# Ping-Pong: A Networked Multi-Player Game[1]

In this assignment, you have to design and develop a networked multi-player Ping-Pong game that will be played on a desktop machine. Important deadlines for this assignment:

1. Design document: 11:55 PM on April 7
2. Final code submission: 11:55 PM on April 28
3. Demo's : April 29 to April 31

Game description:

Here is how a simple 4-player Ping-Pong would look like:
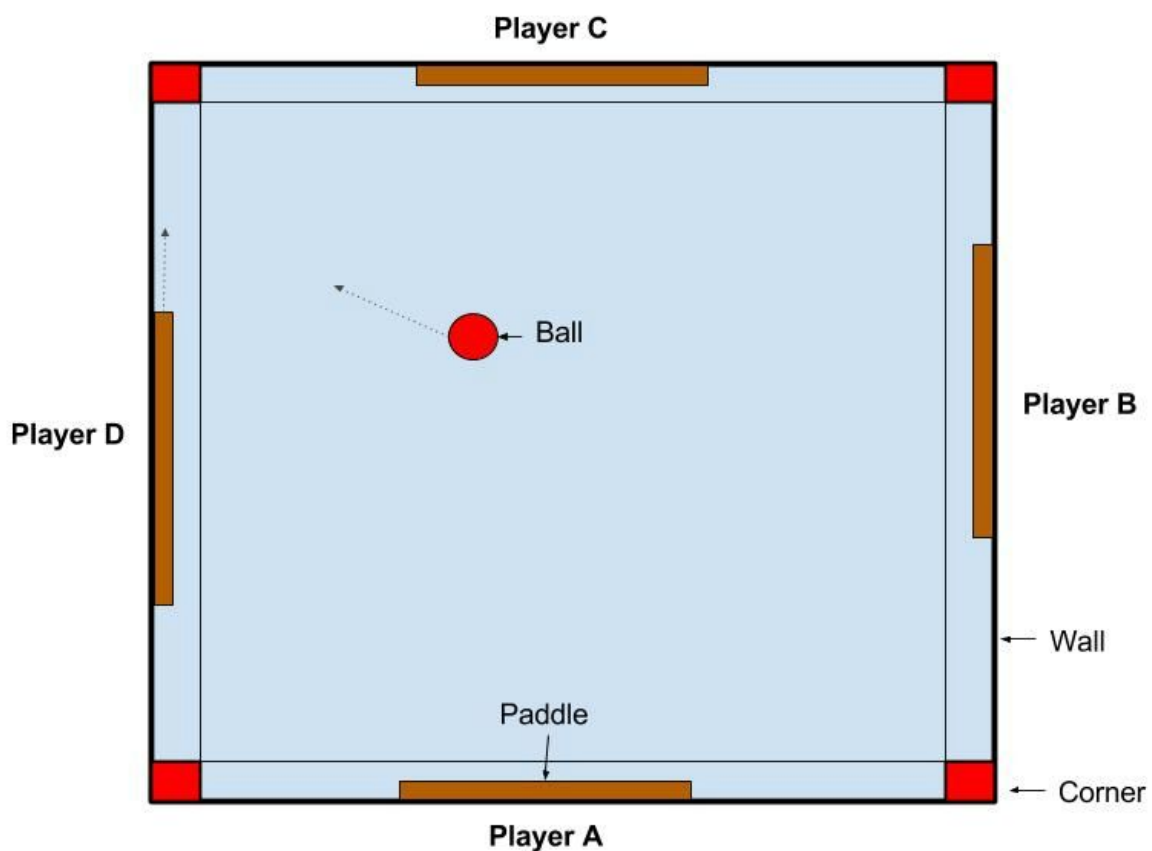
The game can have maximum four players where each player guards her wall from the ball. If the ball touches the player's wall 3 times, the

---

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.

Connection:
- 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, and your game should handle such events gracefully. Students should decide what happens if a host's connection is gone.
  - Should her paddle be replaced by a *computer player* for other players. This way other players still enjoy a seamless game experience
  - What happens to computer logic

You can add a simple countdown to the start of the game to make sure all players are ready when the ball starts moving.

Implementation Details:
As discussed before, this game has to be developed for desktops. The tool set for the assignment has been restricted to the below mentioned list:

1. Programming language: Java
2. Communication: Socket programming
3. GUI: Swing library

Design Document:

You must submit the design document latest by 11:55 PM on April 7.
The design document should answer these questions:

1. Scope of the game
   a. Does your game allow us to play with more than one ball?
   b. Does your game allow special powers on the bat (spin, or extra speed) to make the game interesting?
2. Physics of the game
   a. What are the physics equations that you will use? Are there any corner cases that you have thought of already?
3. Algorithm for the *computer* player
   a. What is the algorithm of your *computer* player? How does it vary across different levels of difficulty?
   b. Where does the computer player run? On one central server? Or on all machines in a distributed fashion?
   c. What are the events that a computer player has to handle? How are these events handled?
4. Network communication
   a. What information is exchanged between different machines?
   b. How do you plan to make sure that the same game state is seen by all the players?
   c. What is the event flow for each network message?
   d. What is the trigger event to decide that a player's machine has been disconnected from the network?
   e. What happens when your code concludes that a player's machine has been disconnected from the network?

Guidelines:

You have one month to design and implement the entire game. So do a **proper assessment of your team's constraints** while deciding the scope of the assignment. Spend a lot of time in designing the assignment. This will help you implementing it in less time.

Grading:

The grading for this assignment will be done on a scale of 100. For this assignment, the breakup of the grade is:

1. Design document [20 marks]: You should submit the design of the application in the latex format. The design should at least answer the questions that have been mentioned above.

2. Basic code [50 marks]:
   a. GUI and physics [15 marks]
   b. Logic of the computer player [10 marks]
   c. Networking [25 marks] There should be no lag in the display, when four people are playing the game over network.

3. Software practices [20 marks]
   a. Code indentation + Commenting + Modularity + Version control system [5 marks each]

4. Design credits [10 marks]: Submissions which distinguish themselves will be rewarded. Here are a few points which can be considered:
   a. Good GUI
   b. Extended the scope of the application
   c. Created a good test environment for the application

Submission:

You should submit the assignment on Moodle. The assignment should be bundled in a zip file whose name is *entryno1_entryno2_entryno3.zip*. The zip must contain three folders:

1. *src:* This should contain the source code of the assignment. This folder should also contain a file called "game.jar". This is the file that contains your game.

2. *doc:* This contains a file main.tex that contains the design of the application. You should also include a "makefile" that builds a main.pdf file from the main.tex file.

If your submission does not adhere to the given submission format, you will receive a penalty of 10 marks. At the time of demo, you should bring your laptop(s) on which this application can run. We will use your Moodle submission to create a *jar* file and we will evaluate your assignment.

Academic Integrity: All the submissions will be checked for plagiarism. If a student group is found to be copying code from other group or a third party source, all the concerned students (including the students who made their code available to other groups) will get 'F' grade and a disciplinary action will be initiated against them. You can use the code obtained from Android examples on the Internet; however, while doing so, you must clearly mention the source of the code in comments.