Review the assignment due date

Lab 4 - gRPC

> Distributed and Networking Programming – Spring 2025

Overview

In this lab, you will implement a **multiplayer Connect Four** game using **gRPC**. A complete **client implementation is provided** to you. Your task is to:

1. Write the `game.proto` file that defines the gRPC services and messages.
2. Implement the `server.py` gRPC server that uses the generated stub files and provides the game logic.

Client (Provided)

The provided client handles all user interactions and calls the remote methods from the server using gRPC.

It supports:

- Creating a new game.
- Connecting to an existing game.
- Playing interactively or in an automated mode (`--automate`).
- Choosing a player mark (`--player R|Y`).
- Connecting to a specific game by ID (`--game_id`).

Do **not modify the client** — it will be used as-is for testing.

Your Tasks

1. Create `game.proto`

Write the game.proto file describing the structure of the messages and services used in the game. The client interacts with the server using this protocol definition.

2. Implement `server.py`

Your server must:

- Accept a port number via CLI argument (e.g., `python server.py 50051`).
- Listen on `0.0.0.0:<port>` and register the gRPC servicer.
- Implement full game logic:
  - Validate moves.
  - Track game state.
  - Determine the winner or a draw.
- Log all received requests in the console (including errors).

Error Handling Requirements

- `GetGame`: `NOT_FOUND` if game doesn't exist.
- `MakeMove`:
  - `NOT_FOUND`: game not found.
  - `INVALID_ARGUMENT`: invalid column.
  - `FAILED_PRECONDITION`: game is finished.
  - `FAILED_PRECONDITION`: not player's turn.
  - `FAILED_PRECONDITION`: column is full.

How to Compile Proto and Run the Game

Install dependencies (in a virtual environment):

```
pip install grpcio grpcio-tools
```

Compile:

```
python -m grpc_tools.protoc -I. --python_out=. --grpc_python_out=. game.proto
```

Run server:

```
python server.py 50051
```

Run client (example):

```
# Player RED
python client.py localhost:50051 --automate --player R

# Player YELLOW
python client.py localhost:50051 --automate --player Y --game_id 1
```

Checklist

- [ ] Required files are pushed to classrooms repository on time. Other files are not modified: `server.py` and `game.proto`
- [ ] Code runs successfully under the latest stable Python interpreter
- [ ] Code only imports dependencies from the Python standard library and grpc
- [ ] `game.proto` compiles without errors
- [ ] Server handles all RPCs
- [ ] Game logic for Connect Four is correct
- [ ] Server logs all requests
- [ ] All gRPC error cases are handled