

Network Protocol for Online Rubik's Cube

ECE 462

10/5/2010

Adapted from "Network Protocol for
Online Super Tetris"

Objective

We need a network protocol because...

- △ people talk only when they speak the same language
- △ make design work easier for both server and client sides

What is a Protocol?

...“a set of instructions for transferring data.”

—Wikipedia

Two protocols you need to worry about:

TCP

- communication between server/client
- works through *sockets*

Socket/Game

- game receives commands from socket

General Rules

- △ All operations must be acknowledged by server.
- △ Client must not act on its own.
⇒ Client relays player actions to server.
- △ Server decides whether or not to take the actions, and send the action back to client if permitted.
- △ Client acts only upon receiving an action from server.

General Message Format

- ▲ `<COMMAND>?`: ask for client information
- ▲ `<COMMAND> [<argument>]`: control messages (player join, quit, start, etc.)
- ▲ `<id>:<COMMAND>`: action triggered by player `<id>`, or a control message particularly for player `<id>`

Commands - Setup

Server commands and responses to begin game

- **"ID?\n"** - first command after connection to establish identity
 - **"ID=playerid\n"** - Client response where *playerid* is the letter 'g' followed by your group number
- **"ACCEPTED.\n"** - initializes beginning of game
 - Client will not respond but await first game command

Commands - In Game

- **"READY?\n"** - initializes new puzzle to allow program time to reset variables. Client will respond with **"READY\n"**.
- **"SOLVE *cubestate commands*\n"** - begins a new puzzle to solve where *cubestate* is a character string describing the cube state is GPA1 and GPA2, and *commands* is a string of the commands which you can use to solve the cube.
 - NOTE: space between *cubestate* and *commands*, and *commands* will be omitted if all commands are allowed
- **"*playerid*:WIN\n"** || **"*playerid*:LOSE\n"** - signals end of tests. This state will need to be displayed by the GUI.

Commands - In Game

Solving, client responses:

- "*playerid:rotation\n*" - *playerid* is your group id and *rotation* is the rotation being performed (i.e. "**F\n**" or "**B'\n**")
- "*playerid*□:**DONE\n**" - signals the cube is solved

Conclusion

1. Connect to server
2. Shake hands (ID, GAMETYPE, READY)
3. Perform only actions from server
4. Relay local actions to server (and wait for feedback)

Questions?