Standard Notation

These are the names of the six layers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Up [**U**] | Down [**D**] | Right [**R**] | Left [**L**] | Front [**F**] | Back [**B**] |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&stage=ll&fc=yyyyyyyyyoooddddddgggddddddddddddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dddddddddddddddoooddddddgggdddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=ddyddyddyoooooooooddgddgddgdddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yddyddydddddddddddgddgddgdddddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=ddddddyyyoddoddoddgggggggggdddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyddddddddoddoddoddddddddddddddddddddddddddddddddd |

A single letter by itself denotes **a 90 degree** **clockwise turn** of that face.

A single letter proceeded by an apostrophe (') denotes **a 90 degree counterclockwise** turn of that face.

A single letter proceeded by a "2" denotes **a 180 degree turn**,in either direction, of that face.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-U.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Uprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-U2.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-D.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Dprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-D2.png** |
| **U** | **U'** | **U2** | **D** | **D'** | **D2** |
| **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-R.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Rprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-R2.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-L.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Lprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-L2.png** |
| **R** | **R'** | **R2** | **L** | **L'** | **L2** |
| **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-F.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Fprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-F2.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-B.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-Bprime.png** | **C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\move-B2.png** |
| **F** | **F'** | **F2** | **B** | **B'** | **B2** |

There are also three "slices", notated by M, E and S. Only M will be used in this guide.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&alg=y2M&arw=U1U7,F1F7&ac=p  **M** | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&alg=y2M'&arw=U7U1,F7F1&ac=p  **M'** | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&alg=y2M2&arw=U7U1,F7F1&ac=p  **M2** |  |  |  |

Beginner's Method

This is what the solve process will look like.

First Two Layers:

|  |  |  |
| --- | --- | --- |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dwdwwwdwddgddgdddddoddodddddddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dddddddddddddodoooddddgdgggddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=ddddddddddddoooooodddggggggddddddddddddddddddddddd |
| Get a cross on D (or U) | Complete the D layer | Complete the equatorial slice |

Last Layer:

|  |  |  |  |
| --- | --- | --- | --- |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyyydyddddoooooodddggggggddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyyyyyyydddoooooodddggggggddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyyyyyyyodooooooogdgggggggddddddddddddddddddddddd | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyyyyyyyooooooooogggggggggddddddddddddddddddddddd |
| Orient the U edges | Orient the U corners | Permute the U corners | Permute the U edges |

**1. The Cross (First Layer Edges)**

This is an intuitive step that requires no algorithms.

**2. First Layer Corners**

This is where you complete the first layer.

Find a white corner in the U layer and turn the U face so that the corner is directly above where it needs to go. You will run into three cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dddddddddddddodoodddddgddggddddddddddddddddddddddd | C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-2-case-1.png  *R U R'* | C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-2-case-2.png  *F' U' F* | C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-2-case-3.png  *R U2 R' U' R U R'* |

If you have no white corner in the U layer, then the unsolved ones are in the wrong place in the D layer. Take one out so it is in the U layer, using any one of those algorithms. Repeat until all four first layer corners are in the correct place.

**3. Second Layer Edges (Equatorial Slice)**

This is where you complete the equatorial layer, and thus the first two layers.

Find a non-yellow edge in the U layer and turn the U face so that the edge is diagonal to where it needs to go. You will run into two cases.

|  |  |  |
| --- | --- | --- |
| Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=ddddddddddddoodoooddddgggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | Case 1:  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-3-case-1.png  *U R U R' U' F' U' F* | Case 2:  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-3-case-2.png  *U' F' U' F U R U R'* |

If you have no non-yellow edge in the U layer, then the unsolved ones are in the wrong place in the second layer. Take one out so it is in the U layer. Repeat until all four second layer edges are in the correct place.

**4. Orientation of the U Edges**

In this step, you will form a "cross" on the U layer. There are three cases. The images on the top show what the cube would look with respect to the F, U and R faces, and the images on the bottom show what the U layer would look like.

|  |  |  |  |
| --- | --- | --- | --- |
| Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=dydyyydydddddddddddddddddddddddddddddddddddddddddd  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyyydyddddoooooodddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | Case 1:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=dydyydddddyddddddddydddddddddddddddddddddddddddddd  *F U R U' R' F'*  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyydddddydoooooodydggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | Case 2:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=dddyyydddddddddddddyddddddddddddddddddddddddddyddd  *F R U R' U' F'*  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dddyyyddddddoooooodydggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | Case 3:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=ddddydddddyddddddddydddddddddddddddddyddddddddyddd  Do either the case 1 or 2 algorithm, then it will reduce to case 1 or 2.  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=ddddydddddydoooooodydggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p |

**5. Orientation of the U Corners**

The next step involves making the whole U face yellow. There are seven cases, but you only need two algorithms.

|  |  |  |
| --- | --- | --- |
|  | Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyyyyyyydddoooooodddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb |  |

Case 1: One Corner Oriented

|  |  |
| --- | --- |
| "Sune"  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&alg=RU2R'U'RU'R'  *R U R' U R U2 R'*  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyyyyydddyooooooddyggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | "Antisune"  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&alg=RUR'URU2R'U'&stage=oll  *R' U' R U' R' U2 R*  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yydyyydydyddooooooyddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p |

Case 2: Two Corners Oriented

Position the cube so that there is one misoriented corner at FLU. 

|  |  |  |
| --- | --- | --- |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&case=R'F2RU2RU2R'F2RU2R'  Sune + Antisune  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yyyyyydyddddooooooydyggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&case=rUR'U'r'FRF'  Sune + *U2* + Antisune  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dyyyyydyydddooooooyddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&alg=rUR'U'r'FRF'  Sune + *U* + Antisune  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=yydyyydyyddyooooooyddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p |

Case 3: Zero corners oriented

Position the cube so that there is one misoriented corner at LUF.

|  |  |
| --- | --- |
| http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&case=RU2R2U'R2U'R2U2Ry'  Sune + U' + Sune  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyyydyddddooooooddyggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p | http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyddddddddddddddddddddddddddddddddddddddddddddd&alg=RU2R'U'RUR'U'RU'R'  Sune + Sune  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&fc=dydyyydydydyoooooodddggggggwwwwwwwwwrrrrrrrrrbbbbbbbbb&ac=p |

**6. Permutation of the U Corners**

This step involves completely solving the last layer corners. There are two cases. The algorithms are slightly longer than the previous ones, but they are not difficult to learn because of how "finger-trickable" they are. The brackets show groupings of the moves, to make them easier to learn.

|  |  |  |
| --- | --- | --- |
| Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&fc=yyyyyyyyyodooooooogdgggggggwwwwwwwwwrdrrrrrrrbdbbbdbbb | "Headlights"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-6-case-1.png  *(R U R' U')(R' F R2)*  *(U' R' U')(R U R' F')* | "No Headlights"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-6-case-2.png  Headlights + *U2* + Headlights |

**7. Permutation of the U Edges**

The very last step is to solve the remaining edges, completing the cube.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Goal:  http://cube.crider.co.uk/visualcube.php?fmt=gif&size=100&view=plan&alg=y2 | "U(a) perm"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-7-case-1.png  *F2 U' M' U2*  *M U' F2* | "U(b) perm"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-7-case-2.png  *F2 U M' U2*  *M U F2* | "H perm"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-7-case-3.png  *M2 U M2 U2*  *M2 U M2* | "Z perm"  C:\Users\Sarah\Documents\GitHub\sa967st.github.io\images\beginners-method\step-7-case-4.png  U(a) + *U'* + U(a) |