Rubik’s Cube Solver

6.111 Final Project
Fall 2013

Katharine Daly
Jack Hutchinson
Rubik’s Cubes are difficult to solve

POSITION THE YELLOW CORNERS CORRECTLY

1st Step: Position yellow corners correctly
While holding your cube as shown above, look at the top (U) face. Place the 2 correct corners in the back A, B or diagonally A, D or B, C across from each other.

If Corners A and B are in the right place then to switch C and D, do the sequence below.

If you need to switch diagonal corners like B and C or D and A, then do the sequence once. Then, orient the cube so the 2 correct corners are in the back and do the sequence again.

Congratulations!
If your cube looks like this picture you can move to the 2nd Step!
An animated solution would be helpful
The solution is divided into three parts

- Get Initial State
- Solve Cube
- Show Animation
The cube configuration must be represented consistently.

- 6 colors -> 3 bits
- 54 cubelet faces
- $3(54) = 162$ bit representation
The rotations must be specified consistently.

- 18 rotations -> 5 bits
Step 1: Gather the starting configuration

- Cross image (opposite) displayed on-screen.
- Scroll through cubelet faces using arrow buttons.
- Input colour with correct button.
- When complete system checks to ensure configuration is valid.

G = Green
W = White
O = Orange
R = Red
B = Blue
Y = Yellow
Step 2: Choose an algorithm

Algorithm Tradeoffs:

- Number of rotations vs. Memory required
- More than $4.3 \times 10^{19}$ starting configurations

Decision:

- 7 step solution
- Solve top, then middle, then bottom layer
Step 2: Use a state machine
Step 3: Animate solution

- Display cube in 3D on screen
- Use matrix multiplication to rotate image

\[
R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & -\sin \theta \\
0 & \sin \theta & \cos \theta
\end{bmatrix}
\]

- Rotator sub-module to carry out animation
  - sends rotation matrix values to 3D renderer
- 2 buffers
  - one to store calculation results
  - one to display cube
  - switch every frame
Step 3: Rotation Sequence
Step 3: Animation Block Diagram
Timeline

Week of 11/11:
  Katharine: Solve top layer of cube
  Jack: Complete basic animation module

Week of 11/18:
  Katharine: Solve entire cube
  Jack: Complete Get_initial_state module

Week of 11/25:
  Katharine: Debug cube solving
  Jack: Improve animation module & debug

Week of 12/2: Debug