

Rahma Seid

Professor Sainju

CSCI 3240

Final Project Proposal

For my final project, I would like to implement a text-based Pacman game for single players, which is designed to operate over a client-server model. The server will manage the core game logic between the server and the client. The client will represent the player, allowing users to connect and interact with the Pac-Man game in real-time.

Description of what client does:

The client.c will allow the player to interact with the game using a text-based interface. The roles client.c will include:

- User Interaction: receive player input (movement through keyboard arrows/WASD keys)
- Rendering: render the game board, including Pac-Man, ghosts, walls, and pellets. The rendering will be done using a simple text-based interface to reduce complexity
- Networking: This will maintain a connection to the server, send player inputs to the server, and receive game state updates from the server
- Real-Time Display: Update the display based on game state data received from the server, showing the current position of Pac-Man, ghosts, and pellets
- Communication: Handle error messages and notifications from the server, such as connection issues or game-over

Description of what server does:

The server will manage the complete game environment and sync communication with the connected client. The roles server.c will include:

- Game State Management: Maintains the representation of the game state, including positions of Pac-Man, ghosts, pellets, and walls

- Client Handling: Accept incoming connections from the client, create a separate thread of the connection, and manage communication
- Input Processing: Receives player commands from the client, like moving up, down, left, and right, and validates the moves based on the game rules
- Ghost AI: Manage and update ghost movement, including the classic chase and scatter behavior of ghosts
- Collisions Detection: Detects collisions between Pac-Man and walls, ghosts, and pellets, updating the game state
- Game Logic and Scoring: Handle winning/losing conditions, scorekeeping, and game timer, and send updates to the client

Services provided:

- Game state updates: Regularly updates the client of the entire game state, including positions of Pac-Man, ghosts and remaining pellets
- Movement Validation: Accept movement requests from the client and validate them
- Scorekeeping: Track the player score and send score updates to the client
- Notification System: Notify the client of significant events in the game, such as player deaths, game over, or game start

Programming language and libraries:

Programming Language: C

Libraries:

- Standard C Libraries
- Ncurses Library
- Socket Programming (BSD Sockets)
- Pthread Library