

2016

RubiksCubezüge

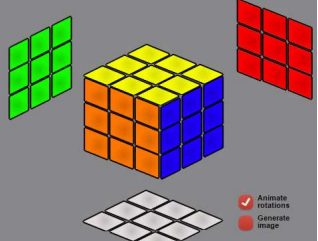
RUBIKS CUBE ROBOTER
STOOP ADRIAN

Inhalt

Welche Drehungen gibt es und welche kann der Roboter	2
Was ist ein Magic Move?.....	4
Aufbau Code, Funktionen und Werte	5
Züge/ Algorithmus.....	7
Was ist ein Stage?.....	7
Stage 0 und 1	7
Stage 0 Abfrage des Codes:	8
Stage 2	9
Stage 2 Abfrage des Codes	10
Stage 3	11
Stage 3 Abfrage des Codes	11
Stage 4	12
Stage 4 Abfrage des Codes, Phase 1.....	12
Stage 4 Abfrage des Codes, Phase 2.....	13
Stage 5	13
Stage 5, Abfrage des Codes	14
Stage 6	14
Stage 6, Abfrage des Codes	15

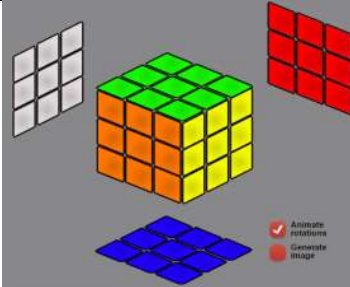
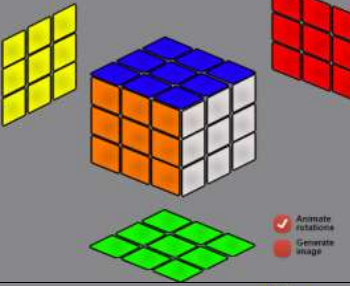
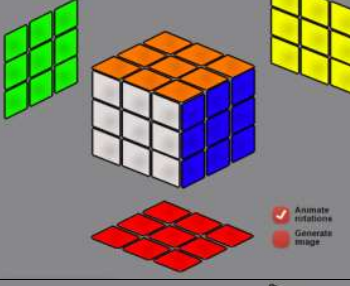
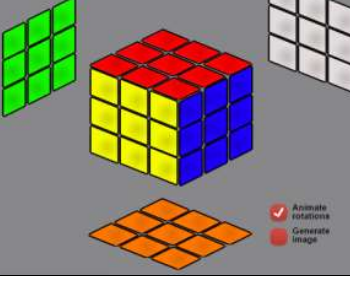
Welche Drehungen gibt es und welche kann der Roboter

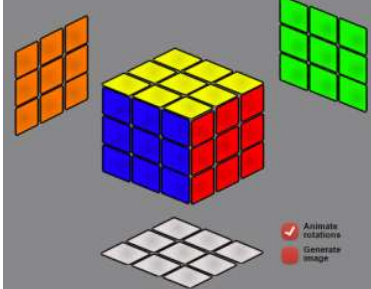
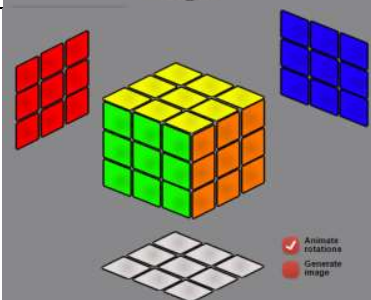
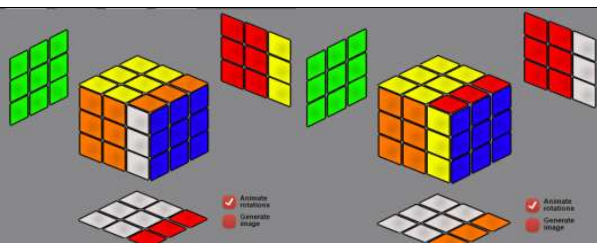
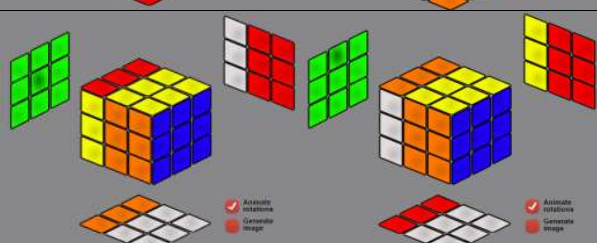
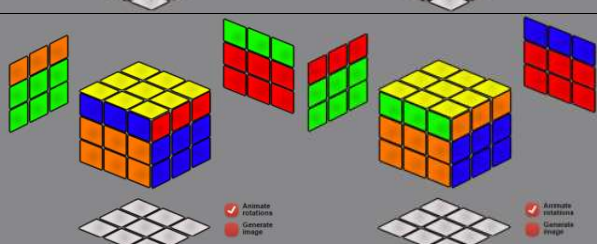
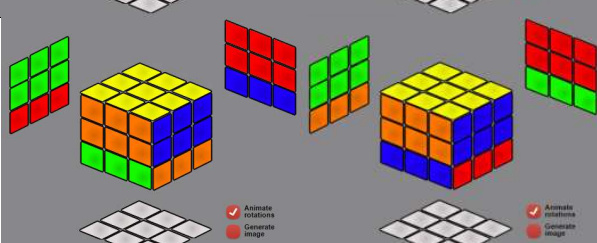
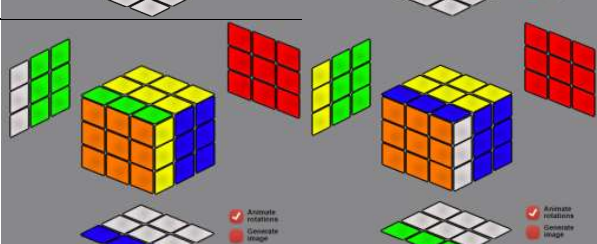
In der Rubiks Fachsprache gibt es folgende Bewegungen:

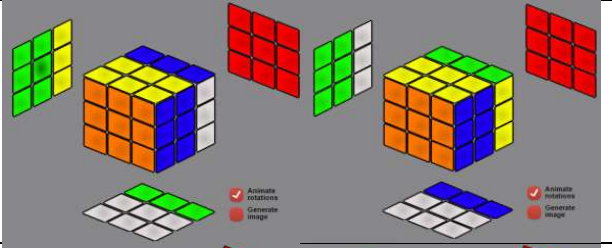
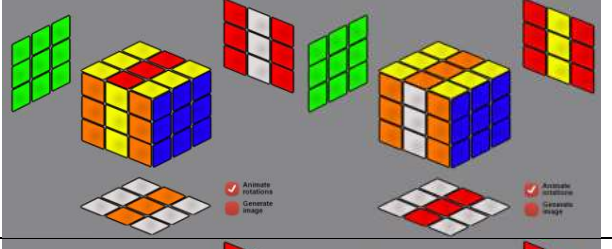
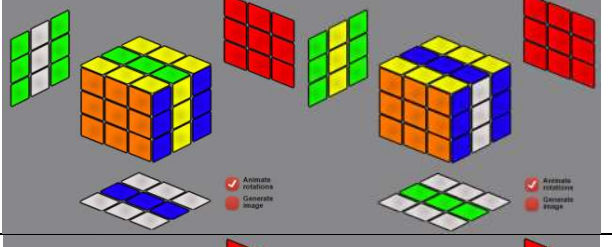
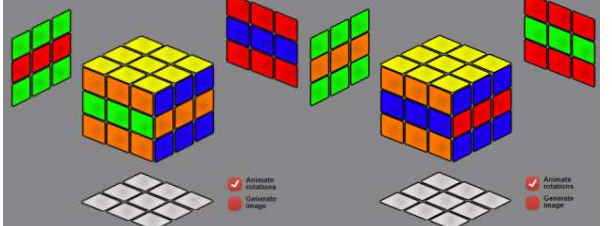
Urstellung:	
-------------	--

*für die Bilder benutze ich die Seite: rubiks-cube-solver.com, da ich die Anzeige sehr hilfreich finde

Jede Bewegung habe ich aus der Urstellung gedreht.

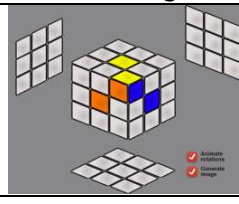
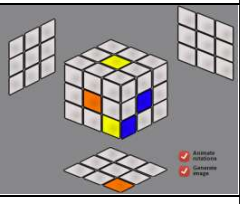
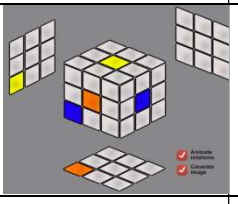
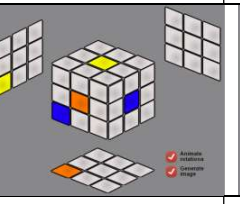
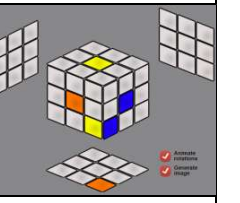
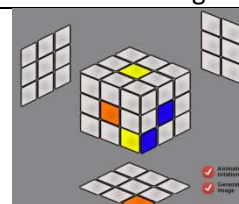
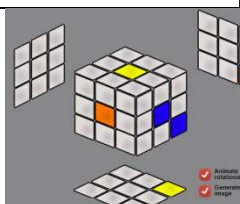
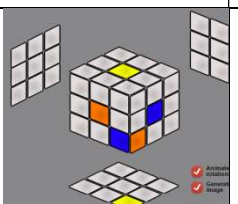
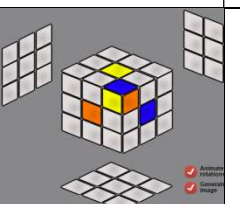
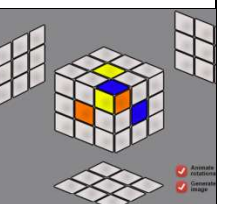
Kürzel oder Bewegung:	Im Code	Bewegung:
Turn Cube Right	MVC_Right	
Turn Cube Left	MVC_Left	
Turn Cube Up	MVC_Up	
Turn Cube Down	MVC_Down	

Turn Cube Clock	MVC_CLK	
Turn Cube !Clock	MVC_NOTCLK	
R & Ri	MC_R & MC_Ri	
L & Li	MC_L & MC_Li	
U & Ui	MC_U & MC_Ui	
D & Di	MC_D & MC_Di	
F & Fi	MC_F & MC_Fi	

B & Bi	MC_B & MC_Bi	
M & Mi	MC_M & MC_Mi	
S & Si	MC_S & MC_Si	
E & Ei	MC_E & MC_Ei	

Was ist ein Magic Move?

Ein Magic Move ist eine Zugsabfolge, die gewisse Steine vertauscht doch andere an ihrer Position lassen. So ein Zug ist z.B. Ri, Di, R und D. Würde man diesen Zug bei einem Gelösten Würfel 6x Anwenden, würde er wieder gelöst sein. Alle 2-mal wäre der Eckstein wieder an seiner Position aber um 120 Grad(gegen Uhrzeigersinn) vertauscht.

Grundstellung	Ri	Di	R	D
				
2. Grundstellung	Ri	Di	R	D
				

Würde man jetzt den oberen Layer drehen und das ganze wiederholen, so wären irgendwann alle 4 oberen Ecksteine richtig in ihrer Position & ausgerichtet, sofern die 4 Ecksteine schon richtig waren nur die Ausrichtung falsch ist.

Aufbau Code, Funktionen und Werte

Um den Würfel mit seinen Farben abzuspeichern habe ich ein Array erstellt das 12x9 Felder Gross ist. Ich habe den Nummern 1-6 den Farbennamen zugeordnet, damit ich einen einfacheren Überblick habe.

```
#define Dc 0 //dont Care
#define Ws 1 //Weiss/schwarz
#define Ro 2 //Rot
#define Bl 3 //Blau
#define Ge 4 //Gelb
#define Gr 5 //Grün
#define Or 6 //Orange

//Arrayfeld mit rubiks farben
char RubiksFeld[9][12] = { //Default stellung/Ziel

    //Oben
    { Dc, Dc, Dc, Ro, Ro, Ro, Dc, Dc, Dc, Dc, Dc, Dc },
    { Dc, Dc, Dc, Ro, Ro, Ro, Dc, Dc, Dc, Dc, Dc, Dc },
    { Dc, Dc, Dc, Ro, Ro, Ro, Dc, Dc, Dc, Dc, Dc, Dc },

    //Links      front      Rechts      Back
    { Bl, Bl, Bl, Ws, Ws, Ws, Gr, Gr, Gr, Ge, Ge, Ge },
    { Bl, Bl, Bl, Ws, Ws, Ws, Gr, Gr, Gr, Ge, Ge, Ge },
    { Bl, Bl, Bl, Ws, Ws, Ws, Gr, Gr, Gr, Ge, Ge, Ge },

    //Unten
    { Dc, Dc, Dc, Or, Or, Or, Dc, Dc, Dc, Dc, Dc, Dc },
    { Dc, Dc, Dc, Or, Or, Or, Dc, Dc, Dc, Dc, Dc, Dc },
    { Dc, Dc, Dc, Or, Or, Or, Dc, Dc, Dc, Dc, Dc, Dc },

};
```

Um die Farben zu Initialisieren muss ich mit dem Sensor an jede Würfelposition gehen und die Farbe einscannen:

```
6
7
8
9     RubiksFeld[x][y] = GetColorHue();
10
11
12
```

Um jetzt die Lösung des Würfels zu ermitteln muss ich nur noch die Steine abfragen die durch einen Magic move an die richtige Stelle gedreht werden. Z.B:

```
if(RubiksFeld[x][y] == RubiksFeld[x][y])
{
    //diese Zwei Steine haben die Gleiche Farbe!
}
```


Danach kann ich dann den Zug im Rubiks-Array nachstellen und weitere vergleiche machen bis der Würfel gelöst ist. Z.B:

```
//drgeht den zug L
void R_MoveCube_L(void)
{
    Temp_1 = RubiksFeld[3][3];
    Temp_2 = RubiksFeld[4][3];
    Temp_3 = RubiksFeld[5][3];

    RubiksFeld[3][3] = RubiksFeld[0][3];
    RubiksFeld[4][3] = RubiksFeld[1][3];
    RubiksFeld[5][3] = RubiksFeld[2][3];

    RubiksFeld[0][3] = RubiksFeld[5][11];
    RubiksFeld[1][3] = RubiksFeld[4][11];
    RubiksFeld[2][3] = RubiksFeld[3][11];

    RubiksFeld[5][11] = RubiksFeld[6][3];
    RubiksFeld[4][11] = RubiksFeld[7][3];
    RubiksFeld[3][11] = RubiksFeld[8][3];

    RubiksFeld[6][3] = Temp_1;
    RubiksFeld[7][3] = Temp_2;
    RubiksFeld[8][3] = Temp_3;

    Temp_1 = RubiksFeld[3][2];
    Temp_2 = RubiksFeld[4][2];

    RubiksFeld[3][2] = RubiksFeld[3][0];
    RubiksFeld[4][2] = RubiksFeld[3][1];

    RubiksFeld[3][0] = RubiksFeld[5][0];
    RubiksFeld[3][1] = RubiksFeld[4][0];

    RubiksFeld[5][0] = RubiksFeld[5][2];
    RubiksFeld[4][0] = RubiksFeld[5][1];

    RubiksFeld[5][2] = Temp_1;
    RubiksFeld[5][1] = Temp_2;

    R_Wechsel = R_Wechsel + 25; R_Zuege++;
}
```

Ist der Würfel im Programm gelöst muss nur noch der Lösungsweg umgewandelt werden um die Motoren richtig anzusteuern:

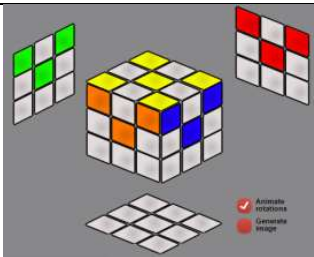
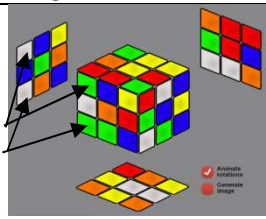
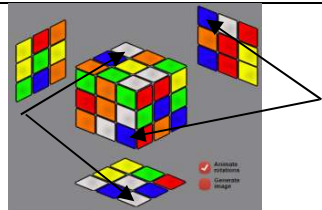
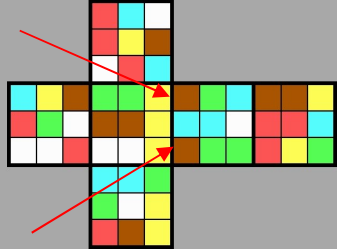
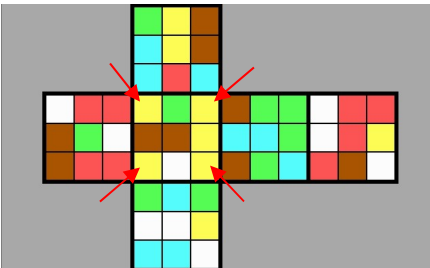
bild@@@@@@@@

Züge/ Algorithmus

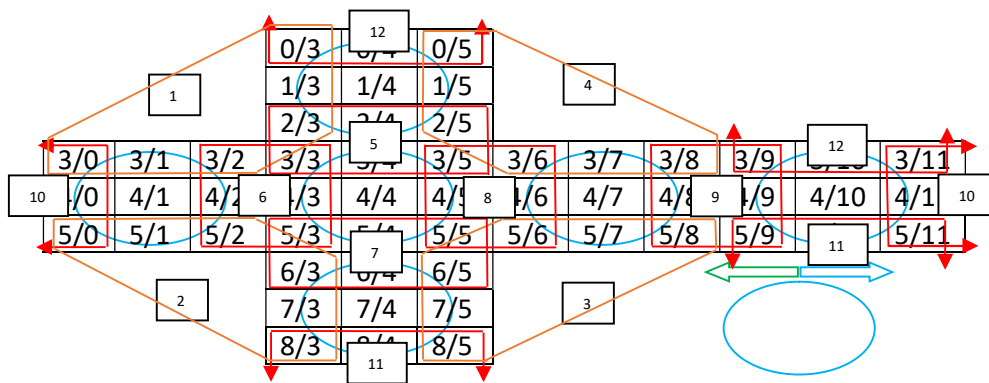
Was ist ein Stage?

Ich habe mir den Algorithmus den ich brauchen möchte angeschaut und habe ihn in verschiedene Teile(Stage) aufgeteilt. Es ist eine Art Zwischenziel, denn ohne die vorherigen Stages kann man nicht den nächsten beenden. Die Farben des Würfels in den folgenden Tabellen muss nicht zwingend diese Farbe sein die angegeben wird(d.h. bei Stage 1 muss nicht mit Gelb gearbeitet werden, bloss die gleiche Farbe). Die Steine die eine Farbe haben sind entweder schon richtig in ihrer Position oder sie sind Betroffen und müssen abgefragt werden um den Sollzustand zu erreichen.

Stage 0 und 1

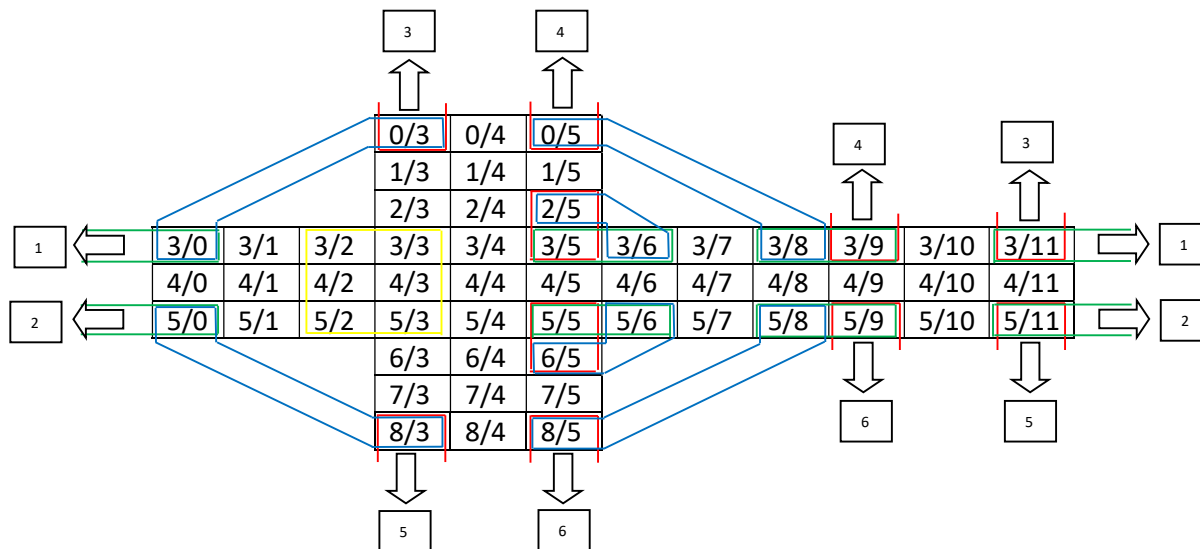
Sollzustand:	Möglicher Zustand:	Massnahme(Algorithmus):
	 <p>Eck paar</p>	Eck paare suchen, die schon richtig in ihrer Position sind(pfeile). Danach die andren Ecksteine (mit z.B. Ri, Di, R und D) an die Richtige Position bringen.
Dieser Stage ist in bis zu 6 Zügen zu meistern. Man muss nur die 8 Ecksteine beachten und 4 davon richtig platzieren & Ausrichten	 <p>Kein Paar</p>	Eck Paar bilden und obere Anweisung folgen(hier 2R 2Ri).
In Stage 0 werden 2 Ecksteine ausgerichtet(oder gefunden) und in Stage 1 die anderen 2, sodass alle 4 Ausgerichtet sind.		Bild von Muster nach Stage 0 mit einem richtigen ungelösten Würfel.
		Bild von Muster nach Stage 1 mit einem richtigen ungelösten Würfel.

Stage 0 Abfrage des Codes:



Zuerst wir das 2er Paar gesucht(rot & orange, z.B. $3/2 == 5/2$ && $3/3 == 5/3$).

Wenn keins gefunden wurde, drehe eine der Seiten und teste es nochmal bis jede Seite(12 Möglichkeiten) getestet wurden(blau & Grün). Wenn kein 2er Paar gefunden wurde wird die Seite zurückgedreht, damit der Lösungsweg möglichst kurz ist.

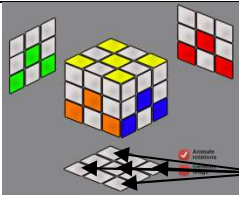
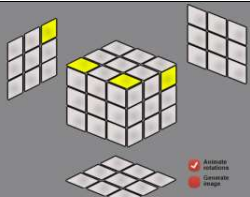
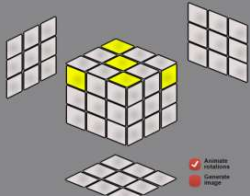
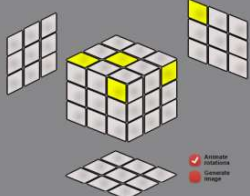
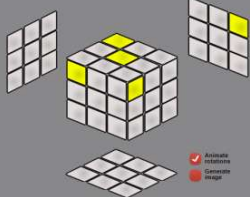
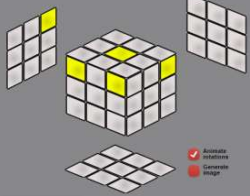
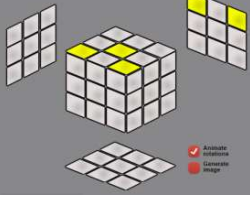
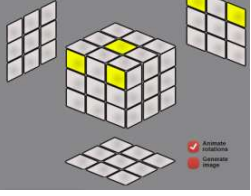


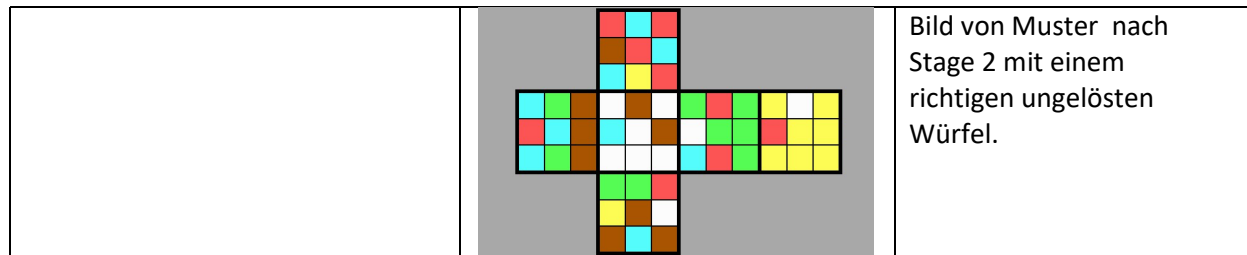
Das 2er Paar wird nun (nur im Code) in die gelbe Markierung gedreht. Der Lösungsweg muss auch mit gedreht werden! Von dort werden die zwei anderen Ecksteine gesucht (rot & blau & grün) und an ihre Position gedreht (24 Möglichkeiten, davon 2x3 schon benutzt (gelb, z.B. $5/2$, $5/3$ und $6/3$)).

Am Schluss wird das 2er Paar wieder an seinen Ursprung gedreht. Der Lösungsweg muss auch mit gedreht werden.

Stage 2

Würfel nur im Code um 180 Grad drehen und Mittelsteine Richtig Positionieren.

Sollzustand:	Möglicher Zustand:	Massnahme(Algorithmus):
 <p>gleiche Farbe</p>	 <p>T-Zustand</p>	R U Ri Ui Fi Ui F
Ziel dieses Stage ist, dass alle 8 Ecksteine an ihrer Position sind. 4 davon sind schon im Stage 0 erledigt worden(Position & Ausrichtung).	 <p>L-Zustand</p>	F Ri Fi Ui Ri U R
Die anderen werden mit folgenden Algorithmen direkt richtig ausgerichtet, nicht richtig Positioniert. Max 7 Züge.	 <p>M-Zustand</p>	R U Ri U R 2U Ri
Die 4 Ecksteine unten behalten bei jedem Algorithmus ihre Position d.h. der Stage 0 wird nicht verschoben.	 <p>Mi-Zustand</p>	Ri Ui R Ui Ri 2U R
	 <p>Pi-Zustand</p>	B U2 B2 Ui B2 Ui B2 U2 B
	 <p>U-Zustand</p>	Ri Fi Ui F U R
	 <p>H-Zustand</p>	2R 2U Ri 2U 2R



Stage 2 Abfrage des Codes

A grid of fractions arranged in a cross shape. The central row contains fractions from 3/0 to 3/11, and the central column contains fractions from 0/3 to 8/5. Some fractions are highlighted with colored boxes (red, green, blue) and connected by lines.

			0/3	0/4	0/5									
			1/3	1/4	1/5									
			2/3	2/4	2/5									
3/0	3/1	3/2	3/3	3/4	3/5	3/6	3/7	3/8	3/9	3/10	3/11			
4/0	4/1	4/2	4/3	4/4	4/5	4/6	4/7	4/8	4/9	4/10	4/11			
5/0	5/1	5/2	5/3	5/4	5/5	5/6	5/7	5/8	5/9	5/10	5/11			
			6/3	6/4	6/5									
			7/3	7/4	7/5									
			8/3	8/4	8/5									

Rot markierte Steine sind schon an ihrem richtigen Platz und richtig Ausgerichtet.

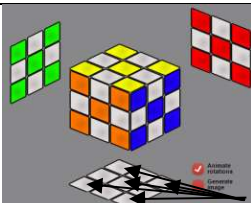
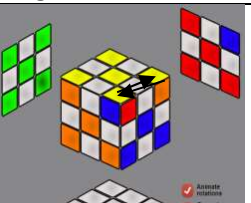
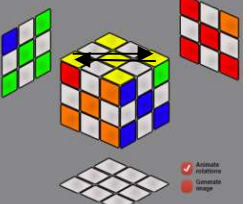
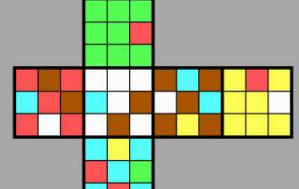
Alle 4 steine je Farbe werden mit dem Mittleren(violett) verglichen. Sind sie alle gleich so kann man sie alle mit einem Algorithmus mit der richtigen Farbseite oben, wie (4/4), Platzieren (hier blau).

Dies wird für alle 4 Seiten (schwarz, MVC_CLK & MVC_NOTCLK) getestet und eins geht zu 100%, da alle möglichen Kombinationen getestet werden.

$$\begin{array}{l} (3/2) (3/6) (5/3) (5/5) \quad | \quad (3/3) (5/5) (6/3) (3/6) \quad | \quad (2/3) (5/3) (6/5) (3/6) \quad | \\ (3/3) (6/3) (5/6) (2/5) \quad | \quad (3/2) (6/3) (6/5) (3/6) \quad | \quad (5/3) (5/5) (2/3) (2/5) \quad | \\ (6/3) (6/5) (2/3) (5/3) \quad = \quad (3/3) (3/5) (5/3) (5/5) \end{array}$$

Danach wird der Würfel im Code wieder um die Anzahl MVC_CLK | MVC_NOTCLK gedreht und 180 Grad in die Urform. Der Lösungsweg muss auch mit gedreht werden.

Stage 3

<p>Sollzustand:</p>  <p>gleiche Farbe</p>	<p>Möglicher Zustand:</p>  <p>Seite-falsch</p>	<p>Massnahme(Algorithmus):</p> <p>2R Ui 2R U 2R 2B U 2B Ui 2B (U(nicht nötig aber Top ist 90 Grad daneben wenn nicht U ausgeführt wird))</p>
<p>Dieser Stage richtet die letzten 4 Ecksteine richtig aus. Es kann 3 Möglichkeiten geben: richtig, diagonal tauschen oder nur eine Seite tauschen.</p>	 <p>Diag. Falsch</p>	<p>2B 2U 2R U 2R 2U 2F U 2F 2U 2L(Di(nicht nötig aber down ist 90 Grad daneben wenn nicht Di ausgeführt wird))</p>
		<p>Bild von Muster nach Stage 3 mit einem richtigen ungelösten Würfel.</p>

Stage 3 Abfrage des Codes

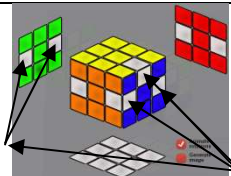
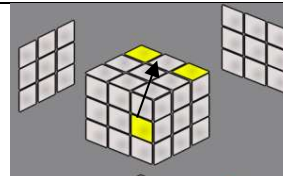
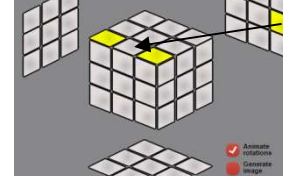
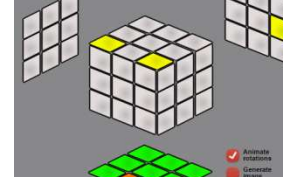
[illegible]

Hier wird auch wieder alle 4 Seiten (schwarz, MVC_CLK & MVC_NOTCLK) auf 2 Stellungen getestet. Eine der 8 stimmt mit dem jeweiligen aussenstein (rot & blau) überein und kann so alle 4 Ecksteine Richtig Ausrichten und Positionieren.

Wenn nötig Up & Down richtig zu ihren Mittelsteine (z.B. 1/4) gedreht im Code sowie im Lösungsweg.

Danach wird der Cube im Code wieder in die Urform gedreht.

Stage 4

<p>Sollzustand:</p>  <p>nicht richtig</p>	<p>Möglicher Zustand:</p> 	<p>Massnahme(Algorithmus):</p> <p>Es gibt kein Algorithmus, man muss die Steine nur „Rein“-drehen wenn sie so wie im Bild vorbereitet werden: Bi E B</p>
<p>Eine Seite wird komplett gemacht(hier Unten) und bei der Anderen alle bis auf einen Mittelstein</p>		<p>Es gibt kein Algorithmus, man muss die Steine nur „Rein“-drehen wenn sie so wie im Bild vorbereitet werden: F Ei Fi</p>
<p>Zu beachten ist, dass bei der 2ten Seite der Untere Stein der Falsche sein muss, sonst würde man ihn wieder raus drehen!</p>		<p>Hier wird Orange raus gedreht wenn Gelb rein gedreht wird. Die anderen 3 Grünen Steine bleiben an ihrem Ort.</p>
	<p>@@@bild</p>	<p>Bild von Muster nach Stage 4 mit einem richtigen ungelösten Würfel.</p>

Stage 4 Abfrage des Codes, Phase 1

[illegible]

Rot markierte Steine sind schon an ihrem richtigen Platz und richtig Ausgerichtet.

Nun muss die orangen gesucht werden und an einer der blauen Positionen gedreht werden, um dann an ihre richtige Stelle mit einem Algorithmus reinzudrehen.

Dies muss für 3 Seiten gemacht werden(violett), dann wird der Cube um 180 Grad gedreht, mit Lösungsweg.

Stage 4 Abfrage des Codes, Phase 2

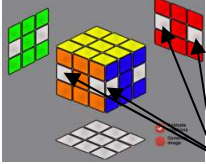
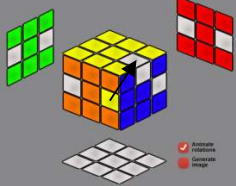
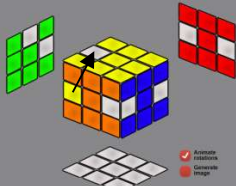
[illegible]

Rot markierte Steine sind schon an ihrem richtigen Platz und richtig Ausgerichtet.

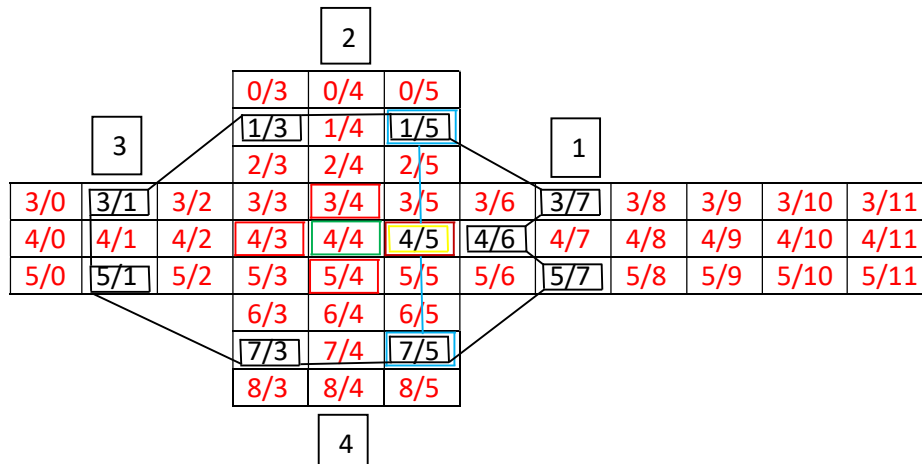
Wenn wie oben in Phase 1 ein Stein gefunden wurde (sucht in blau), der in Orange reinpasst, muss die grüne Fläche unterhalb des Cubes sein (schwarz drehen) damit die 3 Seiten aus Phase 1 nicht verändert werden.

Sind alle 4 Seiten gemacht so wird der Cube wider um 180 Grad gedreht mit Lösungsweg in die Urform (damit der Lösungsweg vom realen Cube stimmt)

Stage 5

Sollzustand:	Möglicher Zustand:	Massnahme(Algorithmus):
 <p>nicht richtig</p>	 <p>R</p>	R E Ri Ei Ri E R
<p>Der Letzte Stein muss nun mit einem Zug reingedreht werden, der nicht den unteren verschiebt. Danach sind nur noch 4 Steine falsch</p>	 <p>L</p>	Li Ei L E L Ei Li
	@@bild	Bild von Muster nach Stage 4 mit einem richtigen ungelösten Würfel.

Stage 5, Abfrage des Codes

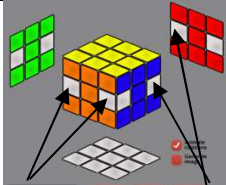
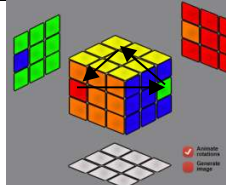
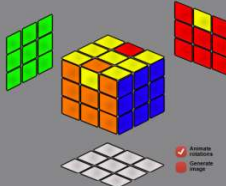


Rot markierte Steine sind schon an ihrem richtigen Platz und richtig Ausgerichtet.

Dies muss für alle 4 Seiten gemacht werden: Suche grüne Farbe bei schwarz und bewege es nach blau. Einer der zwei Algorithmen setzt den Stein zu dem anderen 3(rot), die schon drin sind.

Wenn der Stein schon drin ist (gelb, Change von 1:10), dann überspringe diesen teil.

Stage 6

Sollzustand:	Möglicher Zustand:	Massnahme(Algorithmus):
 <p>richtige Position</p>	 <p>3 Falsch</p>	Ei 2B E 2B
Die letzten Steine an die richtige Position bringen, nicht richtig ausgerichtet	 <p>2 Falsch</p>	Mi Ui Mi Ui Mi 2Ui M Ui M Ui M 2Ui
Bei 2 Falsch können es auch die Diagonalen sein, dann einfach eine Seite um 180 Grad drehen um in den 2. Möglichen Zustand zu kommen.		

Stage 6, Abfrage des Codes

			0/3	0/4	0/5							
			1/3	1/4	1/5							
			2/3	2/4	2/5							
3/0	3/1	3/2	3/3	3/4	3/5	3/6	3/7	3/8	3/9	3/10	3/11	
4/0	4/1	4/2	4/3	4/4	4/5	4/6	4/7	4/8	4/9	4/10	4/11	
5/0	5/1	5/2	5/3	5/4	5/5	5/6	5/7	5/8	5/9	5/10	5/11	
			6/3	6/4	6/5							
			7/3	7/4	7/5							
			8/3	8/4	8/5							

Rot markierte Steine sind schon an ihrem richtigen Platz und richtig Ausgerichtet.

Nun wird entweder blau oder 1-2 mal grün abgefragt auf allen 4 Seiten. Nach diesem Algorithmus ist der Cube im Code fertig und der Lösungsweg kann umcodiert werden in Bewegungen des Roboters. Da der Roboter nur die untere Seite drehen kann und den Cube entweder festhalten, drehen (VCM_CLK & VCM_NOTCLK) oder nach rechts(VCM_Right) kippen kann. Der Lösungsweg wird somit länger...