

# Dokumentation zu



Von Lukas Jesche  
Klasse 10/2

## KURZREFERAT

Es ist wohl eines der berühmtesten Zahlenrätsel überhaupt: das Sudoku. Es ist auf keine Sprache festgelegt und jeder kann es lösen. Doch schon seltener zu finden und noch schwieriger zu lösen sind größere Sudoku's, welche fast jedes beliebige Format erreichen können. Ein eventuell weniger bekannte und verbreitete Art ist das HexaDoku. Bei dieser Sonderform ist nicht nur die Größe anders, sondern es verwendet statt unserem alltagsgebräuchlichen dekadischen Zahlensystem das Hexadezimale, wodurch wie beim normalen Sudoku nur eine Ziffer pro Zelle gefüllt wird. Um dieses 16 \* 16 Felder große Abwandlung eines Sudoku's zu lösen ist es Ziel dieser Arbeit und des Programms eben dieses dem Nutzer zu ermöglichen. Als Anreiz für den Spieler soll dabei noch die Zeit gestoppt und so der Ehrgeiz geweckt werden. Dabei soll dem Nutzer natürlich auch viele Einstellungsmöglichkeiten eingeräumt und das Lösen einfach gemacht werden. Dabei soll natürlich das gesamte Programm portable bleiben um dem Nutzer überall unterwegs die Nutzung zu ermöglichen. Die Einstellungen, HighScores und gespeicherten Spiele so zu speichern, dass der Nutzer sie ebenfalls portable nutzen kann ist dabei selbstverständlich.

## ABSTRACT

It is definitely one of the famous puzzle: the Sudoku. It is not written in only one language and everybody can solve it. But rarer to find and even much harder to solve are bigger Sudoku's which can reach nearly every Format. One maybe barely known and not found everywhere is the HexaDoku. In this Special Form not only the size is different, it uses the hexadecimal System instead of the decimal number System which is used everywhere and in which only one number per line is filled in just like in the normal Sudoku.

To solve this 16 times 16 fields big Sudoku it is the Goal of this work and this Program to make this possible for the user.

To motivate the Player the time should be stopped and so he is under pressure to solve it. Therefore the user should get so many possibilities so that he really wants to solve it and many possibilities should be there to solve it. All this possibilities should be portable and should make it also possible to Play from everywhere else. The high scores and the back up games should be possible to save so that the user can also use portables which is absolutely normal.

# INHALTSVERZEICHNIS FÜR HEXADOKU - DAS SPIEL

Kurzreferat .....	0
Abstract.....	0
1. Einleitung.....	1
2. Ergebnisse.....	3
2.1. Erreichte Ziele.....	3
2.1.1. Spielen bzw Lösen des oder der HexaDoku's.....	3
2.1.2. Hilfen an den Nutzer beim Lösen.....	3
2.1.3. Laden und Speichern eines angefangenen Spieles.....	4
2.1.4. Die eigene aktuelle Zeit und die Pause.....	4
2.1.5. Die HighScore- bzw. Bestenliste.....	4
2.1.6. Eigene Farben festlegen.....	5
2.1.7. Windows HTML-Hilfdatei mit Index und Volltextsuche.....	5
2.2. Nicht erreichte Funktionen.....	6
2.2.1. beliebige HexaDoku's Lösen.....	6
2.2.2. HexaDoku's generieren.....	7
2.2.3. Verschiedene Benutzerprofile.....	7
2.3.4. Packen der Datei in ein zwei Dateien Projekt.....	7
3. Methoden.....	8
3.1. Spielen bzw Lösen des oder der HexaDoku's.....	8
3.2. Hilfen an den Nutzer beim Lösen.....	10
3.3. Laden und Speichern eines angefangenen Spieles.....	11
3.4. Die eigene aktuelle Zeit und die Pause.....	12
3.5. Die HighScore- bzw. Bestenliste.....	13
3.6. Eigene Farben festlegen.....	14
3.7. Windows HTML-Hilfdatei mit Index und Volltextsuche.....	15
3.8. Die HexaDoku's.....	16
4. Diskussion der Ergebnisse.....	17
Quellenverzeichnis.....	18
Websites: .....	18
Fremdquelltext:.....	18
Programme:.....	18
Selbständigkeitserklärung.....	19

# 1. EINLEITUNG

Jeder kennt sie, denn sie sind in fast jeder Zeitschrift und das wohl bekannteste Japanische Zahlenrätsel überhaupt: Sudokus. Sie sind bekannt, sie sind einfach und sie bedürfen nur des logischen Denkens. Es wird kein Wissen oder Rechenkünste benötigt. Doch genauso stark wie seine Popularität stieg, genauso sind die Anzahl der Varianten in die Höhe geschossen. Gab es anfangs nur das klassische Sudoku, so kamen im Lauf der Zeit die verschiedensten Varianten mit den unterschiedlichsten Regeln dazu. Eins der weniger bekannten ist das "Fudschijama", welches statt aus 9x9 Feldern mit je 3x3 Blöcken aus 16x16 Felder mit 4x4 Blöcken besteht. Doch noch weit weniger bekannt ist eine Variante des "Fudschijama", welches dazu noch in ein anderes Zahlensystem umsteigt, in das Hexadezimalsystem. Das Nützliche dabei ist, dass jede dann eintragbare Zahl genau wie beim normalen Sudoku nur aus einer Ziffer besteht und so übersichtlicher wird. Und die Variante, die dies in sich vereint ist das HexaDoku. Das Wort setzt sich aus dem ersten Teil von Hexadezimalsystem und dem zweiten Teil von Sudoku zusammen. Genau wie das klassische Sudoku sind die Regeln einfach und schnell erklärt. Alle Zahlen von 0 bis F (Hexadezimalsystem) müssen ohne doppelt in einer Spalte, einer Spalte oder einem Block vorkommen, entsprechend der schon vorhandenen eingesetzt werden, sodass jedes freies Feld am Ende mit einer Zahl belegt ist. Jede Spalte, jede Zeile und jeder Block sind also eine Einheit, in welcher die Zahlen von 0 bis F genau einmal vorkommen. Das Ganze klingt komplizierter als es in der Praxis ist, denn dort bedeutet es Zahlen, welche noch nicht in einer dieser Einheiten stehen zu ergänzen. Dabei zu beachten keine Zahlen doppelt zu schreiben, ist jedoch einfacher als die Zahlen herauszufinden, welche in eine Zelle können. Dieses Programm soll nun zum einen dem Nutzer helfen HexaDoku's zu lösen, aber ihn auch anreizen es schnell und bis zum Ende zu lösen. Dabei muss natürlich eine große individuelle Einstellungsmöglichkeit für den Nutzer bestehen, als auch ihm eine sehr übersichtliche und intuitive Oberfläche zu bieten. Diese sollte sowohl von der Erscheinungsform, als auch von ihrer Bedienbarkeit und Nutzbarkeit überzeugen.

Der Nutzer soll auf keinen Fall erst groß Hilfe-Texte lesen, Tutorials durchlaufen und sich überfordert fühlen, sondern soll sich einfach intuitiv und logisch sofort beginnen und lösen können. Dies alles wird bei den Ergebnissen der Arbeit vorgestellt. Wie diese genau im Programm implementiert wurden, ist dann bei den Methoden zu lesen.

Wie genau dann aber die Ergebnisse und die Ziele ausgefallen sind und welche Gründe es dafür gibt, wird anschließend im letzten Teil der Arbeit diskutiert.

## **2. ERGEBNISSE**

### ***2.1. ERREICHTE ZIELE***

#### **2.1.1. SPIELEN BZW LÖSEN DES ODER DER HEXADOKU'S**

Der Nutzer kann um ein Spiel zu beginnen, einfach eine von 6 verschiedenen Schwierigkeiten wählen und anschließend wird zufällig eines der im Spielordner vorhandenen HexaDoku's der entsprechenden Schwierigkeit eingelesen und geöffnet.

Anschließend kann der Nutzer damit beginnen das HexaDoku zu lösen. Falls ihm dies gelingt wird er darüber informiert und kann seine Zeit speichern (mehr dazu unter dem Punkt: 'Die HighScore- bzw. Bestenliste').

#### **2.1.2. HILFEN AN DEN NUTZER BEIM LÖSEN**

Um den Nutzer das Lösen seines HexaDoku's möglichst einfach zu gestalten, sind einige Hilfen eingebaut. So wird das aktuelle Feld hervorgehoben, damit sich der Nutzer orientieren kann, wo er ist. Außerdem wird ihm die aktive Spalte, Reihe und der aktive Block angezeigt. Dies erhöht die Orientierung ebenfalls. Damit der Nutzer komplett mit der Tastatur das HexaDoku lösen kann, ist es ebenfalls möglich nicht nur sein aktives Feld über die Maus zu bestimmen, sondern es ebenfalls komplett über die Tastatur zu steuern.

Ebenfalls mit eingebaut ist eine Funktion, mit welcher der Nutzer bequem alle Zahlen angezeigt bekommt, welche noch in ein Feld eingetragen werden können. Dies erleichtert dem Nutzer den Lösungsvorgang.

Eine weitere Hilfe ist, bei Problemen seine Lösungen überprüfen, und sich seine Fehler anzeigen lassen zu können.

### **2.1.3. LADEN UND SPEICHERN EINES ANGEFANGENEN SPIELES**

Da das Lösen eines kompletten HexaDoku's sehr viel Zeit in Anspruch nimmt, wird dem Nutzer die Möglichkeit gegeben sein aktuelles HexaDoku zu speichern und später wieder zu laden. Dabei wird natürlich auch seine Zeit mitgespeichert. Wo der Nutzer sein Spiel speichern möchte, ist dabei ihm überlassen, allerdings wird der Spiele-Ordner empfohlen, damit der Nutzer nur diesen kopieren muss um überall spielen zu können.

### **2.1.4. DIE EIGENE AKTUELLE ZEIT UND DIE PAUSE**

Der Nutzer hat die Möglichkeit während des gesamten Lösungsprozesses seine aktuelle Zeit zu betrachten und so genau zu wissen wie lange er schon löst.

Dabei hat er jederzeit die Möglichkeit das Spiel zu pausieren. Falls der Spieler das Programm minimiert, wird die Pause natürlich auch aktiviert. So geht dem Nutzer keine Zeit verloren, welche beim erfolgreichen Lösen für die HighScoreliste genutzt wird (siehe nächster Punkt: 'die HighScore- bzw. Bestenliste').

### **2.1.5. DIE HIGHSCORE- BZW. BESTENLISTE**

Ob als Anreiz, als Trainingzeitensammlung oder einfach um bei seinen Freunden angeben zu können, schneller als dieser ein HexaDoku zu lösen, eine HighScoreliste ist vielseitig. Die Liste lässt sich jeder Zeit anschauen und wird natürlich sortiert und die Platzierung angegeben. Damit der Nutzer sich später orientieren kann wer denn welche Zeit hat, kann dieser natürlich beim Erreichen einer neuen Zeit seinen Name beziehungsweise sein Kürzel hinterlassen. Die HighScore's werden natürlich gespeichert.

### **2.1.6. EIGENE FARBEN FESTLEGEN**

Um den Nutzer eine nicht nur nützliche, sondern auch individuelle Oberfläche zu bieten sind im Einstellungsmenü die Möglichkeiten (fast) jede Farbe der Spieloberfläche selbst auszuwählen und anzupassen. Damit lässt sich für einen Benutzer seine eigene Oberfläche kreieren, genau nach seinen Wünschen. Die Farben werden selbstverständlich gespeichert und beim Programmstart geladen.

### **2.1.7. WINDOWS HTML-HILFEDATEI MIT INDEX UND VOLLTEXTSUCHE**

Um den Nutzer bei Fragen oder Unklarheiten zu beantworten beziehungsweise sie zu klären ist dem Programm eine Hilfe beigelegt. Diese ist mit dem Microsoft HTML Help Workshop erstellt und sollte deshalb auf jedem Windows lesbar und nutzbar sein. Diese Hilfe verfügt dann über einen Index und eine Volltextsuche welche bei Problemen hilfreich ist.



## **2.2. NICHT ERREICHTE FUNKTIONEN**

### **2.2.1. BELIEBIGE HEXADOKU'S LÖSEN**

Zum Lösen von HexaDoku's lassen sich prinzipiell jegliche Algorithmen von Sudoku's übernehmen, da die Regeln praktisch dieselben sind. Nur der Unterschied, der unterschiedlichen Größen muss dabei beachtet werden.

Doch genau wegen dieser Größe beginnen schon die ersten Methoden um ein HexaDoku zu lösen wegzufallen. Mit einem einfachen Backtrack Algorithmus, welcher praktisch durch probieren und Try und Error's jede Mögliche Lösung finden kann, funktioniert bei Sudoku's noch recht schnell. Dort sind die  $9 * 9$  großen Rätsel in wenigen Sekunden gelöst. Da aber das Verfahren des Backtracking mit jedem Feld, was dazu kommt exponentiell an Rechenleistung und damit auch an Rechendauer steigt, sind die Zeiten für HexaDoku's zum bestimmen jeder Lösung viel zu langsam. Auch die Möglichkeit spezielle Backtracking Algorithmen zu nutzen, welche im Schnitt schneller eine Lösung liefern, indem sie gezielter probieren scheidet aus, da dem Nutzer mitunter trotzdem sehr lange Wartezeiten nicht zumutbar sind.

Ein anderer Algorithmus zum schnellen Lösen von Sudoku's und auch zum schnellen Lösen von HexaDoku's, ist die Dancing Links Variante (siehe Quellenverzeichnis: 'Dancing Links'). Diese Variante ist, aber sehr kompliziert und langwierig zu programmieren, weshalb sie ebenfalls nicht erreicht wurde.

Die letzte Variante, die ich hier anbringen möchte ist einfach einige mögliche Strategien, welche benutzt werden können um Sudoku's zu lösen einfach als Algorithmus für das Programm zu schreiben. Anschließend müssten diese dann bloß immer wieder aufgerufen werden. Bei dieser Variante ist dann sogar noch die Angabe des Lösungsweges möglich. Auch eine Bewertung eines Sudoku's ist so vorstellbar. Allerdings ist diese Variante auch komplex, weshalb sie nicht vorhanden ist.

### **2.2.2. HEXADOKU'S GENERIEREN**

Die schwierigste Hürde beim Generieren von Sudoku's und damit auch beim Generieren von HexaDoku's ist die Tatsache, dass geltende Sudoku's eindeutig sein müssen, was bedeutet, dass nachdem wie auch immer Zahlen auf einem leeren Sudoku verteilt sind, man überprüfen muss das es auch eindeutig ist. Und dieser Schritt bedeutet alle Möglichen Lösungen bestimmen zu können, was bei HexaDoku's entweder nur sehr langsam oder sehr kompliziert möglich ist. Um ein HexaDoku, aber beispielsweise zu erzeugen könnte man das gesamte HexaDoku mit zufälligen Ziffern füllen. Dies ist die Lösung. Um nun das eigentliche HexaDoku daraus zu erhalten, entfernt man solange zufällig Zahlen wie es noch eine eindeutige Lösung gibt. Da dies mit einfacheren Mittel wie erklärt nicht schnell genug geht wurde darauf verzichtet.

### **2.2.3. VERSCHIEDENE BENUTZERPROFILE**

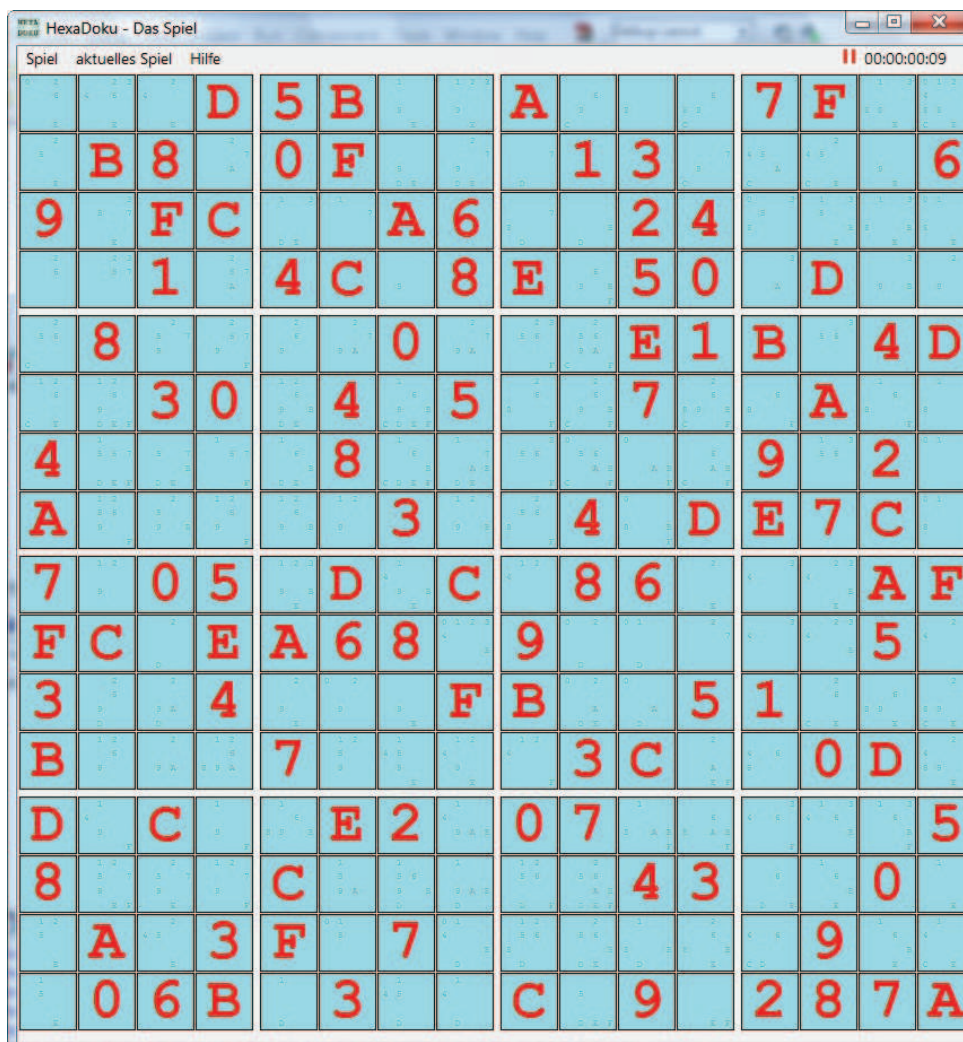
Um verschiedene Einstellungen in Profilen speichern zu können würde es theoretisch reichen mehrere Dateien zu speichern. Da aber häufig jede Person nur ein Profil benötigt und es bei mehreren Personen aufgrund der geringen Programmgröße keine große Rolle spielt wurde die Funktion nicht eingebaut. Die Arbeit für ein fast nie benutztes Feature wäre zu groß.

### **2.3.4. PACKEN DER DATEI IN EIN ZWEI DATEIEN PROJEKT**

Theoretisch wäre es möglich den gesamten Spielordner zu einer Zip zu packen und sie nur zur Laufzeit zu entpacken. Dies hätte den Vorteil eines geringeren Speicherplatzbedarfs und des einfacheren Handlings beim Kopieren für den Nutzer. Allerdings wird zum (un)zippen .dll's benötigt, welche am Ende der .exe Datei des Programms eingefügt werden müsste um nicht noch zusätzlich immer bereitgehalten werden müsste. Da dies allerdings nicht trivial und vor allem nachträglich nach dem Kompilieren angefügt werden müsste ist es eingebaut worden.

### 3. METHODEN

#### 3.1. SPIELEN BZW LÖSEN DES ODER DER HEXADOKU'S



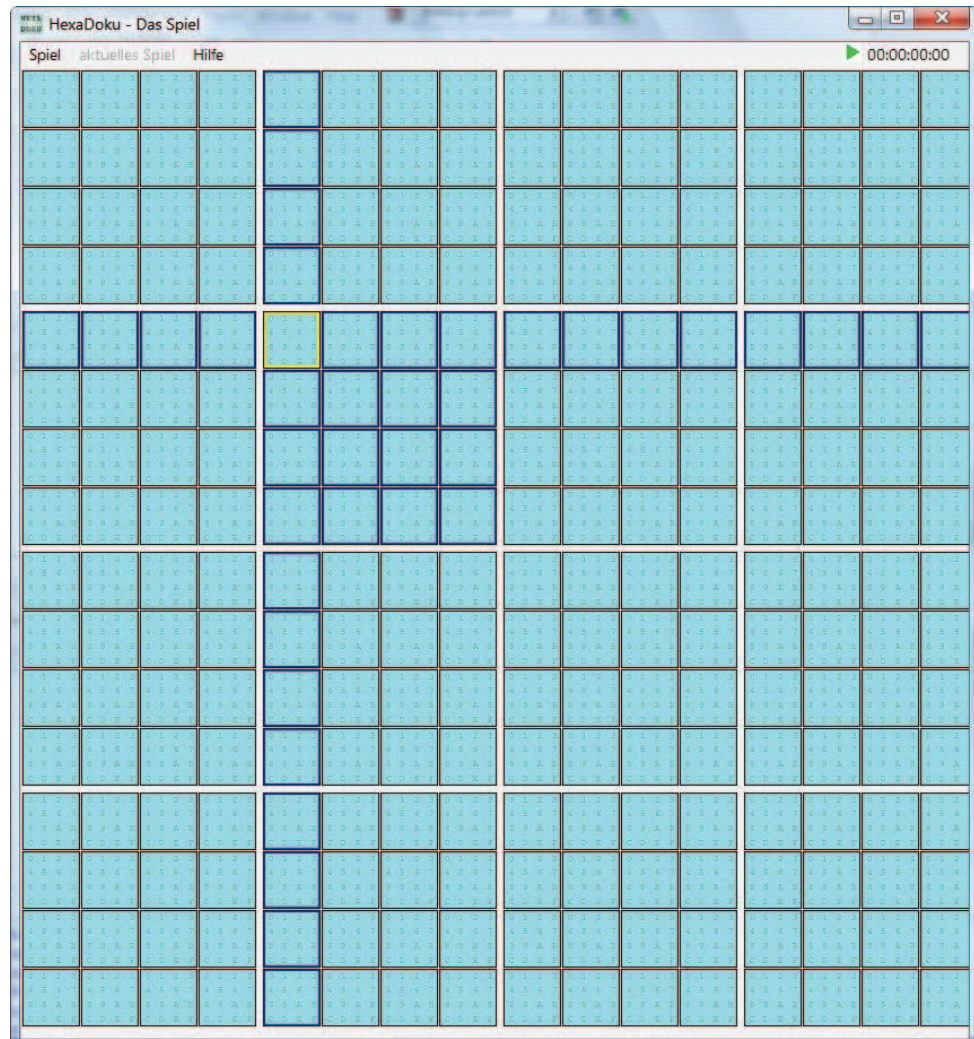
ScreenShot 1: Die Benutzeroberfläche während eines Spiels

Wie in der Abbildung oben zu sehen ist, ist das Formular vollkommen ausgefüllt, um keinen Platz zu verschenken den der Nutzer nutzen kann.

Um die Oberfläche anzuzeigen wird einfach auf der Form für jedes Feld ein TImage gezeichnet. Um ein Feld zu zeichnen gibt es eine eigene procedure, da sie häufig aufgerufen werden muss. Um später die Ein- und Ausgabe zu vereinfachen fassen wir die TImages zu einem Array zusammen. Um später die herauszufinden welches gerade das aktive TImage, ist versehen wir den Tag mit den Infos über die Spalte und die

Zeile. Da sich die TImages zwar zum Zeichnen eignen aber für Operationen wie Abfragen welche Ziffer in welchem Feld steht ungeeignet ist übernimmt diese Aufgabe ein virtuelles Array. In diesem stehen die Zahlen dann auch im Integer und Dekadischen Format, da damit der Rechner ohne ständige Umwandlung arbeiten kann. Anschließend wird noch ein Array eingefügt in dem steht welche Zahlen vom Nutzer eingetragen wurden und welche Vorgegeben waren. Damit der Nutzer nun beginnen kann Ziffern einzutragen, bestimmen wir stets das aktuelle Feld. Wenn dann eine Taste gedrückt wird, überprüfen wir welche Taste es ist, und reagieren dementsprechend. Wenn es eine der Tasten zum Steuern durch die TImages ist, dann wird das entsprechende Feld aktiv gesetzt. Wenn aber eine andere Taste gedrückt wurde, schauen wir ob denn der Nutzer überhaupt eine Zahl setzen kann, oder ob das Feld mit einer Vorgabe gefüllt ist. Wenn der Nutzer eine Zahl setzten kann, dann bestimmen wir entweder welche er gerne setzten möchte oder ob er nur ein Zeichen löschen möchte. Wenn der Nutzer einen Wert eingegeben hat, können wir gleich noch prüfen ob er schon alle Felder ausgefüllt hat. Wenn dies der Fall ist, vergleichen wir das eingegebene HexaDoku mit der Lösung und kontrollieren so ob er es erfolgreich gelöst hat.

### 3.2. *HILFEN AN DEN NUTZER BEIM LÖSEN*

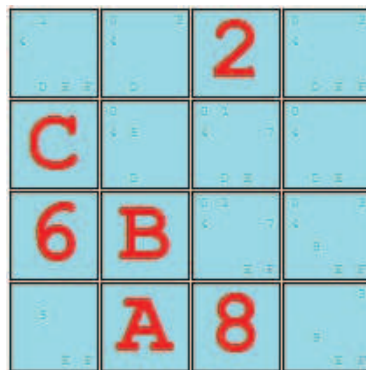


*ScreenShot 2: Programm nach Start; Zeigt alle Möglichkeiten; Zeigt die aktive Spalte/Zeile/Block/Feld*

Wie man sehr schön sehen kann und wie bei den Ergebnissen beschrieben wird also zum einen die Spalte, die Zeile, den Block und das gerade ausgewählte Feld. Um dies zu realisieren werden zum einen die procedures zum Zeichnen der Umrandungen benötigt. Diese werden natürlich bei jeder Änderung des aktiven TImage aufgerufen. Um vom aktiven TImage die Spalte, die Zeile und den Block zu bestimmen, können wir nun unsere Vorteile nutzen, welche wir beim Erzeugen deklariert haben. Durch den Tag kennen wir die genaue Position, durch das Zusammenführen in einem Array erleichtern wir uns das Bestimmen der Felder einer Einheit.



Außerdem zeigt das Programm die noch möglichen Ziffern an. Dies wird am folgenden Screenshot deutlich:



*ScreenShot 3: Ausschnitt aus Programm; zeigt die Möglichkeiten*

Um dies zu Erreichen habe ich eine procedure entwickelt, welche ähnlich wie schon bei der Markierung die Einheiten entlang geht und dabei bestimmt, welche Zahlen vorkommen und welche dementsprechend noch verfügbar sind. Zur Laufzeit

wird diese procedure dann jedes mal beim Zeichnen eines leeren Feldes aufgerufen.

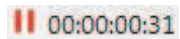
### **3.3. LADEN UND SPEICHERN EINES ANGEFANGENEN SPIELES**

Um dem Nutzer das Speichern und das Laden eines angefangenen Spieles zu ermöglichen muss nicht viel beachtet werden. Zuerst muss das Spiel gespeichert werden um später geladen werden zu können. Beim Speichern werden zuerst alle vorgegebenen Ziffern gespeichert. Danach werden die eingetragenen Ziffern gespeichert und anschließend die Lösung. Um die Zeit weiterlaufen zu lassen, wird außerdem anschließend die aktuelle Zeit gespeichert. Um später noch die Schwierigkeit zu kennen, speichern wir diese auch noch am Schluss. Dies geschieht flott, indem wir ein TextFile erstellen, für jede Zeile einen String befüllen und diesen einfach mit WriteLn anhängen.

Beim Laden ist dann auch nicht viel mehr zu beachten. Es müssen nur alle vorher gespeicherten Dateien wieder eingelesen werden. Dies geschieht analog zum Speichern. Wir lesen eine Zeile ein und nehmen diese auseinander und ordnen sie den Werten zu.

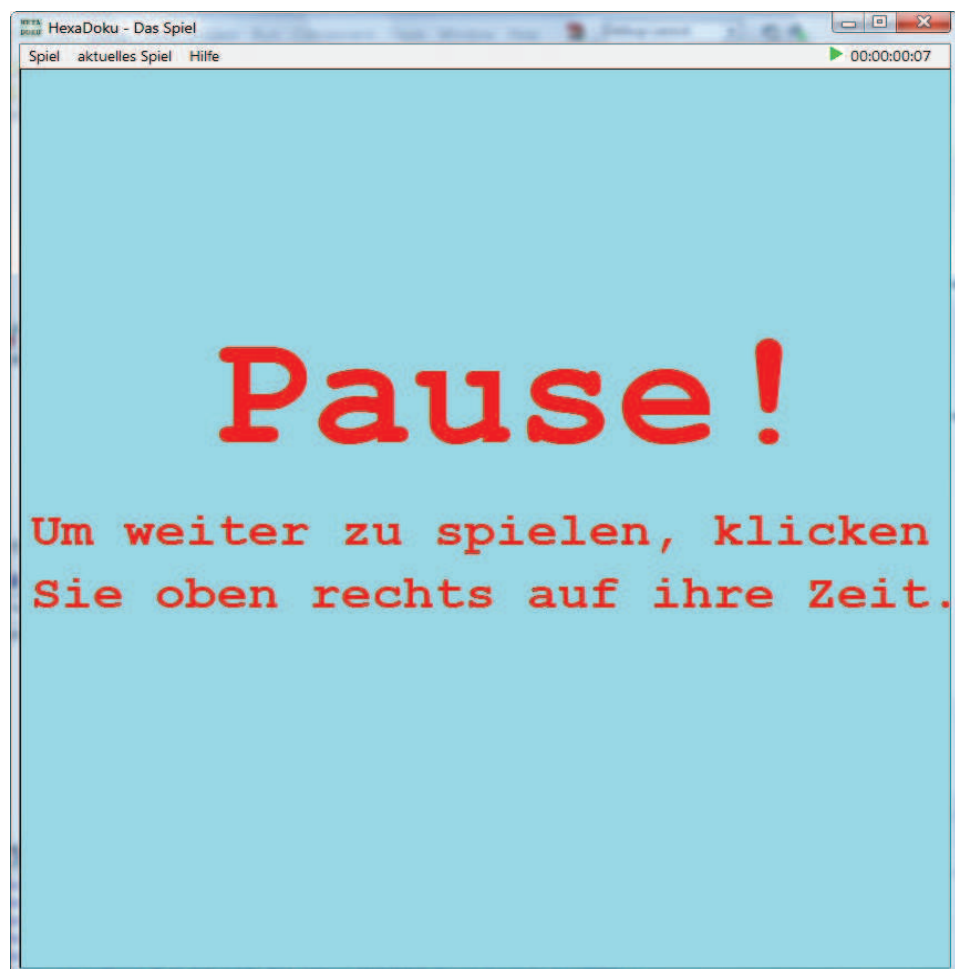
Bei beiden ist es dem Nutzer möglich über einen Dialog die Speicherorte zu bestimmen.

### 3.4. DIE EIGENE AKTUELLE ZEIT UND DIE PAUSE



Die Zeit, welche der Nutzer jederzeit im Blickfeld haben kann ist schlicht gehalten und im Format dd:hh:mm:ss aufgebaut. Realisiert ist, dies über den Timer welcher jede Sekunde die aktuelle Zeit um eins erhöht. Für die Ausgabe wird anschließend dieser Wert umgerechnet, was eine vorher erstellte procedure übernimmt.

Wenn der Nutzer nun darauf klickt, beginnt der Pause-Bildschirm zu erscheinen welcher so aussieht:



ScreenShot 4: Programm während einer Pause

Die Zeit ist angehalten, was dadurch geschieht, dass der Timer gestoppt wird. Außerdem wird der Bildschirm übermalt, damit keine Nutzer schummeln können. Wenn der Nutzer nun wieder auf seine Zeit klickt, kann er ganz normal weiterspielen.

### 3.5. DIE *HIGHSCORE*- BZW. *BESTENLISTE*



ScreenShot 5: HighScore Fenster mit zufälligen Werten

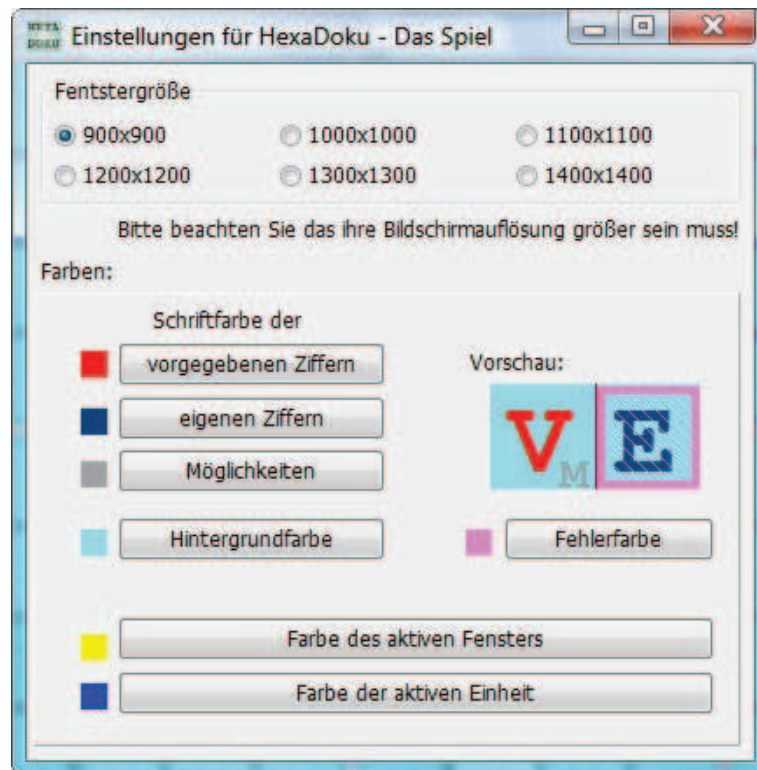
Bei der HighScore Liste ist im Wesentlichen bestehend aus einigen Tlisten, eine für jede Schwierigkeit. Wenn ein Element hinzugefügt wird, wird die Liste sortiert und ausgegeben. Der Hauptunterschied zwischen der Virtuellen und der Ausgegebenen ist, dass im Virtuellen das Programm die Zeit in Sekunden gespeichert hat, welche auf 9 Zeichen verlängert wurde, damit es zu keinen Problemen beim Sortieren kommt. Bei der Ausgabe wird der Wert dann im gewohnten Format angezeigt. Außerdem wird der String vor der Ausgabe formatiert, um ein ordentliches Layout zu zeigen.

Natürlich wird auch die Platzierung ganz vorne angezeigt.

Gespeichert wird es einfach mit dem Befehl der TList im Spiele-Ordner.



### 3.6. *EIGENE FARBEN FESTLEGEN*

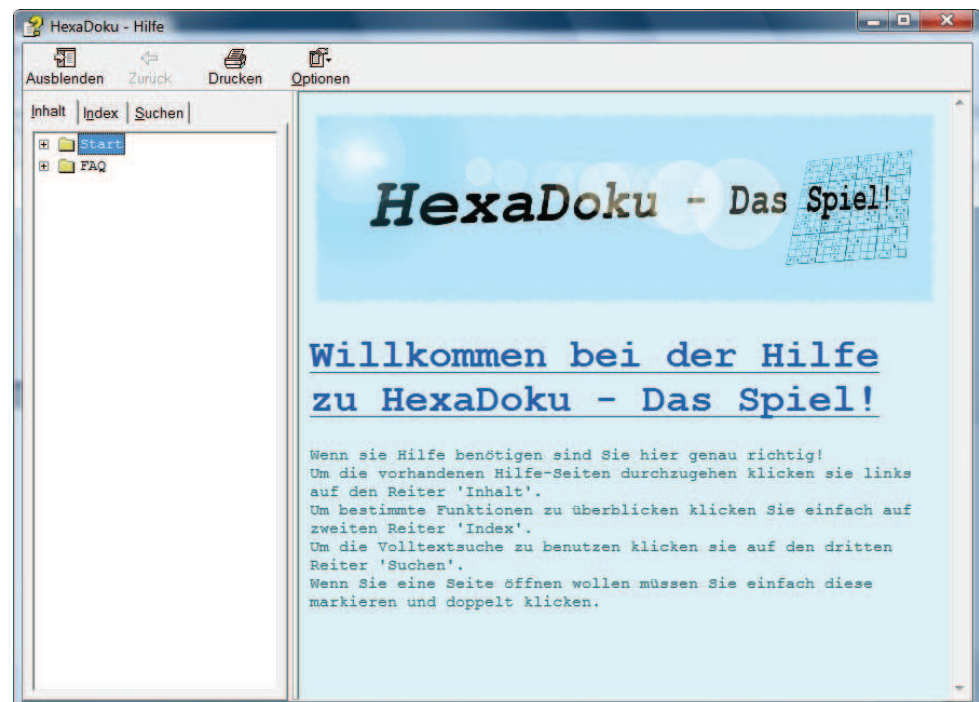


ScreenShot 6: *Einstellungen*

Wie sehr schön zu sehen ist, kann der Nutzer alle Farben ändern und diese bekommt er in einer Vorschau angezeigt. Die Farben werden bei ButtonClick mit einem ColorDialog ausgewählt. In der Vorschau wird ein Beispiel gezeichnet, um den Nutzer einen Eindruck zu vermitteln. Die Daten werden sofort übernommen und beim Programmschluss in einer .ini gespeichert. Diese hat den Vorteil, dass sie auf jeden Rechner übernommen werden kann. Zum Speichern werden die Farben in Integer umgewandelt, da Delphi keine TColor in eine Ini-Datei schreiben kann.

Die Fenstergröße kann der Nutzer ebenfalls anpassen, je nachdem wird dann die Fenstergröße verändert. Die Schriftgröße sollte sich dann genau wie die TImage anpassen.

### 3.7. WINDOWS *HTML-HILFEDATEI* MIT *INDEX* UND *VOLLTEXTSUCHE*



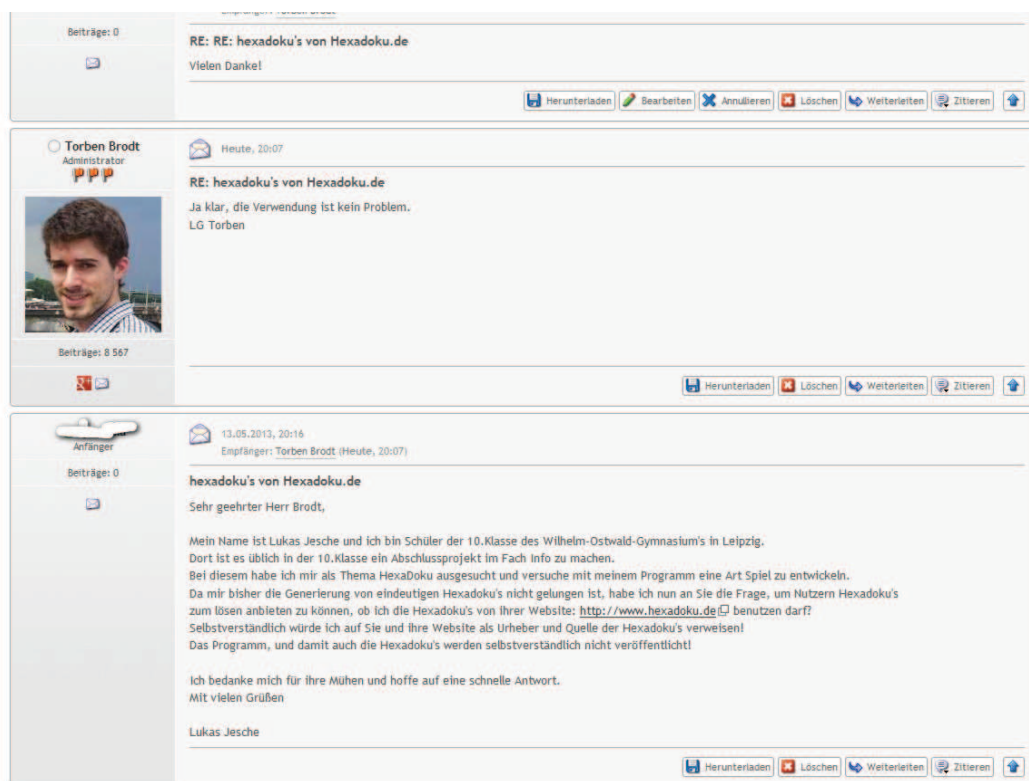
ScreenShot 7: Hilfe zum Programm

Die Hilfe, so wie sie oben zu sehen ist, lässt sich mit HTML-Kenntnissen und dem „Microsoft HTML Help Workshop“ erstellen. Anschließend müssen noch Seiten IDs angelegt werden. Die so entstandene Alias-Datei lässt sich anschließend im HTML Help Workshop öffnen und mit kompilieren. Damit sie sich allerdings in Delphi Programmen nutzbar ist lässt sich zum Beispiel das "HTML Help Kit for Delphi" von der Helpware Group unter Leitung des Australiers Robert Chandler nutzen, welche als Freeware verfügbar ist. Nun werden die HelpContexte normal eingebunden wie normale Hilfedateien angelegt und anschließend mithilfe des Help Kit an die .chm Datei umgeleitet.

### 3.8. DIE HEXADOKU'S

Alle HexaDoku's sind, da es mir bisher noch nicht gelungen ist einen funktionierenden Generator zu entwickeln, der Website <http://www.hexadoku.de> entnommen. Fast alle der dort über 2000 verfügbaren HexaDoku's sind dabei downloadet, umgewandelt und gespeichert worden. Dies geschah mit 2 weiteren kleineren Programmen. Das eine bestand aus einer TWebbrowser-Komponente und öffnete jede der Websites, um dann den Quelltext zu speichern. Das zweite Programm hatte dann die Aufgabe die HexaDoku's in ein für das Spiel nutzbares Format umzuwandeln. Dazu las es alle Spieldateien in eine Liste, welche anschließend abgearbeitet wurde. Über die Stringverarbeitung wurden so die HexaDoku's entnommen, und anschließend die Schwierigkeit bestimmt, um die Datei richtig speichern zu können.

Die Erlaubnis die HexaDoku's für dieses Projekt downloaden und nutzen zu können habe ich mir selbstverständlich bei dem Betreiber der Seite Tobias Brodt geholt.



## 4. DISKUSSION DER ERGEBNISSE

Ich bin dem Ergebnis sehr zufrieden. Auch wenn das Projekt noch um einige Sachen erweitert werden könnte und auf dem Weg zur Fertigstellung einige Funktionen auf der Strecke blieben, welche später noch eingefügt werden können. Sehr schön habe ich auch einige Grenzen von Delphi entdeckt. Ich denke trotzdem, dass das Programm die Ziele dem Nutzer beim Lösen zu helfen und dabei sehr intuitiv und individuell einstellbar zu bleiben, durchaus erreicht hat. Das dabei der Quelltext durchaus nicht der optimierteste ist, kann sein, allerdings war der wichtigere Punkt die Funktionalität als ein möglichst geringer Quelltext, welcher dafür mehr Fehler enthält.

Dennoch denke ich durchaus das ich meine Fähigkeiten des algorithmieren und des programmieren am Beispiel Delphi durchaus verbessert haben und das Programm als Darstellung der Ergebnisse des Informatikunterrichtes in der 9. und 10. Klasse dient.

Außerdem bin ich mir durchaus bewusst, dass das Programm nicht damit mithalten kann was beispielsweise HoDoKu für normale Sudoku's bietet. Dies war allerdings auch nicht der Anspruch. Jedoch habe ich gemerkt wie viel Arbeit in einem solchem Projekt stecken muss.

## QUELLENVERZEICHNIS

### WEBSITES:

<http://www.delhipraxis.net/>

<http://www.entwickler-ecke.de/>

<http://www.delphi-treff.de/>

> bei Fragen/Problemen über Delphi

<http://www.hexadoku.de/>

> Quelle der HexaDoku's

<http://sudokugarden.de/de/loesen/dancing-links/>

> Lösungsmöglichkeiten von Sudoku's aufgerufen am 16.05.2013

<http://hodoku.sourceforge.net/de/index.php/>

> Ideen, Lösungsstrategien aufgerufen am 16.05.2013

<http://de.wikipedia.org/wiki/Sudoku/>

> allgemeine Informationen zu Sudoku's aufgerufen am 16.05.2013

### FREMDQUELLTEXT:

<http://www.delhipraxis.net/234840-post.html#271833>

> Farbe invertieren aufgerufen am 16.05.2013

<http://www.delphi-treff.de/tipps/komponenten/menues/menueeintrag-rechts-im-formular-ausrichten/>

> MenuEintrag nach rechts ausrichten aufgerufen am 16.05.2013

### PROGRAMME:

<http://www.delphi-treff.de/downloads/edh/>

> Hilfeprogramm

<http://download.microsoft.com/download/OfficeXPProf/Install/4.71.1015.0/W98NT42KMe/EN-US/HTMLHELP.EXE>

> Erstellen der Hilfe

<http://www.delphi-treff.de/tutorials/tools/html-hilfe/einbindung-in-delphi/>

> Download des Help Kit's

## **SELBSTÄNDIGKEITSERKLÄRUNG**

Hiermit erkläre ich, dass ich die vorliegende Arbeit alleine und nur mit den im Literaturverzeichnis angegebenen Quellen erstellt habe.

Lukas Jesche

Leipzig, den 16.05.2013