COP290: Ping-Pong game

Parichay (2013TT10947)
Silky (2013TT10970)
Ujjawal (2013TT10974)

April 7, 2016

Design Document

1 Overview

Design a multi-player pong game with maximum of 4 players connected over a common network, to compete against each other or with computer with one or more balls depending on the difficulty level.

2 Abstract Design Specification

Java is the implementing language.Application provides for deciding the *number of players*, *maximum score* and *time for the game* at the beginning of a game session.Game is twitch-based and uses the Peer-to-Peer model after the connection.

2.1 Game Logistics

- *Game Overview* :
  - Plot consists of a 2-D canvas consisting of 4 paddles along the 4 edges.
  - Each paddle is controlled either manually by a connected player or computer controlled.

  - **Single player**: he will play against the computer player(s).
  - **Multi-player** : one player starts the game and others join the game by providing the IP address of starting machine.
  - If any player looses his network during the game, his paddle and pointer will be removed and will be considered as out.
  - Game can be modified with as **many balls** as one likes depending on *difficulty level*.Also length of paddle and speed of paddle and ball can be varied accordingly.

- *Game Interactions*:
  - If the ball touches a player's wall 3 times , the player is deemed as dead and his paddle is removed from the game-board. His side is replaced by a computer player.
  - Points and score of each player will be stored/displayed and will be used to decide the winner.Players can see their performance and ranking in between the game and also at the end.
  - Player can continue to view the live game even after losing it provide he stays connected to server.

2.2 GUI for Game Setup

- At the starting of game, there would be a host of the gaming session who selects the level of the game and adds other players.
- A menu will be displayed for choosing between different options either to join the game or to host the game.
- The host asks for the IP addresses of all the other players and then transmits this array of IP addresses to all the players.

2.3 Graphics

Swing library is used to construct graphics. A mathematical model of these graphical layout will be constructed using a canvas.
The screen can be thought of as a 2-D grid with each lattice point representing nodes of graph and each latitude and longitude representing its edges.
Paddle and ball movement will be restricted using these nodes of graph.

2.4 Networking

A peer to peer network model is implemented when 2 or more players/PCs are connected and data is shared without going to a separate server system. Thus every connected PC acts as a server and a client.

- **Network Protocol**
  Preferred networking protocol will be UDP(User Datagram Protocol). UDP uses a simple transmission model without implicit handshaking dialogues for guaranteeing reliability, ordering or data integrity.
  We are more concerned about the *time lag* and *speed issues* and *consistency of frames* rather than the reliability of the transmission. Thus UDP is more suitable as compared to that of TCP/IP.

- **Game Initialisation**
  At the instant of initiating the game, there will be a client-server model.One player acting as a host, will create the network(will have a server.java file) and the other players will be joining that network(stored in clients.java file).
  When all the players have joined the game(after providing the **IP address of server** and **Port number**, the useful information(**Player names**, **basic game settings**) will be circulated(through InputStream and OutputStream methods of Socket class) among all the players and our setup will enter the peer to peer mode.

- **Multi-player Code** :
  There will be a multi-player code which is same in all the peers. It will contain necessary functions for receiving and transmitting the messages as well as updating the current positions of the players. In effect, this code will be responsible for reflecting the current status of game.

- **Sending Action messages** :
  For performing various actions over the board a player has to transmit certain messages over the network.
  For every action message there will be a specific code associated with it which will consist of a handler thread class. Handlers are responsible for a dealing with a single client and broadcasting its messages.

Figure 1: Event flow for network message

- **Synchronisation** :
  Every client will be receiving certain messages over the network. In order to get a simulation of all the other players in local machine we need to update their current position using Multi-player code scheme.

- **Latency issues** :
  The time between sending and receiving the response needs to be minimised. For this purpose, we can have two possible solutions:

  - Reduction in the size of messages being transmitted over the network.
  - *Method of extrapolation*: we will predict the trajectory on the basis of current information till we get the exact position. This will result in the continuous movement of players over the arena.

2.5 Artificial Intelligence/Algorithm for computer player

The gameplay involves different types of AI techniques which the computer player follow such as given below throughout the course of the game. It follows weak or strong AI depending on the mode selected.

- **Amatuer Mode :** In this mode there will be more chances to beat computer player as there will be more lag between movement of AI(computer) paddle and ball. Also speed of the ball will be less in this mode.

- **Pro Mode :** In this mode it will be more difficult to beat computer player as there will be less lag between movement of AI(computer) paddle and ball. Also speed of the ball will be more in this mode.Through the time paddle length of player will decrease to a predefined minimum limit.

3 Physics involved

Game will consist of 3 basic elements :

- *Player paddle* :

  - 2 conditions are to be ensured :

* Reflection of ball from paddle surface
* Movement of paddle along the screen frame.

Same can be extended for all players.

- To ensure that the ball is reflected back from the paddle surface we define a Rectangle bounding box and set bounds to it:
  Rectangle boundingBox;
  boundingBox = new Rectangle (x , y, width, height);
  (the four dimensions of the paddle)
  boundingBox.setBounds(x, y, width, height) ;

- Defining the response equations for computer paddle:
  Set PlayerSpeed = 3 ; (a greater value for higher difficulty levels)
  // The paddle moves along the balls co-ordinate
  If ((game.ball.x < x+h) && (x >= 0))
  {
  x-= PlayerSpeed*game.delta;
  }
  // game.delta used to maintain same speed in multiplayer case
  //x+h (h is a constant defined to lag the computer player)

  else if ((game.ball.x > x) &&
  (x < HeightGame - HeightPaddle - HeightCorner))
  {
  x += PlayerSpeed*game.delta
  }

- **AI paddle :**
  3 conditions are maintained: *Reflection from paddle*, *movement along frame* and *ensuring limitations so that the computer player can miss the ball sometimes.*

  - To make the computer player lag so as to make it possible for it to miss the ball in certain cases:
    AIspeed = PlayerSpeed - 1 ;

- To ensure that the ball is reflected back from the paddle surface we define a Rectangle bounding box and set bounds to it:
  Rectangle boundingBox;
  boundingBox = new Rectangle (x , y, width, height);
  (the four dimensions of the paddle)
  boundingBox.setBounds(x, y, width, height) ;

- Defining the response equations for computer paddle:
  // The paddle moves along the balls cordinate
  If ((game.ball.x < x+h) && (x >= 0))
  {
  x -= AIspeed*game.delta;
  }
  // game.delta used to maintain same speed in multiplayer case
  //x+h (h is a constant defined to lag the computer player which changes depending on the difficulty level)

  else if ((game.ball.x > x) &&
  (x < HeightGame - HeightPaddle - HeightCorner))
  {
  x += AIspeed*game.delta
  }

- **Ball :** Ensure reflection of ball when hit by any hard surface.

  - For multiple balls we need to ensure each ball has set bounds :
    Rectangle boundingBox;
    boundingBox = new Rectangle(x , y, size, size);
    boundingBox.setBounds(x, y, size, size);

  - Four variable x, y, vx , vy(coordinates and speeds in x and y direction):
    int speed = 2;

    if (y <= 0){
    Vy = speed;

```
game.score2++;
}
else if (y+size >= game.getWidth() ){
Vy = -speed;
Game.score1++;
}
```

- Similarly for y coordinates:

```
if (x <= 0) {
Vx = speed;
game.score3++;
}
else if (x+size >= game.getHeight()) {
Vx = -speed;
game.score4++;
}
x+=vx;
y+=vy;
```

- PaddleCollide method: This will ensure reflection from paddles and other balls.

```
if ( boundingBox.intersects(paddle1) )
{
Vy = -vy
}
//incase it exists
else if (boundingBox.intersects(paddle2))
{
Vy = -vy
}
//incase it exists
else if (boundingBox.intersects(paddle3))
{
Vx = -vx
}
else if (boundingBox.intersects(paddle4))
{
```

```
Vx = -vx
}
else if (boundingBox.intersects(ball1))
{
Vy = -vy
Vx = -vx
}
else (boundingBox.intersects(ball2))
{
Vy = -vy
Vx = -vx
}
```

Corner Cases : Both the velocities (x and y components) change when a corner is striked. To ensure this separate if conditions for x and y coordinates of each corner (paddles and the corners) is to be defined within the code. The multiple ball collision cases or the cases of collision of ball with the paddle have already been covered in the pseudocode.

4 Extra Features

- There will be an audio response whenever ball touches the wall. Player will lose with another special audio effect of a sound of explosion. There can be extra audio effects when extra powers are gained by the player.

- Paddles will be provided with features like spin, superspeed and reverse mode.
  Players will not be provided with these powers initially but can gain such powers as the game proceeds. For example, if a player hits the ball 15 times in a row with his paddle, he will get power of superspeed which he can execute with keyboard key.
  Similarly when he hits the ball 25 times in a row, he gets the power of spin whereas if he missed the ball 2 times in a row, his paddle movement gets reversed.
  Also the powers gain will be removed as soon as a shot is missed or the time-span specified expires.

- As the game advances, speed of the ball increases and size decreases to increase the chances of a player to die.