



NONOGRAMM

Ilendemli Muhammet

Rathbauer Alexander

INHALTSVERZEICHNIS

Aufgabenstellung	2
Zeitaufteilung	2
Arbeitsschritte	3
UML - Klassendiagramm	3
Model	3
View	4
Testfälle	5
Feld	5
Fehler	6
Farbe ändern	7
Lösung	7
Schwierigkeitsstufen	8
Durchgespielt	11
Probleme	11
Designer	11
Nonogramm – Logik	11
Tools	11
Designer	11
PyQT	12
PyCharm	12
Quellen	12

AUFGABENSTELLUNG

In einem Team (2) soll das Spiel Nonogramm umgesetzt werden.

- Spielfeld: 15 x 15
- Eine Statusleiste mit Anzeige der noch gesuchten Felder,
- Button zur sofortigen Lösung
- Button zum Neustart
- Auswahlfeld zur Einstellung der Schwierigkeit (EASY/200; MEDIUM/150; HARD/125; EXPERT/90; IMPOSSIBLE/50) auf Basis der gesuchten Felder!

Die Farbe rosa ist natürlich nicht Pflicht und könnte vielleicht vom User variabel eingestellt werden.

Viel Erfolg!

Ressourcen:

Unterlagen zu GUI-Programmierung in Python

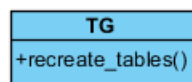
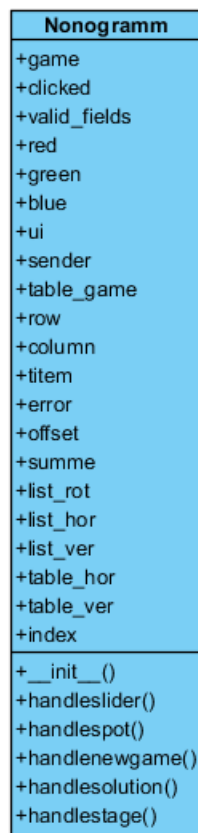
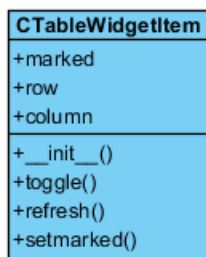
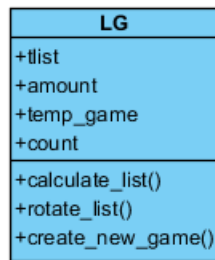
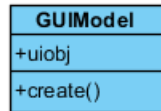
<https://de.wikipedia.org/wiki/Nonogramm>

ZEITAUFTEILUNG

Aufgaben	Geschätzte Zeit	Wirkliche Zeit	Verantwortlicher
<i>Aufgabenstellung analysieren</i>	1h 00 m	1 h 00 m	R , I
<i>Research</i>	3h 00 m	2 h 00 m	R, I
<i>Github einrichten</i>	0h 10 m	0 h 10 m	R
<i>Einarbeitung mit dem Tool</i>	2h 00 m	1 h 00 m	R
<i>Erstellung einer GUI</i>	2h 30 m	2 h 00 m	R
<i>Einbinden von dem UI File in den Pythoncode</i>	1h 00 m	1 h 00 m	I
<i>Kopellung von GUI mit Controller</i>	2 h 00 m	1 h 00 m	I
<i>Schwierigkeitsstufen</i>	3 h 00 m	2 h 00 m	I
<i>Befüllen des Spielfeldes</i>	2 h 00 m	1 h 00 m	I
<i>Berechnung der Zahlen</i>	2 h 00 m	2 h 00 m	I
<i>Lösung</i>	1h 00 m	1 h 00 m	R
<i>Protokoll</i>	2h 00 m	1 h 00 m	R
<i>Zusammenfassen</i>	21 h 30 m	15 h 00 m	

ARBEITSSCHRITTE

UML - KLASSENDIAGRAMM



MODEL

```
class GUIModel(object):
    @classmethod
    def create(cls, parent):
        ui_class, base_class = uic.loadUiType("Nonogramm.ui")

        uiobj = ui_class()
        uiobj.setupUi(parent)

        return uiobjLösung
```

VIEW

```
class CTableWidgetItem(QTableWidgetItem):
    marked = False

    def __init__(self, row, column):
        QTableWidgetItem.__init__(self)

        self.row = row
        self.column = column

    def toggle(self):
        if self.marked:
            self.marked = False
        else:
            self.marked = True

        self.refresh()

    def refresh(self):
        if self.marked:
            self.setBackground(QColor(100, 100, 150))
        else:
            self.setBackground(QColor(255, 255, 255))

    def setmarked(self, marked):
        self.marked = marked
        self.refresh()
```

TESTFÄLLE

FELD

Hier sieht man wie unser Spielfeld aufgebaut ist.

Form

1	8	1	8	4	6	2	7	1	
8	3	4	1	9	6	13	4	2	2
4	2	15	8	14	4	5	3	1	3

15									
2	8	2							
15									
15									
7	2	4							
15									
3	5	5							
10	3								
1	3	9							
6	6	1							
4	4	3							
7	1	2	2						
1	13								
8	1	4							
4	8	1							

offene Felder: 200 falsche Felder: 0 Lösung LEICHT / 200 Neustart

FEHLER

Hier wurde getestet ob das Feld „Falsche Felder“ erhöht wird wenn ein Falsches Feld aktiviert wird.

Form

The screenshot shows a Minesweeper game window titled "Form". The main grid is 15x15. The top 10 rows are mostly empty, with some numbers appearing in the bottom rows. The bottom 5 rows of the main grid are filled with numbers. To the right of the main grid is a 10x10 grid. The status bar at the bottom shows "offene Felder: 50" and "falsche Felder: 15". The status bar also contains buttons for "Lösung", "UNMÖGLICH / 50", and "Neustart".

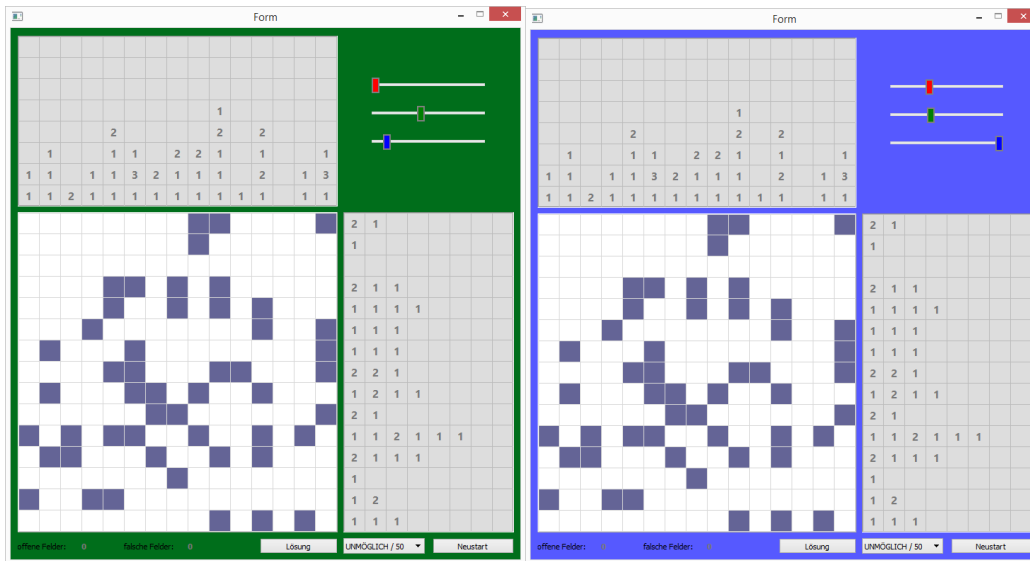
1	1	1			3		1		1	1				
3	1	1			3	2	2		1	1	2	2	1	1
1	1	1	1	2	2	1	1	1	1	1	1	1	1	1

1	1	1							
2	1								
1	1	1							
2									
1	1	2							
1	2	1							
1	1	1							
1	1	1	1						
1	1	1	1	2					
1	1	1	2						
1									
1	1								
1	1	1	1						
1	2	1							
1	1								

offene Felder: 50 falsche Felder: 15 Lösung UNMÖGLICH / 50 Neustart

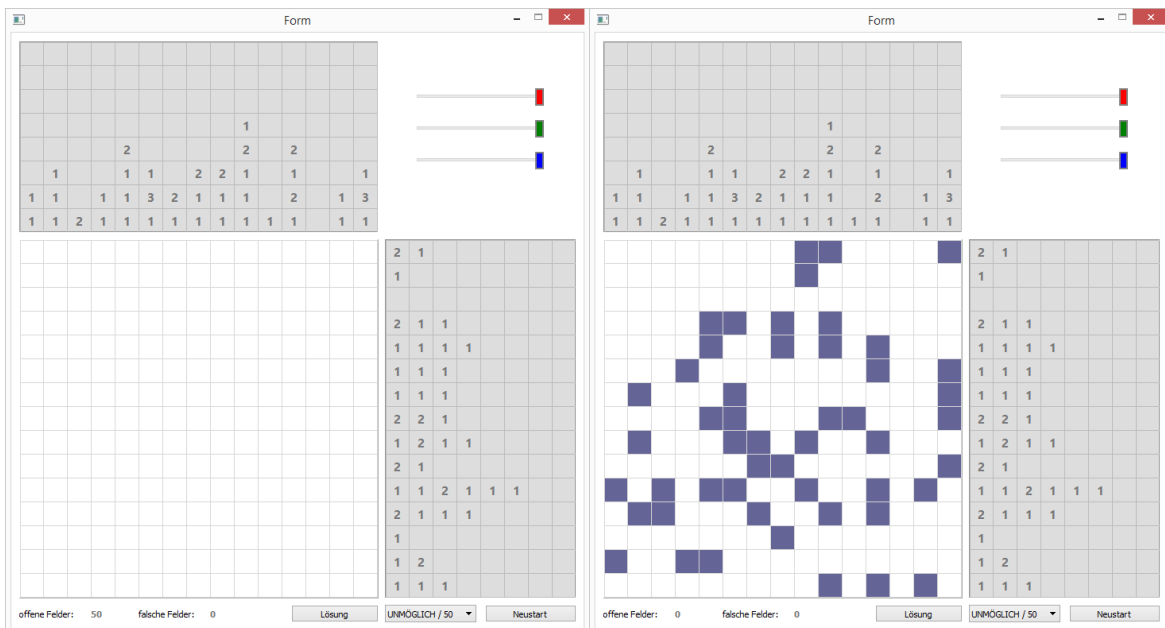
FARBE ÄNDERN

Hier wurde getestet ob es Möglich ist durch die Drei Slider, die Farbe des Spielfeldes zu verändern.



LÖSUNG

Hier wurde der Lösungsbutton getestet, das Spiel zählt dannach nicht als Gewonnen.



SCHWIERIGKEITSSTUFEN

Hier werden die Felder mit ihren 5 verschiedenen Schwierigkeitsstufen aufgerufen.

Leicht:

Form

								4							
								2							
1	8		1		8		4		6	2	7	1			
8	3		4		1	9	6	13	4	2	2	9	9		
4	2	15	8	14	4	5	3	1	3	1	4	3	4	15	

15															
2	8	2													
15															
15															
7	2	4													
15															
3	5	5													
10	3														
1	3	9													
6	6	1													
4	4	3													
7	1	2	2												
1	13														
8	1	4													
4	8	1													

offene Felder: 200

falsche Felder: 0

Lösung

LEICHT / 200

Neustart

Mittel:

[illegible]

Schwer:

Form

3	1	2					
2	1	2	5				
2	1	2	1				
1	1	1	1	1	1		
1	1	3	1				
4	1	1	3	1			
5	1	2					
2	2	3	2				
1	4	1	1	1	1		
2	4	2					
1	1	4	2				
1	4	3	3				
1	2	1	3	1			
4	2	3	1				
1	2	5	2				

offene Felder: 125 falsche Felder: 0

Lösung

SCHWER / 125 ▾ Neustart

Expert:

Form

										1				
		1					1			1	1			
1	4	1		1			1	3	1		1	1	2	
2	1	1		3			1	2	2	5	1	1	1	2
2	2	1	1	1	5	1	1	2	2	1	1	2	1	1
2	1	1	1	5	1	1	2	1	2	2	1	1	1	1

1	2	1	2			
1	1	1	2	2		
2	2	2	1	1	1	
1	3	4				
2	2	1	2			
1	1	3				
2	1	1	1			
1	1					
1	1	1	1			
1	3	1	2	1		
1	1	2	1			
2	1	1	1	1		
1	1	1	1	2	1	
1	1	2				
1	3	3				

offene Felder: 90

falsche Felder: 0

Lösung

EXPERTE / 90 ▾

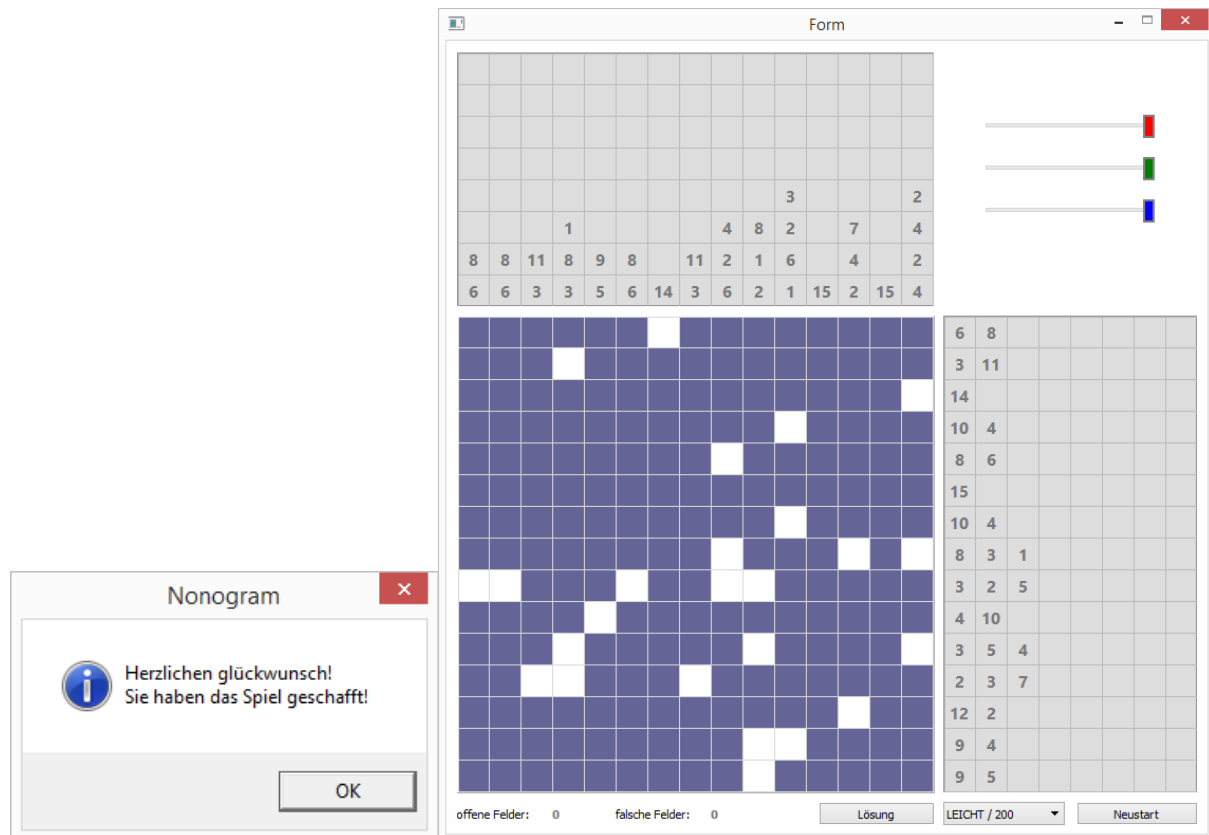
Neustart

Unmöglich:

[illegible]

DURCHGESPIELT

Nachdem man das Spiel beendet hat taucht eine Textnachricht auf.



PROBLEME

DESIGNER

Da uns das Programm Designer, welches wir für unsere GUI benutzten am Anfang noch völlig unbekannt war. Entstanden am Anfang Probleme da man nicht wusste wie dieses Programm funktioniert. Nach ein paar Tutorials im Internet war dieses Problem jedoch behoben.

NONOGRAMM – LOGIK

Nonogramm ist ein Spiel das aus dem Östlichen Bereich der Welt kommt (Japan). Da dieses Spiel kaum bekannt im Europäischen Bereich ist, und wir es daher kaum kennen, war die Spielelogik hinter diesem Spiel nicht von Anfang an bekannt. Nach mehreren Stunden nachforschen und auch ausprobieren dieses Spiel, war die Spielelogik soweit bekannt, das man dieses Spiel umsetzen konnte.

TOOLS

DESIGNER

“Qt Designer is Qt's tool for designing and building graphical user interfaces (GUIs) from Qt components. You can compose and customize your widgets or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and test them using different styles and resolutions.

Widgets and forms created with Qt Designer integrated seamlessly with programmed code, using Qt's signals and slots mechanism, that lets you easily assign behavior to graphical elements. All properties set in Qt

Designer can be changed dynamically within the code. Furthermore, features like widget promotion and custom plugins allow you to use your own components with Qt Designer”[3]

PYQT

“PyQt is a set of Python v2 and v3 bindings for Digia's Qt application framework and runs on all platforms supported by Qt including Windows, MacOS/X and Linux. PyQt5 supports Qt v5. PyQt4 supports Qt v4 and will build against Qt v5. The bindings are implemented as a set of Python modules and contain over 620 classes.

Digia have announced that support for Qt v4 will cease at the end of 2015. PyQt5 and Qt v5 are strongly recommended for all new development.

PyQt is dual licensed on all supported platforms under the GNU GPL v3 and the Riverbank Commercial License. Unlike Qt, PyQt is not available under the LGPL. You can purchase the commercial version of PyQt here. More information about licensing can be found in the License FAQ.

PyQt does not include a copy of Qt. You must obtain a correctly licensed copy of Qt yourself. However, a binary Windows installers of the GPL version of both PyQt5 and PyQt4 are provided and this includes a copy of the LGPL version of Qt.”[1]

Weitere Informationen welche wir verwendet haben, befindet sich auf [4].

PYCHARM

“PyCharm is one of the most popular Python IDEs and deservedly so. It’s packed with features such as incredible code completion, code analysis, code navigation, top-notch Django, JavaScript, HTML, and CSS support, great debugger, and much more.”[2]

QUELLEN

- [1] <http://www.riverbankcomputing.com/software/pyqt/intro> zuletzt gesehen: 09-02-2015
- [2] <http://pedrokruger.net/choosing-best-python-ide/> zuletzt gesehen: 10-02-2015
- [3] <http://qt-project.org/doc/qt-4.8/designer-manual.html> zuletzt gesehen: 10-02-2015
- [4] <http://pyqt.sourceforge.net/Docs/PyQt5/index.html> zuletzt gesehen : 09-02-2015
- [5] <https://wiki.python.org/moin/PyQt> zuletzt gesehen: 10-02-2015