Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A31

Game C/S Model

Team:

[Neeraj Kumar Bansal] - Id: [041000185] / [Jay Patel] - Id: [041028206]

**NumPuz Proposal**

***This template is suggested (not mandatory) to answer A31 Specification.***

| **Part**  **1** | **C/S Architecture** |
| --- | --- |

* 1. **Server Model**

*Describe how your server interface should be organized and the main methods to be defined*

* + - ***Example****:*

**Example** (see A31 specification)

INTERFACE:

Class: NumPuzServer

# Components:

→ JLabel: labPort

→ JTextField: txtPort

→ JButton: start, end, results,

→ JRadioButton: finalizeButton

→ ImageIcon: gamelogo

CONTROLLER:

Class: NumPuzServer – Object: “**server**”

→ Method: Start:

try (

NumPuzServer **server** = new NumPuzServer (portNumber);

NumPuzClient client = **server**.accept();

}

→ Method: End:

Closing my connection (including all threads) and disabling the **EXIT\_ON\_CLOSE** in JFrame.

→ Method: Result:

Output my results (Including ranking of players or clients and their moves, points and time in which they finished the game).

→ Method: Finalize:

Finalize the application when all clients have finished.

* ***Note****: The professor interface continues being a proposal. Focus on your ideas using the best user experience.*
  1. **Client Model**

*Describe aspects of your client (interface and methods) considering the proposed idea.*

**Example** (see A31 specification)

INTERFACE:

Class: NumPuzClient

# Components:

→ JLabel: labUser, server, port

→ JTextField: txtUser, txtServer, txtPort, description

→ JButton: connect, end, play, sendGame, etc.

→ ImageIcon: gamelogo

CONTROLLER:

Class: NumPuzClient – Object: “**client**”

→ Method: Start:

try {

NumPuzClient **client** = new Socket(hostName, portNumber);

}

→ Method: Connect:

Initialize the connection between client and server by receiving one proper thread.

→ Method: End:

Finish connection by sending a message for the server that close the thread.

→ Method: Play:

To be able to invoke the Game (using a previous configuration) and transmit details when the game is executed (Play mode).

→ Method: Send:

To transmit and receive the game configuration created (remember the Design mode in the MVC).

* 1. **Protocol Proposal**

*Finally, what is your idea to define the protocol to be used.*

**Example** (using the string definition mentioned in the A21 specification)

CONFIGURATION STRING:

Class: NumPuzModel

→ Property: String: gameConfig:

→ Format: <dim><dataSeparator><dataConfig>, where:

→ <dim> = integer (from 2, 3, etc.)

→ <dataSeparator> = comma (,)

→ <dataConfig> = chars (example: 1-9), obeying the formula (dim2)2.

→ Example:

numerical;1,2,3,4,5,6,7,8,0

text;M,y, ,g,a,m,e,!,∙.

PROTOCOL P0:

→ protocolSeparator: hashtag (#)

→Format:<clientPortNumber><protocolSeparator><serverIPAddress><protocolSeparator><serverPortNumber>

PROTOCOL P1:

→ configSeparator: (:)

→ gameConfiguration: <type><configSeparator><dimension><configSeparator><tile>

<type> = string -“Number” | “Text”

<dimension> = integer -range from 3 to 9

<tile> =String -integer | character

→ protocolSeparator: hashtag (#)

→ Format: <clientId><protocolSeparator><gameConfiguration><protocolSeparator><serverId>

→ Example: 1#Number:3:123456780#4

PROTOCOL P2:

→ protocolSeparator: hashtag (#)

→ Format: <serverId><protocolSeparator><clientId><protocolSeparator><responseData>

PROTOCOL P3:

→ gameDataSeparator: (:)

→ protocolSeparator: hashtag (#)

→ gameData: <userName><gameDataSeparator><points><gameDataSeparator><time>

<userName>: String

<points>: integer

<time>: integer (seconds)

→ Format: <clientId><protocolSeparator><gameData><protocolSeparator><data>

→ Example: 2#xyz:8:200#5

| **Part**  **2** | **Game Evolution** |
| --- | --- |

* 1. **Notes about upgrading the game**
  + *Describe the main modifications to be proposed in the C/S version of the game.*
    - *What are the differences between the original proposal (A11 / A21) and the current project to be developed (A31).*
    - *If so, explain why you need to do some adjustments.*

**Example** (About MVC modifications)

MODEL component:

Public methods to change private data (ex: dataConfig), that can receive inputs, but evaluate if they are valid.

VIEW component:

Send client information to server (ex: time, points, moves and port of client).

CONTROLLER component:

Talk with client and play game accordingly on request.

SERVER component:

Store client information (including time, points and moves), set port number, start or give permission to play and finalize results.

CLIENT component:

To establish connection and finish the client execution; new game definition, communication with server (send / receive game config), data communication (passing game results to the server) and game play.

* 1. **GitHub / Database Integration (Bonus)**
  + *GitHub link: https://github.com/JyP2275/Numpuz-Game*
  + *Considering this proposal for 3-tier architecture using Databases, define:*
    - *What to persist.*

Requests will pass through middleware (ex: application layer).

Control presentation layer at client’s device while middleware and server handle application and database respectively.

* + - *What is the DB datatype to be used.*

SQL\*Net

* + - *How frequently to update.*

Every time on request, update the server first and then the UI when the server responds with success.

**References**

[*https://intellipaat.com/blog/what-is-client-server-architecture/#no8*](https://intellipaat.com/blog/what-is-client-server-architecture/#no8)

Algonquin College

Spring / Summer, 2022