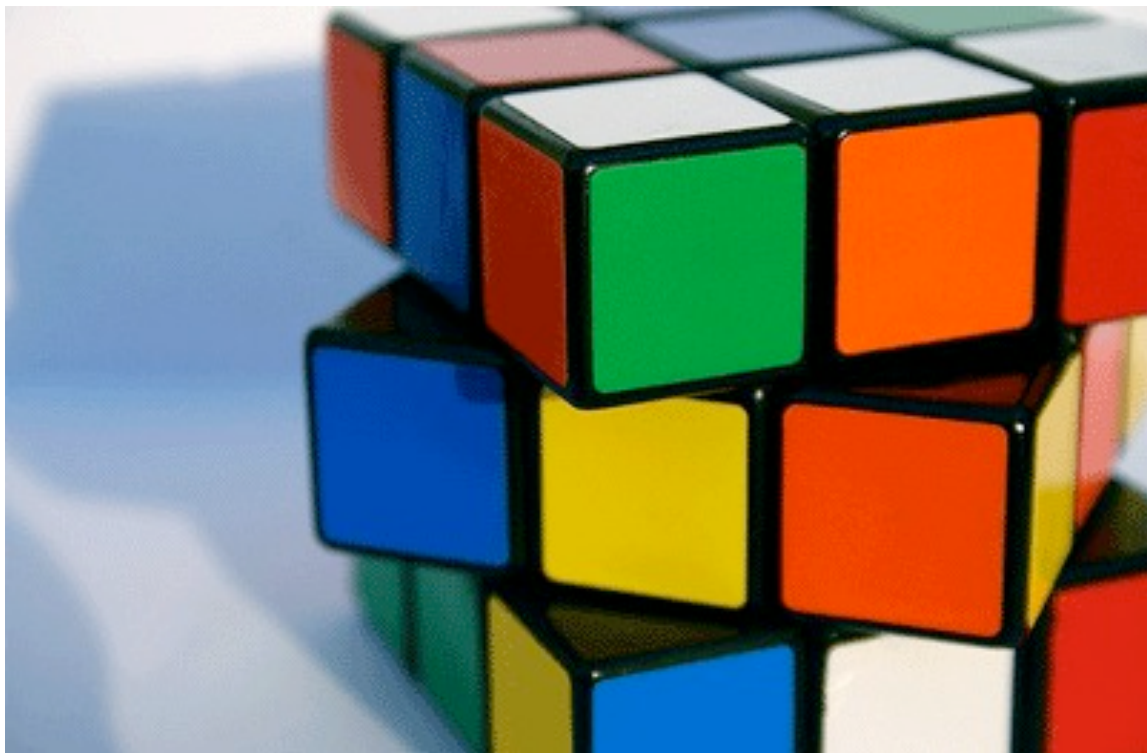

Rubik's Cube Solver

Informatikprojekt

Noëlle Rosenberg & Selim Öksüz
Hochschule für Technik Zürich • June 14, 2011



Inhaltsverzeichnis



Einleitung	4
<i>Aufgabenstellung</i>	4
<i>Ziele</i>	5
Der Rubik's Cube	6
<i>Ein bisschen Geschichte</i>	6
<i>Rund um den Aufbau des Cubes</i>	7
<i>Wir nehmen den Rubik's Cube noch genauer unter die Lupe</i>	7
Wir lösen den rubik's Cube	9
<i>Begriffe</i>	9
<i>Programmstruktur</i>	10
<i>Übersicht</i>	11
<i>Model</i>	11
<i>Cube</i>	11
<i>Cubie</i>	11
<i>HistoryItem</i>	12
<i>Controller</i>	12
<i>CubeSolver</i>	12
<i>View</i>	12

Algorithmen	13
<i>Die sieben Schritte zur Lösung des Rubik's Cubes</i>	13
1. Schritt: Mache ein Kreuz in der Top Surface	13
2. Schritt: Platziere die Eck-Cubies der Top Surface	14
3. Schritt: Positioniere die Kanten-Cubies der mittleren Cube-Reihe	14
4. Schritt: Mache ein Kreuz in der Bottom Surface	15
5. Schritt: Drehe die Kanten-Cubies an die richtige Position	16
6. Schritt: Positioniere die Eck-Cubies der Bottom Surface	17
7. Schritt: Drehe die Farben der Eck-Cubies an die richtige Position	17
Graphical User Interface	19
Übersicht	19
Bedienungs-Panel	20
Elemente	20
Lösungs-Panel	21
Auflistungs-Ansicht	21
Schritt-für-Schritt-Ansicht	22
Elemente	22
Kleines Schlusswort	24
Literaturverzeichnis	25

EINLEITUNG

Schon mancher hat sich stundenlang mit ihm beschäftigt, sich mit ihm herumgeschlagen, ihn gedreht, gewendet und geschraubt. Viele Nerven beim verzweifelten Versuch, alle gleichfarbigen Würfelchen auf die korrekte Ebene zu bringen gebraucht und ihn dann schlussendlich doch müde und erschöpft in der hintersten Schubladenecke verstaut.



Genau, die Rede ist vom Rubik's Cube!

Wie funktioniert er denn nun wirklich? Welcher bzw. welche Algorithmen stecken hinter diesem magischen Zauberwürfel?

Diesen Fragen wollen wir in unserem Informatikprojekt und damit in der folgenden Dokumentation nachgehen.

Aufgabenstellung

In unserer Projektarbeit dreht sich alles rund um das Lösen eines solchen Rubik's Cubes.

Wir erstellen eine kleine Applikation, welche es dem Benutzer ermöglicht, einen beliebigen Rubik's Cube einzugeben und diesen per Knopfdruck lösen zu lassen. Zudem soll die Lösung des Rubik's Cubes dem Benutzer so präsentiert werden, dass er die einzelnen Schritte auch gleich selbst nachvollziehen kann und zum Schluss seinen richtigen, physischen Cube gelöst vor sich liegen hat.

Ziele

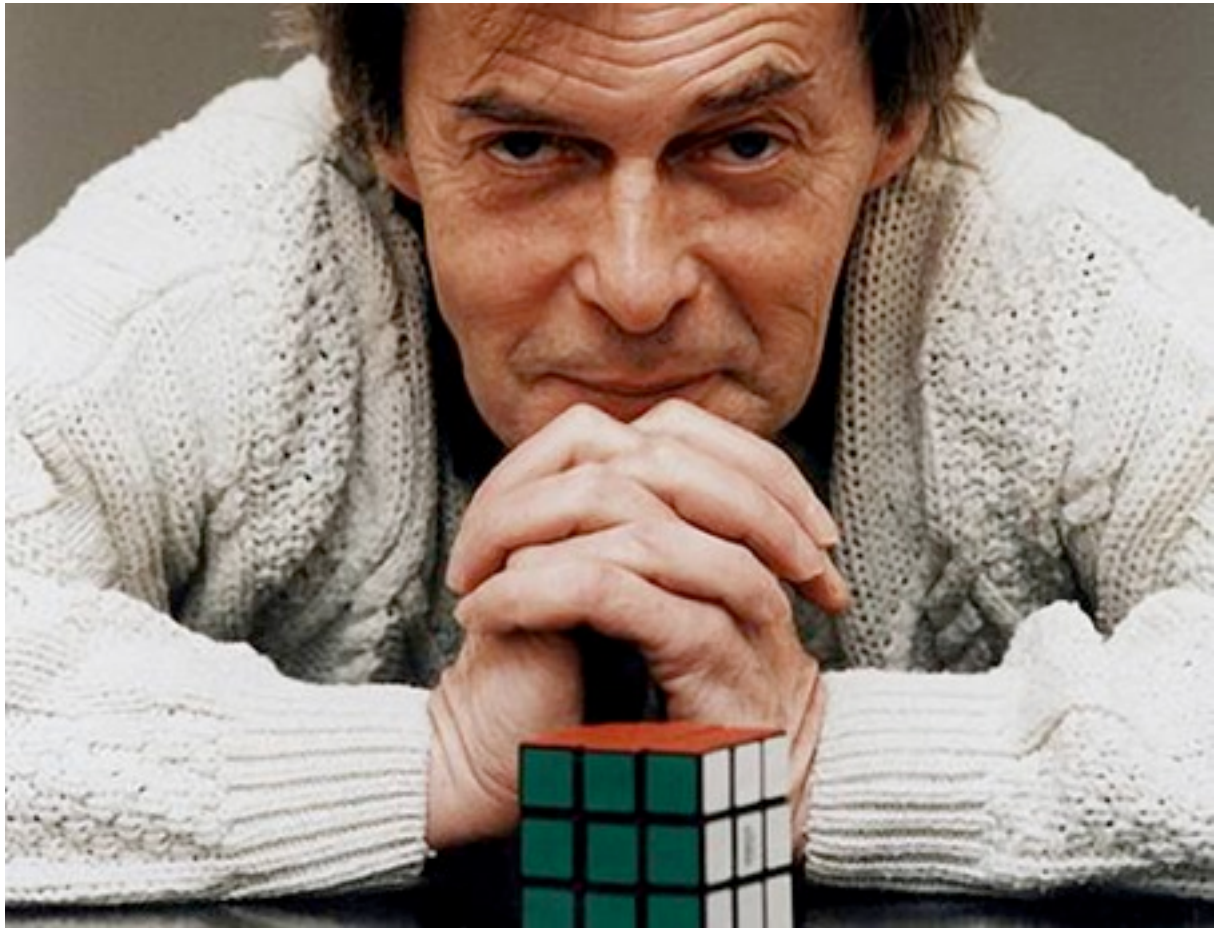
- Verstehen, Nachvollziehen und Anwendung des Algorithmus, bzw. der Algorithmen zur Lösung eines Rubik's Cubes.
- Entwurf einer geeigneten Datenstruktur zur Lösung eines Rubik's Cubes
- Implementierung der Algorithmen mit C# und dem .NET Grafik-Framework Windows Presentation Foundation (WPF)
- Implementierung eines benutzerfreundlichen graphischen User Interfaces zur Eingabe eines beliebigen Zauberwürfels und der Darstellung der einzelnen Lösungsschritte in 2D

DER RUBIK'S CUBE

Ein bisschen Geschichte

Der Rubik's Cube ist ein dreidimensionales, mechanisches Geduldspiel. Er wurde im Jahre 1975 von dem ungarischen Bauingenieur und Architekten Ernő Rubik erfunden und sollte den damaligen Studenten Rubik's die Möglichkeit geben, ihr räumliches Denkvermögen zu trainieren.

Schon bald gewann der Zauberwürfel an weltweiter Beliebtheit von Gross und Klein und wurde 1980 gar mit dem Sonderpreis *Bestes Solitärspiel* des Kritikerpreises *Spiel des Jahres* ausgezeichnet.





Rund um den Aufbau des Cubes

Der Rubik's Cube setzt sich aus 26 Würfelchen zusammen, welche in der richtigen Position die sechs verschieden farbigen Seiten des Cubes ergeben.

Alle sechs Seiten können mit dem, sowie gegen den Uhrzeigersinn rotiert werden. Der Zauberwürfel lässt sich also ebenenweise vertikal, sowie horizontal, gegen den, sowie mit dem Uhrzeigersinn drehen.

Wir nehmen den Rubik's Cube noch genauer unter die Lupe

Folgende Elemente werden ersichtlich:

Das Achsenkreuz

Das Achsenkreuz ist das Herzstück des Rubik's Cubes.

Die sechs Zentrums-Würfelchen sind fix miteinander verbunden. Daher besitzen sie konstruktionsbedingt immer dieselbe relative Lage zueinander und bestimmen somit, welche Farben aneinandergrenzen müssen.



Die Eck-Würfelchen



Diese acht Würfelchen verbinden je drei angrenzende Flächen in den Ecken des Rubik's Cubes miteinander.

Die Kanten-Würfelchen



Die Zwölf Kanten-Würfelchen verbinden je zwei angrenzende Flächen in den Kantenmitten des Rubik's Cubes miteinander.

WIR LÖSEN DEN RUBIK'S CUBE

Nun wird es spannend. Und technisch!

Im folgenden Kapitel werden wir uns mit der Implementierung der komplexen Lösung eines Zauberwürfels auseinandersetzen.

Begriffe

Sie werden nun auf viele Rubik's Cube spezifische Begriffe stossen.

Damit die Verwirrung nicht allzu gross wird, sind diese im Folgenden in einer Begriffsübersicht - Tabelle aufgelistet:

Begriff	Beschreibung
Cube	Der eigentliche Rubik's Cube, bzw. Zauberwürfel.
Cubie	Eines der 26 Würfelchen, aus welchen der Cube zusammengesetzt ist.
Surface	Eine der sechs Oberflächen des Cubes
Top	Die obere Surface des Cubes
Front	Die von vorne her betrachtete Surface des Cubes
Bottom	Die untere Surface des Cubes
Left	Die linke Surface des Cubes
Right	Die rechte Surface des Cubes
Back	Die hintere Surface des Cubes
Rotation	Die Rotation beschreibt eine der Möglichkeiten, den Cube ebenenweise zu drehen bzw. zu rotieren. Jede Surface lässt sich horizontal, sowie vertikal, gegen den und mit dem Uhrzeigersinn drehen.

Begriff	Beschreibung
TR / TL	Abkürzungen, welche für die Beschreibung der Algorithmen verwendet werden. Sie beschreiben, welche Surface in welche Richtung gedreht (rotiert) wird. TR steht hierbei für Top Right, TL für Top Left. FR/ FL => Front Right/ Front Left BoR/ BoL => Bottom Right/ Bottom Left LR/ LL => Left Right/ Left Left RR/ RL => Right Left/ Right Right BaR/ BaL => Back Right/ Back Left
FR / FL	
BoR / BoL	
LR / LL	
RR / RL	
BaR / BaL	

Datenmodell

Sehr wichtig ist, eine passende Datenstruktur zu finden, was nicht so einfach war, wie zu Beginn des Projektes angenommen.

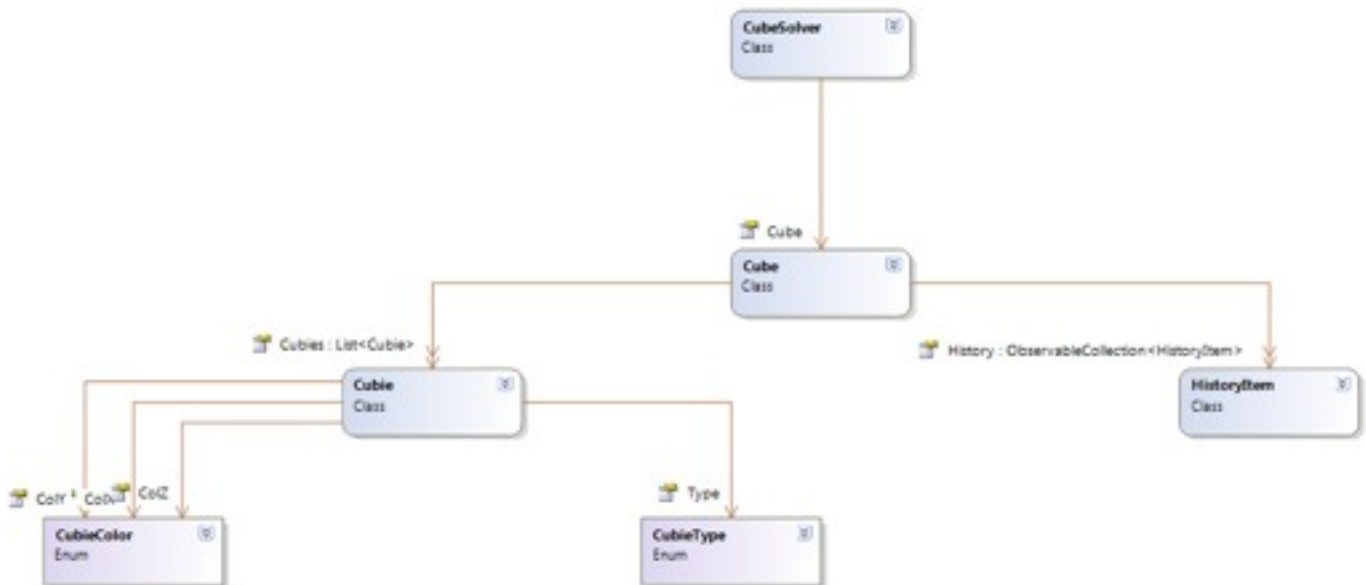
Es gibt zwei verschiedene Ansätze, einen Rubik's Cube zu betrachten.

1. Man betrachtet den Cube als Zusammensetzung der sechs Flächen (Surfaces) Top, Front, Bottom, Left, Right und Back, welche matritzenförmig aufgebaut sind (3x3 Matrix) und welche dann korrekt miteinander verknüpft die Form des dreidimensionalen Würfels ergeben. Bei jeder Rotation einer Surface des Cubes werden die Farben der 9 Sub-Flächen der betroffenen Surfaces jeweils neu gesetzt. Dieser Ansatz beruht auf der Vektor- und Matrizen-Rechnung, wobei die Rotationen anhand von Permutationsmatrizen erfolgen.
2. Der zweite Ansatz ist, den Cube als Zusammensetzung seiner 26 Cubies zu betrachten, welche in Eck-, Kanten- und Zentrums-Cubies unterteilt werden. Werden alle Cubies korrekt herum gedreht ihren bestimmten Positionen im Cube zugeteilt, erhält man den gelösten Cube. Dieser Ansatz entspricht einem Objekt orientierten Ansatz. Bei der Rotation einer Surface, erhalten die betroffenen Cubies dann einen neuen Platz im Cube.

Wir haben uns für den zweiten, vielleicht auch etwas moderneren, Objekt orientierten Ansatz entschieden, da uns dieser Weg eine einfachere, vorstellbarere Abbildung des Würfels aus der realen Welt in das Datenmodell ermöglicht.

Übersicht

Unsere Applikation ist nach dem Model, View, Controller (MVC) Entwurfsmuster aufgebaut.



Model

Cube

Die Klasse **Cube** stellt den eigentlichen Rubik's Cube dar. Der Cube besteht aus seinen 26 Cubies, welche alle eine bestimmte Koordinatenposition im Cube einnehmen. Jeweils 9 Cubies zusammengefasst ergeben die sechs Surfaces, Top, Front, Bottom, Left, Right und Back des Cubes. Man kann den Cube drehen und von einer beliebigen Seite (Surface) her betrachten. Auch ist es möglich ihn ebenenweise (surfaceweise) zu rotieren.

Eine spezielle Eigenschaft des Cubes ist sein Gedächtnis. Er weiss stets, wieviel mal und welche seiner Surfaces in welche Richtung rotiert wurde.

Cubie

Die **Cubie** Instanzen stellen die Würfelchen, aus welchen der Cube aufgebaut ist dar. Sie besitzen alle einen bestimmten Typ, Eck-, Kanten- oder Zentrums-Cubie. Jedes Cubie besitzt eine, zwei oder drei Oberflächenfarben. Zentrums-Cubies haben eine, Kanten-Cubies zwei und Eck-Cubies drei farbige Oberflächen.

Cubies wissen an welcher Koordinaten-Position sie sich im Cube befinden. Ihr eigenes Koordinatensystem benötigen sie, um ihre Oberflächenfarbe (nach Rotationen) richtig setzen zu können.

HistoryItem

Die HistoryItems entsprechen dem Gedächtnis des Cubes.

Jede Cube Rotation wird in einem HistoryItem mit seiner Nummerierung , einer textuellen sowie bildlichen Beschreibung des Lösungsschrittes festgehalten.

Controller

CubeSolver

Die CubeSolver Klasse stellt den Controller dar, welcher fähig ist, einen Cube zu lösen. Dabei sucht er verschiedene Cubies und wendet die Algorithmen an, die zur Lösung des Rubik's Cubes erforderlich sind.

View

Die View entspricht dem User Interface.

Sie befindet sich in einem eigenen Projekt, damit Businesslogik und User Interface getrennt voneinander behandelt werden.

Wir werden im Kapitel Graphical User Interface genauer darauf eingehen.

ALGORITHMEN

Um einen beliebigen Cube lösen zu können, benötigt unser CubeSolver verschiedene Algorithmen. In diesem Kapitel werden wir diese Algorithmen nun genauer erläutern.

Die sieben Schritte zur Lösung des Rubik's Cubes

Es gibt verschiedene Wege den Rubik's Cube zu lösen. Wir haben uns für einen Weg entschieden, welcher zwar nicht der effizienteste, jedoch ein gut verständlicher und einfach in Code zu überführender Weg ist. Er besteht aus sieben Hauptschritten.

1. Schritt: Mache ein Kreuz in der Top Surface

Hierfür müssen die vier Kanten-Cubies der Top Surface des Würfels an die korrekte Position verschoben werden, sodass ein Kreuz entsteht.

D.h. Wir suchen ein Kanten-Cubie nach dem anderen und positionieren es an der richtigen Kanten-Position der Top Surface.

Um anschliessend Kanten-Cubies, deren Farben noch verkehrt herum in der Position liegen, in die richtige Stellung zu drehen, verwenden wir den folgenden Algorithmus genau ein mal:

FR RL BoL RR FR FR



Wir erkennen das Kreuz in der Top Surface. Die Farbe stimmen auch mit den jeweiligen Zentrums-Cubies überein.

2. Schritt: Platziere die Eck-Cubies der Top Surface

Nun werden die Eck-Cubies mit den entsprechenden Farben gesucht und in die jeweiligen Eckpositionen der Top Surface platziert.

D.h. Wir suchen ein Eck-Cubie nach dem anderen und positionieren es an der richtigen Ecken-Position der Top Surface, ohne das zuvor gemachte Kreuz zu zerstören.

Um anschliessend Eck-Cubies, deren Farben noch verkehrt herum in der Position liegen, in die richtige Stellung zu drehen, verwenden wir den folgenden Algorithmus solange, bis die Farben in der richtigen Stellung liegen:

RL BoL RR BoR



Die Top Surface ist nun gelöst und wenn wir die angrenzenden Surfaces betrachten, erkennen wir ein "T" auf jeder Seite.

3. Schritt: Positioniere die Kanten-Cubies der mittleren Cube-Reihe

Die Kanten-Cubies werden nun an die richtigen Positionen neben den "T"s gebracht. Natürlich wieder ohne das zuvor Gelöste zu zerstören. Dazu muss der Cube zuerst um 180° in vertikaler Richtung gedreht werden.

D.h. Wir stellen den Cube also zuerst auf den Kopf, sodass wir die "T"s nun verkehrt sehen.



Nun suchen wir ein Kanten-Cubie nach dem anderen (von der Front her betrachtet das rechte Cubie) und platzieren es immer zuerst (wieder von der Front her betrachtet) in der oberen Kanten-Position. Nur dann können wir den folgenden Algorithmus anwenden, um den Cubie von der oberen Kanten-Position in die rechte Kanten-Position neben dem "T" zu verschieben. Kommt ein Kanten-Cubie falsch herum in der korrekten Position zu liegen, wenden wir den Algorithmus weiter an, positionieren das Cubie dann wieder in der oberen Kanten-Position und führen den Algorithmus nochmals aus.

TR RR TL RL TL FL TR FR



Alle an die Top Surface grenzenden Flächen (Surfaces) haben nun zwei korrekt gelöste Reihen.

4. Schritt: Mache ein Kreuz in der Bottom Surface

Nun soll auch in der Bottom Surface ein Kreuz aus dem Zentrums-Cubie und den vier Kanten-Cubies erstellt werden. Die Kanten-Cubies müssen noch nicht an der korrekten Position liegen, ausschliesslich das Kreuz muss ersichtlich sein.

D.h. Wir müssen vor jeder Anwendung des folgenden Algorithmus drei Fälle unterscheiden, bevor wir diesen Algorithmus ausführen können.

1. Fall: Ein Kanten-Cubie befindet sich bereits an einer Kreuz-Position.

In diesem Fall können wir den Algorithmus einfach ausführen.

2. Fall: Zwei Kanten-Cubies befinden sich bereits an Kreuz-Positionen, die Cubies bilden eine "Linie".

In diesem Fall müssen wir den Cube so drehen, dass wir die Linie waagrecht, also horizontal anschauen, anschliessend können wir den Algorithmus anwenden.

3. Fall: Zwei Kanten-Cubies befinden sich bereits an Kreuz-Positionen, die Cubies bilden eine "Ecke" (oder ein kurzes "L").

In diesem Fall müssen wir den Cube so drehen, dass sich die "Ecke" (das "L") oben links befindet. Anschliessend können wir den Algorithmus anwenden.

FR RR TR RL TL FL



Nun bilden die Kanten-Cubies ein Kreuz in der Top Surface, sie liegen jedoch noch nicht an den richtigen Positionen im Cube.

5. Schritt: Drehe die Kanten-Cubies an die richtige Position

Die Kanten-Cubies werden nun auch noch in die korrekte Position gedreht.

D.h. Wir gehen wieder von einem Kanten-Cubie zum nächsten und wenden den folgenden Algorithmus an:

RR TR RL TR RR TR TR RL TR



Das Kreuz in der Bottom Surface ist noch immer zu erkennen und die Kanten-Cubies befinden sich an der korrekten Position im Cube.

6. Schritt: Positioniere die Eck-Cubies der Bottom Surface

Die Eck-Cubies werden nun an die korrekte Koordinaten-Position im Cube verschoben. Die Farben dürfen falsch herum sein.

D.h. Wir suchen ein Eck-Cubie nach dem anderen und positionieren es an der richtigen Ecken-Position der Bottom Surface. Hierfür verwenden wir den folgenden Algorithmus:

TR RR TL LL TR RL TL LR



Nun sind die Eck -Cubies an der richtigen Koordinaten-Position im Cube platziert, die Farben können aber noch falsch herum liegen.

7. Schritt: Drehe die Farben der Eck-Cubies an die richtige Position

Im letzten Schritt werden die Farben der Eck-Cubies so gedreht, dass sie richtig herum an ihren Eck-Positionen liegen.

D.h. Wir verwenden den folgenden Algorithmus, um die Eck-Cubies, deren Farben noch verkehrt herum in der Position liegen, in die richtige Stellung zu drehen, bis die Farben in der richtigen Stellung liegen.

Wichtig ist hierbei, dass nach jedem richtig gedrehten Eck-Cubie, nicht der ganze Cube sondern nur die Top Surface eine Ecke weiter nach rechts gedreht wird.

RL BoL RR BoR

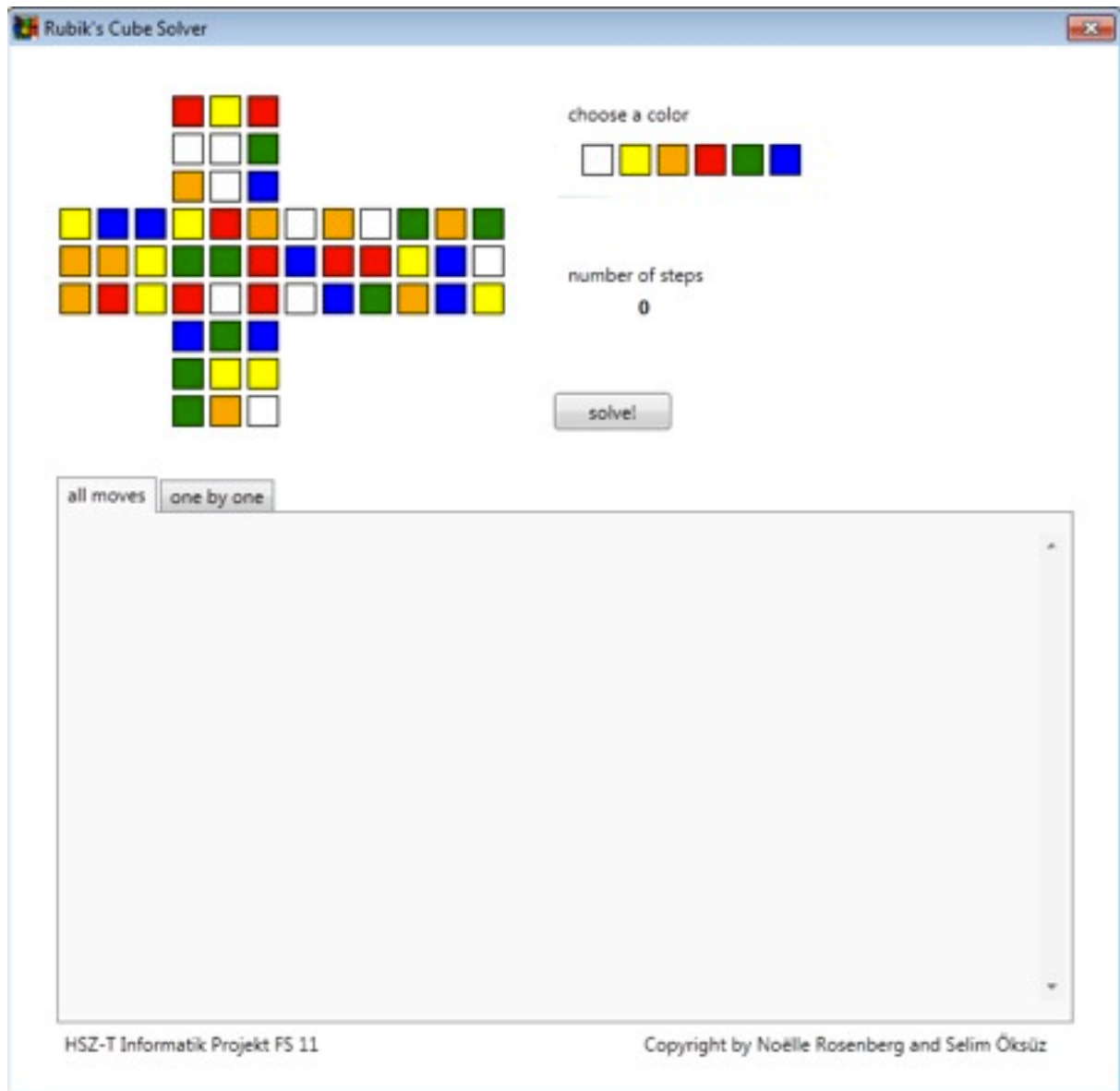


Und fertig! Der Rubik's Cube ist gelöst!

GRAPHICAL USER INTERFACE

In diesem Kapitel schauen wir uns das Graphical User Interface genauer an. Es dient als eine kleine Bedienungsanleitung unseres Rubik's Cube Solver Programms.

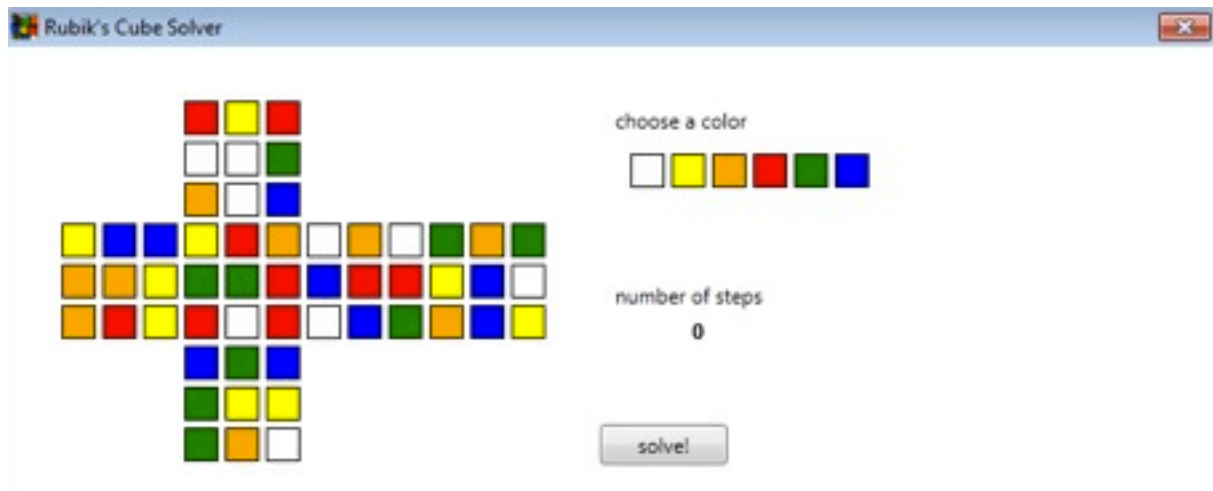
Übersicht



Der Rubik's Cube Solver ist in zwei Teile gegliedert.

Einem Bedienungs-Panel, in welchem man die Farbauswahl des Cubes setzen und diesen Würfel dann auch lösen lassen kann, und einem Lösungs-Panel, in welchem die verschiedenen Schritte zur Lösung des gewünschten Cubes dargestellt werden.

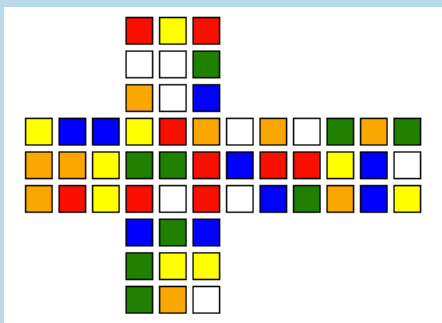

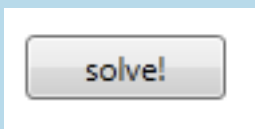
Bedienungs-Panel



Das Bedienungs-Panel ermöglicht die Eingabe eines beliebigen Rubik's Cubes, indem man seine verschiedenen Farben setzt.

Hat man die gewünschten Farben gesetzt so wird der Cube durch einen Klick auf "solve!" gelöst und im Lösungs-Panel angezeigt.

Elemente

Element	Beschreibung
	<p>Der aufgeklappte Rubik's Cube ermöglicht durch Klick auf die Cubie Oberflächen eine beliebige Farbauswahl.</p> <p>Der Benutzer gibt so einen beliebigen, zu lösenden Rubik's Cube ein.</p>
	<p>Durch Klick einer Farbe im "choose a color" Element wählt der Benutzer seine Farbe, welche er im aufgeklappten Rubik's Cube setzen möchte aus.</p>
	<p>Ein Klick auf den "solve!" Button löst den zuvor zusammengestellten Rubik's Cube.</p> <p>Die Lösungsschritte werden im Lösungs-Panel dargestellt.</p>

Element	Beschreibung
<div>number of steps</div> <div>0</div>	Das "number of steps" Element dient ausschliesslich zur Information. Es gibt die Anzahl Schritte, die zur Lösung des gewählten Rubik's Cubes gebraucht werden an.

Lösungs-Panel

Das Lösungs-Panel stellt die Schritte, die zur Lösung eines gewünschten Rubik's Cubes gemacht werden müssen übersichtlich dar. Dabei kann zwischen zwei Ansichten gewählt werden, einer Auflistungs-Ansicht und einer Schritt-für-Schritt-Ansicht.

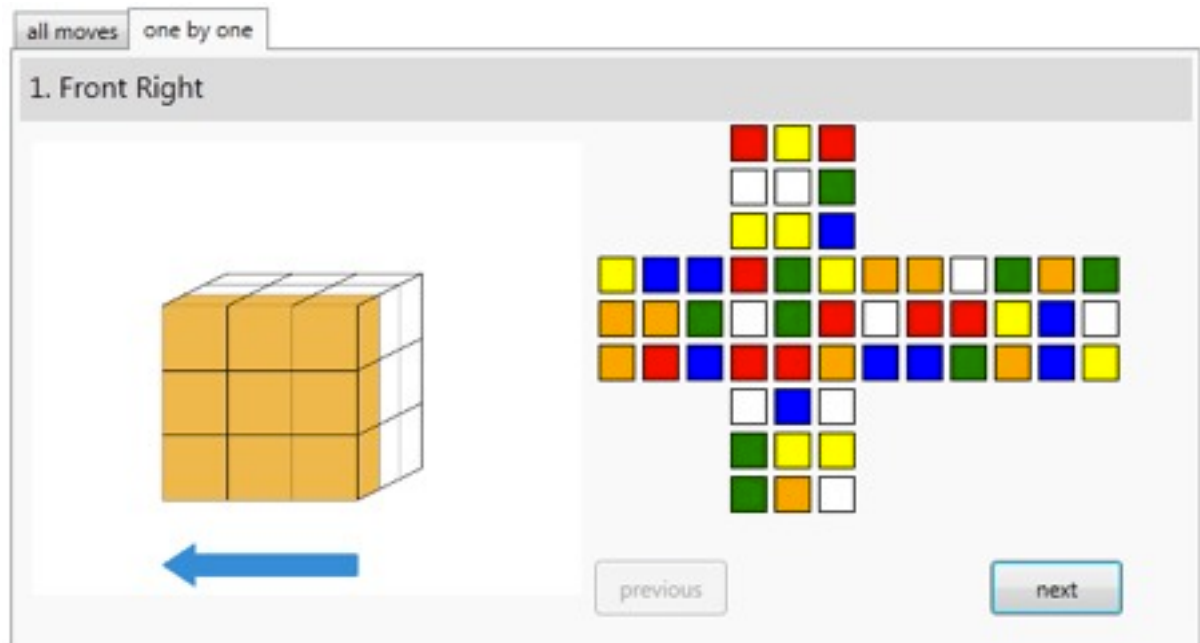
Auflistungs-Ansicht



In der Auflistungs-Ansicht sind alle einzelnen Schritte zur Lösung des Cubes von links nach rechts aufgelistet. Mit dem Scroll-Balken kann weiter nach unten gescrollt werden.

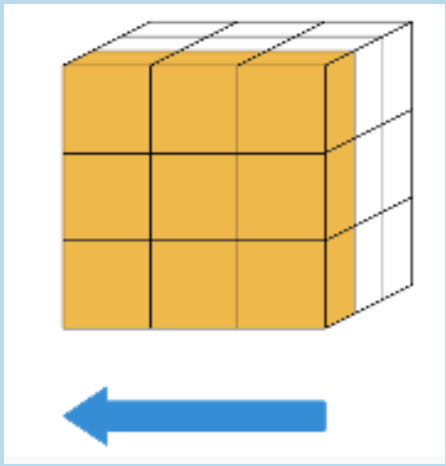
Jeder Schritt besteht aus einer textuellen Information, wie der Cube oder eine Cubes Surface zu rotieren ist und einem Cube Bild, welches die Änderungen der Cubies graphisch darstellt.

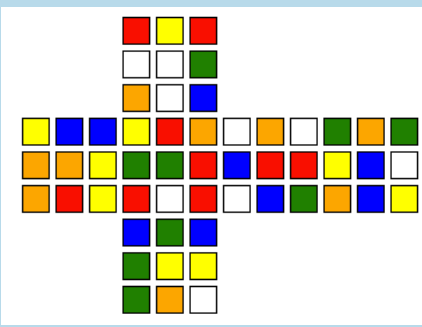
Schritt-für-Schritt-Ansicht



Die Schritt-für-Schritt-Ansicht präsentiert jeweils einen einzelnen Schritt zur Lösung des zuvor zusammengestellten Rubik's Cubes. Durch Klick auf "next" bzw. "previous" kann durch die einzelnen Lösungs-Schritte navigiert werden.

Elemente

Element	Beschreibung
1. Front Right	Textuelle Beschreibung , die angibt was für diesen einzelnen Schritt zu tun ist.
	Visuelle Darstellung, die angibt was für diesen einzelnen Schritt zu tun ist.

Element	Beschreibung
	<p>Dieser aufgeklappte Rubik's Cube zeigt die Vorschau, wie der Cube aussehen wird, nachdem der Lösungs-Schritt ausgeführt wurde.</p>
<div>next</div> <div>previous</div>	<p>Die "next" und "previous" Buttons dienen der Navigation durch die einzelnen Lösungs-Schritte.</p>

KLEINES SCHLUSSWORT

Das Rubik's Cube Solver Projekt war eine sehr spannende und herausfordernde Arbeit.

Aller Anfang ist schwer, so mussten wir uns zu Beginn enorm in die dreidimensionale Datenmodellierung mit den absoluten und relativen Koordinatensystemen eindenken, was mehrere Anläufe einer brauchbaren Datenstrukturfindung zur Folge hatte. Ein richtiges logisches und abstraktes Denkvermögens-Training also. Genau das, was Ernő Rubik zum Ziel hatte, als er den diesen verflixten Zauberwürfel erfand.

Genug, bzw. viel Zeit in die Entwicklung der Datenstruktur zu investieren hat sich jedoch gelohnt. Alle weiteren Schritte, wie die Implementierung der Rotationen und Algorithmen gingen wunderbar auf. Auch das Graphical User Interface konnte ohne grössere Probleme an das Datenmodell geknüpft werden.

Wirklich eine sehr interessante Arbeit! Es hat Spass gemacht!

Nun wünschen wir viel Erfolg beim Ausprobieren und Zauberwürfel lösen! :)

LITERATURVERZEICHNIS

Wikipedia Zauberwürfel (2011)

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

Youtube, Rubik's Cube: Zauberwürfel lösen (Teil 1 von 3); firefox82; (2011);

<http://www.youtube.com/watch?v=mfygeOarPBo>

Youtube, Rubik's Cube: Zauberwürfel lösen (Teil 2 von 3); firefox82; (2011);

<http://www.youtube.com/watch?v=fVxn5geTOpM>

Youtube, Rubik's Cube: Zauberwürfel lösen (Teil 3 von 3); firefox82; (2011);

http://www.youtube.com/watch?v=bYsUfQ_KTxo