Sudoku multiplayer game

DS - homework 1 documentation

Contributors:

- Juhan Raedov
- Rain Eric Haamer
- Ragnar Margus
- Joonas Merisalu

For both the server and the client side we took shared_state_guessword_game as the basis and modified it. More thorough descriptions of classes and methods have been written in the code as comments.

How to get the server running:

1. Navigate to the "../MultiplayerSudoku/server " folder
2. Run the command: "**python2.7 serverMain.py**"
    a. additionally flag "-a IP.ADRESS" may be used, by default server will listen on 127.0.0.1
3. The server is running and waiting for players.

How to get the client running:

1. Navigate to the "../MultiplayerSudoku/client " folder
2. Run the command: "**python2.7 clientMain.py**"
3. The client program will be running and guide the player through joining and playing process - see below...

Playing the game:

1. Player will be asked for username.
2. Program asks server's IP and tries to connect to the the server. Client will ask server for name validation, if it succeeds, it jumps to 3rd point, otherwise new name will be asked.
3. Server sends a notification of currently available sessions on the server. Player may:
    a. Create a new session - by specifying how many players must connect to the game till it starts and a session name.
    b. Join an existing session by specifying its name.

    In both cases name is checked (is it alphanumeric) and server notifies if the game is already full or if a game with such name actually exists.

4. If the game session has all the required players, the game will start, otherwise players wait.
5. Player will be asked to enter values to the sudoku, in the form of three numbers. (Example: 123 will request the server to put number 3 into x=1, y=2 cell.) The user input will also be tested for correctness (numbers in range 1...9).

    The server takes action if input:

    a. **Correct** - Sends updated scores and changed Sudoku table to all the players. The client who put the number also gets a 'correct' notification. (Player's score incremented)

   b. **False** - Sends updated scores and changed Sudoku table to all the players. The client who put the number also gets a 'wrong' notification. (Player's score decremented)
      i. Wrong numbers can not be inserted to the sudoku.
      ii. In case game ends (Sudoku is finished) winner(s) will be declared and players will return to lobby and get current session list.
6. In case a player disconnects and only one player is left in the game session, the winner is declared and the remaining player will be returned to lobby.

Brief description of the client files:

1. clientMain.py - main client file. Upon running asks input from the player and communicates with the server.
2. clientIO.py - contains classes SyncConsoleAppenderRawInputReader and AbstractSyncIO which provide input and output capabilities for the player in Terminal

Brief description of the server files:

1. serverMain.py - The main server file. Creates a listener socket, keeps tract of connected clients and game sessions creates clientHandler object for each client.
2. clientHandler.py - The clientHandler will take action based on client's actions and also allows to notify the changes in game session.
3. sessionClass.py - Allows clientHandlers to interact with Sudoku instance. Keeps track of game status and notifies clients about changes.
4. Sudoku_new.py - Has the Sudoku creation tools and the Sudoku class, numbers can be added to the 'current' Sudoku, game end is checked here, the Sudoku board is designed and made to string.