**[Type the company name]**

Fall 2012

**Multiplayer Pac-Man**

**Concurrent programming**

**Arjun ,Anshu,Yashovardhan**

Concurrent Programming

Project Report on Multiplayer PacMan

# Abstract:

In a regular game of pacman only the pacman is human controlled and the ghosts are computer controlled (AI). This project attempts to make a multiplayer pacman game where all the characters are human controlled. There are two teams: pacman’s team and the ghost’s team. Multiple players can play at a time and they can join in either of the teams. The aim for the pacman’s team is to eat all the pellets on the board, while the aim for the ghost’s team is to catch any of the pacman before all the pellets are finished.

# Motivation:

The motivations behind this project are stated below:

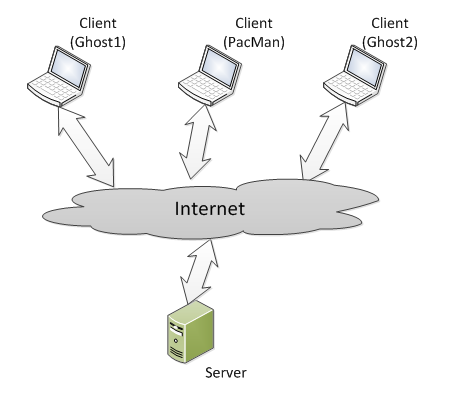
* To build an application in which multiple people can interact concurrently.
* Implement the concurrent programming techniques learnt during the coursework.
* Have fun while learning something new.

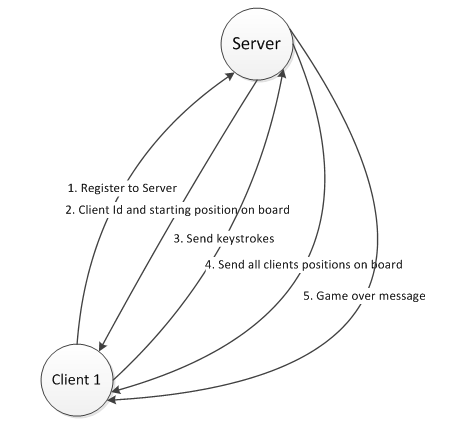
# Controls of the game:

The Pacman/Ghost moves using the arrow (up, down, left and right) keys on the keyboard. The player cannot go through the grid blocks. The players continue to move in their direction until they hit a wall in the maze. Player direction is queued; this means that a player changes its directions after pressing an arrow key only when the grid allows them to. Meanwhile, the player continues to move in the direction he is moving in. Wherever possible the both ghost and pacman can tunnel across the screen and come from the other side of the screen.

# Design structure of the application:

Server – Client architecture is followed. Clients are connected to the server using UDP protocol. Each player initially registers itself with the server by sending the information whether he wants to be pacman or ghost. Once registered, the server sends the initial position to every player. Every client implements key listeners to sense the keystrokes. They send their respective key information and their current position to the server. The server collects every client’s position and broadcast it to everyone else so that clients get to know about every other player’s current position. If all the pellets are over the server announces team pacman as winner and if any pacman is tagged by a ghost the server announces team ghost as the winner.

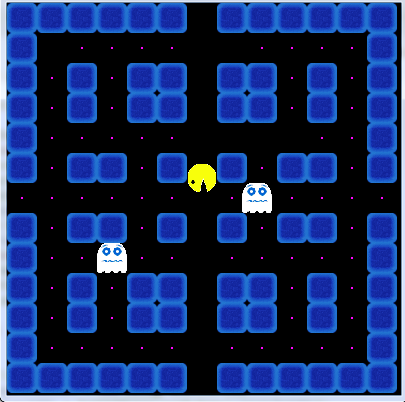
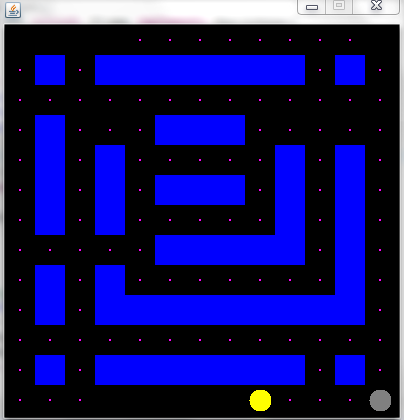




# UI Design:

We started our project with a very basic UI. The walls of the maze in the game were just squares and the pacman and ghosts were simple circles of different colors drawn using fillOval and fillRect functions. The framework of the maze is made using a 2D matrix which is initialized with 1s and 0s. The walls are constructed by drawing squares in place of 1s and pellets in place of 0s. Once the pacman eats a pellet the 0 is replaced by -1 indicating empty path.

Once we were done with main functionality of the code we decided to improve the UI by replacing the circles and squares with images. We replaced the fillRect and fillOval functions with drawImage. Note that the pacman turns its face in the direction it moves so different images are displayed for pacman according to the direction pacman is moving. We tried various different types of mazes and images and finally settled with the one on the right as we believed that this provides a better user experience and is a good balance of difficulty for both the pacmans and the ghosts.



# Evaluation