

# Von der Anfängermethode zur Fridrich-Methode – Überblick

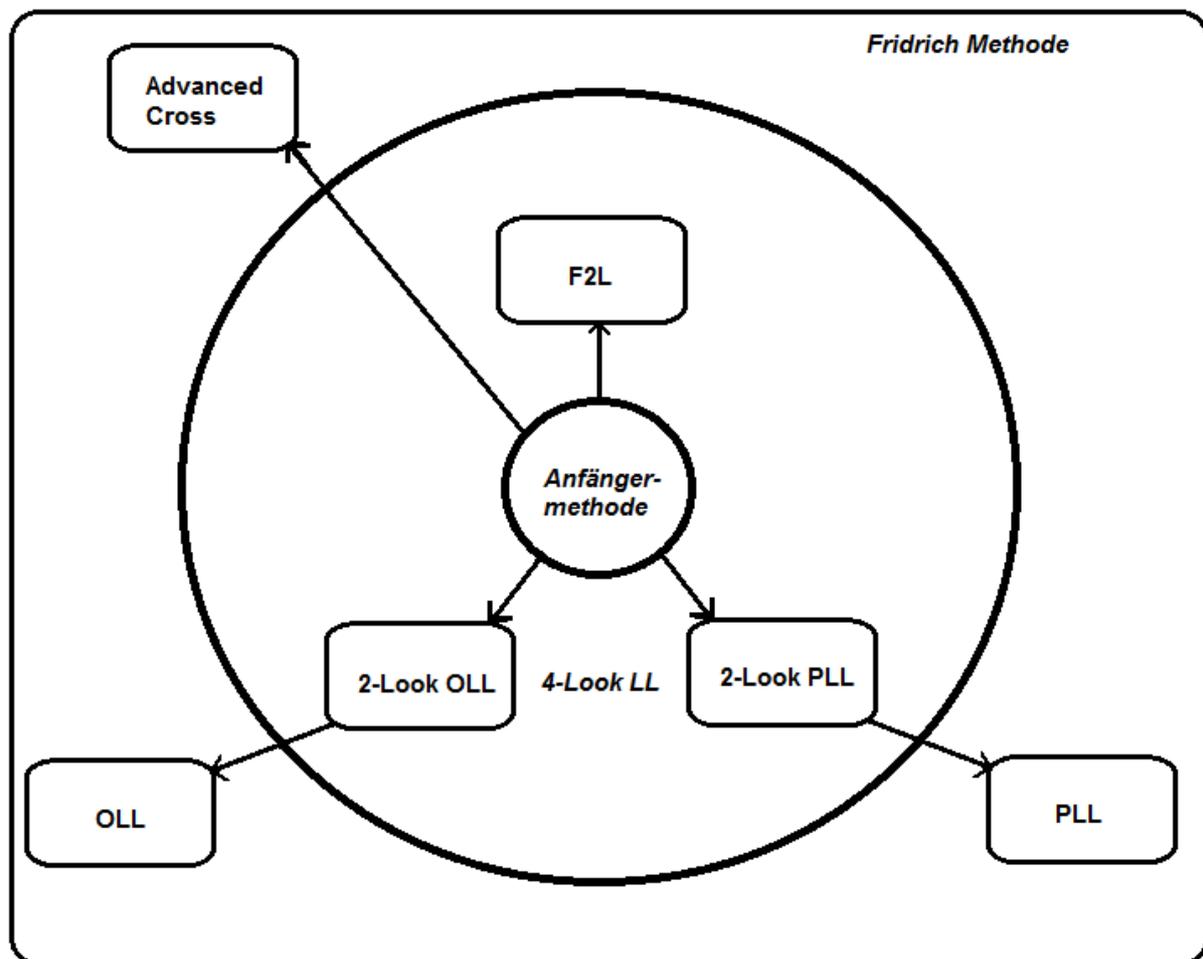
## 1. Organisation der Arbeitsschritte

Anfängermethode	4-Look LL	Fridrich-Methode
1. weißes Kreuz bilden	1. weißes Kreuz bilden	1. weißes Kreuz bilden
2. weiße Ecken einfügen	2. F2L	2. F2L
3. mittlere Kanten einfügen		
4. gelbe Kanten orientieren	3. 2-Look-OLL a) gelbe Kanten orientieren b) gelbe Ecken orientieren	3. OLL
5. gelbe Ecken orientieren		
6. gelbe Ecken positionieren	4. 2-Look PLL a) gelbe Ecken positionieren b) gelbe Kanten positionieren	4. PLL
7. gelbe Kanten positionieren		

Abkürzungen:

- LL = Last Layer (letzte Schicht)
- F2L = First 2 Layers (Lösen der ersten beiden Schichten)
- OLL = Orientation Last Layer (Orientierung der letzten Schicht)
- PLL = Permutation Last Layer (Permutation / Positionierung der letzten Schicht)
- 2-Look = zweimaliges Hinsehen, d.h. der gesamte Arbeitsschritt wird in zwei Teilschritten / durch zwei Anwendungen entsprechender Algorithmen erledigt

## 2. Lernzirkel zum Aneignen der Fridrich-Methode



Die Bausteine innerhalb des Kreises sind für den Projektkurs verpflichtend, die Bausteine außerhalb sind optional.

### **3. Anzahl der Fälle bzw. der erforderlichen Algorithmen**

Anfängermethode:

<i>Arbeitsschritt</i>	<i>Fälle</i>	<i>Algorithmen (&gt; 4 Züge)</i>	<i>Teilschritte "worst case"</i>	<i>intuitiv lösbar</i>
1. weißes Kreuz bilden	6	-	4	ja
2. weiße Ecken einfügen	3	1	8	teilweise
3. mittlere Kanten einfügen	2	2	8	bedingt
4. gelbe Kanten orientieren	4	2	2	nein
5. gelbe Ecken orientieren	4	1	3	nein
6. gelbe Ecken positionieren	3	1	2	nein
7. gelbe Kanten positionieren	3	1	3	nein
Summe:		8	30	

4-Look LL:

<i>Arbeitsschritt</i>	<i>Fälle</i>	<i>Algorithmen (&gt; 4 Züge)</i>	<i>Teilschritte "worst case"</i>	<i>intuitiv lösbar</i>
1. weißes Kreuz bilden	6	-	4	ja
2. F2L	3 / 41	-	4	ja
3. 2-Look-OLL				
a) gelbe Kanten orientieren	4	2	1	nein
b) gelbe Ecken orientieren	8	7	1	nein
4. 2-Look PLL				
a) gelbe Ecken positionieren	3	2	1	nein
b) gelbe Kanten positionieren	5	4	1	nein
Summe:		15	12	

Von diesen 15 Algorithmen sind 5 bereits aus der Anfängermethode bekannt.

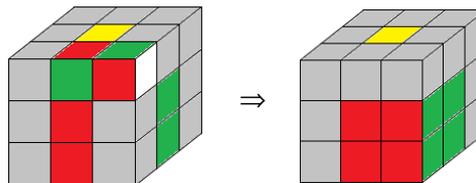
Fridrich-Methode:

<i>Arbeitsschritt</i>	<i>Fälle</i>	<i>Algorithmen (&gt; 4 Züge)</i>	<i>Teilschritte "worst case"</i>	<i>intuitiv lösbar</i>
1. weißes Kreuz bilden	6	-	4	ja
2. F2L	3 / 41	-	4	ja
3. OLL	58	57	1	nein
4. PLL	22	21	1	nein
Summe:		78	10	

Von diesen 78 Algorithmen sind 15 bereits aus 4-Look LL bekannt und 5 sogar aus der Anfängermethode.

## F2L – Leitfaden (Ergänzung zum Workshop)

Die Abkürzung steht für "First 2 Layers" und bezeichnet den zweiten Baustein der Fridrich-Methode. Nach der Fertigstellung des weißen Kreuzes werden im Gegensatz zur Anfängermethode nicht erst die vier Ecken der ersten Schicht und anschließend die vier Kanten der zweiten Schicht eingebaut. Statt dessen werden jeweils eine Ecke und die zugehörige Kante als Pärchen *gleichzeitig* in einen der 4 "Slots" platziert.



### Vorteile von F2L

- Es sind nur halb so viele Arbeitsschritte nötig (maximal vier).
- Die Arbeitsschritte sind teils sehr kurz (im Idealfall 3 Züge).
- Es ist möglich, F2L komplett intuitiv auszuführen, ohne Algorithmen.
- Man kann sich je nach individueller Vorliebe trotzdem Algorithmen aneignen, z.B. um unbequeme Spezialfälle effektiv(er) zu lösen.
- F2L ist sehr flexibel – eher ein ganzheitlicher Prozess als eine stumpfe Mechanik.
- Man kann das entstehende OLL-Bild beeinflussen.

### Standardtechniken

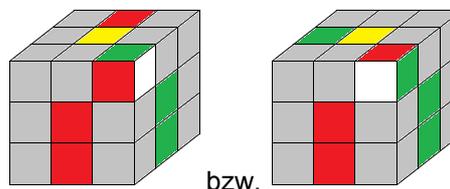
Die Strategie ist stets, Ecke und Kante in eine bestimmte Lage zueinander zu bringen, so dass einer der unten aufgeführten Fälle eintritt. Folgende Prinzipien/Techniken sind dafür nützlich:

- Die obere Schicht dient als Arbeitsfläche.
- Ecken und Kanten können in der mittleren Schicht "versteckt" werden.
- Ecken und Kanten, die falsch zusammenhängen, können getrennt werden.
- Freie Slots können als Zwischenablage genutzt werden.
- Am Anfang hat man viele freie Slots zur Verfügung. Später sollte man es vermeiden, gelöste Slots wieder kaputt zu machen.
- Wenn Ecken falsch in der unteren Schicht bzw. Kanten in der mittleren stecken, können sie in die Arbeitsschicht hochgeholt werden.

### Die drei grundlegenden Fälle

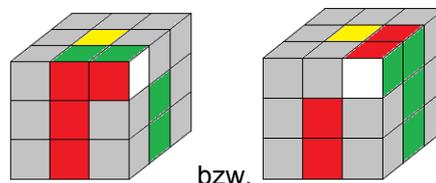
Fall 1: Weiß zeigt nach außen, die oben liegenden Farben von Ecke und Kante stimmen *nicht* überein.

Dies ist der günstigste Fall. Man positioniert die Kante so gegenüber der Ecke, dass ein "L" entsteht, wobei Weiß am Ende des kurzen Balkens liegt. Wenn man nun die Ecke gemäß der Anfängermethode (Schritt 2) einbaut, wird die Kante automatisch mit gelöst.



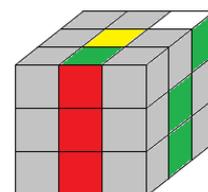
Fall 2: Weiß zeigt nach außen, die oben liegenden Farben von Ecke und Kante stimmen überein.

In diesem Fall bildet man einen zusammenhängenden Block, der analog zum 3. Schritt der Anfängermethode eingebaut wird.



Fall 2: Weiß zeigt nach oben.

- Dies ist der unbequemste Fall. Hier geht man wie folgt vor:
0. Falls Ecke und Kante zusammenhängen, werden sie zunächst getrennt.
  1. Man platziert die Kante so, dass die seitliche Farbe mit der Mitte übereinstimmt (siehe Bild rechts).
  2. Man versteckt die Kante, indem man sie vom Ziel-Slot wegdreht.
  3. Man dreht die obere Schicht, bis Ecke und Kante aneinander liegen.
  4. Man holt die Kante wieder hoch und setzt mit Fall 2 fort.



### Nützliche Links

<http://www.badmephisto.com/f2l.php>

<http://speedcube.de/fridrich.php#step2>

<http://www.opticubes.com/cubing/f2l/>

(Video-Tutorials und Techniken für Fortgeschrittene, englisch)

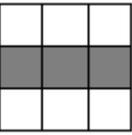
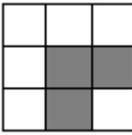
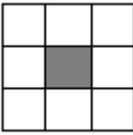
(Beschreibung der intuitiven Vorgehensweise, deutsch)

(Algorithmensammlung, englisch)

## 2-Look OLL

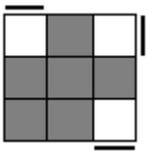
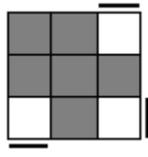
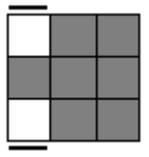
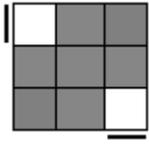
### a) gelbe Kanten orientieren

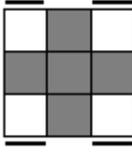
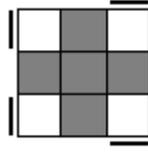
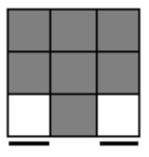
Dieser Schritt wird unverändert aus der Anfängermethode übernommen.  
D.h. folgende drei Fälle sind relevant:

		
$F (R U R' U') F'$	$f (R U R' U') f'$	$F (R U R' U') F'$ $f (R U R' U') f'$

### b) gelbe Ecken orientieren

Nun werden alle Ecken auf einen Schlag richtig orientiert. Dabei sind 7 Fälle möglich:

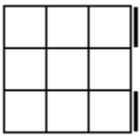
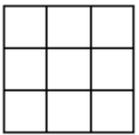
Fall 1: "Fisch" (im Uhrzeigersinn)	Fall 2: "Anti-Fisch" (gegen Uhrzeigersinn)	Fall 3: "Chamäleon"	Fall 4: "Fliege"
			
$R U R' U R U^2 R'$ (siehe Anfängermethode)	$R' U' R U' R' U^2 R$	$r U R' U'$ $r' F R F'$	$F' r U R' U'$ $r' F R$

Fall 5: "Auto"	Fall 6: "Blinker"	Fall 7: "Scheinwerfer"
		
$F R U R' U'$ $R U R' U'$ $R U R' U' F'$	$f R U R' U' f'$ $F R U R' U' F'$	$R R D R' U^2$ $R D' R' U^2 R'$

## 2-Look PLL

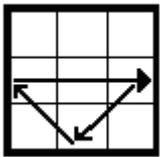
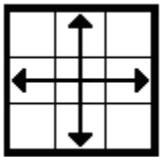
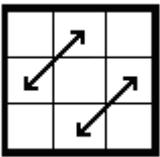
### a) gelbe Ecken positionieren

Dieser Schritt wird - mit einer Ergänzung - aus der Anfängermethode übernommen.  
D.h. folgende zwei Fälle sind relevant:

Scheinwerfer auf einer einzigen Seite	gar keine Scheinwerfer
	
$F' L F' R^2 F L' F' R^2 F^2$  (siehe Anfängermethode)	$R R U R' U'$ $y R U R' U'$ $R U R' U'$ $R U R' F U' F^2$

### b) gelbe Kanten positionieren

Nun werden alle Kanten auf einen Schlag richtig positioniert. Dabei sind 4 Fälle möglich:

Fall 1: (im Uhrzeigersinn)	Fall 2: (gegen Uhrzeigersinn)	Fall 3: (gegenüber)	Fall 4: (benachbart)
			
$L' U L' U' L' U' L' U$ $LU L^2$  (siehe Anfängermethode)	$RU' RU RU RU'$ $R' U' R^2$	$M^2 U M^2 U^2$ $M^2 U M^2$	$M^2 U M^2 U$ $M' U^2 M^2 U^2$ $M' U^2$

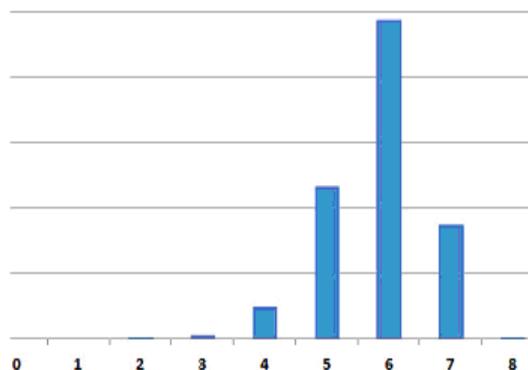
## Advanced Cross

Dieser Baustein umfasst mehrere Möglichkeiten, wie man das Bilden des weißen Kreuzes (1. Schicht) effektiver gestalten kann.

Zuerst sollte man wissen:

Es ist immer möglich, das weiße Kreuz in maximal 8 Zügen zu bilden, und in der überwiegenden Zahl der Fälle sind sogar nur 6 oder weniger Züge nötig (siehe Statistik rechts)

Im Grunde ist es nicht erforderlich, tatsächlich stets die optimale Möglichkeit zu finden. Die Reduzierung der benötigten Zeit ist viel entscheidender, wobei die Anzahl der Schritte natürlich eine Rolle spielt.



Quelle:

<http://www.badmephisto.com/cross.php>

Auf der Seite finden sich auch nützliche Video-Tutorials (englisch).

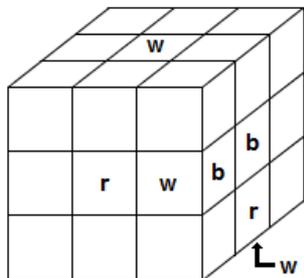
Hier einige Techniken, die dabei helfen können:

### 1. Die obere Schicht als Arbeitsfläche nutzen

Man muss nicht sofort jede Kante an die endgültige Position bringen. Es reicht, wenn die *relative* Position der Kanten *zueinander* stimmt. Beispielsweise muss weiß-blau lediglich gegenüber von weiß-grün platziert werden. Das fertige Kreuz kann dann in einem letzten Schritt endgültig ausgerichtet werden.

### 2. Mehrere Kanten gleichzeitig bearbeiten

Oft lohnt es sich, nicht sofort die naheliegendste Möglichkeit zu wählen, sondern Zwischenzüge einzubauen, um damit beim nächsten Zug Arbeit einzusparen. Ein Beispiel:



Wie man sieht, könnte man durch ein einfaches R die weiß-blaue Kante sofort an die richtige Stelle bringen. Allerdings würde dadurch die weiß-rote Kante in die mittlere Schicht wandern, von wo aus sie erst wieder mühsam befreit werden müsste.

Die elegantere Lösung wäre, zuerst ein vorbereitendes D' auszuführen. Dadurch bringt man die rot-weiße Kante in eine sehr günstige Position. Im Endeffekt kann man so zwei Kanten in nur 4 Zügen (D' R F2) vollständig bearbeiten.

### 3. Genaues Planen des Vorgehens

Auf Wettbewerben hat man in der Regel eine Inspektionszeit von 15 Sekunden, bevor man den Würfel lösen muss. In dieser Zeit kann man die Lösung des Kreuzes planen: Mit welcher Kante fängt man an? Welches ist die zweite? Wo landen die restlichen Kanten? Im Idealfall kann man das Kreuz dann blind lösen.

### 4. Lösen des Kreuzes auf der Unterseite des Würfels

Dies bedeutet eine gewisse Umgewöhnung. Insbesondere sieht man nicht mehr, was man schon gelöst hat. Trotzdem bietet diese Technik einige Vorteile:

- man hat einen besseren Blick auf die *noch nicht gelösten* Teile
- man erspart sich eine Rotation des gesamten Würfels, wenn das Kreuz fertig ist
- man hat einen fließenden Übergang zu F2L ("Lookahead")
- man kann Fingertricks besser einsetzen

## **OLL**

In diesem Schritt werden alle Ecken und Kanten der letzten Schicht gleichzeitig orientiert, so dass eine gelbe Fläche entsteht.

Es gibt 57 mögliche Fälle.

Weblinks:

<http://www.badmephisto.com/oll.php>

(strukturierte Darstellung und Erläuterung der Algorithmen, strategische Tipps zum Auswendiglernen, englisch)

[http://speedcube.de/fridrich\\_oll.php](http://speedcube.de/fridrich_oll.php)

(bietet nur wenige Erläuterungen, dafür vielfach mehrere Algorithmen zur Auswahl, abweichende Notation, deutsch)

## **PLL**

In diesem Schritt werden alle Ecken und Kanten der letzten Schicht gleichzeitig an ihre richtige Position gebracht.

Es gibt 21 mögliche Fälle.

Weblinks:

<http://www.badmephisto.com/pll.php>

(strukturierte Darstellung und Erläuterung der Algorithmen, englisch)

[http://speedcube.de/fridrich\\_pll.php](http://speedcube.de/fridrich_pll.php)

(bietet nur wenige Erläuterungen, dafür eine Vielzahl an Algorithmen zur Auswahl, abweichende Notation, deutsch)