Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A31

Game C/S Model

Team:

[Thomas Stanley] - Id: [41037136]

Picross Proposal

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

* 1. **Server Model**

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

INTERFACE:

Class: GameServer

→ Components: JLabel: labPort, JTextField: txtPort, JButton execute, JButton results, JButton end, JScrollPanel scrollPanel

CONTROLLER:

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

→ Method: Start()

try (

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

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

}catch(Exception e){

System.out.print(‘Connection error’);

}

Method : End() :

Try

Server.close();

System.out.println(‘Server closed’

} catch(Exception e){

System.out.print(‘Connection error’);

}

* 1. **Client Model**

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

INTERFACE:

Class: GameClient

→ Components: JLabel: labUser, JTextField: txtUser, JScrollPanel newGame, JButton sendGame, JButton receiveGame, JButton sendData, JButton play, JTextField userText, JTextField portNum, JTextArea terminalTextArea

CONTROLLER:

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

→ Method: Start:

try {

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

Client.connect();

} catch(Exception e){

System.out.print(‘Connection error’);

}

* 1. **Protocol Proposal**

CONFIGURATION STRING:

Class: GameModel

→ 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: 00001,10111,00110,11111,00011

PROTOCOL P1:

→ protocolSeparator: hashtag (#)

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

→ Example: 1#3; 00001,10111,00110,11111,00011

PROTOCOL P2:

→ GameData

→ Example : 00001,10111,00110,11111,00011

PROTOCOL P3:

→ Protocol Separator : ( , )

→ Format <userName><protocolSeparator><pointsNum>><protocolSeparator><time>

→ Example: “Tom”,2000,00:34

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

*The main modifications to be made is the added functionality of a high score table, including names, scores. And times. Currently my game has time and score displayed at the end of a successful game, so I will need to add a name portion. I will also need to program a client-server framework, to be able to save these details for later viewing.*

*I will also need to implement threading to handle the timing of the data sending/receiving that will be happening.*

* 1. **GitHub / Database Integration (Bonus)**
  + *The use of GitHub is also a bonus to be considered:*
    - *Be sure that you can inform the updated repository and branch.*
    - *TIP: To avoid problems, also include the document (template answer) in the BrightSpace.*
    - *Database should be updated weekly to keep track of scores.*

Algonquin College

Winter, 2022