

Multiplayer Number Guessing Game (UDP or TCP)

Overview:

Build a multiplayer number guessing game using **TCP (reliable)** or **UDP (challenging due to statelessness)** where multiple clients try to guess a number chosen by the server. The server provides hints ("Too high", "Too low", "Correct") and tracks which client wins first.

Learning Outcomes:

- Understand socket communication (TCP or UDP).
 - Handle multiple clients in real-time.
 - Manage simple game logic and client states.
 - Explore fairness and concurrency in games.
-

Assignment Description:

🚀 Server (`game_server.py`):

- Randomly selects a number between 1 and 100.
- Accepts guesses from multiple clients.
- For each guess, sends a response: "Too high", "Too low", or "Correct".
- Announces the winner to all connected clients.
- Resets the game after a round ends (optional).
- Keeps track of connected players and scores (e.g., 3 rounds max).

💻 Client (`game_client.py`):

- Connects to the server.
- Receives a welcome message and instructions.
- Submits guesses via terminal.
- Receives feedback on each guess.
- Displays game-over message and leaderboard.

Optional Features:

- **Leaderboard** with names and scores.
 - **Chat mode** before the game starts.
 - **Difficulty levels:** narrow or widen number range.
 - **Timeout:** players must guess within X seconds.
-

TCP vs UDP Options:

- **TCP version:** Easier, with persistent connection and ordered delivery.
- **UDP version:** Harder, must handle retries and associate guesses with players using address tracking.

Deliverables:

- `game_server.py`
The server script that manages the game logic, receives guesses, and handles multiple clients.
- `game_client.py`
The client script that connects to the server, submits guesses, and receives feedback.
- `README.md`
Setup instructions, gameplay guide, and usage examples.
- *(Optional)* `game_log.txt`
A server-side log file tracking player activity and round results (if implemented).