Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A21

Game MVC

Team:

John Vinh - Id: 041004759

NumPuz Proposal

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

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

* 1. **MVC Details**

*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**

*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.*

Class: JFrame

* Object: “GameFrame”

Class: JPanel

* Object: “GameBoard”

Class: JButtons

* Objects: “GameMoveButton”, “ShowSolution”, “HideSolution,”, “Reset”

Class: JLabels

* Objects: “Mode”, “TimeElapsedLabel”, “TimeElapsed”, “DimLabel”, “SolutionLabel”, “TypeLabel”, “MovesLabel”, “Moves”, “PointsLabel”, “Points”

Class: JComboBox

* Objects: “Dim”
  1. **Controller Component**

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

Object: resetButton

* Event: mouseClicked -> method: resetGame()

Object: typeChoice

* Event: actionPerformed -> method: changeType()

Object: showButton

* Event: mouseClicked -> method: showSolution()

Object: hideButton

* Event: mouseClicked -> method: hideSolution()

Object: dimComboBox

* Event: actionPerformed -> method: changeDim()

Object: designButton

* Event: actionPerformed -> method: enableDesignMode()

Object: playMode

* Event: actionPerformed -> method: enablePlayMode()

**Example**

Object: “BSave”

→ Event: actionPerformed → method: saveGame()

Etc.

* 1. **Model Component**

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

Value: timeElapsed

* Method: updateTime()

Value: points

* Method: updatePoints()

Value: moves

* Method: incrementMoves()

Value: dim

* Method: updateBoardDimensions()

Value: mode

* Method: updateMode()

Value: tileType

* Method: updateTileType()

**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.*

With the new model, I needed to separate my game into three separate classes: the Model, View, and Controller. I needed to do this adjustment because it will make the future development a lot easier for me by following the MVC design pattern. With all three of those components in a single class, as my current development was heading towards, it would be a lot harder to keep my code clean.

* 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.*

**References**

*[Include eventual references used here]*

Algonquin College

Spring / Summer, 2022