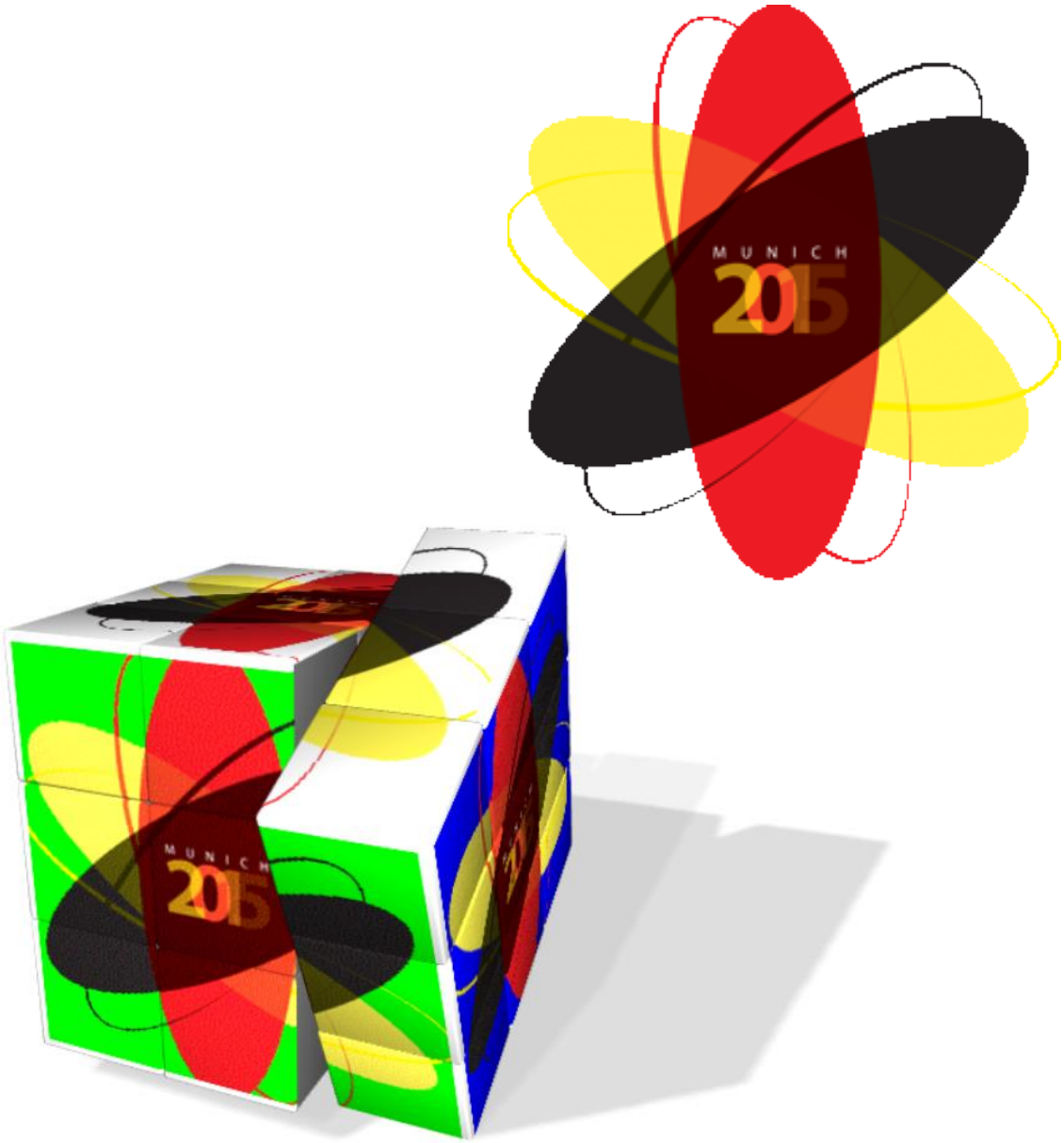
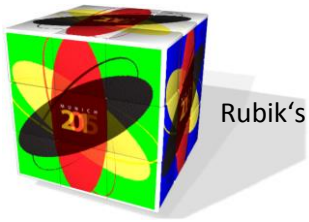
Rubik's Cube Lösungshilfe

European Schools Science Symposium 2015

7HS ESMunich
Maria Despoina Mariggi





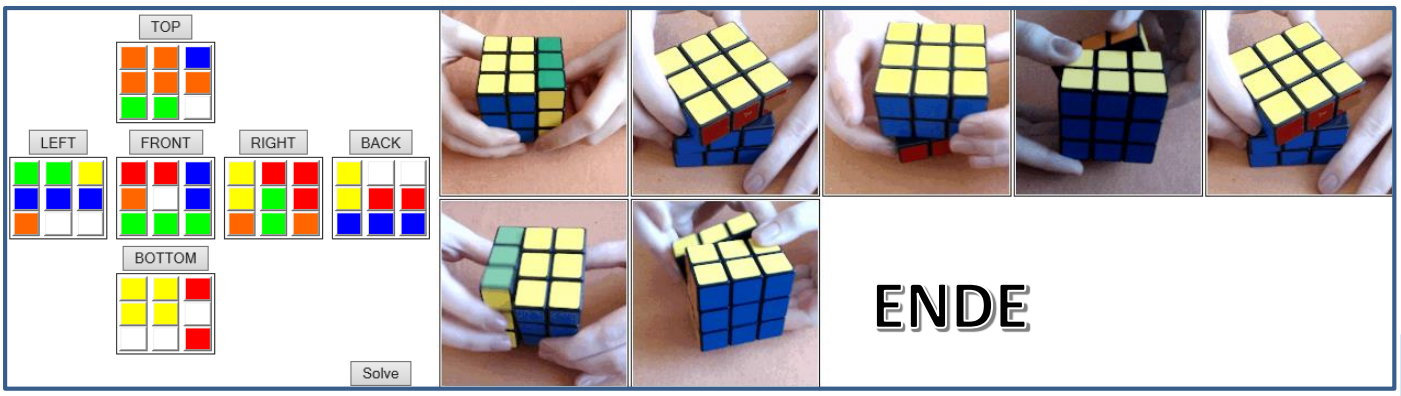
Inhalt

Abstract

Das Projekt „Rubik‘s Cube Lösungshilfe“ ist ein Programm auf den Sprachen C/C++ und HTML/JavaScript. Das grafische Output sind animierende GIF–Bilder, die genauen Drehungen von Seiten anzeigen. Der Benutzer des Programmes soll diese befolgen, um seinen Würfel zu lösen. Die Reihenfolge der anzuwendenden Drehungen ist auf jede als Bezugspunkt beliebig gewählte Farbe (der Würfel-Mittelsteine) anwendbar.
Das Programm versucht anfänglich den Würfel in bis zu sieben Drehungen von Seiten zu lösen (1). Falls dies sich als nicht möglich herausstellt, wird der Würfel Ebenen-Weise in sieben Schritten gelöst (2).

Vom Benutzer aus gesehen erfolgt das Ganze auf folgender Weise:
Ein Internet Explorer Fenster öffnet sich und Layout eines gelösten Würfels erscheint, wie es im ScreenShot unten links abgebildet ist. Der Benutzer wählt einen Bezugspunkt aus und gibt im Fenster die Reihenfolge der Farben ein, die dem echten Würfel entsprechen. Diese werden dann auf einer Text-Datei eingeschrieben, die als Input für die Exe-Datei dient und so den virtuellen Würfel darstellt.

(1) Das Programm führt verschiedene Drehungs-Sequenzen nacheinander durch und vergleicht immer den virtuell neu entstandenen Würfel mit einem bereits gelösten virtuellen Würfel, der als Vorlage fungiert. Wurde eine Drehungs-Sequenz entdeckt, die eine Lösung herbeiruft, endet das Programm an dieser Stelle und es öffnet sich ein zweites Internet-Fenster. Die Lösung wird in animierten GIF-Bildern im Fenster angezeigt.
(2) Wird keine Sequenz bis zu sieben Drehungen gefunden, dann wird der virtuelle Würfel auf die ursprüngliche ungelöste Form zurückgesetzt und anschließend in sieben Schritten gelöst. Diese Schritte entsprechen den Reihenfolgen von Drehungen, die der Benutzer mithilfe handlicher Gebrauchsanweisungen befolgen würde. Wird auf letzterer Weise eine Lösung gefunden, wird nun diese in ein neues Internet-Fenster in Form von GIF-Bildern angezeigt.



Einführung

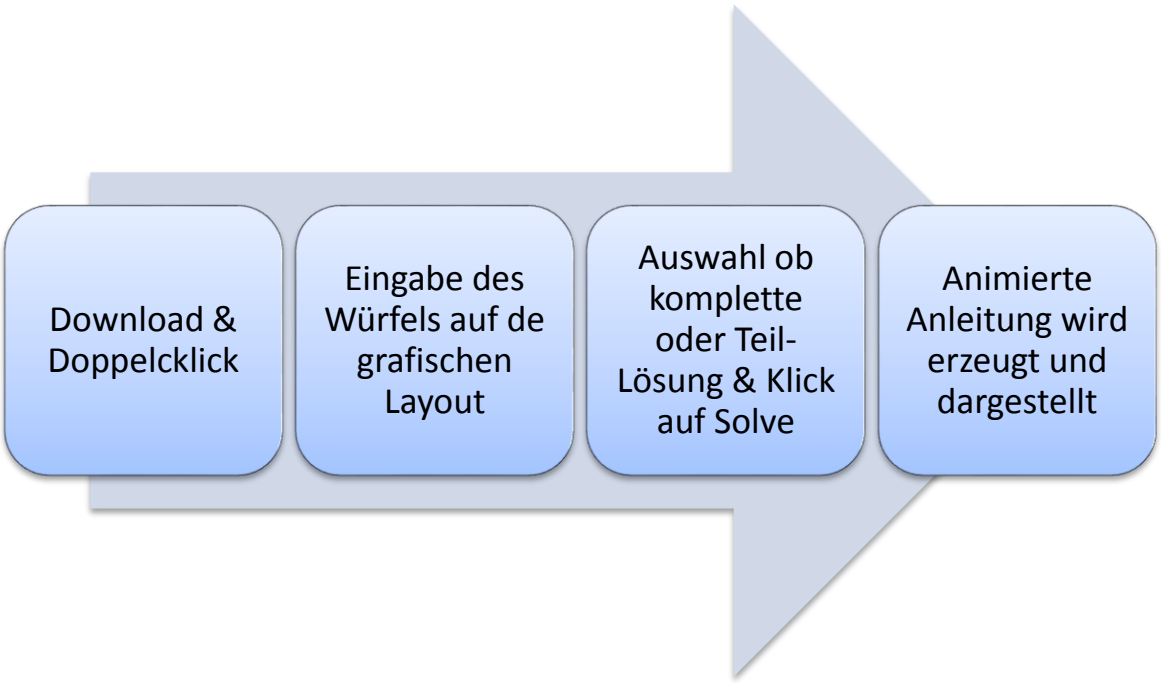
Die Idee ist es, ein Projekt zu entwickeln, das Benutzern hilft, einfacher und interaktiver zu lernen, wie man die handlichen Drehungs-Sequenzen anwenden soll, um den Rubik’s Cube zu Lösen.

Es gibt zahlreiche Hilfsmittel im Internet, die ausführlich beschreiben, wie man den Würfel manuell lösen kann. Oft ist es aber der Fall, dass es Menschen schwer fällt, die angebotenen Strategien zum Lösen auf ihren eigenen Würfel zu übertragen.

Dieses Programm erlaubt es dem Benutzer interaktiv an seinem eigenen Würfel die korrekte Anwendung der Sequenzen schnell und ohne falscher Interpretation zu lernen. Es werden dem Benutzer die manuelle Lösung in animierter Form vorgestellt und die Drehungs-Sequenzen, die angewendet werden sollen, genau verdeutlicht.

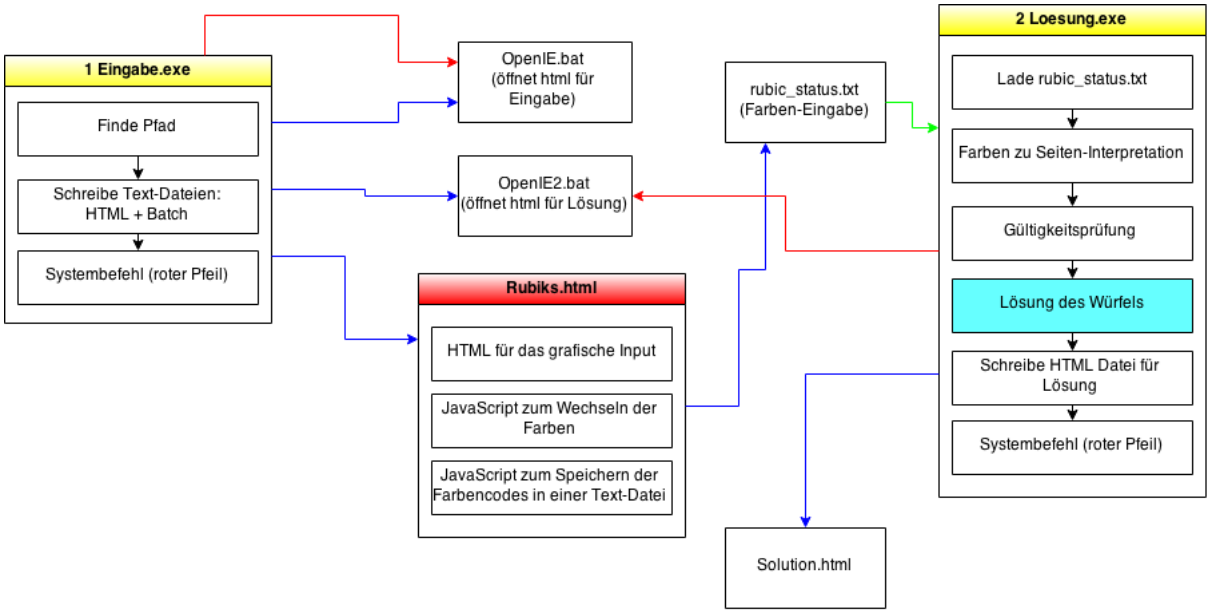
Natürlich dient dieses Programm auch denjenigen Benutzern, die Ihre Rubik’s Cubes einfach und ohne Mühe gelöst haben wollen.

Das Programm ist zudem noch so aufgebaut, dass es leicht bedienbar ist:



Kern des Systems

Das folgende Diagramm zeigt den Kern des Systems. Das Projekt „Rubik's Cube Lösungshilfe" besteht aus zwei Programmen, die durch ein drittes Programm miteinander verknüpft werden. Später wird unter „Das gesamte System" in einem zweiten Diagramm diese Verknüpfung verdeutlicht. Diese bezweckt, dass der Benutzer nur einmal doppelklicken muss, um das erwünschte Ergebnis zu erhalten. Das Diagramm hier ist die Darstellung des Systems ohne das dritte Programm, aber sie vereinfacht das Prinzip zum wesentlichen Verstehen.



Programm zum Starten der Eingabe

Dieses Programm ist notwendig, damit man das ganze System (alle Programme, Seiten und Dateien) von jedem beliebigen Pfad aus ausführen kann und dass der Benutzer auch in der Lage ist das System nach einem Download durch einmaliges Doppelklicken (der Programm-Datei) zu starten. Dieses Programm schreibt drei notwendige Dateien:

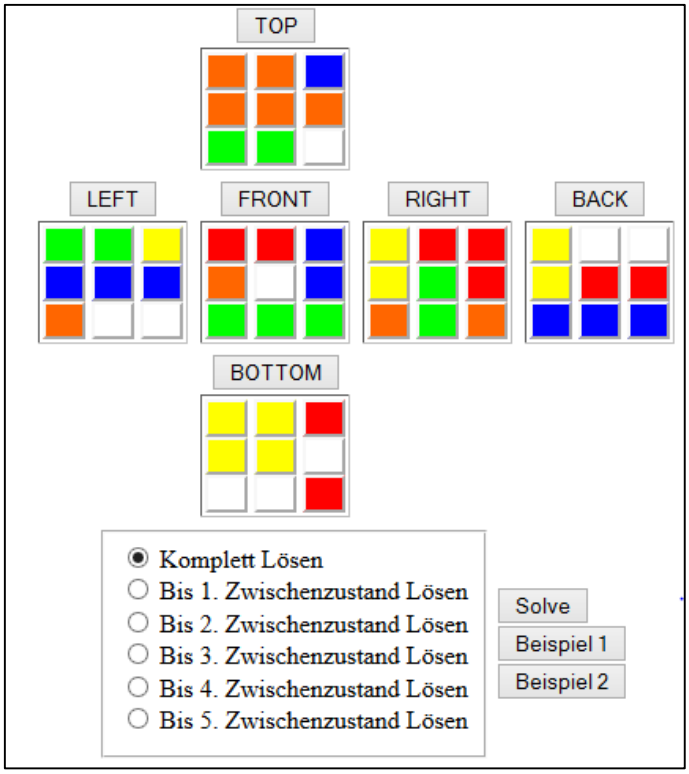
- Zwei davon sind Batch-Dateien zum Öffnen der HTML-Dateien, eine für die Eingabe und eine für die Lösung
- Die dritte Text Datei ist die HTML-Datei, die für die Eingabe zuständig ist

Das meiste was dieses Programm macht ist einfach nur viel Text schreiben, welcher bereits vordefiniert ist. Da es sich aber nicht nur um Text handelt, sondern auch System-Befehle damit zusammenhängen, müssen die Dateipfade auch stimmen, auf die zugegriffen wird. Dieses Programm findet durch eine Bibliotheks-Funktion den eigenen Pfad heraus und schreibt diesen in den richtigen Stellen im Quelltext der drei Text-Dateien. So sind die Befehle in diesen Dateien an den richtigen Ordnern gerichtet.

Bevor dieses Programm abläuft, führt es noch einen System-Befehl zum Öffnen der Eingabe-Datei „Rubiks.html" aus.

Grafische Eingabe

Der Benutzer gibt den Anfangszustand des Würfels in das grafische Fenster im Browser ein. Dieses Layout sieht folgender Maßen aus:



HTML Teil

Die grafischen Elemente werden hier generiert. Dafür gibt es für die verschiedenen Eingabemethoden und Darstelllungen bestimmte Funktionen, die HTML unterstützt. Die Felder mit den Farben sind Buttons und jede Seite (3x3 Felder) in der Darstellung des Würfels befindet sich in seiner eigenen HTML-Tabellenzelle.

Die Felder „solve", „Beispiel 1" und „Beispiel 2" sind ebenfalls Buttons und befinden sich in einer zweiten Tabelle direkt unter der ersten Layout-Tabelle. Die Radiobuttons, die dem Benutzer ermöglichen zu entscheiden, bis wohin er seinen Würfels gelöst haben möchte, befinden sich ebenfalls in der zweiten Tabelle.

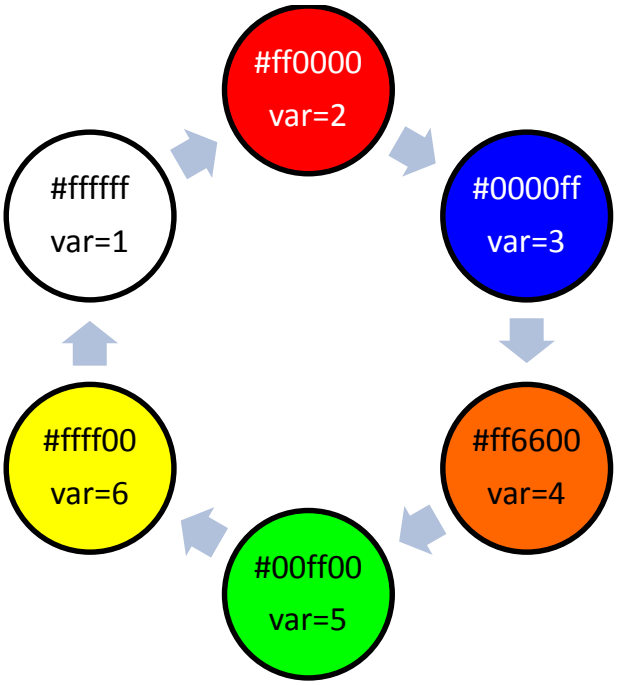
Für alle Buttons gibt es eine Eigene „onclick"-Funktion und für alle Radiobuttons eine eigene „onchange"-Funktion. Diese sind Funktionen, die im JavaScript-Teil definiert werden und jedes Mal wenn ein Button geklickt wird, aufgerufen werden.

JavaScript-Teil

Hier werden die ganzen Funktionen definiert, die durch Anklicken der Grafischen Elemente aufgerufen werden. In dem JavaScript-Teil gibt es bestimmte Variablen, die global definiert wurden, damit sie in jeder Funktion enthalten sind. Letztere bestimmen den Status der einzelnen Kästchen-Felder und auch die Auswahl des Benutzers, mit der er entscheidet bis zu welchem Zwischenzustand gelöst werden soll.

Rubik's-Kästchen-Funktionen

Wenn eine dieser Funktionen aufgerufen wird, dann ändert sich die Farbe im Button und auch die dazugehörige globale Variable. Während die anfängliche Farbe für jeden Bereich (Front, Back,…) unterschiedlich ist, rotieren die Farbe und die Variable wie folgend dargestellt:



Die Radiobuttons: Hier wird bei jedem Klick eine Variable umgeschrieben, die dem Radiobutton entspricht.

Die „Beispiel 1,2"-Buttons: Hier werden die Status-Variablen so beschrieben, dass es zwei echten Beispielen entspricht. Beim Klicken werden die Funktionen für die Kästchen-Felder aufgerufen, sodass sich die Farben auf den Feldern entsprechend ändern. Diese Buttons dienen der schnellen Demonstration des Projektes.

Der „solve"-Button: Wenn dieser Button geklickt wird, dann schreibt die Funktion alle Status-Variablen auf einer Text-Datei und die Auswahl-Variable in eine zweite Text-Datei, in einem der Unterordner. Die Auswahl-Variable bestimmt bis wohin der Würfel gelöst werden soll.

Lösungs-Programm

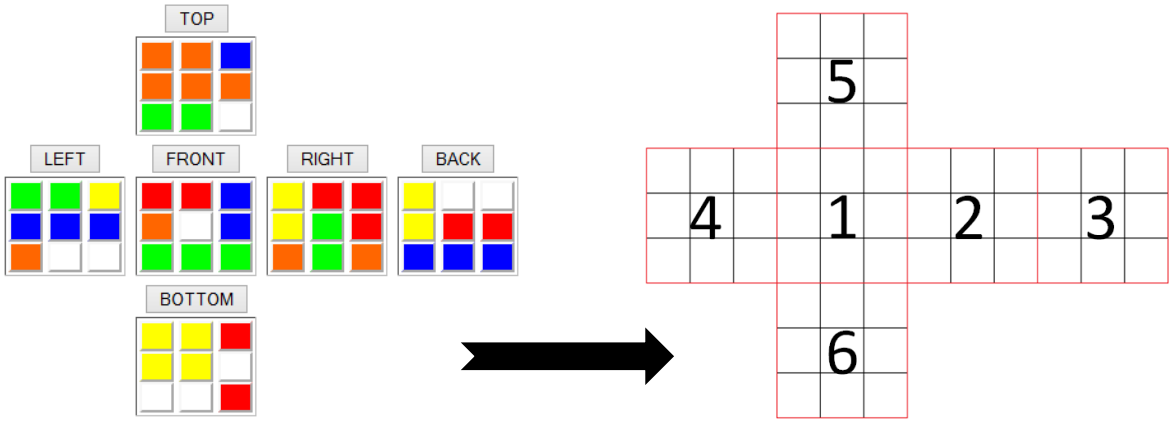
Eingabe-Interpretation und „Gültigkeitscheck":

Hier passieren einige Teilschritte. Der wichtigste ist natürlich die eigentliche Lösung des Würfels, wie er vom Benutzer eingegeben ist.

Als erstes liest das Programm die zwei Text-Dateien um die Benutzereingabe als Input zu nehmen. Die eine Text-Datei ist eine „int"-Variable, die angibt bis wohin der Würfel gelöst werden soll (Eingabe durch die Radiobuttons).

Die zweite Text-Datei wird gelesen und muss danach noch interpretiert werden. Das ist deshalb, weil der Benutzer Farben eingibt, das Programm aber unabhängig von Farben funktioniert. Dies ist

natürlich in dem Sinne sehr hilfreich, dass sich der Benutzer nicht einschränken muss, welches mittlere Kästchen (also Farbe) als Bezugspunkt wählt. Im Programm hat z.B. der Vorderbereich (FRONT) immer den Wert 1. Dieser Wert kann je nach Eingabe eine verschiede Farbe sein (die Farbe die der Benutzer als Bezugspunkt gewählt hat).



Bevor das Lösungsverfahren beginnen kann, wird noch überprüft ob es sich um eine gültige Würfeleingabe handelt, indem sichergestellt wird, dass alle Bereiche (FRONT, BACK,…) eine verschiedene Farbe im mittleren Kästchen haben und dass alle Farben genau 9-mal auftreten.

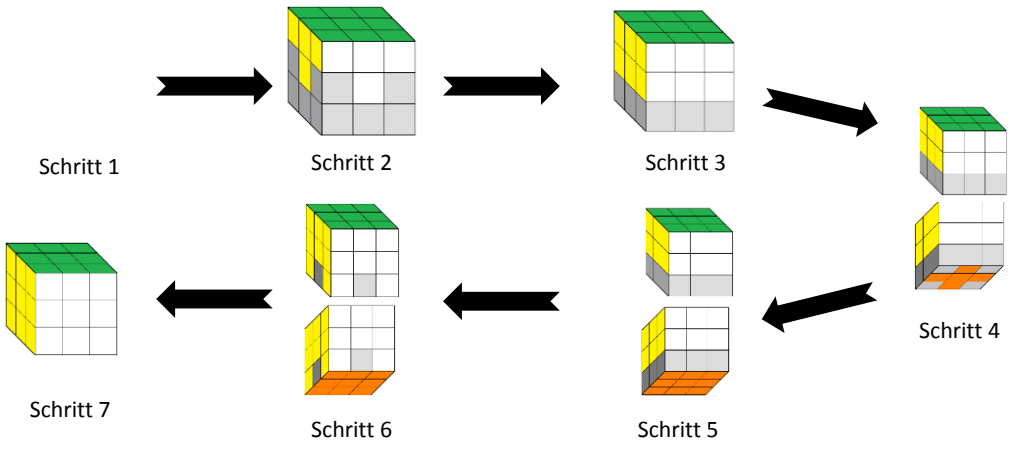
Nun kann man den Lösungsalgorithmus laufen lassen. Dieser löst den Würfel in den 7 Schritten, die man durch handliche Gebrauchsanweisungen befolgen würde und die auch in vielen Webseiten zu finden sind. Im Folgenden ist mit einer Drehungs-Sequenz eine Folge von vordefinierten Drehungen gemeint, für jeden Schritt anders sind.

Lösungsalgorithmus

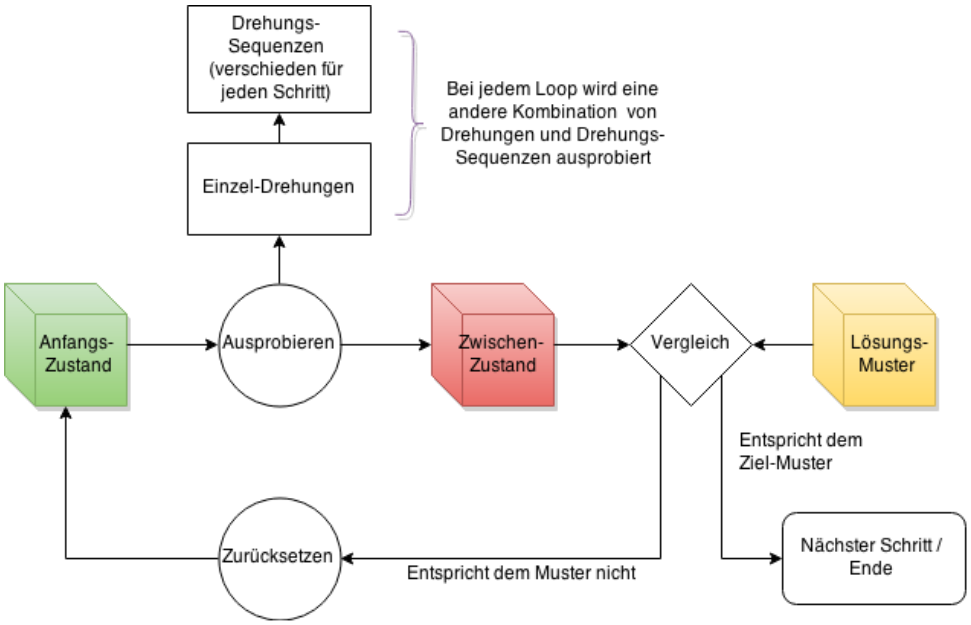
Der Lösungsalgorithmus, kurz erklärt:

> Der Lösungsalgorithmus löst den Würfel in 7 Schritten. In jedem Schritt werden (nach einem passenden Schema) verschiedene Drehungen und Drehungs-Sequenzen ausprobiert bis die Lösungsbedingung erfüllt ist. Am Ende des 7. Schrittes ist die gesamte Lösung vorhanden.

Die Schritt-Endbedigungen sind in der folgenden Abbildung illustriert:



Die genaue Vorgehensweise bei den einzelnen Schritten wird nun kurz folgender Grafik demonstriert:

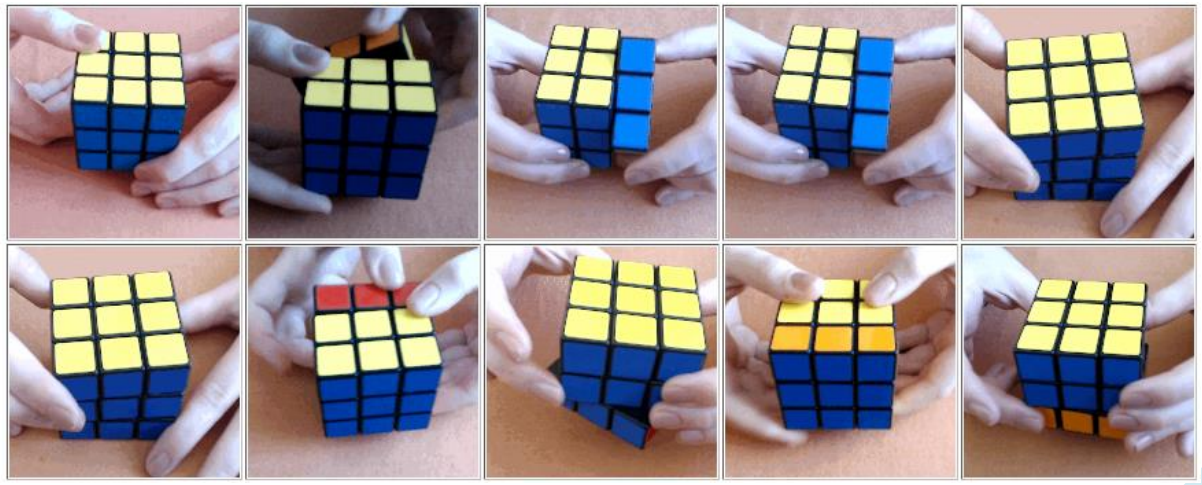


Schnelles Lösen

Wie bei den ersten beiden Schritten schon erwähnt worden ist, gibt es eine Schleife im Programm die alle Möglichkeiten ausprobiert, bis die Eckfelder dem gesuchten Muster entsprechen. Diese Schleife kann für allgemeinere Zwecke benutzt werden, und alle Möglichkeiten durchsuchen mit dem Ziel den gesamten Würfel zu Lösen. Es ist nicht garantiert, das eine Lösung in akzeptabler Zeit gefunden wird, aber oft ist der eingegebene Würfel nur etwas vermischt und kann mit bis zu 7 Drehungen gelöst werden. Diese Schleife läuft bis zu 7 Drehungen, da es unter diesen Bedingungen nur einige Sekunden dauert um das Ziel zu erreichen. Für größere Drehungs-Folgen dauert es wesentlich länger.

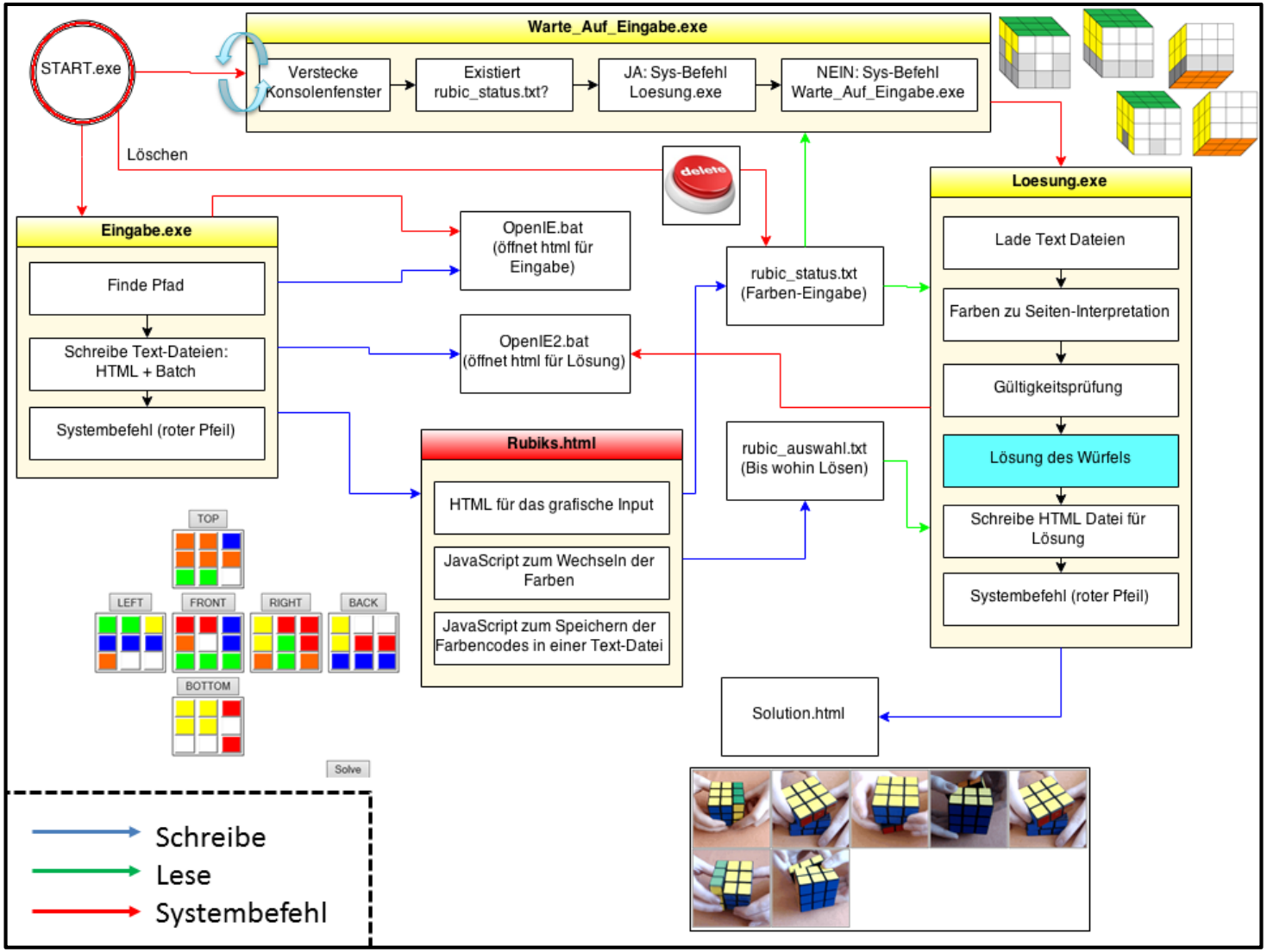
Die Schleife zum schnellen Lösen wird nur dann ausgeführt, wenn der Benutzer den kompletten Würfel gelöst haben möchte und falls der Würfel in 7 Schritten gelöst werden kann.

Grafische Lösungs-Darstellung



Das gesamte System

Das folgende Diagramm zeigt die Funktionsweise vom gesamten System. Das Einzige was der Benutzer machen muss ist auf die „START.exe“ doppelzuklicken und der Rest läuft unabhängig im Hintergrund weiter. Die Funktionsweise von den zwei extra Programmen die das ermöglichen ist weiter unten ausführlicher beschrieben.



Startprogram

Durch Doppelklicken auf die START.exe beginnt man den ganzen Prozess. Das was hier passiert, sind drei System-Befehle:

1. Lösche rubic_status.txt (damit das Warten auf die Eingabe auch richtig läuft)
2. Starte Eingabe.exe
3. Starte Warte_Auf_Eingabe.exe

Programm zum Warten auf die Eingabe

Was dieses Programm macht ist einfach, aber sehr hilfreich damit der Benutzer sich nicht Gedanken machen muss, was im Hintergrund passiert.

Der JavaScript auf der HTML-Datei kann zwar Text-Dateien schreiben, aber er ist nicht in der Lage System-Befehle durchzuführen. Der Benutzer wäre dafür zuständig, nachdem er den Button zum Lösen geklickt hat, noch separat die Loesung.exe auszuführen. Dies ist aber umständlich und deswegen habe ich noch ein weiteres Programm geschrieben, das gleichzeitig mit dem Programm für die Eingabe startet und auf die Eingabe des Benutzers wartet.

Es funktioniert, indem es einfach versucht auf die „rubic_status.txt"-Datei zuzugreifen, die nur dann existiert, wenn der Benutzer die Eingabe fertiggestellt hat (da es vom Startprogramm gelöscht wird falls es vorhanden ist). Falls die Datei existiert, wird ein System-Befehl zum Öffnen des Lösungsprogrammes ausgeführt. Falls nicht, dann wird der Versuch auf die Datei zuzugreifen wiederholt, indem das Programm einen System–Befehl ausführt, um sich selber wiederzueröffnen nachdem es geschlossen wurde.

Das Programm wird im ausgeblendeten Modus ausgeführt, durch Aufrufen der „Stealth"-Bibliotheksfunktion und es führt einen „Sleep"-Befehl von einer halben Sekunde aus und verbraucht somit sehr wenig CPU.

Quellenangabe

Mao, Tyson; Lee, Jasmine; Harris, Dan; **Rubik's Cube Solution Guide**; http://rubiks.com/uploads/general_content/Rubiks_cube_3x3_solution-en.pdf 07.2012 - 08.2012

07.2013 - 08.2013

How to solve a Rubik's Cube; http://www.rubikssolver.com/ 07.2012 - 08.2012

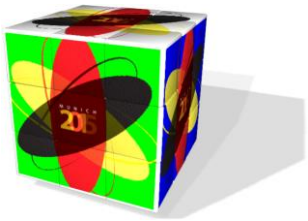
07.2013 - 08.2013

Stackoverflow; http://stackoverflow.com/ (Sehr oft für den Quelltext besucht)

W3schools; www.w3schools.com (Sehr oft für das Schreiben von HTML und JavaScript besucht)

Draw.io (Diagramm-Tool); https://www.draw.io/ 28.12.2014

Design Your Cube; http://www.designyourcube.com/designer 17.02.2015

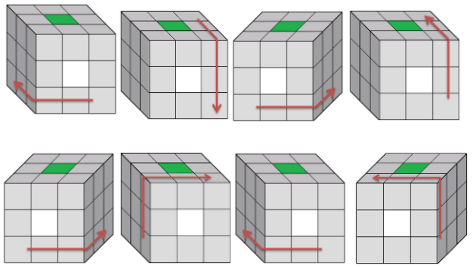


Anhang

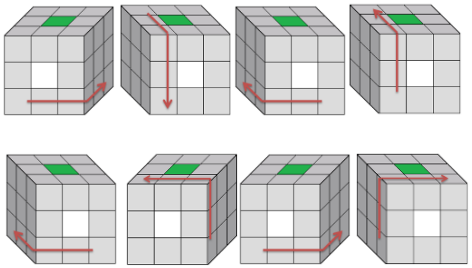
Die vordefinierten Drehungs-Sequenzen für die einzelnen Schritte im Lösungsalgorithmus.

Schritt 3:

Sequenz 1:

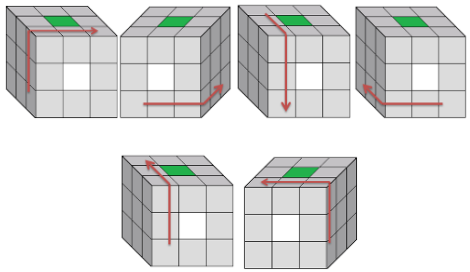


Sequenz 2:

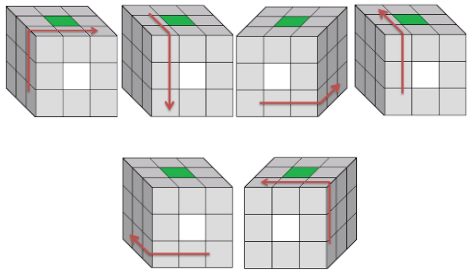


Schritt 4:

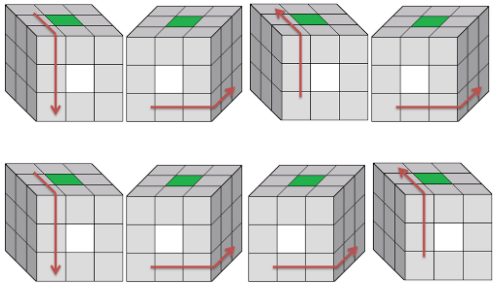
Sequenz 1:



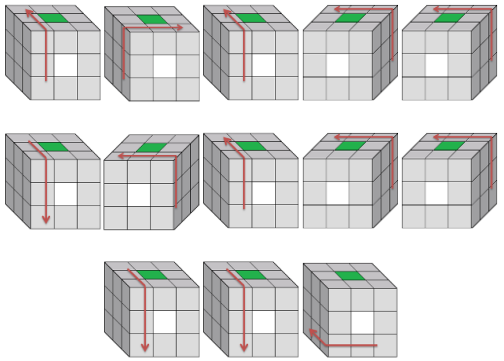
Sequenz 2:



Schritt 5:

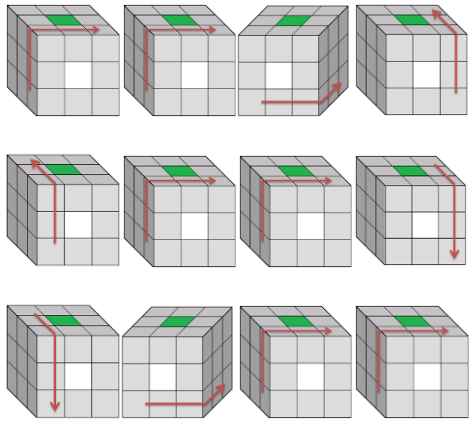


Schritt 6:



Schritt 7:

Sequenz 1:



Sequenz 2: