COP290: Ping Pong
A Networked Multi-Player Game

Aditi Singla (2014CS50277)
Shardul Tripathi (2014CS10253)
Dhairya Sandhyana (2014CS10217)

October 4, 2018

Contents

1 Introduction

This desktop app implements a 4-player networked Ping-Pong game for people of all age groups. It involves a ball and a square board with a paddle attached to each side such that the paddle is constrained to move only along the edge. Each player gets one side and controls the corresponding paddle. The main aim of each player is to guard his/her wall (side of square) from the ball hitting it, using his paddle. The player can also use some techniques (powers like spinning,speed,etc) to trick the opponents and score over them. The designing of the app involves the following key points:

1. The application would be programmed in Java.

2. The GUI component would involve the use of Swing Library [2].

3. Connection is done using socket programming [1]:

   - **Single Player:** If there's only one manual player, he/she has to play against computer players on their local machine.
   - **Multi-player:** In this case, one player starts the game, and others join the game by providing the IP address of the starting machine. The IPs of all machines involved in the game will be exchanged at this time. Once started, the game is completely peer-to-peer, meaning there is no central server.

2 User Interface

The game has a square playing board of 600x600 pixels, with raised edges (the walls). Each edge has a paddle attached to it which is of size 100x25 pixels. Each paddle is constrained to move along its corresponding edge only and remain in the interior of the board. There is a ball of diameter equal to 30 pixels, which can freely move in any direction on the board. The board surface and the surface of the ball have negligible friction, keeping the speed of the ball practically constant.

There are 4 concentric circles in the middle. They are divided into 4 sectors by two diagonals of the square. Each sector now represents the performance of the corresponding player. Every time the player misses a shot, one of the 3 segments gets shaded to red. 3 penalties, i.e., all 3 segments of red colour for any player means his/her death (the player loses and his/her paddle is removed from the board).

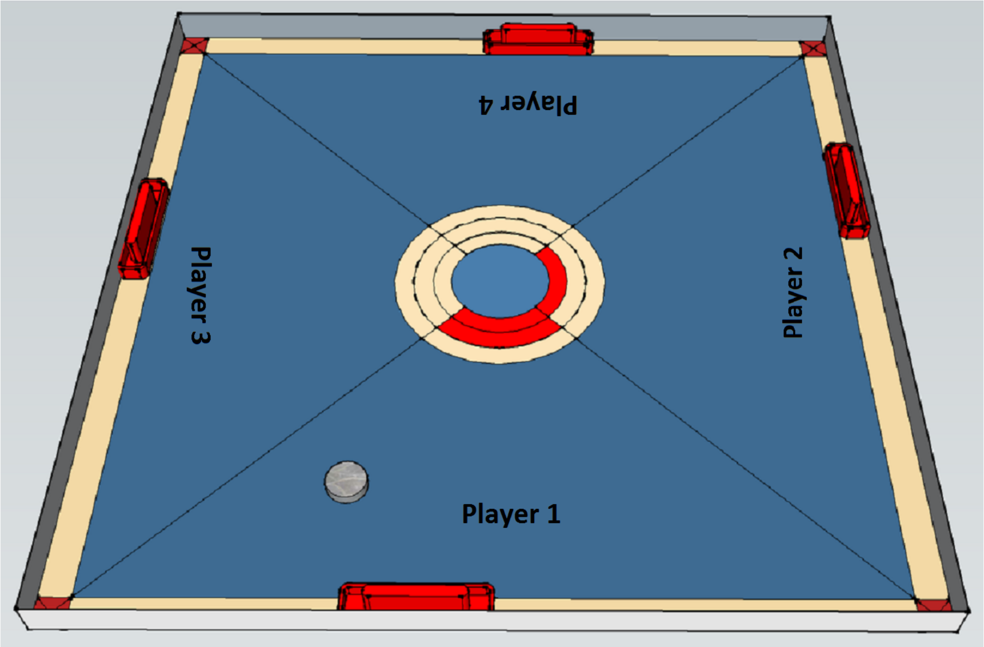

Figure 1: Expected User Interface

There is a safe zone at each of the 4 corners. The paddles can only move outside these safe zones. The ball might touch two sides at the same time, i.e., can hit a corner; leading to ambiguity. To avoid this, none of the two players gets a penalty. The ball simply rebounds from the corner at an angle of 195 degrees rather than 180 degrees. This is to avoid the oscillation of the ball between two opposite corners.

3 Implementation

3.1 For a player

The player controls the paddle using arrow keys. Whenever a ball approaches his/her wall, if the player is able to guard it by the paddle, the ball rebounds from the paddle with a coefficient of restitution. The mass of the paddle is assumed to be much larger than that of the ball and hence, while rebounding, the velocity of the paddle remains the same. In case the paddle is moving at the time of collision, the ball gets some extra velocity in the direction of the paddle as well.

If the player misses the ball, the ball rebounds from the wall. This leads to a penalty for the player. A maximum of 3 penalties are allowed, after which the player is dead and hence, out of the game. The paddle of the dead player is removed from the board and the game continues until 3 players are dead and the last player is considered as the winner of the game. In a scenario where one player is playing against 1 or more computer players, the death of the player would result in the end of the game.

There are certain special powers in the game which the players can use, whenever available, using space bar and A, S and D keys (ensuring convenience for the player as one hand controls the movement of the paddle while the other is in control of the special powers). The player can impart an extra speed i.e a spring hit, to the ball using the space bar in which case the ball will move with an increased speed without any effect on its direction. A and D keys can be used to spin the ball in the clockwise and anti-clockwise directions respectively, making the ball follow a parabolic path. Using key S will enable the player to use the special power 'invisibility'. In the case of invisibility, the ball is visible for a moment after rebounding, and then disappears from the scene. It becomes visible again only once it has reached quite close to one of the other players' paddle. The powers are explained in greater detail in Section 5.

Another feature of the game, designed to add to the difficulty of the game is reduction in paddle size as "number of lives" left decreases. As the game begins, the player is asked to choose the difficulty level of the game. Easy means absence of this feature and easy level for computer player(s) (if any). Similarly, Medium means absence of the feature but hard level for the computer player(s) and Hard means presence of the feature as well as hard level for the computer player(s). Levels of difficulty are explained in Section 3.2.

3.2 For computer

For the computer, the movement of paddle is decided by the direction of the approaching ball. If the ball is approaching the wall such that its projected point of target on the line comes in the marked range of A to B, the paddle will move else it will remain stable (with the paddle's motion still restricted between the two safe zones at the corners). The probability of overshoot i.e the paddle moving beyond the target (and missing the shot) and undershoot i.e the paddle unable to reach the target, will be p each. Consequently, the probability of the paddle hitting the ball will be (1-2p). Also, the amount of overshoot and undershoot distance will move on a bell curve which will ensure the accuracy of the distance as well as the fact that same distance won't be left out every time.

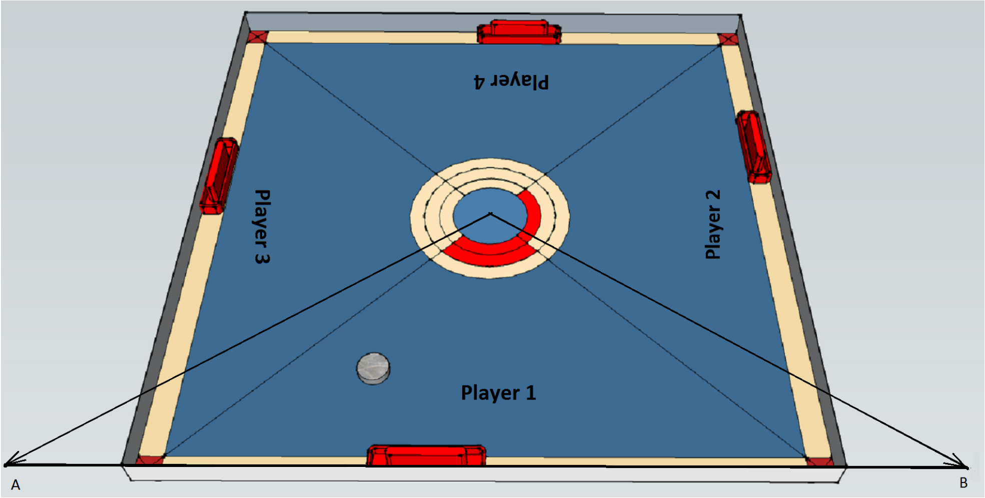

Figure 2: Range of ball for which computer will respond

The value of p decides the accuracy of the computer and hence changes with the level of the game. The value of p decreases as the level increases. This improves the accuracy of the computer as the chances of missing the ball reduce substantially, from easy to hard level. Similarly, as a improves the efficiency of the computer since it moves only when the ball is actually approaching the wall and there are chances of hitting, it will also decrease as the level increases.

For the connection purposes, each of the player in the game would be exchanging its data with each and every other player. Thus we shall implement a peer-to-peer (P2P) connection. To control AI components of the game, we shall be using the concept of a "temporary server" or a "virtual server". A randomly chosen person will be chosen as the "temporary server". This player will act as the AI of the game and will send messages to all the other players accordingly. If a player other than this one disconnects, no change is required. If this player disconnects, another active player will be chosen to act as the "temporary server". When the connection of a player is lost, he/she is replaced by a computer to ensure seamless game experience for the other players.

3.3 Physics behind the motion of ball

1. Collision of ball with wall/paddle:

$$e = \frac{v_{2f} - v_{1f}}{v_{1i} - v_{2i}}$$

and, $$mv_{1i} + mv_{2i} = mv_{1f} + mv_{2f}$$
where the value of the coefficient of restitution e is taken to be 1 so that the game can continue to stay the same without any decrease in the ball's speed.

2. Motion of the ball on the board:

   The ball moves freely on the board without any friction so that the speed of the ball remains constant throughout the game. Hence, distance covered by the ball (s) in time t in case of straight line motion is given by,

   $$s = vt$$

3. Motion of the ball after the special power 'spin' has been used:

   If θ represents the angle of incidence of the ball at paddle/wall, the equations of (parabolic) trajectory of the ball after deflecting is as follows:

   (a) Spin in anti-clockwise direction

   $$y = usin\theta t$$

   $$[sin\theta(y - xcot\theta)]^2 = 4a_1[xcosec\theta + (y - xcot\theta)cos\theta]$$

   (b) Spin in clockwise direction

   $$x = ucos\theta t$$

   $$(y + 2a_2cot\theta)^2 = 4a_2(x + a_2cot^2\theta)$$

   The exact values of the constants will be determined as and when the game is being tested.

4 Networking

4.1 Network Implementation

We will use the Transmission Control Protocol (TCP) to design and implement the network aspect of the assignment. TCP allows for reliable communication between two

applications and is typically used over the Internet Protocol, which is referred to as TCP/IP.
Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket.
The variables which will be written to the sockets for the rest of the players, per frame, include:

- Current position of paddle (x and y coordinates)
- Current position of ball (x and y coordinates)
- Velocity of paddle (x and y coordinates)
- Velocity of ball (x and y coordinates)
- Integer specifying special power used [0 for no power, 1 for spin left, 2 for spin right, 3 for invisibility, 4 for extra speed]
- Boolean for collision with wall
- Boolean for collision with paddle

This ensures that the same game state is seen by all the players because the current paddle positions etc, for individual players is sent to the rest. This information will be used to render the game for every player.

4.2 Event Flow

1. Peer 1 connects.
2. Peer 2 accepts connection; starts handler thread.
3. Peer 1 sends message.
4. Peer 2 sends ACK; message dispatched, connection closed.
5. Peer 2 connects.
6. Peer 1 accepts connection; starts handler thread.
7. Peer 2 sends message.

4.3 Network Outages

In the event of network outages, we shall replace control of the paddle of the lost In the event of network outages, we shall replace control of the paddle of the lost player with computer of the same level. The level of computer player would be calculated on the basis of number of lives left just before the player was lost(If all three lives remain, then hard level is selected, else easy). Once the player reconnects, he automatically regains control of his paddle.
The following are the trigger events to determine whether a machine has been disconnected from the network:

1. read() returns -1
2. A write will throw an IOException: 'connection reset by peer', eventually, subject to buffering delays.
3. If the connection has dropped for any other reason, a write will throw an IOException, eventually, as above, and a read may do the same thing.

5 Extra Features

5.1 Special Powers

Each player is provided with three special powers throughout the game, the availability of which is decided by a timer which starts from zero at the start of the game. The timer takes 10 seconds (may change when the game is actually implemented) to load fully and restarts when the player misses a ball or uses a power.

5.1.1 Spring Hit

The player can impart the ball an extra speed (1.5 times the current speed) when hitting the ball. This gained speed remains until the next collision, after which the previous speed is regained.

5.1.2 Invisibility

This is a special power to confuse the opponent, even the computer player. The ball becomes invisible after the first-fifth journey of the approximate calculated path of the ball and becomes visible again in the last-fifth part of the journey. When this power is used, the probability p of undershooting or overshooting is also increased for the AI of computer.

5.1.3 Spin

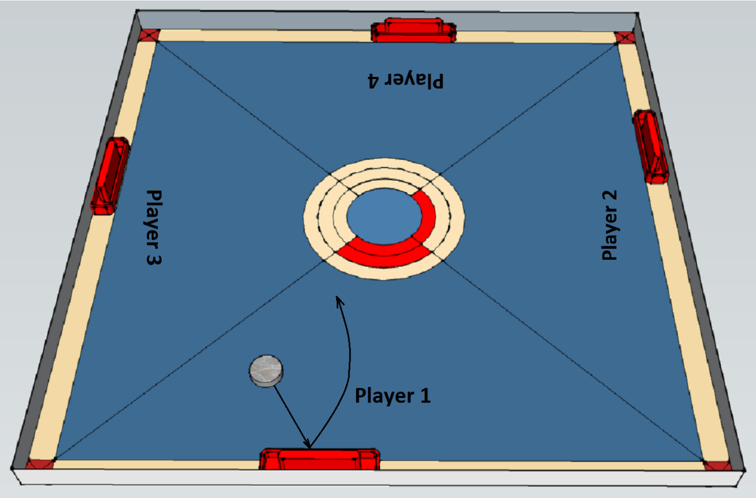

Figure 3: Anti-clockwise spin

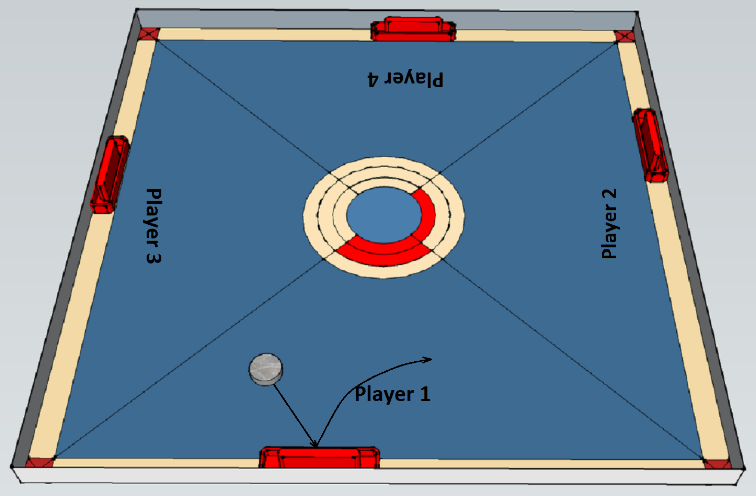

Figure 4: Clockwise spin

The player can spin the ball either in clockwise or anticlockwise direction, when reflecting the ball with the plank, using the keys 'A' and 'D' respectively; making the ball follow a parabolic path after deflection. The complete equations are provided in Section 3.3.

5.2 Difficulty Level

When the game begins, the player is asked to choose the level of difficulty:

1. Easy: This level includes a basic version of the game, with no extra features (apart from already mentioned). The level of computer player(s) (if any) is easy.

2. Medium: This level again does not incorporate any feature other than the ones already mentioned. However, the difficulty level of the computer player(s) (if any) is hard.

3. Hard: This level includes a **new feature**, along with the ones already mentioned, i.e., reduction in paddle size as the number of lives remaining decreases. As the game begins, the paddle size of each player is 100x25. Missing the ball once reduces the size to 80x25 and missing it again further reduces it to 60x25; thereby increasing the difficulty more and more. Along with this added feature, the difficulty level of computer player(s) (if any) is hard.

   Details of difficulty levels of the computer in Section 3.2

6 Future Endeavours

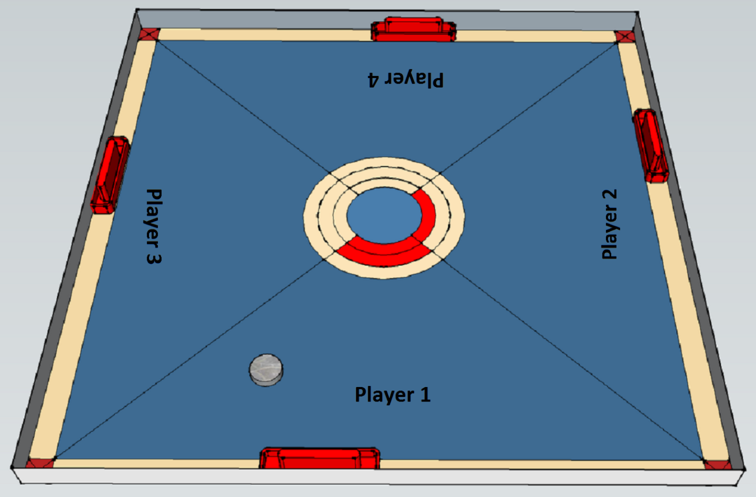

Figure 5: Game implemented using 1 ball

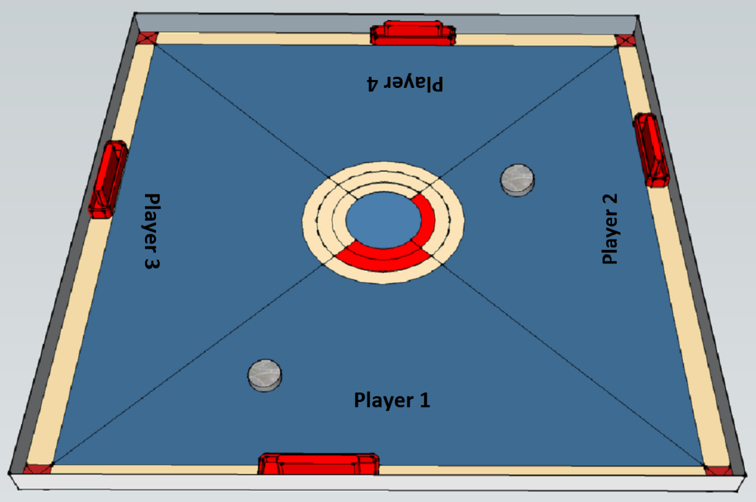

Figure 6: Game implemented using 2 balls

We aim to implement the game with a choice of playing with one or two balls in the future. More than two balls can be implemented but that would make the game end very soon and hence, quite boring.

References

[1] Socket programming:. https://docs.oracle.com/javase/tutorial/networking/sockets/.

[2] Swing library:. https://docs.oracle.com/javase/tutorial/uiswing/.