https://www.tutorialspoint.com/python/python_multithreading.htm>
* <https://docs.python.org/3/library/pickle.html>

7. Github Repository

<https://github.com/bluespex/Air-Hockey-Multiplayer>

Appendix

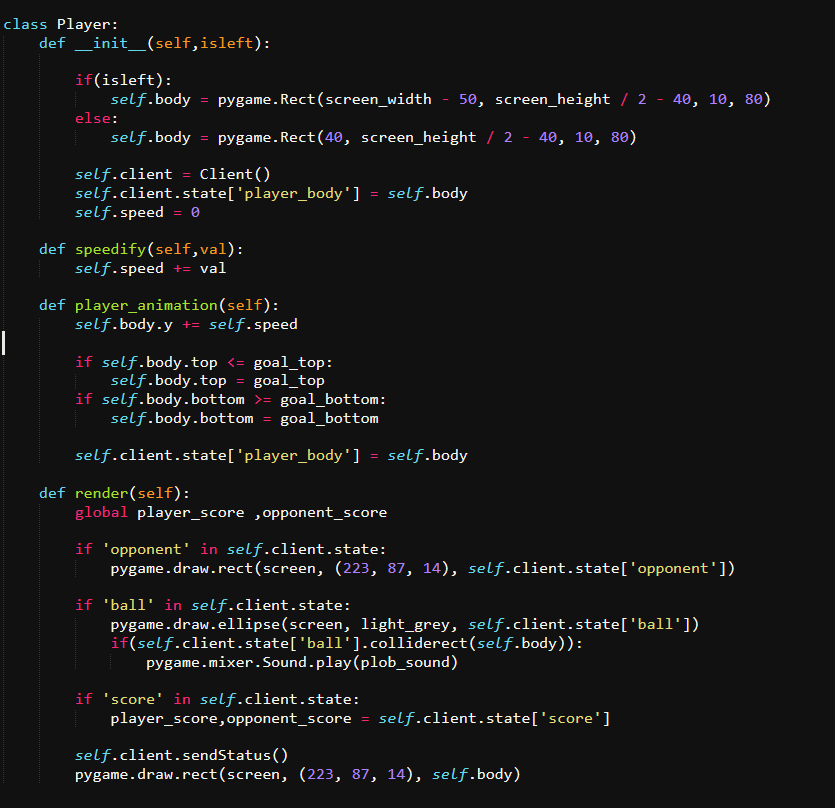
# CODE SNIPPETS

# The Client class



#### The players are objects of this class.

# The Player class



#### The initialization and the motion of the players.

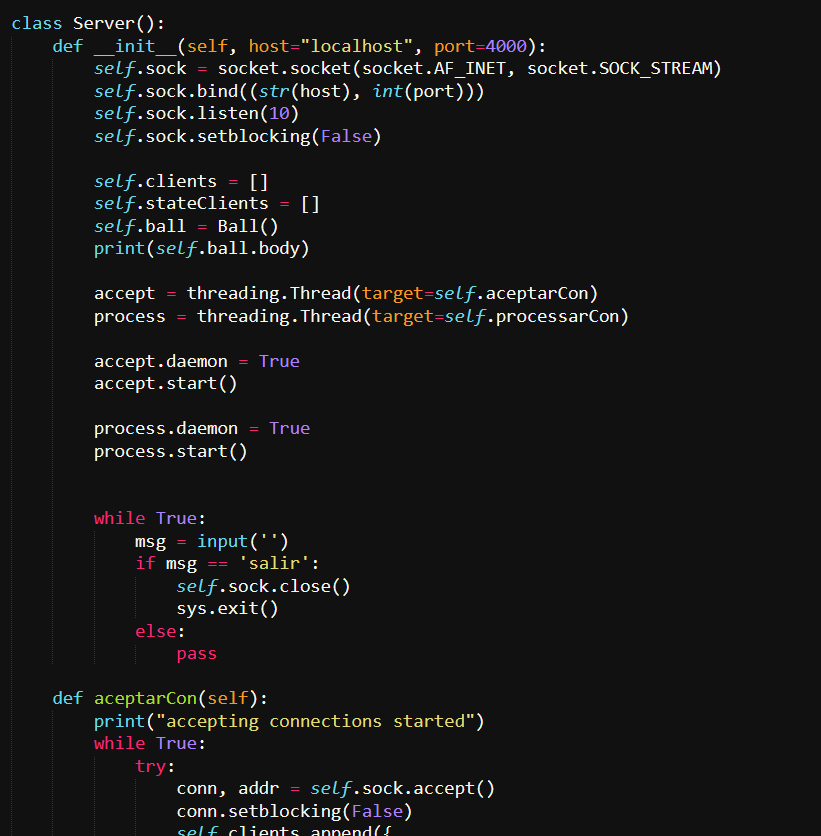
# The Ball object class

#### 

#### The ball is an instance of this class.

#### The mechanics of the ball is implemented here.

# The Server class



#### The overall game state is managed here.