

Rubik's Cube

MatLab Praktikum

Elisa Junghans

Mirko Dransfeld

1 Aufbau des Programms

Zur erleichterten Bearbeitung und Übersichtlichkeit haben wir das Programm in zwei Dateien geteilt. Die Datei **RubiksCube.m** beinhaltet das Erstellen der grafischen Oberfläche und die Datei **generate_solution.m** den Lösungsalgorithmus.

1.1 RubiksCube.m

Zu Beginn sei gesagt, dass alles in einer großen Funktion mit sehr vielen *nested functions* geschrieben ist. Dies hat den Grund, dass die MatLab-Version 2015b in einer Datei entweder nur eine Funktion oder ein Skript unterstützt, nicht beides. Dazu sei gesagt, die Version 2018b hat mit solchen Dateien keine Probleme.

In Zeile 3–17 werden globale Variablen gesetzt. Diese werden dann später von anderen Funktionen aufgerufen und verändert. Die Zeilen 20–42 initialisieren Konstanten, wie zum Beispiel die Positionen der Flächen des Würfels im drei-dimensionalen Raum (Zeilen 22–27).

Dann folgt die Initialisierung des Würfels. Dieser ist in der Variablen **face_color_rgb** gespeichert.

```
44 face_color_rgb=[];  
45 for i = 1:6  
46     for j = 1:8  
47         face_color_rgb(i,j,:) = possible_colors(i,:);  
48     end  
49 end
```

Damit wird für jede der 6 Seiten (**i**) und jede der 8 veränderbaren Flächen pro Seite (**j**) die zugehörige Farbe aus **possible_colors** hinterlegt.

1.1.1 main()

Hier wird zuerst die Figur erzeugt, in der anschließend gezeichnet wird. Die Optionen werden in der Form `figure(Option1, Wert1, Option2, Wert2, ...)` übergeben.

```
53     figure('Name','Rubiks Cube','NumberTitle','off','MenuBar','none','  
        resize','off');
```

- **Name** 'Rubiks Cube' setzt den Namen der Figur
- **NumberTitle** 'off' verhindert, dass MatLab dem Fenster eine nummerierte Überschrift gibt
- **MenuBar** 'none' bedeutet, dass keine Menüzeile gezeichnet wird
- **resize** 'off' verhindert dass Verändern der Größe des Fensters durch den User

Danach werden mit Hilfe von **get_elon_ui()** und **get_azim_ui()** den Variablen **elon** und **azim** ihre jeweiligen *Slider* zugewiesen.

Es folgen Aufrufe von **ui_setup()** und **generate_patch_and_ui_menu()**, die die restlichen Elemente des UIs anlegen.

1.1.2 ui_setup()

Hier werden zunächst mit Hilfe von **patch()** die Elemente des UIs erzeugt und, wenn diese später noch verändert werden müssen, globalen Variablen zugewiesen.

1.1.3 get_elon_ui()

1.1.4 get_azim_ui()

1.1.5 generate_patch_and_ui_menu()

1.2 generate_solution.m

TODO

2 besondere MatLab Funktionen

2.1 patch()