ASSIGNMENT-3
PING PONG- MULTIPLAYER

Shobhit Jain
Electrical Engineering
2014EE10134

Piyush Kumar
Electrical (Power and automation)
2014EE30527

Anshul Basia
Mathamatics and Computing
2014MT60502

ABSTRACT

This game can have maximum four players where each player guards her wall from the ball. If the ball touches the player's wall 3 times, the player is deemed as dead and her paddle is removed from the game board. This is a continuous action game; even if a player misses the ball, the ball continues to move on the game board. Some players in the game can be manual, and some can be backed by the computer (different difficulty levels: easy/medium/hard). The player who remains alive at the end of the game is deemed as the winner. This is a basic ping pong game. There is only one ball which moves in a random direction at the start of the game. The game board is in 2D and the paddles can move left and right (or up and down) along a side only. You can improvise on this, and create a fancier version of the same game. Here is how different players play against each other.

∗ Single Player- If there's only one manual player, she has to play against computer players on her local machine.

∗ Multi-player-In this case, one player starts the game, and others join the game by providing the IP of the starting machine. The IP's 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. Any host can go down during the game.

Following are the points on which our code/game will be based on:

1 Scope of the Game

- The game allows us to play with more than one ball.
- The bat has certain super powers which can be used by the player based on his skill and experience.Whenever, the score of the player reaches a certain level, then he will have an option to speed up

the velocity of his bat for a short time interval. This will be based on the experience of the player which will certainly be an additional point in deciding the VICTOR!

2 Physics of the Game

- We are using certain physics equations to determine the trajectory and speed of the ball and yes, we have considered the cases when ball'd collide with any corner or edge of the bat. Equations' implementation will slightly be deviated from the ideal world to make the game more realistic.

- Specifically, We will use elastic collision always because if inelastic collision is used then ultimately the ball will come to rest which we do not want. On collision with the bat, the perpendicular component of velocity will change its direction and the parallel component will remain the same. One of the case is when the ball strikes the bat at just its edge. In this case, the ball will retrace its trajectory and rebound back in the same direction from where it came. Another corner case is when the ball strikes the tip of corner. In this case the velocity of ball is a function of height from the base of ball where it hits the corner.

3 Algorithm for computer player

- Computer player will use a certain algorithm decided by the level that user selects to play against and that algorithm will be followed throughout the game.The computer player will keep on moving its bat in the direction where the ball is moving. e.g. if the computer player is such that it is moving the bat in +x, -x direction, then it will perceive the direction of ball first and move in that direction always to follow the ball. If the level of play is difficult, the velocity of bat is very high and it will nearly be following the ball always, whereas if the level is easy then the bat will be having lower velocity.

- 1. nextposition = next position of the ball on the bars of the board
  2. csbar = bar belonging to computer slider
  3. vcsx = velocity of computer slider in next direction
  4. vbx = velocity of ball in x direction
  5. vby = velocity of ball in y direction
  6. dcsx = change in position of the computer slider
  7. csx = position of computer slider
  8. lb = length of the bar
  9. v = velocity of bar dependent on the level of difficulty

-
  ```
  if nextposition not on csbar:
  .. dvcx = vbx * timedelay
  .. if 0 less than vcsx + dcsx less than lb:
  ....... csx = csx + dcsx
  if nextposition if on csbar:
  .. tempx = nextposition
  .. vcsx = ( (nextposition-csx)/abs(nextposition-csx) * v )
  .. dcsx = vcsx * timedelay
  .. if( ball hits slider):
  ...... nextposition(vbx , vby , ballposition)
  ...... score increased
  ```

```
.. else
...... life lost
```

- Here is the description about where the computer player will be playing, as in distributed networked fashion or on the central server There is no central server. We are designing the peer to peer network around the equal peer nodes simultaneously functioning as both ”clients” and ”servers” to the other nodes on the network. This model of network arrangement will differ from the client–server model as we are trying to avoid communication to and from a central server. Each structure will act as both client and server, sharing same responsibility and status.

- Events to handle for computer player will be as follows:
  The computer player has to handle events such as percieving the direction of ball, moving in the direction of ball, tune the velocity of bat, tapping and using power ups once they have been activated. They will be activated as soon as the player reaches a threshold score required for power-ups.

4 Network Communication

- Information Exchanged between different machines:
  1. Bat position of all the peers
  2. Difficulty level of play if computers are playing
  3. Power up status of every player
  4. Ball position
  5. Direction of movement of ball
  6. Direction of movement of bat of each player
- Making sure that Game State is seen by all the players playing the game.
  In order to ensure that the game plays out identically on all machines it is necessary for our program to wait until all player’s commands for the next movement are received before simulating that move. This means that each player in the game has latency equal to the most lagged player. Any game action may occur only after the delay has passed. Another optimization that is needed is to minimize the information in the network packets.

- EVENT FLOW for each Network message:
  Network message TO Set up a DatagramPacket to receive the data into to Receive a packet from the server TO Find the coordinates of corresponding objects (ball and sliders) as given by the server TO the next frame of the gameplay

- Trigger Event to decide that any player has left the game:
  When the user presses quit button or the window is closed by the player, then a network message will be sent across to all peers that the player has been disconnected.

- When a player’s machine get disconnected from the network:
  A new computer player will be activated at the position where the player left with zero score and medium difficulty level.