Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A21

Game MVC

Team:

Soomin Lee - Id: 040899389

NumPuz Proposal

***This template is suggested (not mandatory) to answer A21 Specification.***

|  |  |
| --- | --- |
| **Part**  **1** | **GUI Definition** |

* 1. **MVC Details**

**There is design class and game control class.**

**Design is designing all frames, layouts, panels, buttons, text box, combo box..etc. also including splash screen.**

**Game control class has all function that action reacted by buttons.**

**Class: JWindow – Object: “splash”**

**→ Class : ImageIcon - Object : “splashImage”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “numberPanel”**

**→ Class : JButtons – Object : “btnArr”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel – Object : “modePanel”, “dimensionPanel”, “levelPanel”, “typePanel”, “imagePanel”, “timePanel”, “lastPanel”**

*Describe the way you can define the MVC components in your game.*

**Example** (from vision “top-down”)

Class: JFrame – Object: “GameFrame”

→ Class: JPanel → Object: “GameuBoard”

→ Class: JButtons → Objects: “BSave”, “BLoad”, etc.

→ Class: JLabel → Objects: “LabOperation”, “LabName”, etc.

…

* 1. **View Component**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “numberPanel”**

**→ Class : JButtons – Object : “btnArr”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel – Object : “modePanel”, “dimensionPanel”, “levelPanel”, “typePanel”, “imagePanel”, “timePanel”, “lastPanel”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “modePanel”**

**→ Class : JRadioButton – Object : “designButton” , “playButton”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “dimensionPanel”**

**→ Class : JComboBox – Object : “dimension”**

**→ Class : Array – Object : “dimArr”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “levelPanel”**

**→ Class : JComboBox – Object : “type”**

**→ Class : Array – Object : “typeArr”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “imagePanel”**

**→ Class : Image – Object : “ima”**

**→ Class : JLabel – Object : “imaLabel”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “timePanel”**

**→ Class : JTextArea – Object : “timerTextArea”**

**Class: JFrame – Object: “f”**

**→ Class : JPanel - Object : “rightPanel”**

**→ Class : JPanel - Object : “lastPanel”**

**→ Class : JButton – Object : “startButton”, “resetButton”**

*Describe how your interface should be organized using new components. Show the idea about your “top-down” organization.*

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

**Example** (from vision “top-down”)

Class: JFrame – Object: “GameFrame”

→ Class: JPanel → Object: “GameuBoard”

→ Class: JButtons → Objects: “BSave”, “BLoad”, etc.

→ Class: JLabel → Objects: “LabOperation”, “LabName”, etc.

…

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

**Object: “startBtn”**

**→ Event : actionPerformed – method: startGame(), startTimer()**

**Object: “type”**

**→ Event : actionPerfomed – method: designNumberPad(), designTextPad()**

**Object: “designButton”**

**→ Event : itemStateChanged – method: setEnabled()**

**Object: “playButton”**

**→ Event : itemStateChanged – method: setEnabled()**

*Describe aspects of your controller using, for example, one unique action command. Create the “map” to define functions with actions.*

**Example**

Object: “BSave”

→ Event: actionPerformed → method: saveGame()

Etc.

* 1. **Model Component**

**Values: resetButton- method: resetGame()**

**Values: startButton- method: checkDimension() -> designNumberPad() , startTimer()**

**Values: numberButtons- method: moveNumber() -> checkNumberOrder() -> numberInOrder() -> congratsMsg() -> resetGame()**

*Finally, what is your idea to define the model to be used in a “default” (randomized) game.*

**Example**

Data structure used:

→ Values: gridValue → method: updateData()

|  |  |
| --- | --- |
| **Part**  **2** | **Implementation Design** |

* 1. **Game Evolution**
  + *Considering this new model, explain:*
    - *What are the differences between the original proposal (A11) and the current project to be developed (A21).*
    - *If so, explain why you need to do some adjustments.*

**There are some difference between A11 and A21.**

**1. Design : Panels are more specified and detailed. Layout is more clear (use BorderLayout to make number pad larger).**

**2. Interative Actions : if select some buttons, actions start**

* 1. **Others DP**
     + *Define (at least one) additional DP that you could use in your Game application.*
  + *Explain what is this DP and the reason why it could be recommended.*

**Other Design Pattern**

**-- Creational design pattern (Singleton) : create one instance object(game, numberpad) that can access from controller**

**References**

*https://stackoverflow.com/questions/16134549/how-to-make-a-splash-screen-for-gui*

Algonquin College

Spring / Summer, 2022