ALGONQUIN COLLEGE

SCHOOL OF ADVANCED TECHNOLOGY

ICT - Applications & Programming
Computer Engineering Technology – Computing Science

A21

Game MVC (Class Diagram)

Team:
[Abhi Patel] - Id: **[041048533]** / **[RominKumar Patel]** - Id: **[041063127]**

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 description and dividing them into the MVC components.

1.1. Classes specification

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

The Model will store the logic of the Battleship game. The Controller will react to user input. The View will present the Battleship game to the user.

Name
Model (The logic of the game.)
Attributes
dimension: int (from 4 to 7)[1]
board: int (2*dimension)
playerShips: char[ ][ ] (Player's Board)
compShips: char[ ][ ](Comp's Board)
playerShipsHit: int (The number of player's ships squares hit)
compShipsHit: int (The number of computer's ships squares hit)
compSquare: int (The number of squares selected by the player on the computer's board.)
playerSquare: int (The number of squares selected by the computer on the player's board.)
readyToPlay: boolean(There are ships on the player's Board.)
playing: boolean (If the game has starts.)

Methods

getScore():double returns the score of user.

placeShips(int x, int y, int length, Boolean vertical): boolean Places a ship of length length(parameter) at co-ordinate (x,y)

isSpace(int x, int y, int length, Boolean vertical): boolean: Checks if space is available at (x,y) to place a ship of length.

computerMove():int [ ]:returns a point(x,y) that represents the computer;s move.

computerMakesMove():Computer selects at point (x,y) on the player's board.

randomPlayerBoard():Create a random Board on the player's Board.

initializeBoard(char b [ ][ ]):Sets all the square's on a board to a default value.

selectComputer(int x, int y):int: Select a point (x,y) on the computer's board.

selectPlayer(int x,int y):int: Select a point (x,y) on the player's board.

select(char b[ ][ ],int x ,int y):int: Selects a point (x,y) on a board.

placeShip(char b [ ][ ], int length, int x , int y , int vertical, int direction):boolean: Place a ship of length length(parameter) at point (x,y) on a board.

placeShips(char b [ ][ ]):boolean:Randomly places the ships on a board.

Name

View: displays the game to the user.

Attributes

frame JFrame (The main window)

panel JPanel (The main Panel)

leftPanel JPanel (Player's Panel)

leftBoardPanel JPanel (Player's Board)

leftBottomPanel JPanel (Player's status information)

rightPanel JPanel (Computer's Panel)

rightBoardPanel JPanel (Computer's Board)

rightBottomPanel JPanel (Computer's status information)

optionPanel JPanel (Game menu and information)

centerPanel JPanel (Message area and the optional designPanel)

messagesArea JTextArea (Game message notification)

topPanel JPanel (The top part of the centerpanel, containing the logo,languages,design,randomize, and dimensions.)

bottomPanel (The bottom part of the centerpanel, containing the score,time,reset, and Play.)

designPanel Jpanel (Part of the centerPanel when design button is pressed.)

JButtons:

designButton (Design mode)

randButton (Random mode)

resetButton (Reset mode)

playButton (Play mode)

Board Buttons (Arrays of JButtons)

playerButtons (Player's Buttons on player's board.)

compButtons (computer's Buttons on computer's board)

JprogressBars:

leftProgressBar (How many ships the user has.)

rightProgressBar (How many ships the computer has.)

Radio Buttons:

vertical JRadioButton (Place ships Vertically.)

horizontal JRadioButton (Place ships Horizontally.)

JComboBox:

languages (changing the language use by game interface.)

dimensions (Changing the dimension of the player and computer boards.)

JTextFields:

timeField (time information)

scoreField (score information)

JLabels:

Labels for languages, dimension, score, and time.

Labels across the top and left margin of the player board and computer board.

Methods

write(String text): displays a text message in the messages area.

drawBoard(JPanel boardPanel, JPanel panel, square [ ] [ ] array, int dimension): draws a board in the graphical interface of the game.

placeShip(int x,int y):int: Places a ship on the player's board at co-ordinate (x,y) with the length from the dropdown menu.

displaycolors(Square [ ][ ] array, char b [ ][ ]): Sets the colors of a board in accordance with the information stored in the model.

randomPlayerBoard(): Generates random board for the player.

reset(): Clears the board.

resize(): Set the size of the board as per the dimension.

updateLabels():Update the label as per the language.

selectComputer(int x, int y):Display a move on the computer's board.

selectPlayer(int x, int y):Display a move on the player's board.

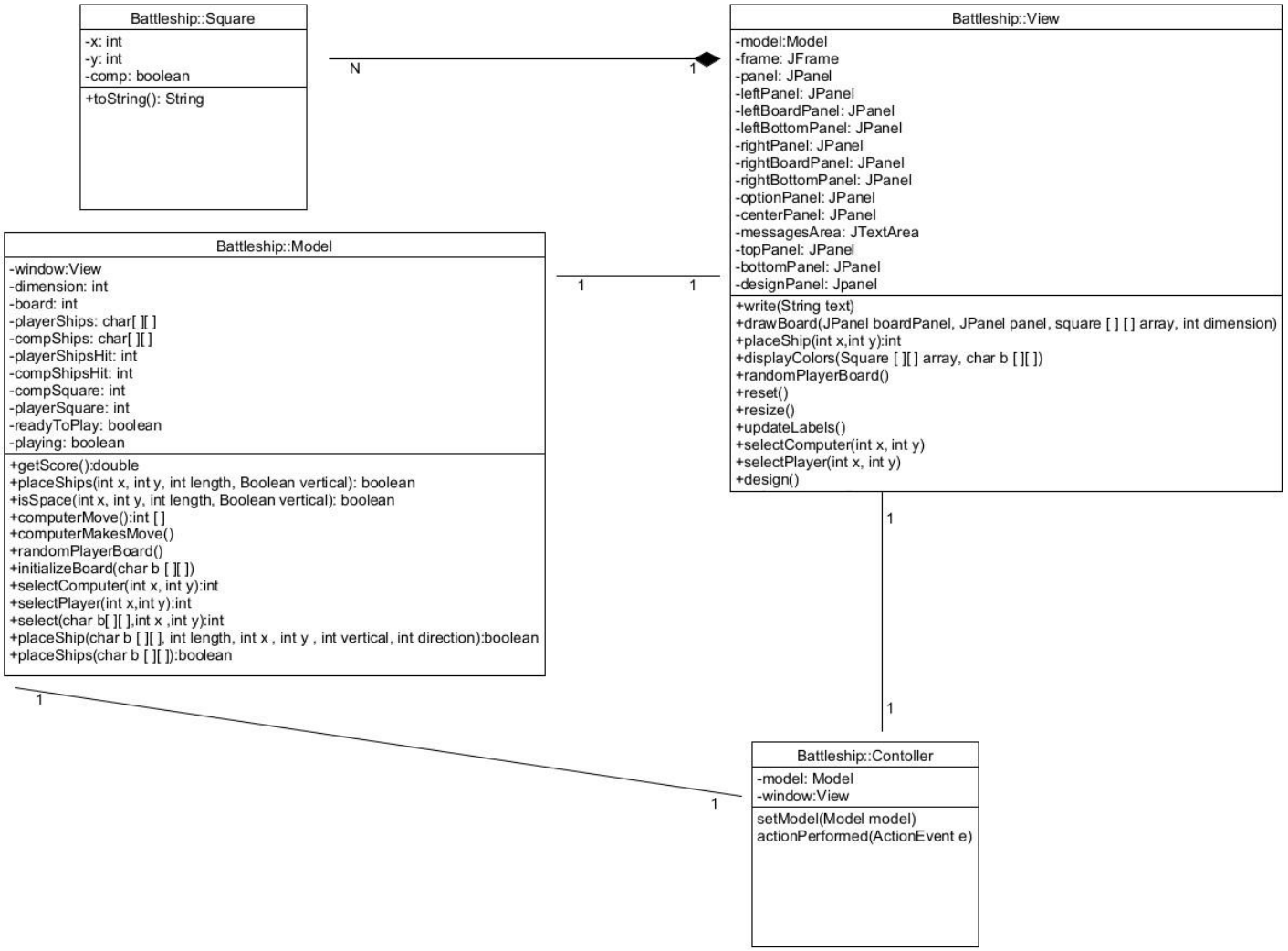
design(): when the user presses the design button.Makes the designPanel visible or invisible.

allShipsPlaced(): when the user has placed all the ships in design mode

Name
Controller: Carries out actions based on user input.
Attributes
model Model (A reference to the game model.)
window View (A reference to the view of the game.)
Methods
setModel(Model model): The setter method for the model.
actionPerformed(ActionEvent e): Reacts to user interface events.

Name
Square: A graphical representation of a game board square.
Attributes
x int (The x coordinate of the JButton on the game board.)
y int (The y coordinate of the JButton on the game board.)
comp boolean (True if the button is a computer board button.)
Methods
toString(): A text based representation of the square.

1.2. Class diagram



References

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/

Algonquin College
Spring, 2023