Programming Technology
Assignment 2 “Rubik Board”

Made by: Zhainagul Altynbek kyzy
Neptun code: v7i2jb

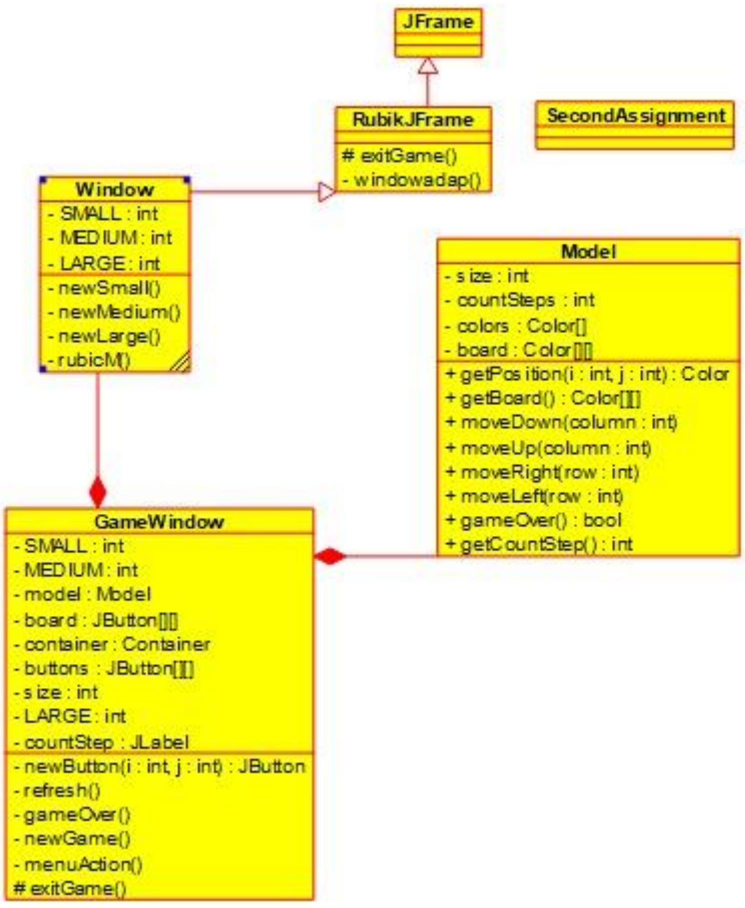
Task 1

Rubik Board

Description:

This game is a two dimension variant of the Rubik cube. The board of the game consists of n x n fields. There are n different colors, and exactly n fields have the same color. The fields are shuffled initially. A player can move cyclically the colors of a row or column (e.g. moving a row to the right, the color of the first field will be the color the last field) to make homogenous rows and columns. The game ends, when each row (or column) contains one color. Implement this game, and let the board size be selectable (2x2, 4x4, 6x6). The game should recognize if it is ended, and it has to show in a message box how much steps did it take to solve the game. After this, a new game should be started automatically.

Class Diagram:



Methods:

RubikFrame class methods:

1. exitGame – method which exits the game
2. windowadapt – Calls the exitGame() method when exit is needed

Window class methods:

1. newSmall – we override the actionPerformed button so that whenever the menu item for small is clicked it generated a new small board game
2. newMedium – We override the actionPerformed button so that whenever the menu item for medium is clicked it generated a new medium board game
3. newLarge – We override the actionPerformed button so that whenever the menu item for large is clicked it generated a new large board game
4. rubikM – We exit the window

GameWindow class methods:

1. newButton– We use this method to generate buttons around the board game this buttons are going to be use to move the corresponding row(to the right or left)or column(up or down)
2. reFresh – This function is used to refresh the board
3. gameOver – This method makes a window pop up when the game will be over informing the user and showing how many steps it took
4. newGame – This method is called when we want to start a new game
5. menuAction – We can either start a new game or quit
6. exitGame – we exit game

Model class methods:

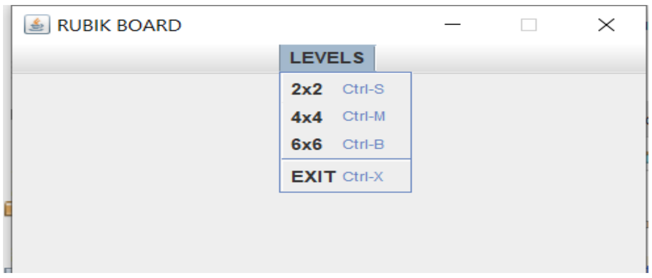
1. getPosition – this method returns the position of the current pressed button
2. getBoard – method to find if there is any winner there, it uses other method to find
3. moveDown – This method is used to move the current column up to meaning it will be the same as what we did moving a row to the right
4. moveUp – method to find if the player placed its all 4 discs vertically
5. moveRight – This method is used to move the current row to the right meaning that the last color will be the first
6. moveLeft – This method is used to move the current column up meaning it will be the same as what we did moving a row to the left
7. gameOver – This method returns weather the game has ended or not
8. getCountStep – it count how many clicks did you do

Event Handlers:

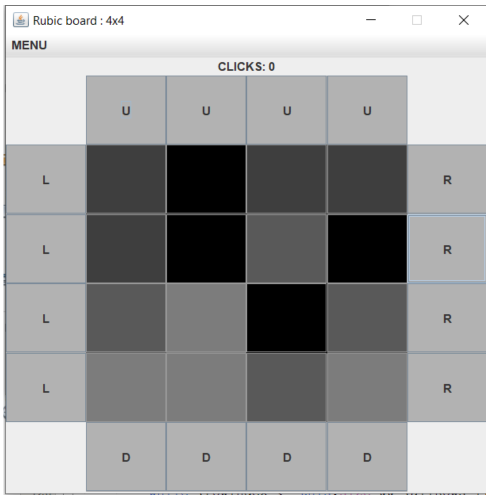
1. addActionListener – whenever medium, large, small is chosen it displays new windows with those sizes
2. getKeyStroke – user chooses one of the size option for a board, so a new game window is being created after clicking the buttons
3. addWindowListener – user can choose to continue or quit the game by clicking yes or not in confirm dialog

Testing

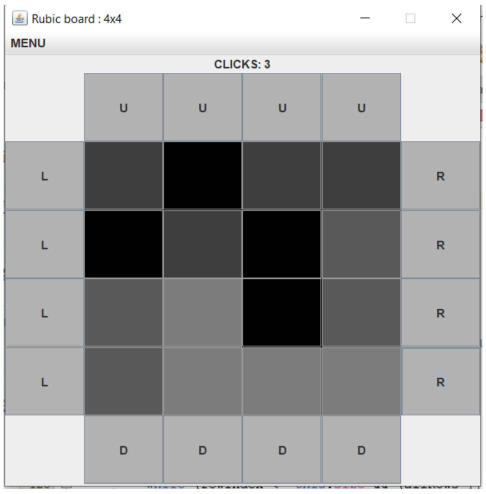
1. if it works all good, the program runs and the cells are clickable:
   Menu to choose the size of the board:



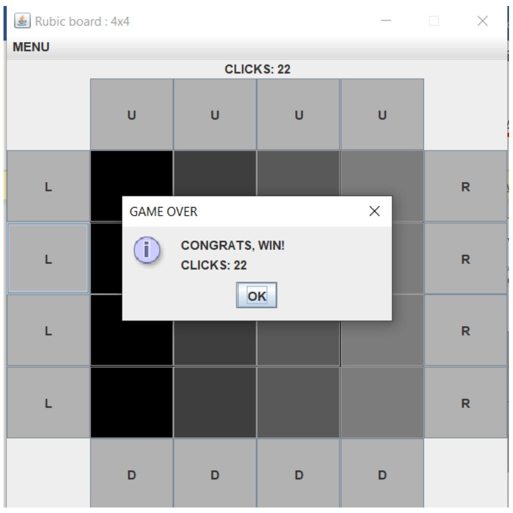
2. Start the game:



3. Rows and columns can be moved well

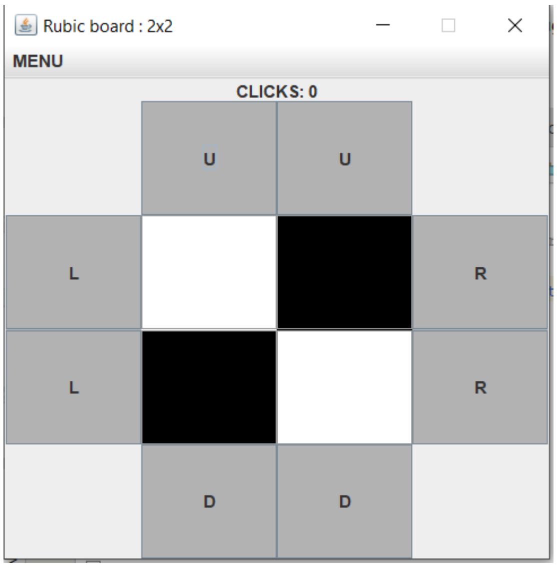


4. Win the game

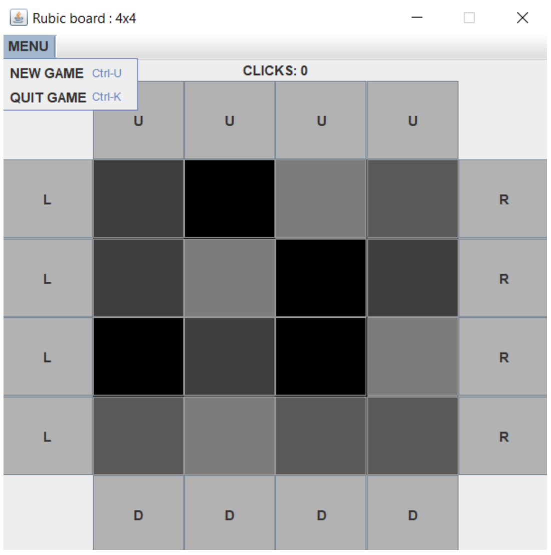


CONGRATS, WIN!
CLICKS: 22

5. Different size can be chosen



6. You can quit the game anytime


Rubic board : 4x4
MENU
NEW GAME Ctrl-U
QUIT GAME Ctrl-K
CLICKS: 0
U
U
U
U
L
L
L
L
R
R
R
R
D
D
D
D