README FILE

Ping Pong: A Networked Multi-Player Game

Team: SPYCODERS

*S*auhard Gupta
2013ME10117

*P*rabhu Prasad Panda
2013ME10859

*Y*ash Kumar Bansal
2013ME10742

April 7, 2016

Contents

1 KEY COMBINATIONS

The User needs only two key

1. **Left Arow key:-** This is to move the paddle left.
2. **Right Arow key:-** This is to move the paddle right.

The key pressed by user has to be converted into corresponding movement of the paddle. Practically we have to control position and velocity of the paddle. So, we can set a constant acceleration(a_r) while the key is pressed, subject to a cutoff speed. Once the key is released, the paddle would decelerate rapidly till it stops or some other key is pressed. Typically, the paddle would move till about 0.5 seconds after key is released, to give it a more realistic effect.

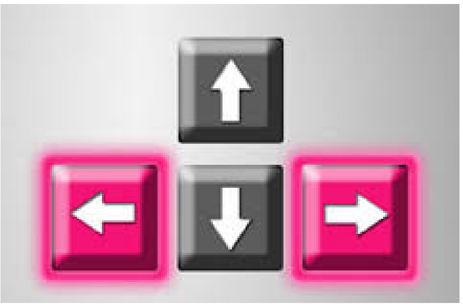

Figure 1: Keys Required

2 RULES

This is a networked ***multi-player Ping-Pong game*** that can be played on a desktop machine.

- **NUMBER OF PLAYERS:-**The game has **maximum four players** where each player *guards his/her wall* from the ball.
- **SCORING:-**The number of remaining lives are visible on the status bar of each player. The lives reduces as soon the ball touches the wall of any player.
- **LIVES:-** If the **ball touches the player's wall 3 times**, the player is deemed as **dead** and **his/her paddle is removed** from the game board and just the wall remains.
- **TYPES OF PLAYERS:-**Some players in the game can be **manual**, and others can be backed by the **computer**.

- **WINNER:-** *The player who remains alive at the end of the game is deemed as the winner.*

- **BALLS:-**There can be **multiple balls**(which ranges from 1 to 4) which moves in a random direction at the start of the game. The balls will appear at the centre of the board symmetrically and after the starting timer blows off. They will be directed arbitrarily towards the players (but it isensured that not more than one ball is directed towards any player)./* The number of the balls can change during the course of the game.*/

- **CONNECTION:-** Here is how different players play against each other;

  1. **Single player:** If there's only one manual player, he/she has to play against computer players on his/her local machine.
  2. **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.

- **TIMER:** In the beginning of the game in order to the allow every player to settle down, we will have a timer 3 → 2 → 1.

- **Speed Control of Ball:** The initial speed of the ball will be set by the player as Setting of the game. With every collision of the ball with the bat, the speed of the ball increases by x%, where x can be changed by the players.

- **SEND BAT INFO TO THE OTHER MACHINES**: Every time, the bat's co-ordinates, velocity or button pressed(left ,right or nothing) changes, this information is sent to other local machines so that these machines could have the latest information about the bat.

- **CONNECTION OUT HANDLING**: If the connection of the controlling machine goes out, the control is transferred to the machine with the next highest priority. This machine resumes operation from the latest information it has about the machine. So this highlights the importance of sending velocity and other parameters apart form co-ordinates at every stage to other machines. Ideally only co-ordinates need to be transmitted for display in other machines. But in order to handle the cases of connection failure, we need to send the entire information so that another machine can resume the control operation.

- **Ensuring that all players must see the same game state** It is somewhat obvious that the game needs to have a global clock to handle events(track inputs, change of co-ordinates of bat or ball etc). We plan to use this clock similar to a flip-flop that is the event is tracked only once in a clock cycle. So to ensure proper synchronisation, we must ensure that the cycle time of the clock is much larger than the network delay of transmitting information. As the state of the game changes only once in each clock cycle, this will ensure that all the local machines are in the same state the next time the clock becomes active.

- *Event Flow for Network Message* As we have used peer-to-peer connection, the event flow basically involves sending the necessary information to the peers at each clock cycle.

This is a continuous action game even if a player misses the ball, the ball continues to move on the game board.

Here is how a simple 4-player Ping-Pong would look like:

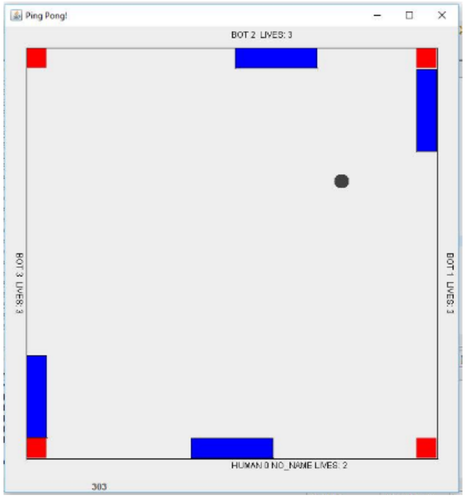

Figure 2: Snap Of Game Board