Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrearea de Laborator#5

Group Project

Autor:
Madiudin Radu

lector asistent:
Gojin Victor

Chisinau-2017

Lucrare de laborator Nr.4

1 Scopul lucrarii de laborator

− Realizarea unui Joc de memorie utilizind limbajul Java

2 Obiective

− Coordonarea proiectului, repartizarea taskurilor, lucru in echipa si obtinerea posibilitatii de a demonstra abilitatile si munca eficienta in echipa

3 Implimentarea lucrarii de laborator

3.1 Sarcini si Obiective

-Realizarea aplicatiei MemoryGame
-Implicarea fiecarui membru de echipa

3.2 Implimentare

Primul punct in realizarea aplicatiei a fost determinarea si crearea GUI, pe baza caruia a fost adaugat functionalul programului. Utilizatorului ii este predispus la alegere nivelul de dificultate si numarul perechilor de combinatii posibile, date memorarii.
La finisarea cerintelor, utilizatorul poate alege continuarea in meniul principal sau iesirea din program.Amin

3.3 Listing Program(GameM class):

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
//import java.io.*;
import java.util.*;
//btn1.setBackground(colors[index]
public class GameM implements ActionListener {
	JFrame frame = new JFrame("Memory Game");
	JPanel field = new JPanel();
	JPanel menu = new JPanel();
	JPanel menu2 = new JPanel();
	JPanel menu3 = new JPanel();
	JPanel mini = new JPanel();
	JPanel start_screen = new JPanel();
	JPanel end_screen = new JPanel();
	JPanel instruct_screen = new JPanel();
	JButton btn[] = new JButton[20];
	JButton start = new JButton("Start");
```

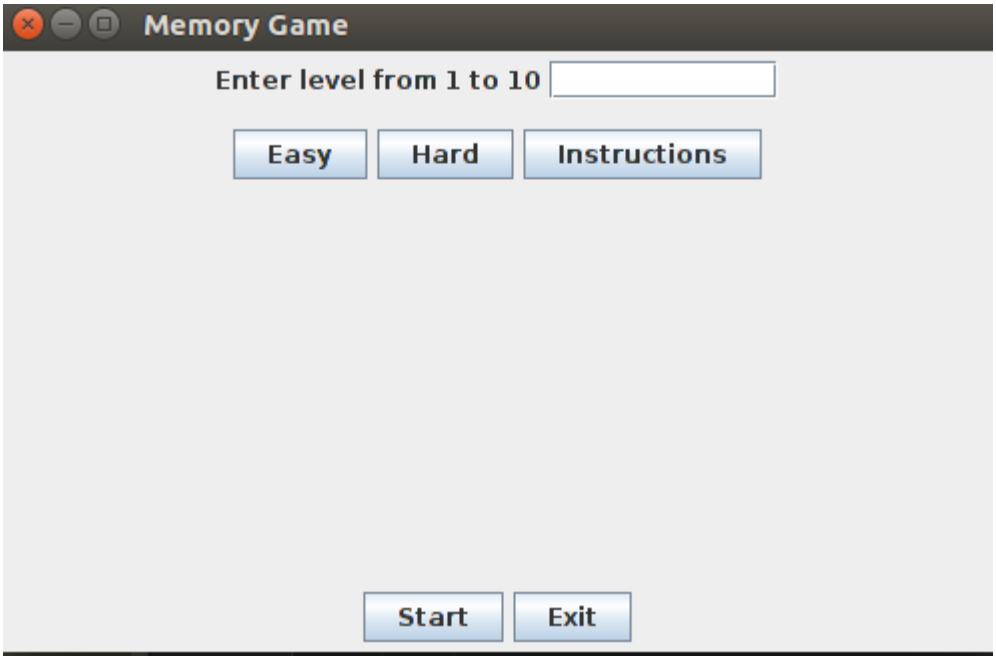
```
	JButton over = new JButton("Exit");
	JButton easy = new JButton("Easy");
	JButton hard = new JButton("Hard");
	JButton inst = new JButton("Instructions");
	JButton redo = new JButton("Play Again");
	JButton goBack = new JButton("Main Menu");
	Random randomGenerator = new Random();
	private boolean purgatory = false;
	JLabel winner;
	Boolean game_over = false;
	int level=0;
	int score=0;
	String[] board;
	int[] boardQ=new int[20];
	Boolean shown = true;
	int temp=30;
	int temp2=30;
	String a[]=new String[10];
	boolean eh=true;
	private JLabel label = new JLabel("Enter level from 1 to 10");
	private JTextField text = new JTextField(10);
	private JTextArea instructM = new JTextArea("When the game begins, the screen
will be filled\nwith pairs of buttons.\n Memorize their placement.\nOnce you press any
button, they will all clear. \n Your goal is to click the matching buttons in a row.\nWhen
you finish that, you win.\nEvery incorrect click gives you a point (those are bad).\n
GOOD LUCK! \n"+"for a single level: enter a level between 1 and 10,\nselect easy or
hard, then press start.");
	//instructM.setEditable(false);
	//instructW.setEditable(false);
	//instructM.setLineWrap(true);
	//instructW.setWrapStyleWord(true);
	public GameM(){
		frame.setSize(500,300);
		frame.setLocation(500,300);
		frame.setLayout(new BorderLayout());
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		start_screen.setLayout(new BorderLayout());
		menu.setLayout(new FlowLayout(FlowLayout.CENTER));
		menu2.setLayout(new FlowLayout(FlowLayout.CENTER));
		menu3.setLayout(new FlowLayout(FlowLayout.CENTER));
		mini.setLayout(new FlowLayout(FlowLayout.CENTER));
		start_screen.add(menu, BorderLayout.NORTH);
		start_screen.add(menu3, BorderLayout.CENTER);
		start_screen.add(menu2, BorderLayout.SOUTH);
		menu3.add(mini, BorderLayout.CENTER);
		menu.add(label);
		menu.add(text);
		mini.add(easy, BorderLayout.NORTH);
		mini.add(hard, BorderLayout.NORTH);
		mini.add(inst, BorderLayout.SOUTH);
```

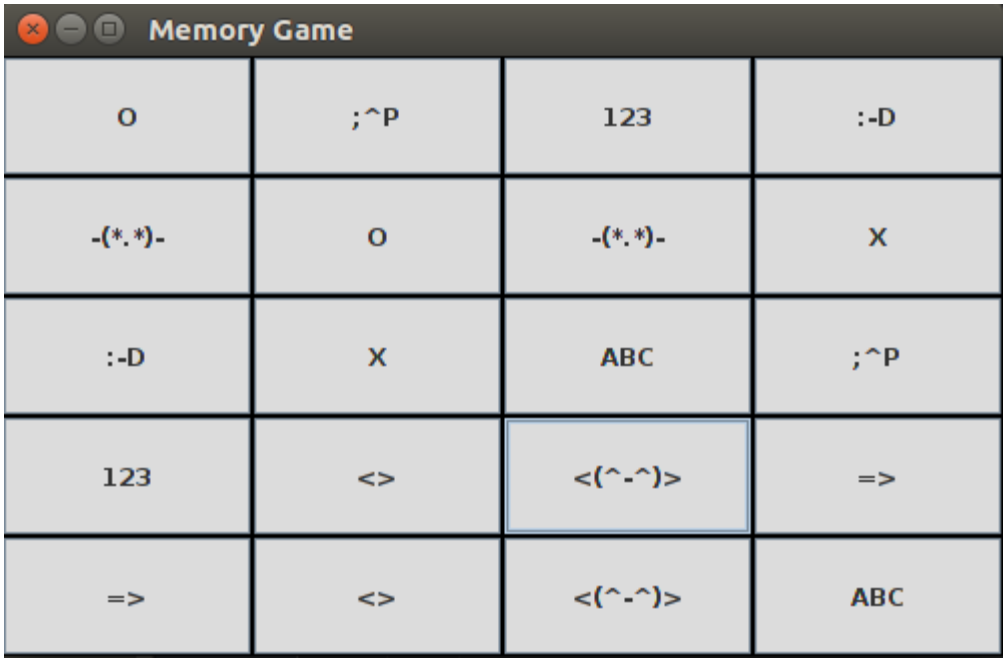
```
        start.addActionListener(this);
        start.setEnabled(true);
        menu2.add(start);
        over.addActionListener(this);
        over.setEnabled(true);
        menu2.add(over);
        easy.addActionListener(this);
        easy.setEnabled(true);
        hard.addActionListener(this);
        hard.setEnabled(true);
        inst.addActionListener(this);
        inst.setEnabled(true);
        frame.add(start_screen, BorderLayout.CENTER);
        frame.setVisible(true);
    }
    public void setUpGame(int x,Boolean what){
        level=x;
        clearMain();
        board = new String[2*x];
        for(int i=0;i<(x*2);i++){
            btn[i] = new JButton("");
            btn[i].setBackground(new Color(220, 220, 220));
            btn[i].addActionListener(this);
            btn[i].setEnabled(true);
            field.add(btn[i]);
        }
        String[] b = {":-D","X","O","-(*.*)-","<>","<(^-
^)>","=>",";^P","ABC","123"};//harder version
        String[] c =
{"square","circle","rectangle","heart","diamond","clover","spade","triangle","polygon","
tetrahedral"};//easier version
        if(what) a=c;//if what is true, make the game easy and use c
        else a=b;//otherwise make it hard and use b
        for(int i=0;i<x;i++){
            for(int z=0;z<2;z++){
                while(true){
                    int y = randomGenerator.nextInt(x*2);
                    if(board[y]==null){
                        btn[y].setText(a[i]);
                        board[y]=a[i];
                        break;
                    }
                }
            }
        }
        createBoard();
    }
    public void hideField(int x){//this sets all the boxes blank
        for(int i=0;i<(x*2);i++){
          /*if(boardQ[i]==0)*/ btn[i].setText("");
```

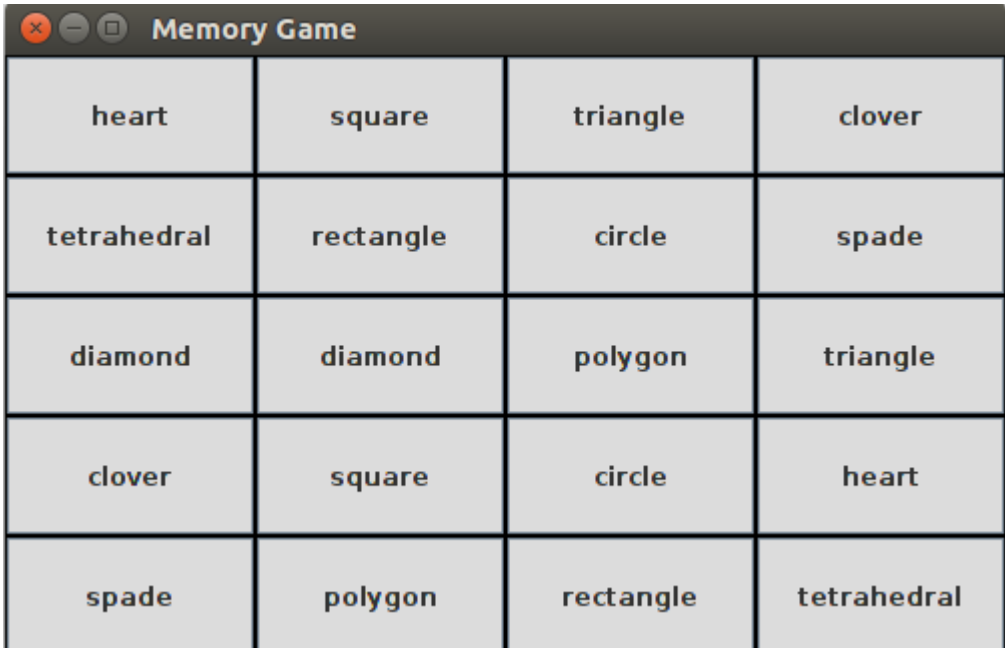
```
            }
            shown=false;
        }
        public void switchSpot(int i){//this will switch the current spot to either blank or
what it should have
            if(board[i]!="done"){//when a match is correctly chosen, it will no longer
switch when pressed
                if(btn[i].getText()==""){
                    btn[i].setText(board[i]);
                    //shown=true;
                } else {
                    btn[i].setText("");
                    //shown=false;
                }
            }
        }
        public void showSpot(int i){
            btn[i].setText(board[i]);
        }
        public void showField(int x, String a[]){//this shows all the symbols on the field
            for(int i=0;i<(x*2);i++){
                btn[i].setText(a[i]);
            }
            shown=true;
        }
        void waitABit(){//this was an attempt at fixing the glitch i told you about
            try{
                Thread.sleep(5);
            } catch(Exception e){
            }
        }
        public boolean checkWin(){//checks if every spot is labeled as done
            for(int i=0;i<(level*2);i++){
                if (board[i]!="done")return false;
            }
            winner();
            return true;
        }
        public void winner(){
            start_screen.remove(field);
            start_screen.add(end_screen, BorderLayout.CENTER);
            end_screen.add(new TextField("You Win"), BorderLayout.NORTH);
            end_screen.add(new TextField("Score: " + score), BorderLayout.SOUTH);
            end_screen.add(goBack);
            goBack.setEnabled(true);
            goBack.addActionListener(this);
        }
        public void goToMainScreen(){
            new GameM();
        }
```

```
	public void createBoard(){//this is just gui stuff to show the board
		field.setLayout(new BorderLayout());
		start_screen.add(field, BorderLayout.CENTER);
		field.setLayout(new GridLayout(5,4,2,2));
		field.setBackground(Color.black);
		field.requestFocus();
	}
	public void clearMain(){//clears the main menu so i can add the board or
instructions
		start_screen.remove(menu);
		start_screen.remove(menu2);
		start_screen.remove(menu3);
		start_screen.revalidate();
		start_screen.repaint();
	}
	public void actionPerformed(ActionEvent click){
		Object source = click.getSource();
		if(purgatory){
			switchSpot(temp2);
			switchSpot(temp);
			score++;
			temp=(level*2);
			temp2=30;
			purgatory=false;
		}
		if(source==start){ //start sets level and difficulty and calls method to set up
game
			try{
				level = Integer.parseInt(text.getText());
			} catch (Exception e){
				level=1;
			}
			setUpGame(level, eh);//level between 1 and 2, eh is true or false
		}
		if(source==over){//quits
			System.exit(0);
		}
		if(source==inst){//this just sets the instruction screen
			clearMain();
			start_screen.add(instruct_screen, BorderLayout.NORTH);
			JPanel one = new JPanel();
			one.setLayout(new FlowLayout(FlowLayout.CENTER));
			JPanel two = new JPanel();
			two.setLayout(new FlowLayout(FlowLayout.CENTER));
			instruct_screen.setLayout(new BorderLayout());
			instruct_screen.add(one, BorderLayout.NORTH);
			instruct_screen.add(two, BorderLayout.SOUTH);
			one.add(instructM);
			two.add(goBack);
			goBack.addActionListener(this);
```

```
            goBack.setEnabled(true);
        }
        if(source==goBack){//backt to main screen
            frame.dispose();
            goToMainScreen();
        }
        if(source==easy){//sets the type. ex. if easy is clicked it turns blue and hard
remains black
            eh=true;
            easy.setForeground(Color.BLUE);
            hard.setForeground(Color.BLACK);
        } else if(source==hard){
            eh=false;
            hard.setForeground(Color.BLUE);
            easy.setForeground(Color.BLACK);
        }
        for(int i =0;i<(level*2);i++){//gameplay when a button is pressed
            if(source==btn[i]){
                if(shown){
                    hideField(level);//if first time, hides field
                }else{//otherwise play
                    switchSpot(i);
                    if(temp>=(level*2)){
                        temp=i;
                    } else {
                        if((board[temp]!=board[i])||(temp==i)){
                            temp2=i;
                            purgatory=true;
                        } else {
                            board[i]="done";
                            board[temp]="done";
                            checkWin();
                            temp=(level*2);
                        }
                    }
                }
            }
        }
    }
}
```


Memory Game
Enter level from 1 to 10
Easy
Hard
Instructions
Start
Exit


Memory Game
O
;^P
123
:-D
-(*.*)-
O
-(*.*)-
X
:-D
X
ABC
;^P
123
<>
<(^-^)>
=>
=>
<>
<(^-^)>
ABC



Concluzie

In aceasta lucrare am putut dezvolta spiritul de echipa prin setarea obiectivelor bine predefinite obtinind o eficienta deosebita in realizarea proiectului.

Ca IDE a fost utilizat IntelliJ IDEA , limbaj de programare JAVA , version control system Git.