Algonquin College Logo

# SCHOOL OF ADVANCED TECHNOLOGY

### ICT - Applications & Programming

### Computer Engineering Technology – Computing Science



A21

Game MVC (Class Diagram)

Team:

[Kiet Tran] - Id: [040487459] / [Gia Bao Tran] - Id: [041048694]

Battleship Proposal

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

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

*This template is very similar to your A11, but going deeper with the components and methods descriptin and dividing them into the MVC components.*

* 1. **Classes specification**

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

*Describe all classes that you imagine to use in your game.*

1. Default package

* A picture containing text, font, screenshot, line

  Description automatically generatedMain.java

1. Component package

* Board.java

A screenshot of a computer

Description automatically generated with low confidence

* Coordinate.java

|  |
| --- |
| Coodinate (class used to coordinate the buttons by rows and columns |
| column (column number)  row (row number) |
| Coordinate(int, int) : constructor  getRow() : get row value  getColumn() : get column value |

* Ship.java

|  |
| --- |
| Ship (class used to represent the ship in game) |
| length (length of the ship)  row (row number)  col (column number) |
| Ship(int) : contructor  coordinateSetter(int, int) : set coordinate of the Ship on board by row and column  getLength() : get the length of the ship  get Row() : get the row of the ship  get Col() : get the column of the ship |

1. game package

* Game.java

|  |
| --- |
| Game (class for main game window) |
| player\_1 (represent player 1)  player\_2 (represent player 2) |
| Game() : constructor |

* Game Controller.java

|  |
| --- |
| GameController (this class is acting like the brain of the game) |
| colorSet1 (set 1 of color)  colorSet2 (set 2 of color)  timer (set the timer ActionListener)  language (English/French/Portugal)  dimension (1 to 12)  img (import image from the resource)  logoLabel (logo label)  logoPanel ( logo panel)  languagePanel (panel for language)  languageLabel (label for language)  languageBox (box of language)  designPanel (panel of design)  designButton (button of design)  randButton (button of random ship)  dimensionPanel (panel for dimension)  dimensionLabel (label for dimension)  dimensionBox (box of dimension)  historyBox (record of scores)  timerPanel (panel for timer)  timerLabel (label for timer)  timerBox (box to display timer)  resetPanel (panel for reset)  resetButton (button for reset)  playPanel (panel for play)  playButton (button to start to play)  dimension (size of the board)  masterGame (main game window)  boardWest (board on the left)  boardEast (board on the right) |
| GameController(JFrame) : constructs the main game window  configurePanels() : configures the panels and components  configurePanel() : configures the specific panel  addComponentsToPanel() : add the component to the panel  actionPerformed (ActionEvent) : handles action events from buttons  removeExistingBoard() : removes the existing board  createGame(int) : create a new game  setLanguage(string) : set the language for the game |

* StartMenu.java

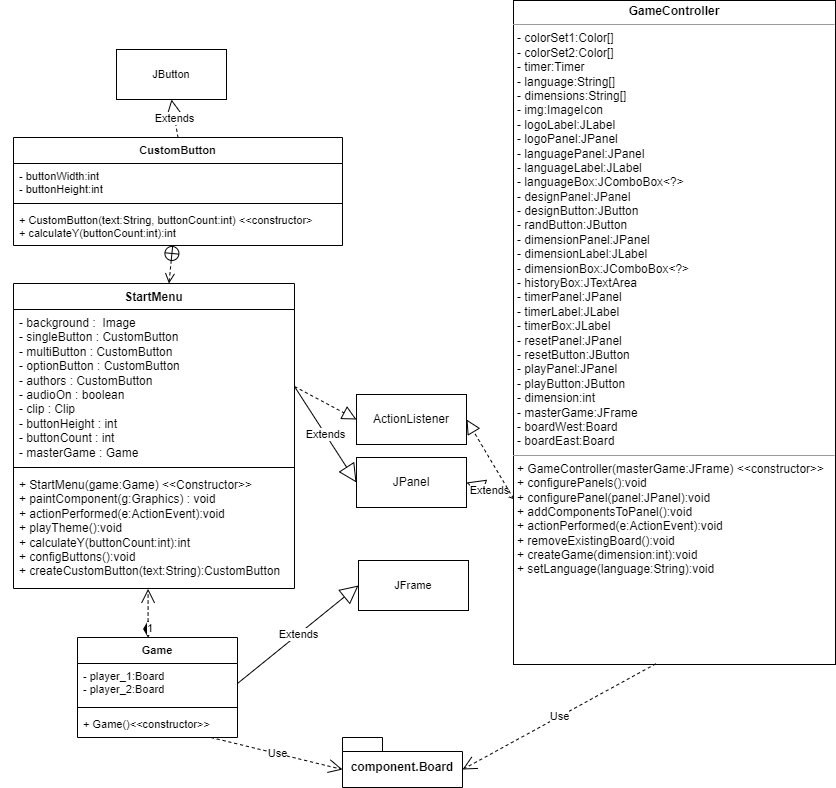
|  |
| --- |
| StartMenu (start screen with background and buttons) |
| backgroundImage (import the image from the resource)  singleButton (button to play the single player)  multiButton (button for multiplayer)  optionButton (button for option)  authors (button to display the author)  audioOn (true/false)  clip (get the clip of audio)  buttonHeight (the height of the button)  buttonCount (number of buttons)  masterGame (call Game object) |
| StartGame(Game) : constructor  paintComponent(Graphics) : draw the background on the panel  actionPerformed(ActionEvent) : handles the action from the buttons  playThemes() : play the theme audio.  calculate(int) : calculate the y-coordinate  configButtons() : configure buttons and their properties  createCustonButton(string) : create custom button |

|  |
| --- |
| CustomButton (class extend from JButton) |
| buttonWidth (width of the button)  buttonHeight (height of the button) |
| CustomButton(string,int) : constructor  calculateY(int) : calculate y-coordinate of button |

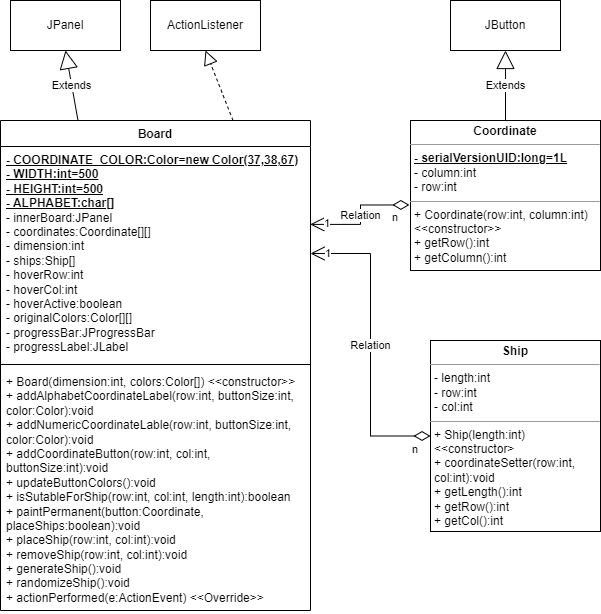
* 1. **Class diagram**

*Describe all classes relationships that you need to define to your game.*

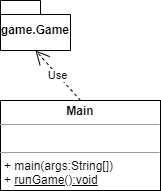
* **Game package**



* **Component Package**



* **Default Package (main)**



**References**

1. <https://stackoverflow.com/questions/9819779/understanding-why-the-extends-arrow-points-in-the-opposite-direction>
2. <https://stackoverflow.com/questions/47588511/uml-diagram-how-to-show-final>

Algonquin College

Spring, 2023