Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A31

Game C/S Model

Team:

Savas Erturk - Id: 040919022

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 startButton, JButton resultButton, JComboBox finilazeCheck JButton stopButton

CONTROLLER:

DataInputStream, DataOutputStream, ArrayList, Socket, Server, ServerSocket, port,

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

→ Method: Start:

try (

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

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

While(!serverSocket.isClosed()){

}

→ Method: closeServer:

try (

if(serverSocket !=null)

}

* ***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: User, JLabel: Server, JLabel: port, JTextField: username, JTextField: serverAddress, JTextField: port, JButton connect, JButton close, JButton singleplayer, JButton newgame, JButton play,

CONTROLLER:

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

→ Method: Start:

try {

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

} …

→ Method: sendData:

→ Method: recieveData:

→ Method: startGame:

→ Method: disconnect:

* 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 P1 (CONNECTION):

→ Format: <clientId><protocolSeparator><server><protocolSeparator><port>

→ Example: savas#localhost#0613

PROTOCOL P2 (SEND SERVER GAME CONFIG):

→ protocolSeparator: hashtag (#)

→ Format: <gameMode><protocolSeparator><data>

→ Example: gameMode#[1,2,3,4,5,6,7,8,0]

PROTOCOL P2 (RECIEVE FROM SERVER):

→ protocolSeparator: hashtag (#)

→ Format: < gameMode ><protocolSeparator><data>

→ Example: gameMode#[1,2,3,4,5,6,7,8,0]

PROTOCOL P2 (SEND GAME RESULT TO SERVER):

→ protocolSeparator: hashtag (#)

→ Format:

<clientId><protocolSeparator><gameMode><protocolSeparator><data><protocolSeparator><score> <protocolSeparator><time>

→ Example: SAVAS#gameMode#[1,2,3,4,5,6,7,8,0]#score#time

|  |  |
| --- | --- |
| **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.

// CONTINUE…

* 1. **GitHub / Database Integration (Bonus)**

*I want to record game results on the database. For this, I only need to use the insert method. A11, I decided to do this game’s unique feature is according to the 5 or 10 matches average score timing needs to start counting from up to down. So, I will record the summary end of the match then I will send to the database. The best option for this project Java JDBC.*

*CREATE TABLE `records` (*

*`id` int(11) NOT NULL,*

*`username` varchar(150) NOT NULL,*

*`score` varchar(150) NOT NULL,*

*`time` varchar(150) NOT NULL,*

*`whenPlayed` timestamp NOT NULL DEFAULT current\_timestamp()*

*) DEFAULT CHARSET=utf8mb4;*

**References**

*[Include eventual references used here]*

Algonquin College

Spring / Summer, 2022