A08

LIGHTS OUT

Sindl|Rasic|Grieshofer|Baum  
3BHIT 2014/15

Inhalt

**Aufgabe3**

Aufgabenstellung3  
Lights Out3  
Umsetzung3

**Team4**

Teammittglieder4

Aufgabenverteilung4

**Bestandteile**5

GUI5

Frame5

Panel6

Controller6

Aufgabe

Aufgabenstellung

Erstellen des Spiels Lights Out.

Lights Out   
Ziel des Spiel ist es alle Lichter aus zu machen bzw. bis alle Kästchen alle Kästchen Schwarz sind.

Umsetzung  
Es soll ein Frame erstellt werden wo die Kästchen im 5\*5 Format ausgerichtet werden.   
Die Panel-Klasse enthält die Buttons.  
Die Controller-Klasse enthält die Spiellogik von Lights Out. Restart zum restarten des Spiels, Aktionen (Was passiert wenn ein Button gedrückt wird), Ändern der Buttons in einen anderen Zustand.

Team

Teammitglieder  
Sindl, Rasic, Grieshofer, Baum

Aufgabenverteilung

Grieshofer: GUI, Starter, Kommentare,

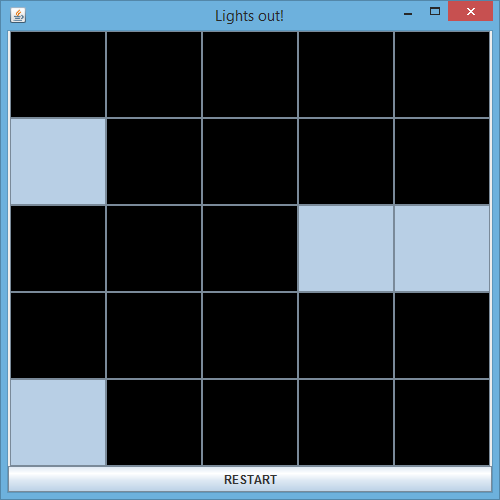
Rasic: Überarbeitung der Kommentare und des Codes

Sindl: Die übrigen gebliebenen Aufgaben, Read Me

Baum: Protokoll

Bestandteile

GUI  
Fertige GUI wie es am Ende aussehen soll



Frame  
Auszug aus dem Frame

public MyFrame(MyPanel panel){

this.setSize(500, 500);   
this.setTitle("Lights out!");   
this.setLocationRelativeTo(null); this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);   
this.add(panel);   
this.setVisible(true);

}  
}

Panel  
Auszug aus dem Panel (Konstruktor)

public MyPanel(MyControl control) {

this.control = control;   
panel1 = new JPanel();  
panel1.setLayout(new GridLayout(5, 5));

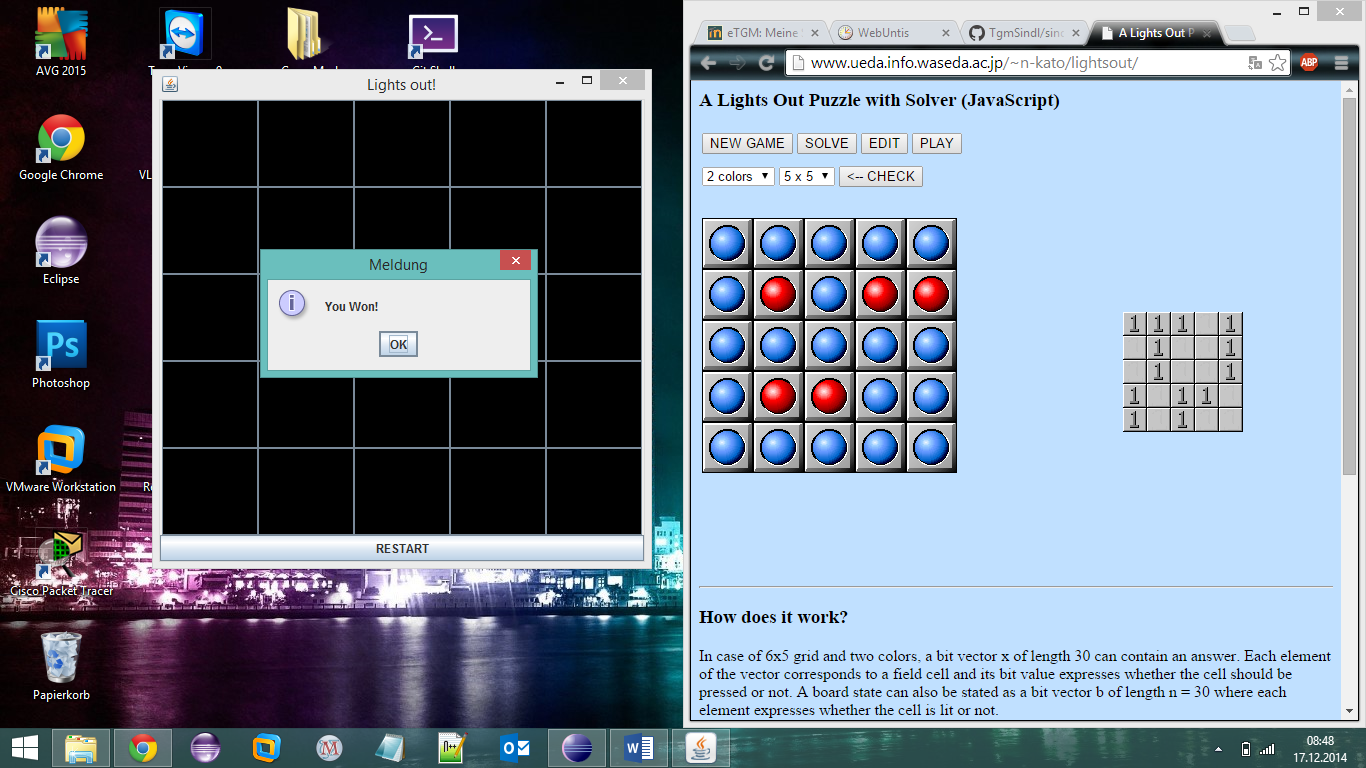
restart = new JButton("RESTART");   
restart.addActionListener(control);

buttons = new JToggleButton[5][5]; //5\*5 = 25 Felder  
JToggleButton temp = null;  
for (int i = 0; i < buttons.length; i++) {   
 for (int j = 0; j < buttons[i].length; j++) {  
 temp = new JToggleButton();  
 temp.addActionListener(control);  
 temp.setBackground(Color.black);   
 panel1.add(temp); //wird dem Panel hinzugefügt  
 buttons[i][j]=temp; //in Array speichern fuer Berechnungen

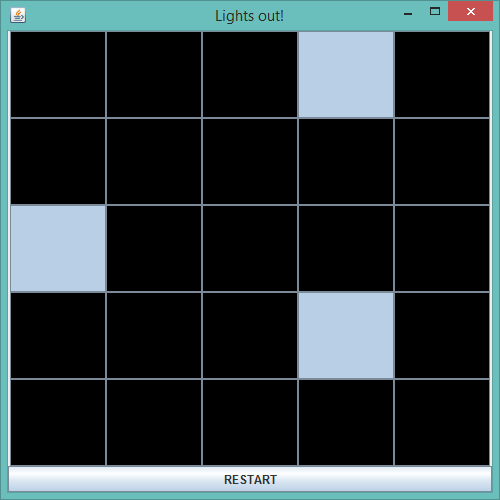
}  
 }

Wenn man gewonnen hat erscheint ein Pop-Up.

public void checkWin(){  
 int counter = 0;  
 for (int i = 0; i < buttons.length; i++) {   
 for (int j = 0; j < buttons[i].length; j++){  
 if(buttons[i][j].isSelected()==false){  
 counter++;  
 if(counter==25){  
 JOptionPane.showMessageDialog(null, "You Won!");  
 }  
 }  
 }  
 }  
}



Controller (Umsetzen der Spiellogik)  
 Hier ein Auszug zum Restarten des Spiels



/\* Wenn der Restart Button gedrückt wird sollen alle Buttons deaktiviert  
\* und danach wieder eine zufällige Anzahl angeschalten werden \*/  
Object source = e.getSource();  
if(source.equals(panel.getRestart())){   
 for (int i = 0; i < panel.getButtons().length; i++) {   
 for (int j = 0; j < panel.getButtons()[i].length; j++) {  
 panel.getButtons()[i][j].setSelected(false);   
 }

}  
panel.lightButtons();  
  
/\* Was passieren soll wen ein SpielButton gedrückt wird \*/  
}else if(source instanceof JToggleButton)   
int[] pos = findButton((JToggleButton) source);   
if(pos==null){  
System.err.println("Button nicht gefunden!");  
return;  
}