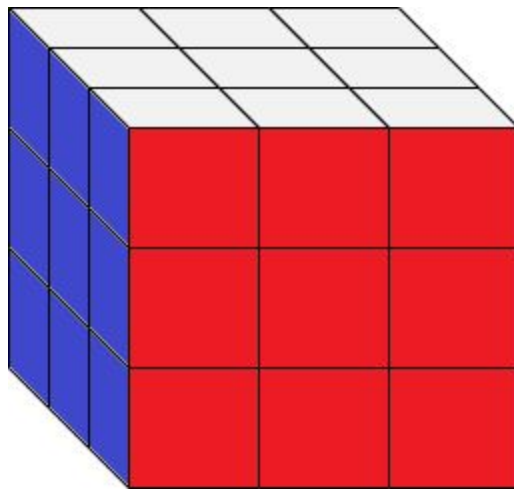


# Solving the Rubik's Cube



David Nelson  
CIS667  
Semester Project  
10/28/2020

<b>Introduction</b>	<b>3</b>
<b>Terms</b>	<b>3</b>
<b>The Rubik's Cube</b>	<b>4</b>
<b>Installation Instructions</b>	<b>5</b>
Installing Anaconda	5
Installing NumPy	5

# Introduction

This document begins by describing the game, including the state of the game and how it is represented, the goal state, and how to perform actions to transition from one state to the next.

Following the description of the game, this document will go over the structure of the rubik's cube solver program. This will begin with installation instructions and prerequisite software and will then dive into usage of the program for the purpose of simulating a cube that the player can rotate via commands.

The document will then go into describing the rubik's cube solver program in detail. (TODO: Next release)

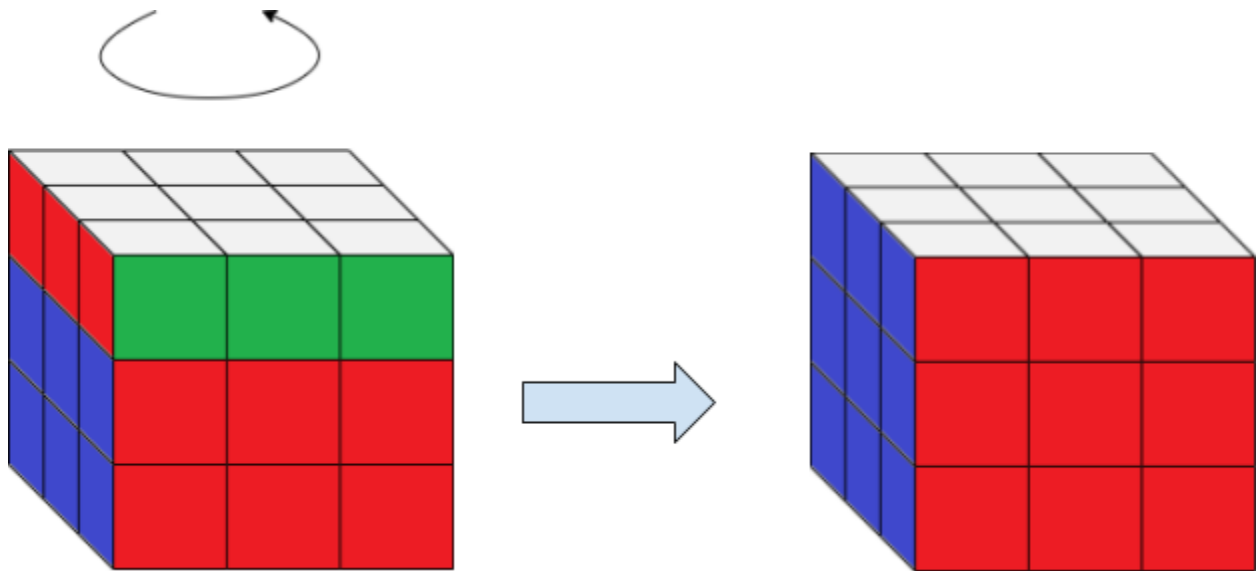
## Terms

Cubie	A single sub-cube of the full rubik's cube. For example, a 3x3x3 rubik's cube contains $(3 \times 3 \times 3) = 27$ cubies.
-------	---

# The Rubik's Cube

The Rubik's cube puzzle is an  $N$ -by- $N$ -by- $N$  ( $N \geq 2$ ) cube composed of smaller cubes known as cubies. The player makes an action by rotating a row or column of cubies. The goal of the game is to solve the puzzle, which is to ensure that each face of the Rubik's cube contains only a single color.

Here is a simple example using a nearly-solved 3x3x3 cube:



**Figure 1: Performing an action and solving the Rubik's cube.**

**In this example, the player makes the action of rotating the top row of cubies to the right 90°. This transitions the cube from the state on the left, to the state on the right (the goal state).**

The player is allowed to rotate any of the rows or columns of cubies by 90°, 180°, or 270° during the course of the game.

# Installation Instructions

Prerequisite Software:

Python interpreter - <https://www.python.org/>

NumPy - Python vector computing library - <https://numpy.org/>

If these are already installed then you are ready to run the Rubik's Cube Solver program. If not, you can follow the instructions below to get started using Anaconda which installs Python automatically and offers a simple NumPy installation procedure.

## Installing Anaconda

1. Go to <https://www.anaconda.com/products/individual>
2. Depending on your OS and hardware, choose the appropriate installer and follow the instructions in the installer.

## Installing NumPy

1. Start the Anaconda command prompt. (On Windows 10, this can be opened by going to Start -> Anaconda3 -> Anaconda Prompt.)
2. Execute the command "conda install numpy".
  - a. If prompted to Proceed with installation, enter "y".

You are now ready to execute the Rubik's Cube program.

# Obtaining the Code

Github repo: [https://github.com/drnelsoniv/CIS667\\_rubiks.git](https://github.com/drnelsoniv/CIS667_rubiks.git)

## Running the Program

1. Start the Anaconda command prompt. (On Windows 10, this can be opened by going to Start -> Anaconda3 -> Anaconda Prompt.)
2. Execute the program: "py [your\_path\_here]/main.py"  
**Replace [your\_path\_here] with the path to the Rubik's Solver script.**

main.py accepts one optional argument indicating the size of the rubik's cube. This must be an integer value between 3-7 inclusive. Other values or adding additional parameters will result in an error and a usage statement is printed.

## Game Setup

**main.py** accepts a single command-line argument indicating the size of the board.

## Playing the Game

A player performs an action by rotating one of the rows/columns of the presented cube. This is achieved by entering a move, such as **OR90**:

0 - Rotate the row/column at index 0.

R - Rotate the row.

90 - Rotate by 90°.

When rotating a row, a positive rotation rotates the selected row from **right-to-left**.

When rotating a column, a positive rotation rotates the selected column from **top-to-bottom**.

Additional usage information if an invalid command is entered.

# Notes to the Professor

These are included to assist with grading:

- Rotation does not quite work correctly yet. It does some instances but there are issues with the perspective that the cube face is being viewed from.
- Pieces of the cost functionality that you requested have been implemented, but no accounting is being done at the present time, so nothing is printed to you.