Solver

By:

Saikat R. Gomes
Scholar's Project
Spring 2008

Contents:

Introduction to the Rubik’s Cube:

The Rubik’s Cube swept across the world like no other puzzle. Millions of people were (and still are) simultaneously bewildered, frustrated and amazed by the problems it posed. In addition, its apparently simple operations concealed a wealth of fascinating mathematics.

The Rubik’s Cube is a mechanical puzzle invented by a Hungarian sculptor and professor of architecture named Erno Rubik in the year 1974. Originally Erno Rubik called this mechanical masterpiece the “Magic Cube”. Ideal Toys renamed it to “Rubik’s Cube” in honor of its inventor in the year 1980. It is said to be the world’s best toy and it is estimated that over 300,000,000 Rubik’s Cube and its imitations have been sold worldwide (Marshall).

There are 43 quintillion ways of configuring the Cube (43,252,003,274,489,856,000 to be precise), but only one solution. At one move per second, it would take 1,400 million million years to go through all the Rubik's Cube's configurations. The whole universe is only 14 thousand million years old! (Rubik 178)

The Rubik’s Cube is a 3-dimensional symmetric cube with 6 sides. The colors of the cube are so obvious that they are easily overlooked, but it is the basis of its great aesthetic appeal. The color patters of the official Rubik’s Cube is standardized so that opposite faces differ by the color yellow: white-yellow, red-orange and blue-green (Rubik viii).

History of the Rubik's Cube

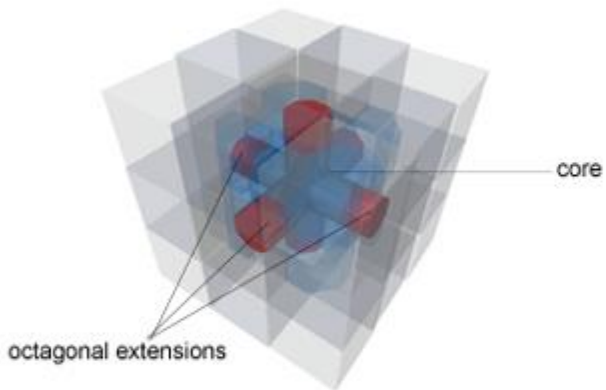
As a teacher, Rubik developed a habit of creating models to demonstrate his designs to students. Rubik's cube was conceived as a teaching tool, not a toy. He wanted to demonstrate structural design by creating a three dimensional object that could rotate on all three axes. Furthermore, he wanted the object to stay intact as it rotated. Rubik originally experimented with a 2x2x2 cube to try and discover a workable mechanism. To keep the pieces together, Rubik tried several techniques: elastic rubbers, magnets, and a system of grooves. But all these methods failed. After much experimentation he came up with the idea of using the shapes of the individual pieces to hold the object together. This mechanism worked and Rubik Erno created the rotatable cube. (Cao)

He marked the faces of the cube with adhesive paper and tested the cube by rotating several sides. With many random movements he was captivated by the way all the 6 colors mixed. After a month of experimenting, Rubik solved the cube. He then demonstrated the cube to his students, the students we fascinated by the magic cube. “The legacy of Rubik's cube spread from this small group of students to the 1979 Nuremberg Toy Fair and eventually to the entire world” (Cao).

Rubik applied for a patent in the year 1975 and this was approved in the year 1977. There were two other people who applied for patents on similar cubes. Terutoshi Ishige of Japan applied for his patent in the year 1976 and an American by the name Larry Nichols had already patented a cube held together with magnets before Rubik's patent. (Bellis)

Mechanics of the Cube

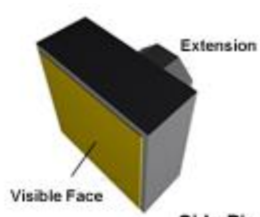
The Rubik's Cube consists of two distinct components: the core and the outer cubes. The core is like a central cube with six attached octagons, one on each face. Each octagon is attached to allow free rotation in either direction.



THE CORE

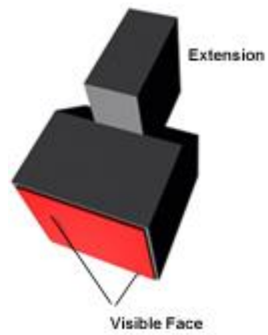
The outer cubes are attached to this central piece. These outer cubes are of three types: the sides, the edges and the corners.

There are six side pieces; each has only one visible face when the entire cube is constructed. Each side piece attaches to one of the octagons of the core; side pieces are only pieces that never move in relation to the core.



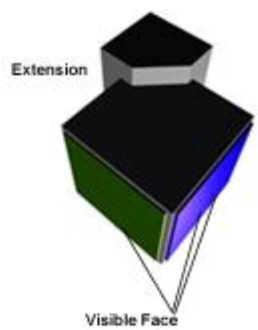
Side Piece

There are twelve edge pieces; each has only two visible faces when the entire cube is constructed. There is an extension off the non-visible faces of each edge piece to connect it to the core. This extension's shape is similar to a smaller cube.



Edge Piece

There are eight corner pieces; each has only three visible faces we the entire cube is constructed. There is also an extension on each corner piece. This extension looks similar to a small cube with a corner cut off. Again, the connecting extension is positioned on the non-visible back corner.



Corner Piece

	Side Piece	Edge Piece	Corner Piece
No. of Pieces	**6**	**12**	**8**
Visible Pieces	**1**	**2**	**3**
Movability	**no**	**yes**	**yes**
Location			
Extension Shape	**octagonal**	**cube**	**cube with chamfered corners**

One face of a Rubik's cube is formed when one side piece, four corner pieces, and four edge pieces fit together. Each face is labeled with a unique solid color. The side piece is the center of the face, and the edge and corner pieces become the edges and corners of the face. The extension off the side piece fits directly on to the core's octagonal face. The other extensions fit on to the edges of the octagon (Cao).

The Rubik's Cube is made up of these twenty-six pieces. When one of the core's octagons turns, it rotates the nine connected outer cubes (one side) with it. As rotation occurs, these cubes are locked with each other and the core thereby marinating the shape and mechanism of the cube.

How to Solve a Rubik's Cube

The goal in solving a Rubik's Cube is to solve it in a systematic yet fast way. There are many different approaches or algorithms available for solving a Rubik's Cube. An algorithm is a procedure or formula for solving a problem. In this paper I will discuss just four important and well known algorithmic approaches used in solving a Rubik's Cube.

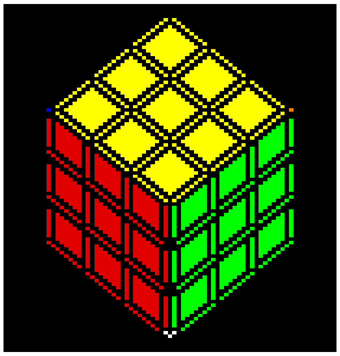
These algorithms/solutions are:

- Optimum Solution – God's Algorithm
- Layer-By-Layer Method (LL method)
- Jessica Fridrich Method (F2L-OLL-PLL)
- F2L with Keyhole-OLL-PLL

But before we take a look at the solutions it is very important we have a good understanding of the moves notation.

Moves Notation

Each face is represented by a letter (F (front), B (back), L (left), R (right), U (up), and D (down)) and moves can be described by the letter of the face that is being turned. For the purposes of the diagrams below from CubeWhiz.com, F is red, B is orange, L is blue, R is green, U is yellow, and D is white:



An upper-case letter means to turn that face in the clockwise direction. If that letter is followed by a prime ('), then that face should instead be turned counter-clockwise. Similarly, if that letter is followed by the number two (2), then that face should be turned twice. Slice turns (M, E, S) are sometimes useful for certain algorithms. In a slice move, the "outside" layers remain stable, and a middle layer will slice through them

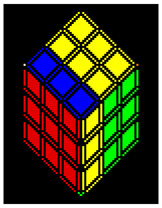
In an antislice turn (Fa, Ra, Ua, Fa', Ra', and Ua'), the listed face and the parallel face are both moved in the same direction (ie - Fa is equivalent to performing F and B simultaneously). Note that Fa and Ba would be exactly the same, so there is only need for one of them to exist. The same applies for La and Da.

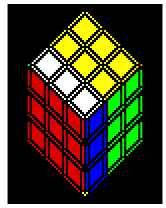
Cube rotations (x, y, and z) do not involve the turning of any layers of the cube. Instead, the entire cube is rotated (imagine an x, y, and z axis going through the cube). These are often used to move the cube into a position that makes an algorithm easier to perform. For this project I have used V instead of y and H instead of x.

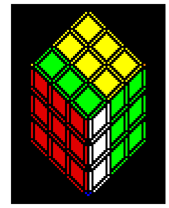
There are other move notations such as lower-case letter (f, b, r, l2, m' etc), but they are not relevant for this paper. A lower-case letter works the same way, except instead of just turning the outside face; the middle layer adjacent to it should be moved as well (in the same direction). For example, "d" is equivalent to performing "D" and "E" simultaneously. The slice turns and anti slice turns can be thought as combinations of other simple moves. For example:

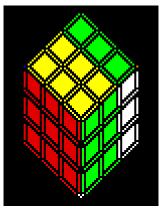
E = U D' V'   M' = L R' H   z = V H' V'

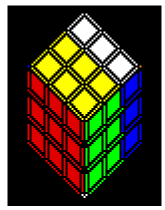
Standard Face-Turns

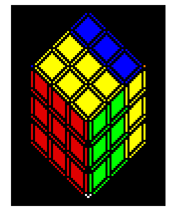
F


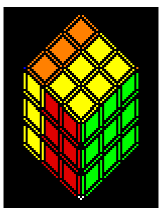
F2


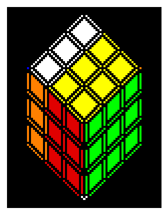
F'


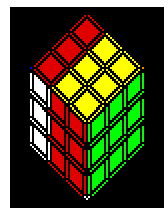
B


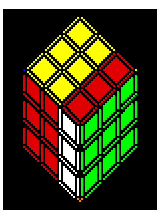
B2


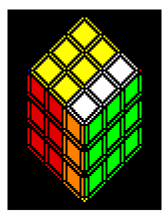
B'


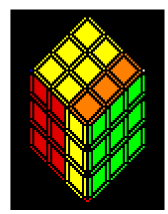
L


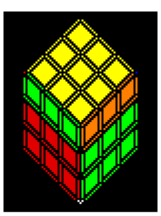
L2


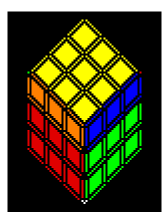
L'


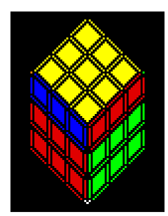
R


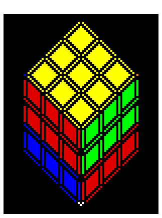
R2


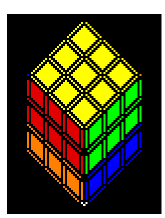
R'


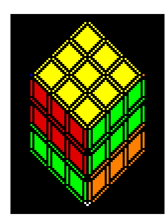
U


U2


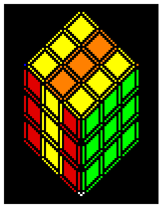
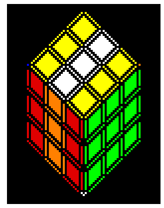
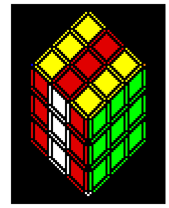
U'


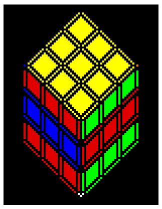
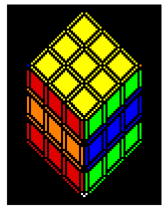
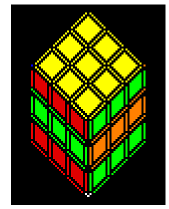
D


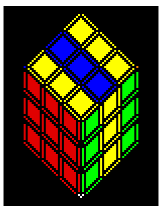
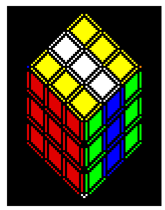
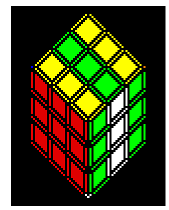
D2


D'


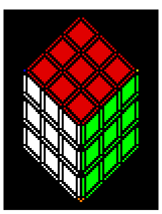
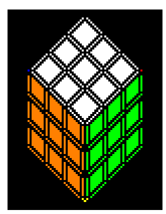
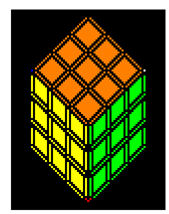
Slice Moves

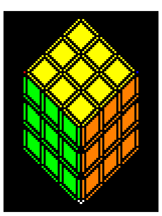
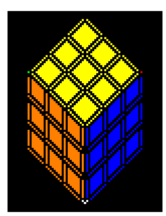
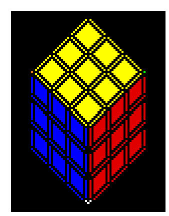
M 
M2 
M' 

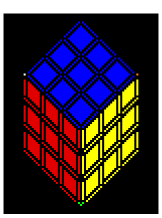
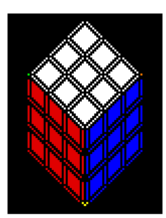
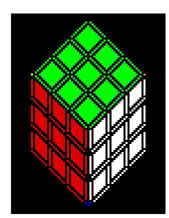
E 
E2 
E' 

S 
S2 
S' 

Cube Rotation

x 
x2 
x' 

y 
y2 
y' 

z 
z2 
z' 

If we take a closer look at the standard face-turn moves, we will notice that the X2 and X' are themselves a multiple of X (where X represents a face move such as F, L, R etc). For example:

X2 = X X (X twice) & X' = X X X (X Thrice)

In fact if we consider all the face-turn moves of a certain face to be elements of a set (example {F,F2,F',I}: where I denotes identity, i.e. no movement), and move multiplying (combination of 2 moves) a binary operation: they from an Abelian Group because:

- The set is closed under - move multiplication
- Move multiplication is associative [a*(b*c)=(a*b)*c]
- I is the Identity element
- Inverse is present (X and X' are inverses of each other and X2 is the inverse of itself)
- Move multiplication is commutative. [a*b=b*c]

*	I	X	X2	X'
I	I	X	X2	X'
X	X	X2	X'	I
X2	X2	X'	I	X
X'	X'	I	X	X2

This property is used for move simplification.

Some times instead of using sides (F, L etc) the color of the side will be used. Such as W for White, B' for Blue Inverse etc. this makes the moves independent of the orientation of the cube.

Optimum Solution – God's Algorithm

This is the name given to the algorithm that solves the cube in the shortest possible moves. In the paper "Twenty-Six Moves Suffice for Rubik's Cube", Daniel Kunkle and Gene Cooperman used mathematical and computer approach and proved that the maximum number of steps needed to solve any Rubik's Cube in the Optimum Solution is 26. On an average almost all Rubik's Cube states can be solved within 20 moves using this algorithm. This algorithm is also known as the two-phase algorithms and uses a database of "pruning tables" to find the optimum solution.

Layer-By-Layer Method (LL method)

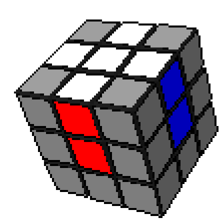
This is the most popular and logical method for solving the Rubik's Cube. It was developed by David Singmaster and published in the book "Notes on Rubik's Magic Cube" in 1981. As the name suggests, this method solves the cube one layer at a time, starting with the top, then middle and finally the last layer. Many derivatives of this method have been developed. This method is very popular among amateurs who are beginners to the art of cubing. The LL method takes more than 100 moves to solve a cube. The basic overview of the algorithm is given bellow:

Step 1: Solving the First Layer:

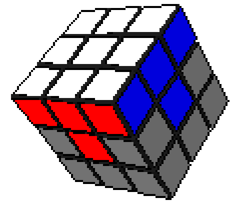
Part A:

Solve the Cross - (with respect to the adjacent sides)

Algorithms Used - intuitive



Part B:

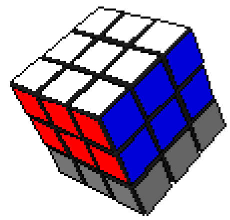


Solve the 4 Corners

Algorithms Used –

- **F' U L' U'**

<u>Step 2: Solve Middle Layer</u>



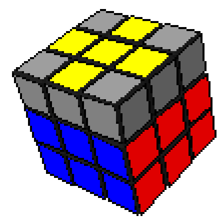
Solve the 4 Middle Layer Edge pieces

Algorithms Used –

- **U R U' R' U' F' U F**
- **U' L U L U F U' F'**

<u>Step 3: Solve the Last Layer</u>

Part A:

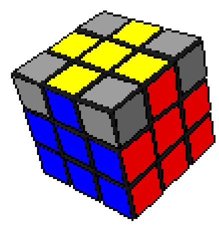


Orient the Last Layer Edges (Cross)

Algorithms Used –

- **F R U R' U' F'**

Part B:

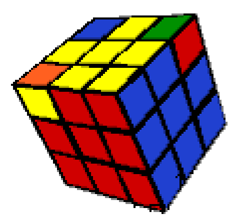


Permute the Last Layer Edges (Cross)

Algorithms Used –

- **R U R' U R U2 R'**

Part C:

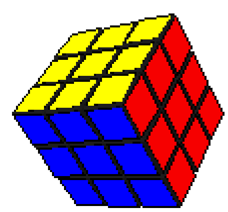


Permute the Last Layer Corners

Algorithms Used –

- **U R U' L' U R' U' L**

Part D:



Orient the Last Layer Corners

Algorithms Used –

- **F' U L' U'**

Jessica Fridrich Method (F2L-OLL-PLL)

This method is a special and efficient modification of the Layer-by-Layer method, developed by Jessica Fridrich, a renowned speed cuber. This method requires a large number of algorithms especially for orienting and permuting the last layer. In this method the first-layer corners and second layers are done simultaneously, with each corner paired up with a second-layer edge piece. The Jessica Fridrich method solves the cube in 55 moves on an average and is used by many speed cubers.

In this method we solve the First Two Layers (F2L) simultaneously, then we Orient the Last Layer (OLL) and finally we Permute the Last Layer (PLL).

First 2 Layers (F2L)

While Solving the F2L, the following situations can occur:

1. The 2 pieces are already in place.
2. The top corner is in place but the middle edge is not.
3. The middle edge is in place but the top corner is not.
4. The top corner and middle edge are on the bottom face.

The 4th case can be further broken down into 2 distinct cases:

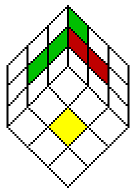
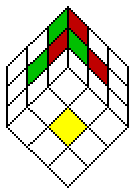
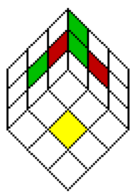
4a) The pieces are next to each other

4b) The pieces are separated.

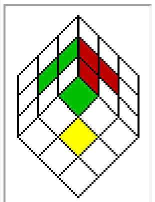
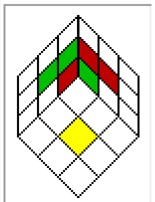
In the following diagrams, colors are used to designate the faces of the cube. Yellow is the bottom face, Green is the front face, Red is the right face, and White is the top face. The goal of this stage is to correctly place and orient a top corner with its middle edge piece at the same time. The diagrams below from

Jessica Fridrich's website show the front, right, and bottom faces along with the 2 pieces that are to be placed. All other pieces are not relevant and so do not appear. The center pieces are marked as a frame of reference.

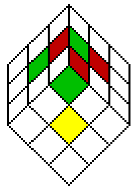
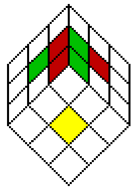
Pieces are in place but not oriented

Pattern	*Algorithm(s)*
	1) R2 D2 R D R' D R D2 R
	2) R2 D2 F' R2 F D2 R D' R
	3) R' D R' D' B' D B R2

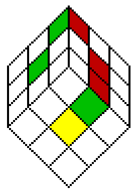
Insert the corner and preserve the edge

Pattern	*Algorithm(s)*
	4) [D] R' D R D2 R' D R
	5a) [D2] R' D B' D' B D' R 5b) R2 D' R2 D' R2 D2 R2

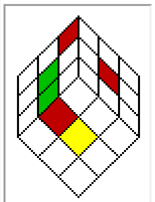
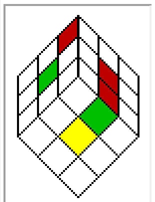
Insert the corner and flip the edge

Pattern	*Algorithm(s)*
	6) [D] R' D' R D' F D F'
	7) R' D R F D2 F'

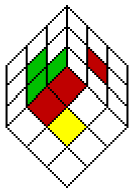
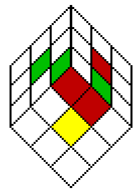
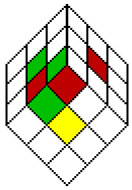
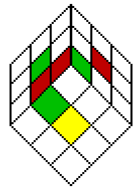
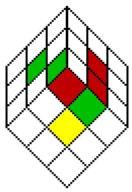
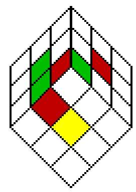
Insert the edge and preserve the corner

Pattern	*Algorithm(s)*
	8) [D] F D' F' D' R' D R

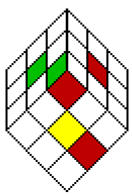
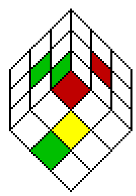
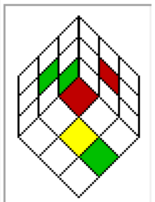
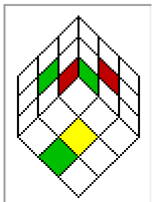
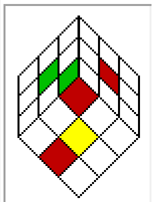
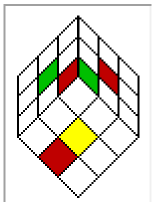
Insert the edge and twist the corner

Pattern	*Algorithm(s)*
	9) F D' F2 R F R'
	10) R' D' R D R' D' R

Connected in bottom layer

Pattern	*Algorithm(s)*	*Pattern*	*Algorithm(s)*
	11) F' R F R'		14a) F L B D' B' L' F' 14b) R' D R D2 F D F'
	12) [D'] F D2 F' D R' D' R		15) [D2] R2 D2 R D R' D R2
	13) (D) R' D R D' R' D' R		16) F D2 F' D' F D F'

Separated in bottom layer

Pattern	*Algorithm(s)*	*Pattern*	*Algorithm(s)*
	17) [D'] F D2 F' D2 F D' F'		20) [D'] F D' F' D R' D' R
	18) R' D' R		21) [D2] R' D' R2 F' R' F
	19) [D'] F D F' D2 F D' F'		22) [D] F D2 F2 R F R'

* It may be necessary to move a piece to the bottom layer. This can happen for the following reasons:

1) The corner to be placed is in one of the other 3 top corner positions

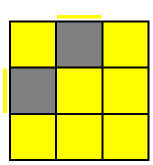
2) The edge to be placed is in one of the other 3 middle edge positions

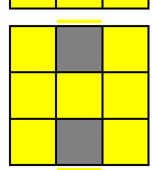
* In that case we have to twist the entire cube so that the piece is in the *FR* position and perform following 3 twist sequence: *R' D R*

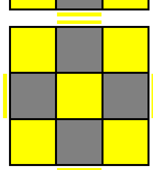
* To save time, if a piece you need is not currently available, move on to another corner/edge position.

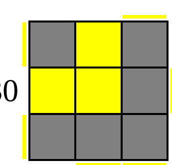
Orientation of the Last Layer (OLL)

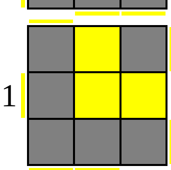
During this step, we correct the Orientation (corner twist and edge flip) of every piece in the Last Layer, which results in the U face being solved, but not the U Layer. This is the toughest stage to learn, and many feel that mastering it is the turning point from being a beginner to becoming an expert. There are 57 cases to learn in this stage. In each diagram below, yellow is the color of the upper face. A yellow "bar" indicates that the last layer color is facing that direction in that location. Grey denotes that a particular piece is not oriented correctly.

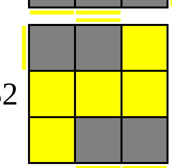
01  (M' U M) U2 (M' U M)

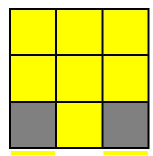
02  (R U R' U') (M' U R U') r'

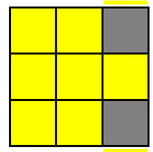
03  r' (R U) (R U R' U' r2) (R2' U) (R U') r'

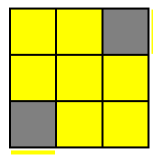
30  F (R U R' U') (R U R' U') F'

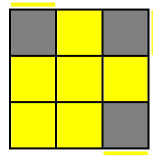
31  F' (L' U' L U) (L' U' L U) F

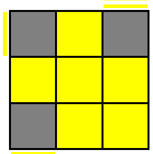
32  (R B') (R' U' R U) y (R U') F'

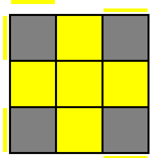
04  (R2' D) (R' U2) (R D') (R' U2 R')

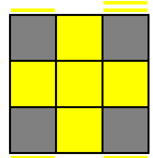
05  (l' U') (L U) (R U') (r' F)

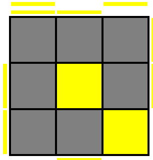
06  (R' F) (R B') (R' F') (R B)

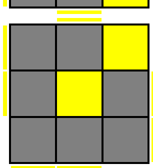
07  (R U R' U) (R U2' R')

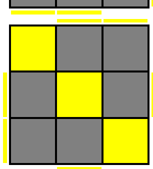
08  (L' U' L U') (L' U2 L)

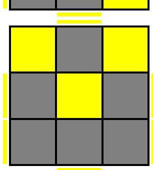
09  (R U2') (R2' U') (R2 U') (R2' U2' R)

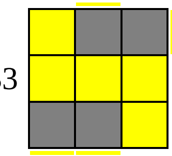
10  F (R U R' U') (R U R' U') (R U R' U') F'

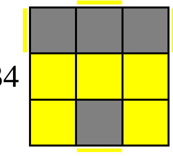
11  f (R U R' U') f' U' F (R U R' U') F'

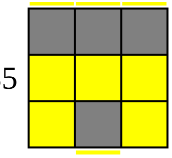
12  f (R U R' U') f' U F (R U R' U') F'

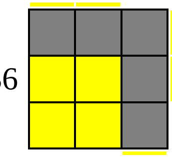
13  (R U R' U) (R' F R F') U2 (R' F R F')

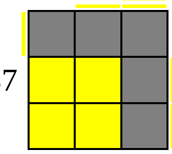
14  (R' U2) F (R U R' U') y' (R2' U2) (R B)

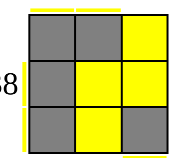
33  (L' B) (L U L' U') y' (L' U) F

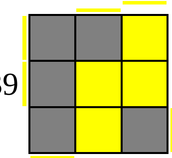
34  (R U R2' U') (R' F) (R U) (R U') F'

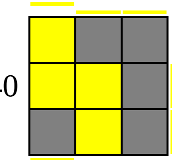
35  B' (R' U' R) y (R U') (R' U2 R)

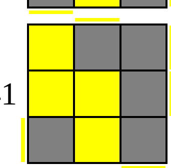
36  B' (R2' F) (R F' R B)

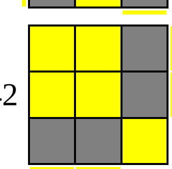
37  (l U2) (L' U' L U' l')

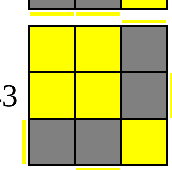
38  (l U L' U) (L U2' l')

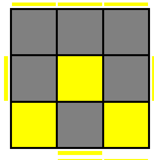
39  F (R U R' U') F' y F (R U R' U') F'

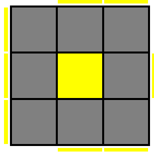
40  (r' U' R U') (R' U2 r)

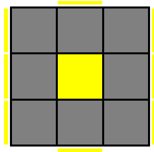
41  F' (L' U' L U) F y' F' (L' U' L U) F

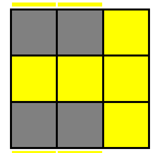
42  F (R U') (R' U' R U) (R' F')

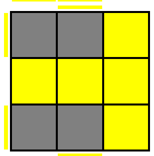
43  (L U2') (L2' B) (L B' L U2' L')

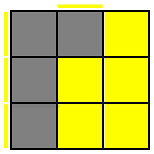
15  F (R U R' U) y' (R' U2) (R' F R F')

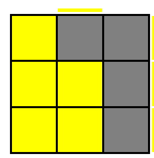
16  F (R U R' U') S (R U R' U') f'

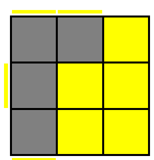
17  (R U) B' (l U [l' R'] U') (R' F R F')

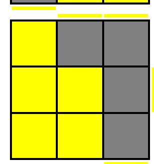
18  (R U R' U') (R' F R F')

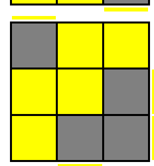
19  F (R U R' U') F'

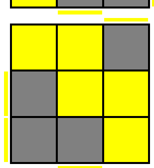
20  f (R U R' U') f'

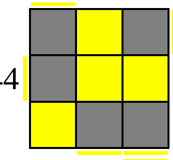
21  f' (L' U' L U) f

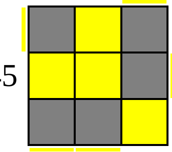
22  (R d) (L' d') (R' U) (l U l')

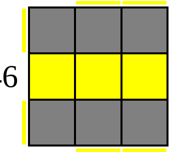
23  (L' d') (R d) (L U') (r' U' r)

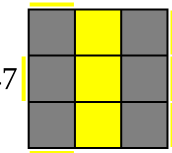
24  (R U R' U) (R U' R' U') (R' F R F')

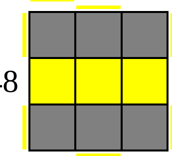
25  (L' U' L U') (L' U L U) (L F' L' F)

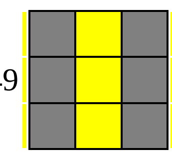
44  (R U R2') y' (R' U' R) y (R2 U') (R2' U2 R)

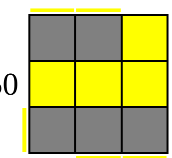
45  (R' U' R) y'x' (R U') (R' F) (R U R')

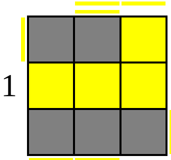
46  f (R U R' U') (R U R' U') f'

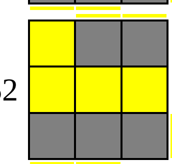
47  (R U R' U R d') (R U' R' F')

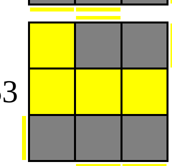
48  F (R U R' U' R) y' (R' F) (R B') (R' F')

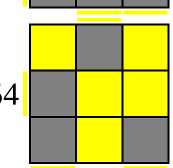
49  (R' U2) (R2' U) (R' U) (R U2') x' (U' R' U)

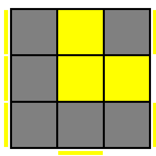
50  x' (R U' R' F') (R U R') xy (R' U R)

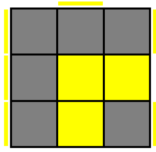
51  (r U r') (R U R' U') (r U' r')

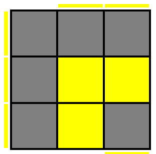
52  x' (L' U L F) (L' U' L) xy' (L U' L')

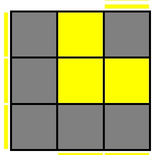
53  (l' U' l) (L' U' L U) (l' U l)

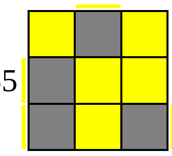
54  (R U') (R' U2) (R U) y (R U') (R' U' F')

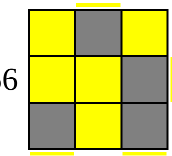
26  (r U) (R' U) (R U') (R' U) (R U2' r')

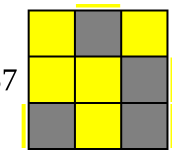
27  (r' U') (R U') (R' U) (R U') (R' U2 r)

28  (R B' R B R2') U2 (F R' F' R)

29  (R' F R' F' R2) U2 y (R' F R F')

55  (R2' U R' B') (R U') (R2' U) (l U l')

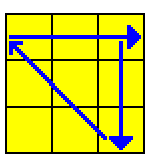
56  (L' U) (L U2') (L' U') y' (L' U) (L U F)

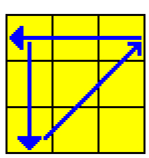
57  (L2 U' L B) (L' U) (L2 U') (r' U' r)

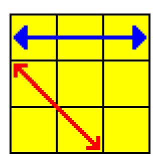
The Pictures are taken from
http://www.cubewhiz.com

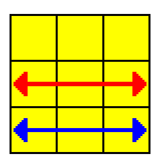
Permutation of the Last Layer (PLL)

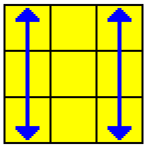
The PLL is the final stage in the System. During this stage, we correct the positions of each cube in the Last Layer, without changing their Orientation. This results in a completed Last Layer, and therefore a solved cube, provided we have kept the F2L in tact. These are the 21 permutation cases for the last layer below. In each diagram, the edges that are being swapped or moved are denoted by the red arrows, while the corners that are being swapped are moved are shown with blue arrows. This stage has the longest sequences to learn

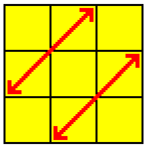
01  (l' U R') D2 (R U' R') D2 R2

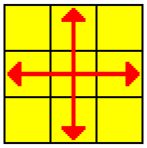
02  x R2 D2 (R U R') D2 (R U' R)

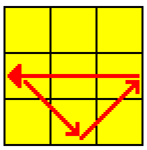
12  (L U2') (L' U2') (L F' L' U' L U) (L F L2' U)

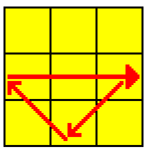
13  (R' U2 R' d') x (l' U R' U') (l R U') (R' U) (l U' F)

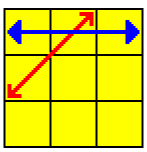
03  x' (R U') (R' D) (R U R') u2' (R' U) (R D) (R' U' R)

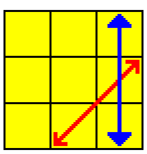
04  (M2' U) (M2' U) (M' U2) (M2' U2) (M' U2)

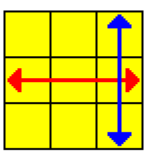
05  (M2' U) (M2' U2) (M2' U) M2'

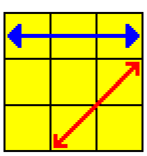
06  (R U' R U) (R U) (R U') (R' U' R2)

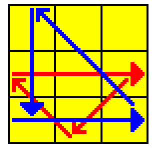
07  (R2 U) (R U R' U') (R' U') (R' U R')

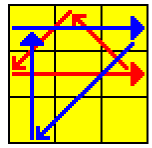
08  (R' U L') U2 (R U' R') U2 (Ra U')

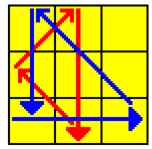
09  (R U R' F') (R U R' U') (R' F) (R2 U') (R' U')

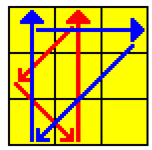
10  (R U R' U') (R' F) (R2 U') (R' U' R U) (R' F')

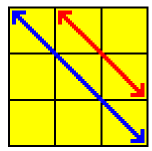
11  (R' U2) (R U2) (R' F R U R' U') (R' F' R2 U')

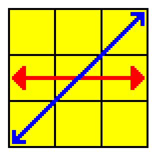
14  (R2' u' R U') (R U R' u R2) y (R U' R')

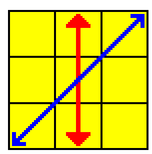
15  (R2' u) (R' U R' U' R u') R2' y' (R' U R)

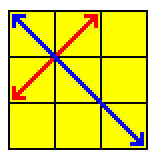
16  (R' U' R) y (R2' u R' U) (R U' R u' R2')

17  (R U R') y' (R2' u' R U') (R' U R' u R2)

18  (R' U R' d') x (l' U R' U') (l R U') (R' U R U)

19  (R U') (R' U) (l U) (F U') (R' F') (R U' R U) (l' U R')

20  [(R' U L') U2 (R U' L)] [(R' U L') U2 (R U' L)] U'

21  (F R U') (R' U' R U) (R' F') (R U R' U') (R' F R F')

The Pictures are taken from
http://www.cubewhiz.com

F2L with Keyhole-OLL-PLL

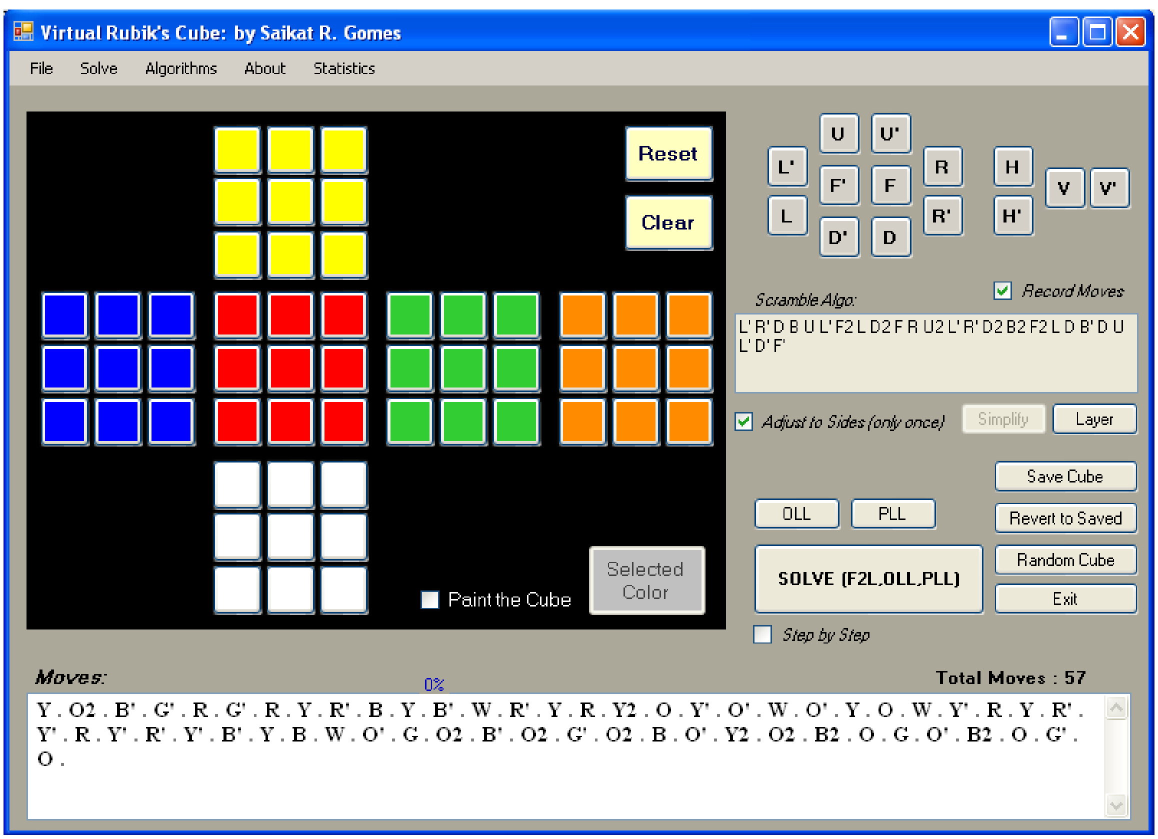
This is a special modification of the Jessica Fridrich Method. It is not as efficient as the Jessica Fridrich Method but is less complex. The average number of moves needed using this method is around 73. The OLL and the PLL in this method is same as the Jessica Fridrich method. They differ in the F2L: Instead of using large number of algorithms to solve the two layers simultaneously, it solves the 2 layers using a keyhole in the top layer. A keyhole is an unsolved corner position in the top layer. By using this keyhole we will be filling in the middle edge pieces and complete both the top and middle layers together. The overview of the F2L with Keyhole-OLL-PLL method is given below:

- Solve the Top Cross
- Solve 2 of the Top Corners
- Solve 2 of the Middle Edge Piece using the Keyhole
- Solve the 3rd Top corner
- Solve a maximum of 3 Middle Edge Piece using the Keyhole
- Complete the Top Layer (Last Top Corner Piece)
- Complete the Middle Layer (Remaining Edges Pieces)
- OLL
- PLL

I have used this method in my software to solve all the cubes.

The Rubik's Cube Solver – Software

The Rubik's Cube Solver is software that I wrote in Visual Basic .Net 2008. This software has many features but most important function of this software is to solve any random Rubik's Cube. The software solves a random cube using the F2L with Keyhole-OLL-PLL method and then simplifies the move sequence and then converts them in term of the color sides. This software also creates a virtual cube that works just like a normal cube and all the functions/moves associated with a normal cube can be performed to it. The user can generate a random cube using the inbuilt randomizing function or can paint the cube himself/herself and obtain a result. Below is a screen shot of the Graphical User Interface of the Software:



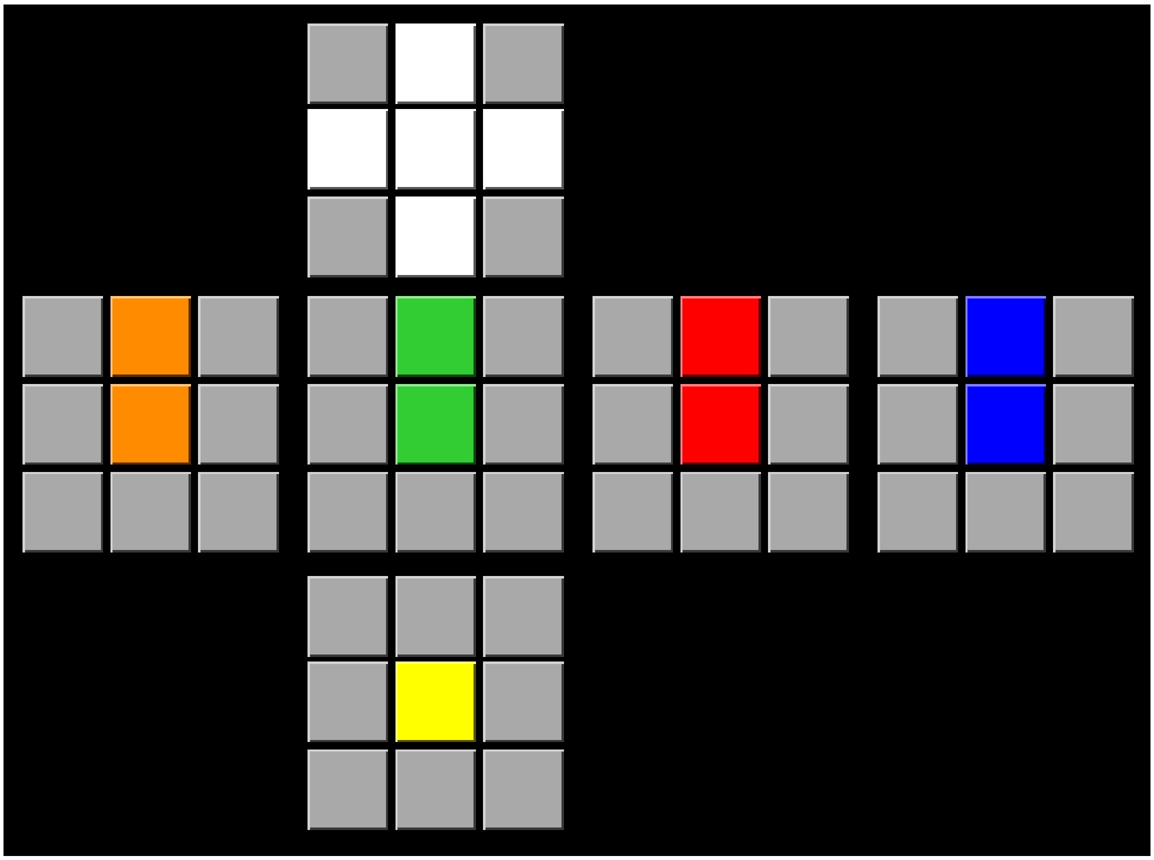
<u>Some of the Functions:</u>

Algorithms:	This is a collection of algorithms that are used to solve the cube using the layer by layer method.
Clear Cube:	This function clears the cube so it can be painted.
Random Cube:	This function generates a random cube. It randomly chooses one of 35 randomizing algorithm and then randomized the cube. To truly randomize a cube this function is done a couple of times in succession.
Revert Saved:	This function reverts back to the saved cube.
Reset Cube:	This function resets the cube to the solved state.
Save Cube:	This function saves the present state of the cube.
Simplify:	This function simplifies the move list by removing redundant move and combining some of them. If the adjust to color checkbox is checked, it converts the moves to the color sides.
Solve:	This function solves the cube using the F2L with Keyhole-OLL-PLL method
Statistics:	This function is used find statistics for solving the cube. It finds the maximum, minimum numbers of moves required to solve random cubes. It also calculates the average number of moves required to solve cube.

An Overview of the Solution Process:

Solving the Top Cross:

The goal of this step is to attain the following state: (i.e. solve all top edge pieces)



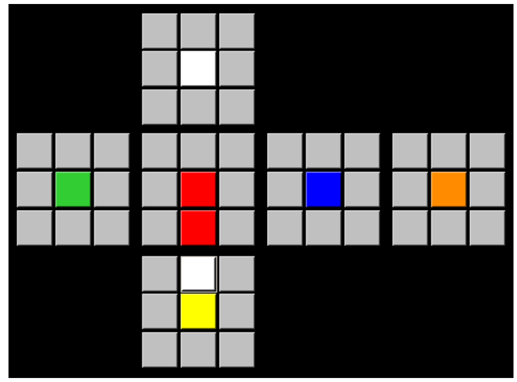
To Solve the Top Cross:

- Align Maximum Pieces in Top Layer
- Align Correctly Oriented Pieces from Bottom Layer
- Align Pieces from Middle Layer
- Align Incorrectly Oriented Pieces from Bottom Layer
- Align Incorrectly Oriented Pieces from Top Layer
- Align Skew Pieces in Top Layer

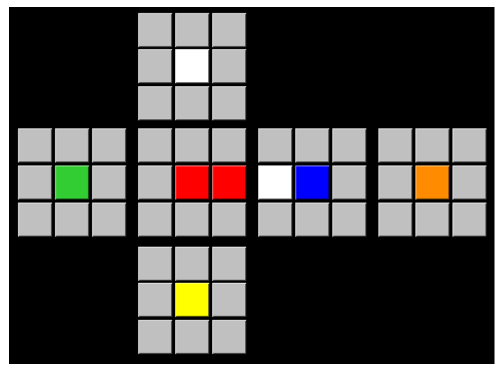
Some of the above steps are repeated more than once till the cross is solved.

The Positions are given below:

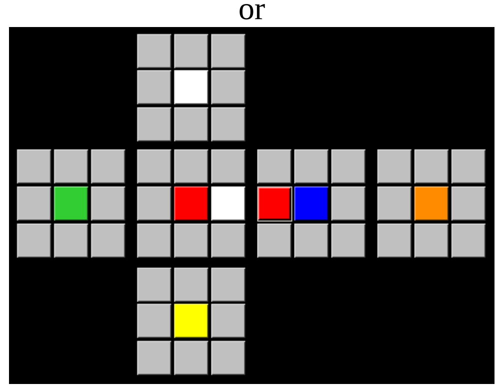
Correctly Oriented Pieces from Bottom Layer



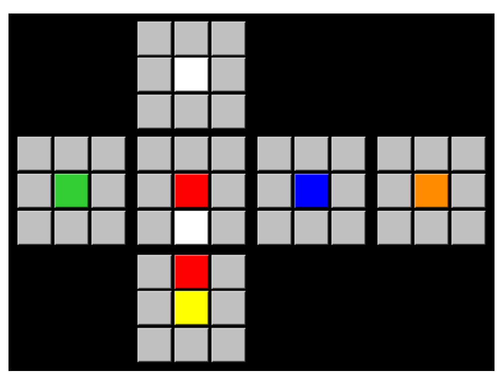
Pieces from Middle Layer



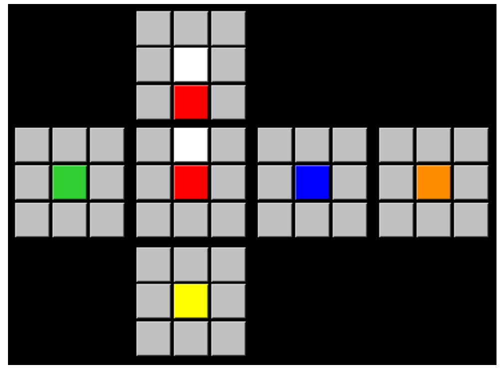
or



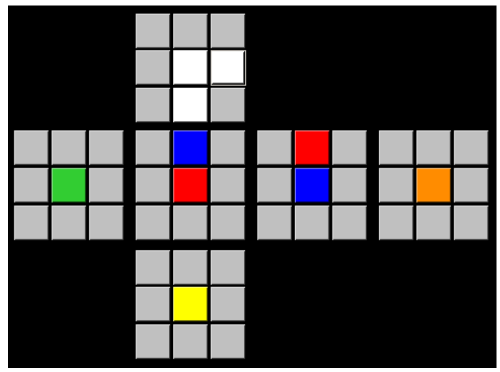
Incorrectly Oriented Pieces from Bottom Layer



Incorrectly Oriented Pieces from Top Layer

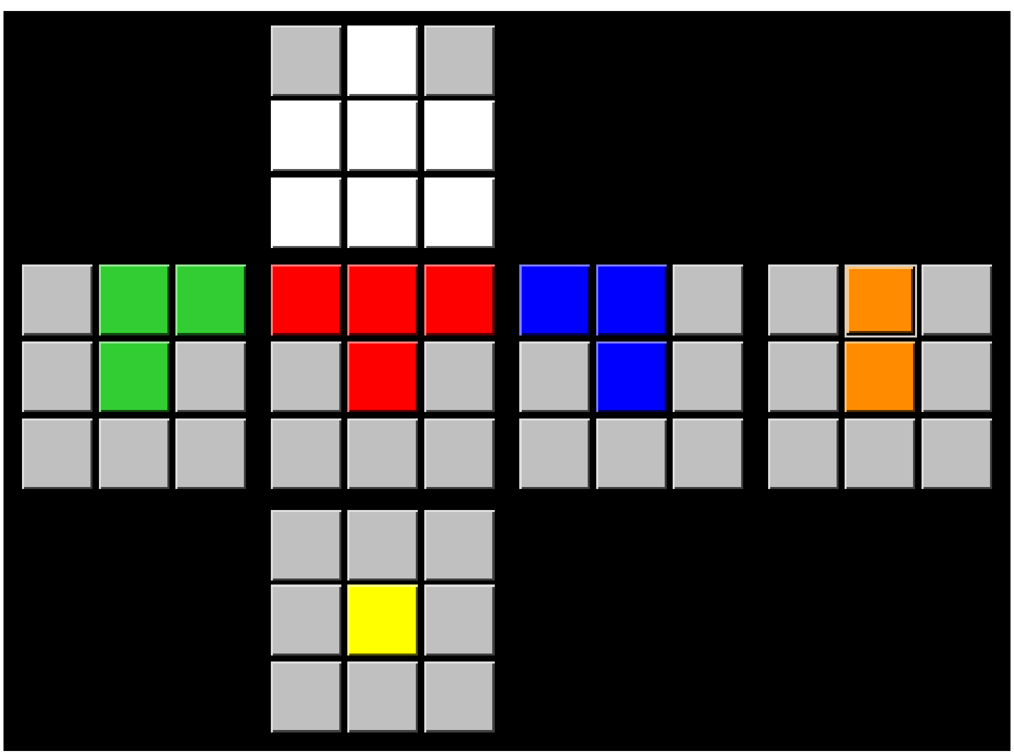


Skew Pieces in Top Layer

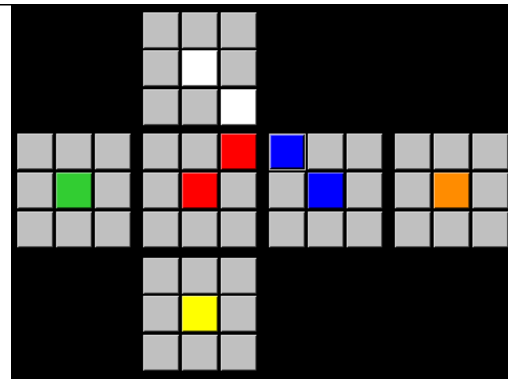
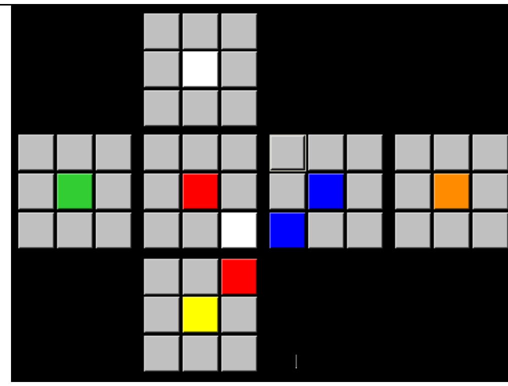
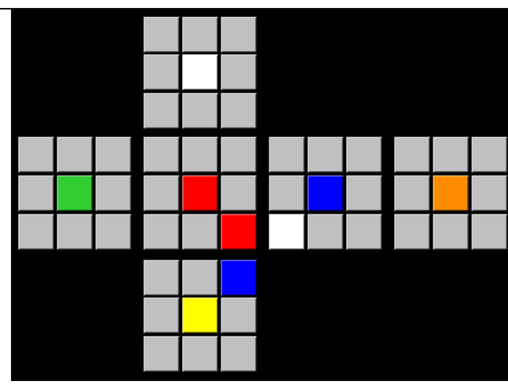
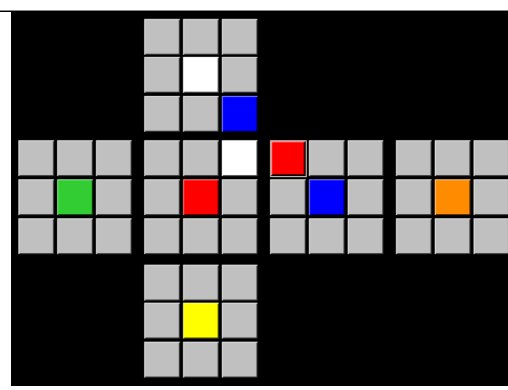


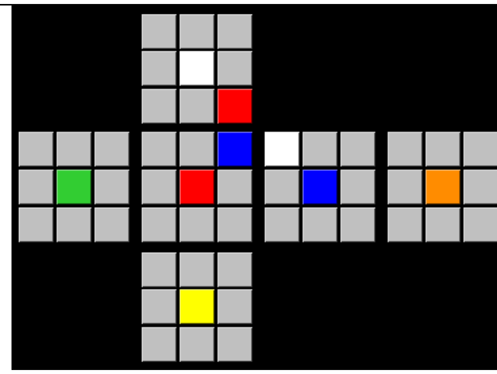
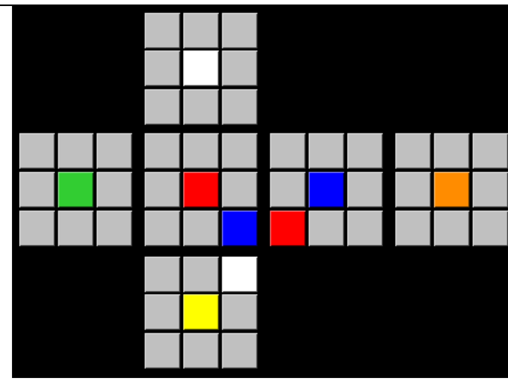
<u>Solving Two Top Corners:</u>

The Goal is to solve at least 2 of the top corners by this step:

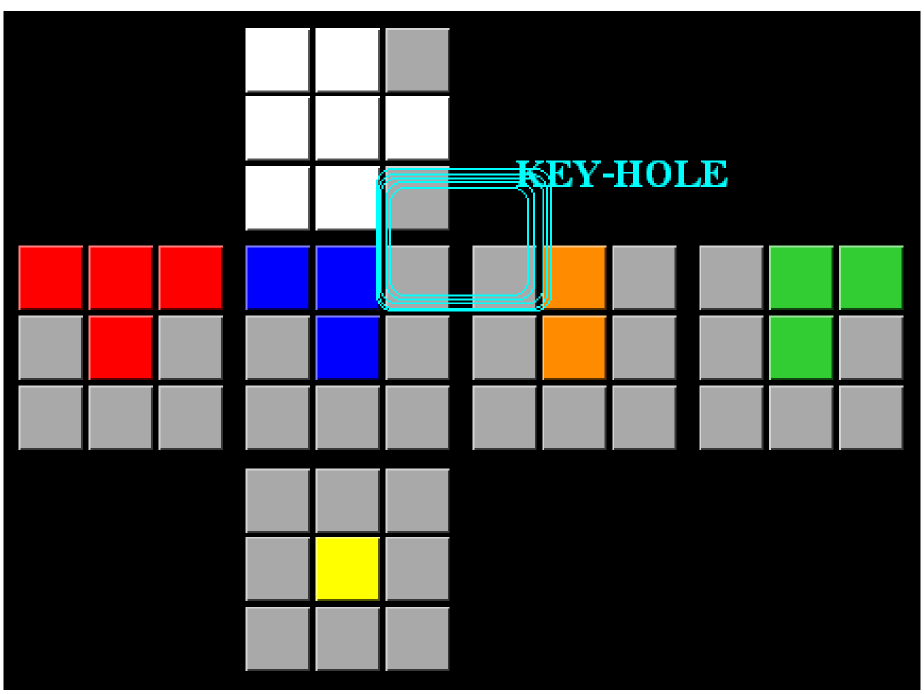


The Top Corner can be found in 6 six different possible positions:

Identity: Corner piece is in position No Algorithm Required (do nothing)	
Type 1a: Corner Piece in the Bottom Layer & White in the Front Side F D F’	
Type 1b: Corner Piece in the Bottom Layer & White in the Right Side R’ D’ R	
Type2a: Corner Piece is in the Top Layer & White in Front Side R’ D R F D F’	

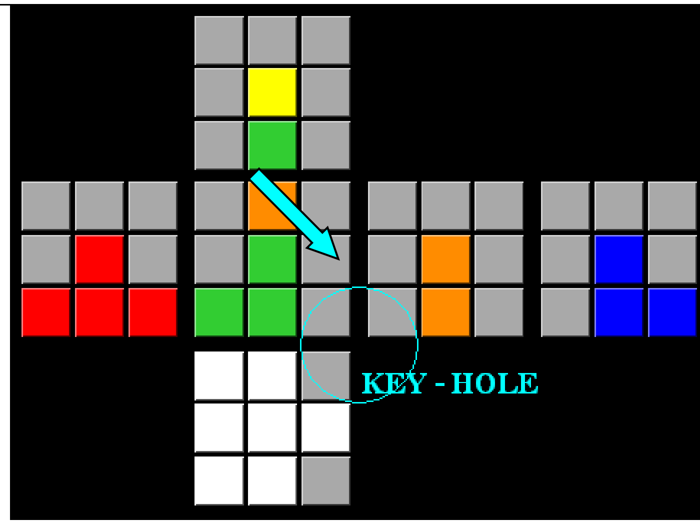
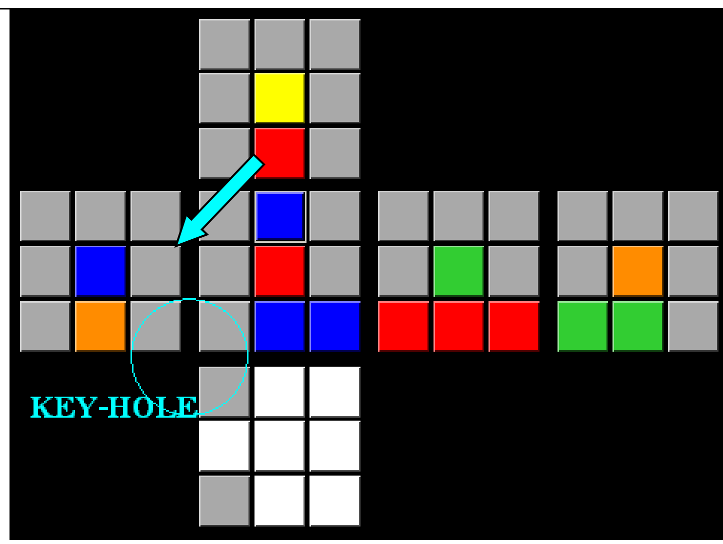
Type 2b: Corner Piece is in the Top Layer & White in Right Side F D’ F’ R’ D’ R	
Type 3: Corner Piece is in the Bottom Layer & White in Down Side R’ D2 R D R’ D’ R	

At this moment we have a high probability of having at least one Keyhole. In most cases we should have two Keyholes. Below is a Keyhole:

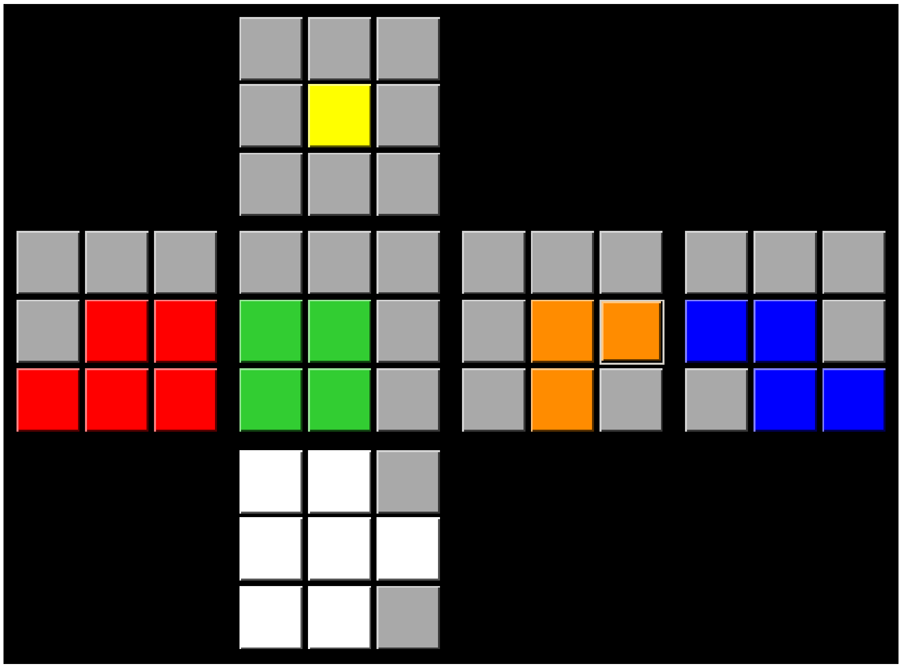


Solving 2 Middle Layer Edges

At this moment put the cube upside down and use the Keyhole to fill in at least 2 of the middle edges pieces. We bring the Keyhole directly under where the edge piece need to go and use one of the situation from below to fill in at least two of the middle edge pieces.

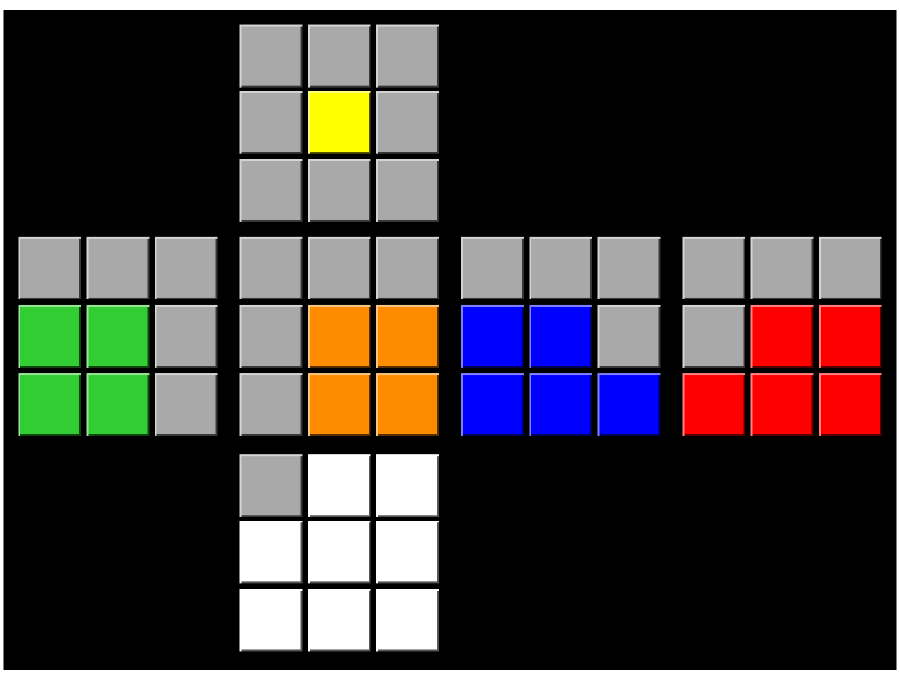
R U' R'	
L' U L	

The goal at the end of this stage is to have at least 2 white corner and 2 middle edges in position:

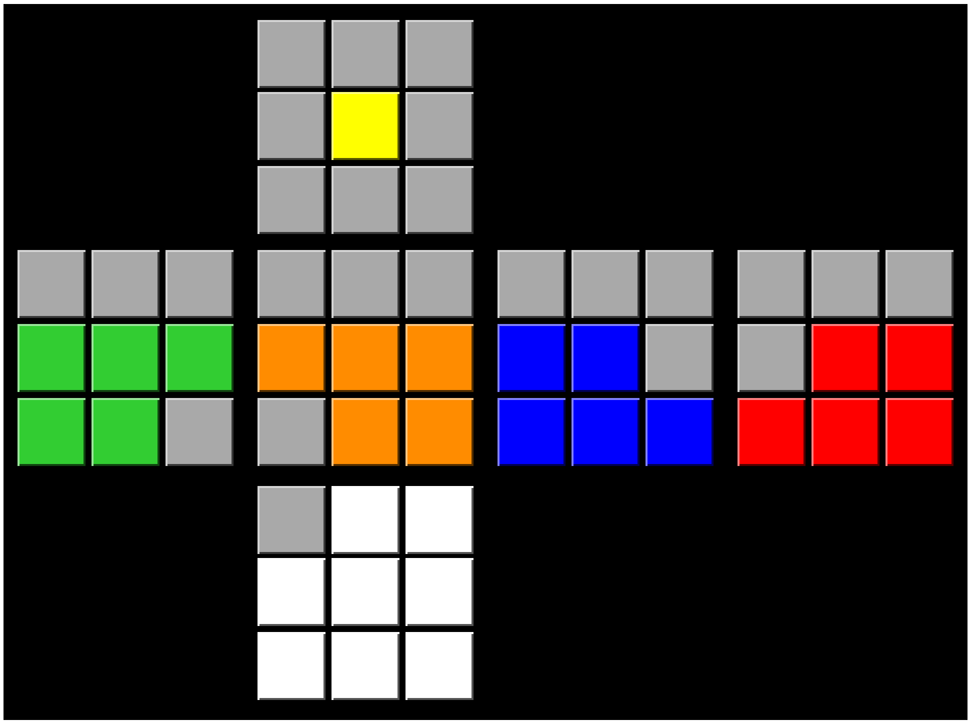


Solve 3rd Corner

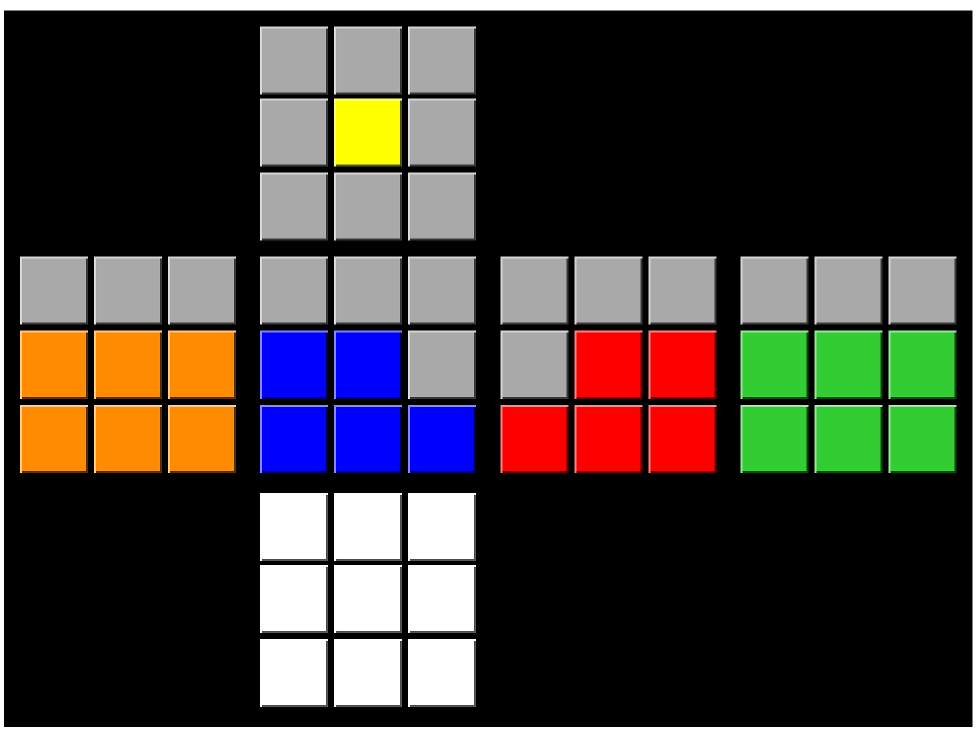
Goal is to achieve at least 3 of the white corners positioned by now.



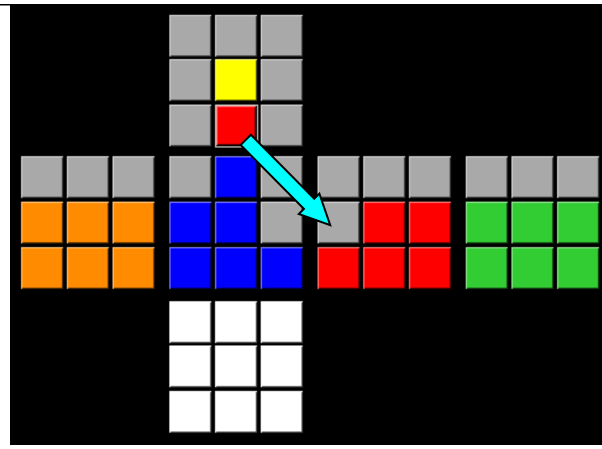
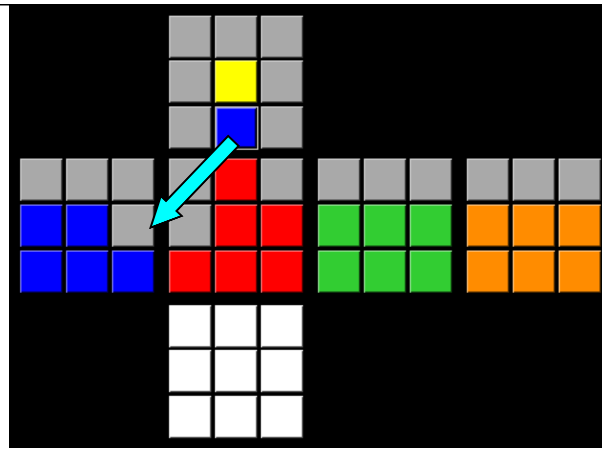
Solve Maximum Middle Edges:



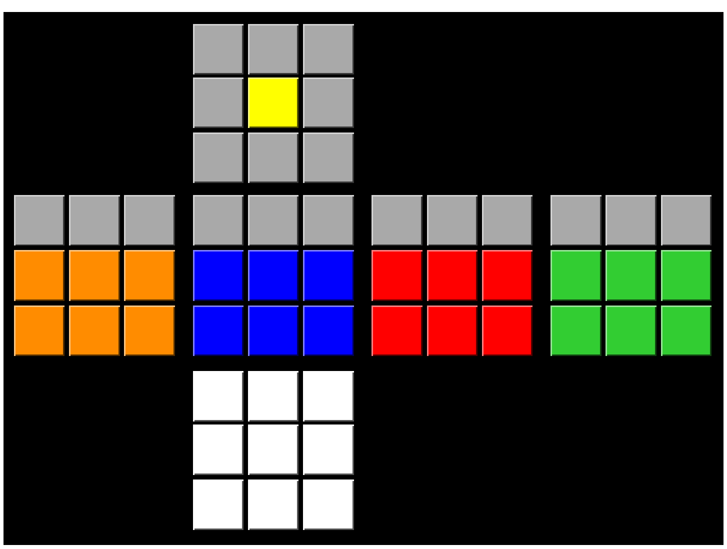
Solve First Layer:



Solve Middle Layer:

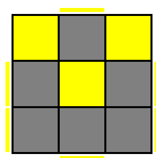
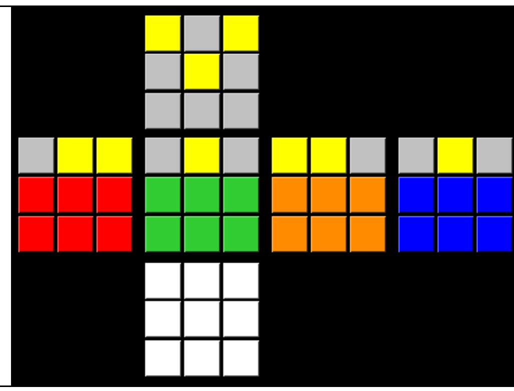
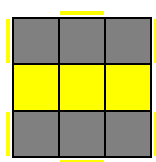
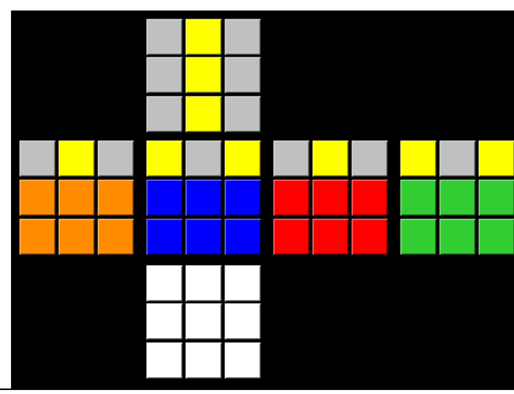
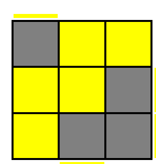
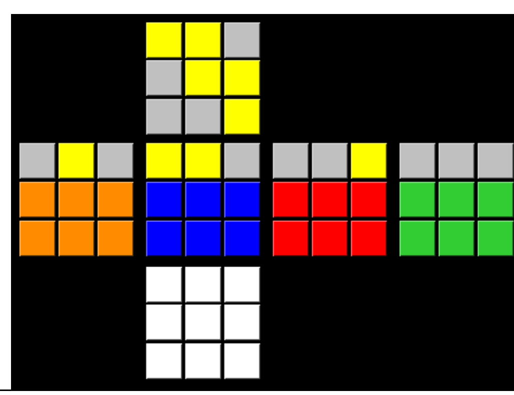
Case 1: Edge Piece on top needs to go on the right	
Case 2: Edge Piece on top needs to go on the left	

By the end of this step the First 2 Layers are Done (F2L):

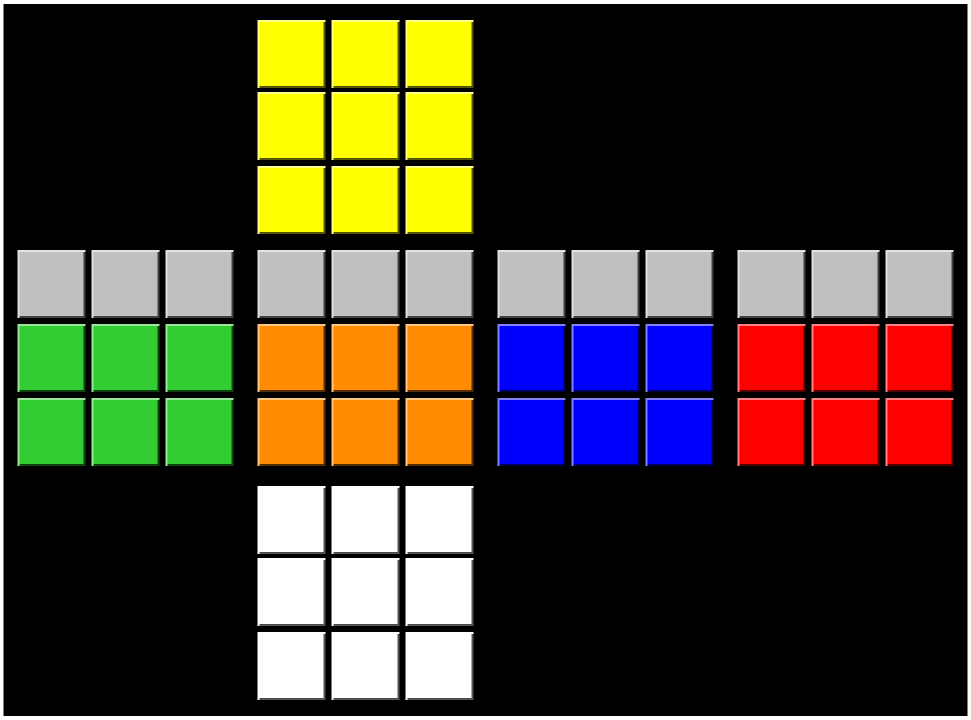


Orientation of the Last Layer:

During this step, the software correct the Orientation (corner twist and edge flip) of every piece in the Last Layer (Yellow), which results in the U face (Yellow) being solved, but not the U Layer. In this the software rotates the entire on cube on the y-axis 4 time and each time it rotates the U (Yellow) side 4 times and check the U face with the 57 OLL possibilities and they uses the appropriate algorithm accordingly. For Example:

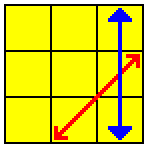
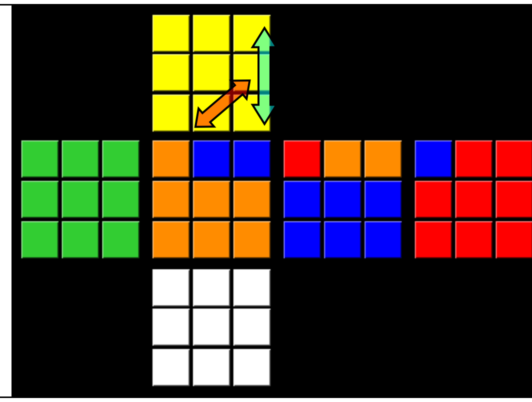
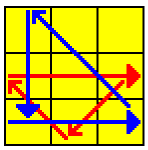
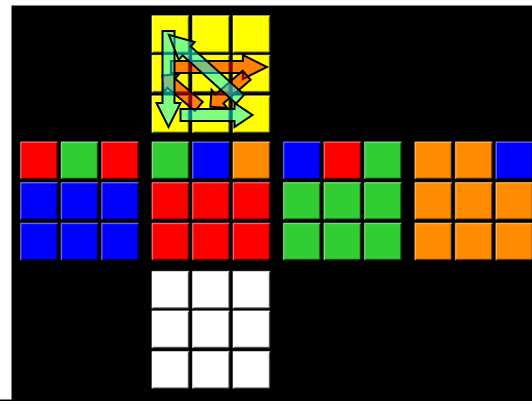
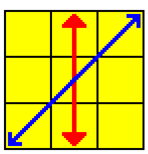
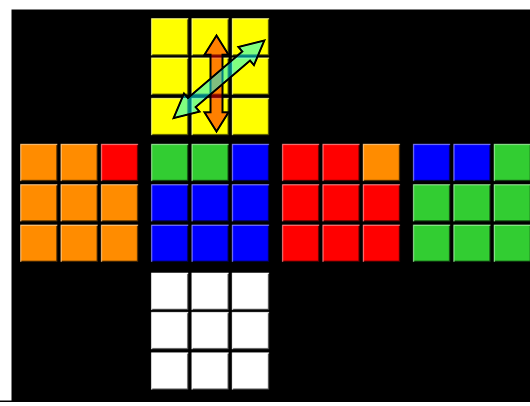
 (R' U2) F (R U R' U') y' (R2' U2) (R B)	
 F (R U R' U' R) y' (R' F) (R B') (R' F')	
 (R U R' U) (R U' R' U') (R' F R F')	

The end of this step the Cube is in the following step:

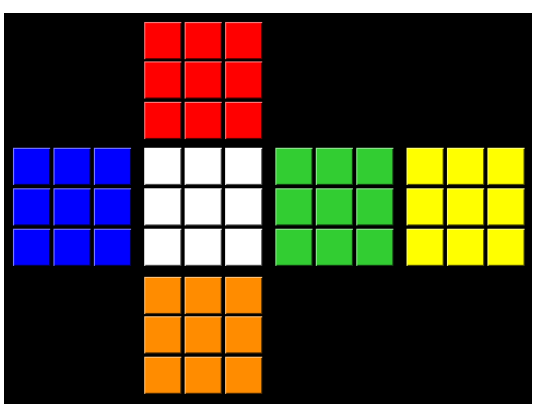


Permutation of the Last Layer (PLL)

The PLL is the final stage in the System. During this stage, the software corrects the positions of each cube in the Last Layer, without changing their Orientation. This results in a completed Last Layer, and therefore a solved cube. Similar to the OLL step, the software rotates the cube 4 time on the y-axis and each time the U (Yellow) layer 4 times and then check for one of the 21 possible cases and uses the appropriate algorithms. For Example:

 (R U R' F') (R U R' U') (R' F) (R2 U') (R' U')	
 (R2' u' R U') (R U R' u R2) y (R U' R')	
 [(R' U L') U2 (R U' L)] [(R' U L') U2 (R U' L)] U'	

At the end of this step the cube is in the solved state:



Statistics:

The following statistics were generated by letting the computer generate 300 random cube positions and then counting the maximum or the upper bound and the average number of moves required to solve a certain step.

Step	Upper Bound	Average
Top Cross	11	5.14
F2L	82	50.48
Total	97	73.22

Note:

- The algorithm of this software is designed to start solving with the white side always. Further efficient and shorter result can be obtained by starting the solving form a different color side. This can be done by solving the cube six times, each time starting form a different side and then using the shorter result. In that case the upper bound and average can be lowered.
- By generating 20,000 random cubes, the upper bound for the entire solution was found to be 107 and the average was found to be 73.52.

Further Research objectives:

Optimization of the software:

The software can be further optimized by the use of better algorithms, pruning table and using trial and error. An advanced element of Artificial Intelligence can be added to it so that the software learns from experience and finds shorter solutions on its own. A database of Pretty Pattern can also be added to this software. I also want to add an animated user friendly output of the solution.

Study the Cube Mathematics:

The most common type of cube study focuses on permutation and algorithms. The Rubik's Cube creates a universe of its own. Rubik's Cube can be studied using the Group Theory. To find the optimum solution the use of advanced mathematics, especially group theory and isomorphic functions is the key. I want to find the optimum solution using the study of advanced mathematics of the Rubik's Cube.

Robot to solve the Rubik's Cube – Automatically

The prototype has already being designed. The software will solve the Rubik's Cube and send the solution to the robot, which will then physically solve the Rubik's Cube. To make the robot I will be using LEGO Robotic Mindstorm NXT. The coding of the robot can also be made in VB.Net.

Conclusion:

The Rubik's Cube is a marvelous piece of invention. Not only is it an amazing mechanical device but in it is locked numerous mathematical and logical puzzles. The Rubik's Cube has undoubtedly sparked the interest of anyone who has handled it, especially me.

While writing the software I had to convert human thinking to of computer logic. Solving the first layer of the Rubik's Cube is very intuitive, therefore converting that intuitive thinking to of if-then logical statements made me understand how complex the human brain is. It is amazing how we humans make complex decisions intuitively. While doing my work with the software I have definitely learned to appreciate the human mind deeply.

I am very much interested in extending my research further in terms of mathematics, programming and robotics. I am sure the Rubik's Cube will be enchanting generations in the future just as it has done in the past.

Works Cites

Bradley Julia Case and Millspaugh Anita. Programming in Visual Basic .Net. McGraw-Hill Irwin, NY, 2005.

Beginner Solution to the Rubik's Cube. 16 January 2008. 20 April 2008. <http://peter.stillhq.com/jasmine/rubikscubesolution.html>

Bellis, Mary. "Rubik and the Cube – Rubik's Cube". Inventors. 14 May 2008. <http://inventors.about.com/od/rstartinventions/a/Rubik_Cube.htm>

Bob Burton's Cube Page. Bob Burton. 20 April 2008. <http://www.cubewhiz.com/>

Cao, Charles. "A Simple Complexity". Illumin. Volume 3, Issue ii. 17 April 2008. <http://illumin.usc.edu/article.php?articleID=11&page=3>

Gilbert Jimmie and Gilbert Linda. Elements of Modern Algebra. 6th Ed. Thomson Books/Cole, CA, 2004.

Marshall, Ray. "Squaring up to the Rubik Challege". Evening Chronicles. 27 July 2005. 15 April 2008. <http://icnewcastle.icnetwork.co.uk/1000expats/expatsfeatures/tm_objectid=15786140&method=full&siteid=50080&headline=squaring-up-to-the-rubik-challenge-name_page.html>

Optimal Rubik's Cube Solver. Michael Reid. 20 April 2008. <http://www.math.ucf.edu/~reid/Rubik/optimal_solver.html>

SpeedCubing Dot Com. 24 April 2008. <http://www.speedcubing.com>

Speedcubing taken one step further. Lars Vanderbergh. 20 April 2008. <http://www.cubezone.be/>

Singmaster, David. Notes on Rubik's Magic Cube. Enslow Publishers, NY, 1981.

Solving the Top 2 layers. Jessica Fridrich. 20 April 2008.
<http://www.ws.binghamton.edu/fridrich/Mike/middle.html>

Solving the 3x3x3 Rubik's Cube. Dan Harris. 20 April 2008.
<http://www.cubestation.co.uk/cs2/index.php?page=3x3x3/3x3x3>

Rubik Erno, Varga Tamas, Keri Gerzson, Marx Gyrgy and Vekerdy Tamas.
Rubik's Cube Compendium. Oxford University Press, NY, 1987.