

2-26-2021

Group Theory Visualized Through the Rubik's Cube

Ashlyn Okamoto
Portland State University

Follow this and additional works at: <https://pdxscholar.library.pdx.edu/honorstheses>



Part of the [Other Computer Sciences Commons](#), and the [Other Mathematics Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Okamoto, Ashlyn, "Group Theory Visualized Through the Rubik's Cube" (2021). *University Honors Theses*. Paper 977.

<https://doi.org/10.15760/honors.1001>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in University Honors Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Group Theory Visualized Through the Rubik's Cube

By

Ashlyn Okamoto

An undergraduate honors thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Science

in

University Honors

and

Computer Science

Thesis Adviser

Andrew Wilson

Portland State University

2021

Abstract

In my thesis, I describe the work done to implement several Group Theory concepts in the context of the Rubik's cube. A simulation of the cube was constructed using Processing-Java and with help from a YouTube series done by TheCodingTrain. I reflect on the struggles and difficulties that came with creating this program along with the inspiration behind the project. The concepts that are currently implemented at this time are: Identity, Associativity, Order, and Inverses. The functionality of the cube is described as it moves like a regular cube but has extra keypresses that demonstrate the concepts listed. Each concept is looked at in the scope of Group Theory and how it can be visualized by this simulation. Each implemented concept is then explained in the context of the Rubik's cube. The code is accessible on GitHub at <https://github.com/asokamoto/GroupTheoryCube>.

Table of Contents

ABSTRACT	2
TABLE OF CONTENTS	3
1 BACKGROUND	4
2 PURPOSE	5
3 METHODS	6
4 FUNCTIONALITY	7
4.1 IDENTITY	8
4.2 ASSOCIATIVE PROPERTY	9
4.3 ORDER	9
4.3 ORDER	9
4.4 INVERSE	9
5 FUTURE EXPANSION AND USE	10
6 RERENCES	11

Background

Rubik's Cubes were invented in 1974 by a Hungarian professor named Erno Rubik. He spent only a few months designing the cube, but it took six years to gain the popularity that has carried it into the 21st century. For those who do not know what a Rubik's cube is, it is a six-sided puzzle with nine stickers on each side. All six sides have different colors: white, yellow, red, orange, blue, and green.



Figure 1: Solved Rubik's Cube

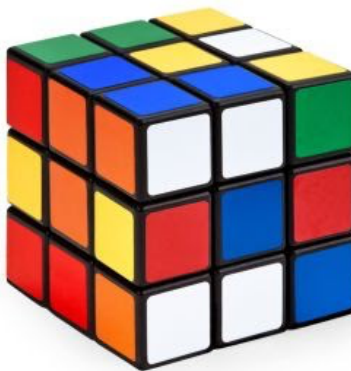


Figure 2: Mixed Rubik's Cube

Using an internal pivot mechanism, each side can spin individually so the colors can be mixed around. The solved state is achieved when all sides are uniform color. Rubik always saw the cube as a piece of art that was more than just plastic with stickers.

Though maybe not Rubik's initial intention, the cube is a great example of a math subject called group theory at work. A Rubik's Cube has 43,252,003,274,489,856,000 possible positions. In English that is 43 quadrillion and some. A single position is known as a "permutation" in group theory terms. A permutation is an arrangement of members in a sequence or order. Permutations are just one of the many group theory concepts that can be visualized by the Rubik's Cube. Group theory is taught in undergraduate and graduate schools and is a part of

abstract algebra. The focus is on an algebraic structure known as groups, which has a different definition than in the English language.

Group theory relies on the understanding of a few basic terms and consistently builds upon those terms. Many definitions include previously defined concepts. For example, to understand the term “group,” the terms “set,” and “operation” must first be defined. To define the word “operation,” the term “function” must be defined. As this is a very wordy and term-based subject, many concepts are easier to understand and grasp when visualized. It is a common idea to use the Rubik’s Cube as a teaching and visualization tool in the group theory discipline. The Rubik’s Cube is even featured on the Group Theory Wikipedia page because it is an easy way for the public to recognize the subject in their everyday lives. A professor named David Joyner wrote a textbook called *Adventures in Group Theory: Rubik’s Cube, Merlin’s Machine, and Other Mathematical Toys*, which is a compilation of revised lecture notes from his classes for students who were capable of learning the content but had

“more immediate pressures in their lives than the long-term discipline required to struggle with the abstract concepts involved (Joyner, ix).”

Professor Joyner wanted to use something as irresistible as the Rubik’s Cube to get students more interested in the subject and help them grasp the concepts that, without the visualization, they would have trouble understanding. Since releasing the textbook, he has created another edition after he found others took interest.

Purpose

The Rubik’s Cube has always been a huge part of my life. I got my first Rubik’s Cube in 4th grade when I saw a 5th grader playing with one after school. As soon as I got into the car that afternoon, I asked my mom to take me to the store and we bought one. I spent a couple months learning the steps and understanding how to get certain pieces in specific places. I would throw around words such as permutation, orientation, and algorithm without truly understanding what they meant. I can remember after about a month of trying to finish the cube I got close to solving it, and I ended up messing the cube up and having to start over. I started to cry and truly almost gave up. I ended up writing my college essay on this exact experience and how it has since shaped me into the person I am today. I kept solving Rubik’s Cubes and I eventually got my solving time down to around 15 seconds with a personal best of 10 seconds. Throughout my life, people knew me as the girl who could solve Rubik’s Cubes and a common question that would come with that was,

“You must be really good at math, right?”

I never understood how math related to the Rubik’s Cube because I had just been memorizing these series of steps that would put the cube into positions that I wanted. I would shrug off the comment by saying,

“I do like math but haven’t found my knowledge of the cube to be particularly helpful!”

I ended up taking Group Theory to satisfy a math requirement within my Computer Science degree, and I was very surprised to see the Rubik’s Cube on one of the professor’s slides. I went up to him after class and asked him about it explaining my experience with the cube. I told him

how I always wondered how the two were related. He gave me a quick overview and throughout the course, I found myself relating the content back to the Rubik's Cube. In the Portland State Honors College, Computer Science students are required to complete a thesis along with the capstone in the Maseeh College. While thinking about potential thesis topics, I was reminded of a program I created in my 10th grade Introduction to Java course, where I created a 1x1x1 version of a Rubik's Cube that used the arrow keys to shift the colors around in a way similar to an actual cube. As it was one of my first real coding projects, I thought it would be a great full circle moment to do my senior year college thesis with a similar premise. My group theory professor, Andrew Wilson was kind enough to be my adviser on this project.

Though the Rubik's Cube is an accepted visualization technique in math, it is not commonly used in a classroom setting because of the unfamiliarity most people have with the puzzle. Even when talked about in group theory classes, the cube is immediately seen as a hard puzzle rather than the math concepts it can represent. It is not practical to expect students to focus on the math when there are other distractions such as the difficulty and mixing up of the cube. It would be much easier to use the Rubik's Cube as a visualization technique if there was a controlled computer version where users are able to control the movement with buttons.

As this project is based on group theory concepts, I had to refresh my memory on many of the concepts that I recreated. I had many helpful meetings with my adviser to talk about which concepts would be easiest represented in the simulation and which were too abstract to be helpful in a project like this. I hadn't taken Group Theory since the winter of 2020, so after a year of not studying the content, I had some trouble grasping the concepts I was trying to recreate. This gave me even more motivation and purpose for this project as I believe that a tool like this would have been helpful in this situation. I found that real world examples were the most helpful in relearning content, which I hoped this project would enforce. A real-world concept that specifically remember is in reference to expanding inverses. When seeing an equation like the following,

$$\text{Eq. 1 } (xy)^{-1} = y^{-1}x^{-1}$$

students may just memorize that when expanding inverses, the expression puts the variables in reverse order. A real-world example to remember this concept is the shoes and socks explanation. When putting on shoes, you put your socks on first, then your shoes. When taking your shoes off and creating an inverse process, you take your shoes off first, then your socks. When I first heard this, I remember thinking about how much sense that made. It was no longer a peculiar group theory concept that didn't have much of an explanation other than that was how it was, but when relating it to something in my real life, I found it was much easier to understand. Though the Rubik's Cube isn't necessarily as real-world as the shoes and socks as not everyone understands how a Rubik's Cube works, but I hoped that seeing a visual representation of these concepts, would help students understand and relate to the concept in a way that would help them remember why concepts are the way that they are.

Methods

I first had to decide which language I wanted to use for this project. I had created that 10th grade project in a language called Processing Java, which is sketchbook software and language that is used for visual art type projects. There is a lot of documentation on the language, and I knew that there would be a lot of support in case I came across a syntax or logic issue. There are a lot of libraries that were useful in the project as well. One library that I used was one called

PeasyCam, which gives the image a 3D aspect. This way, the user is able to move the content around as if it was in a 3D space. This was the type of support I needed in a project like this, especially when I did not have a lot of prior experience in the environment. The ArrayList functionality was used a lot in this project as this is what holds each set of moves for each concept.

When researching for this project on the internet, I came across a set of tutorials by a YouTube channel known as TheCodingTrain. In his project, he explained how to use matrices to represent each side of the cube and how to get each piece to move in a way that simulates a real Rubik's Cube. After getting approval from my adviser, I used these tutorials to create the base of the cube. Daniel, the head of TheCodingTrain, decided that his end goal would just be to create a cube that would mix itself up and then do the same moves in reverse to put itself back to its original position. This was really helpful as my completed project would have a similar concept. The difference between our two projects is the implementation of the Group Theory concepts. I used the knowledge how Daniel completes in his simulation to implement each of the concepts chosen.

In my completed project, I used the ArrayList Java library to create each concept that I wanted. Some concepts had more than one list, but these were very important to the functionality of the cube. I spent a lot of time watching the series of videos that Daniel put out because although I was getting a lot of guidance, I wanted to understand the logic behind the code so that it would make sense when debugging later on. I am extremely appreciative that this set of videos were out there, and I think it's a great lesson on properly using other's code. Many times, in coding, there is someone else out there who has had a similar idea. Libraries and APIs are perfect examples of this. Though Daniel had not created a library, his code was made for teaching purposes, so using it in my project made sense. In most cases, you may not have to completely understand what the previously made code is doing, but for this project, because I had the opportunity to dive deeper into his code, I decided that it would be a good idea. While following the tutorial, I tried to make the code my own as much as I could by using variable names that made sense to me. Therefore, the code does not match exactly to the code listed on his website, but it is very much inspired by that set of tutorials.

Another bump in the road was my struggle with the accessibility of the project. I really wanted my project to reach students so it could be useful to others. I explored the idea of translating the code I had to JavaScript or another language that would be easier to host on a website. After a long month of testing those out, I ultimately decided that I could have a better and more complete final project by continuing to use Processing. I wanted to get as many concepts in as I could and I realized that had I continued down the JavaScript path, I would have spent more time on learning the language than the project itself, which was not my main focus. This was a hard reality to accept, as I had put a lot of weight on making it as accessible to the public (students) as possible. After having a few talks with my adviser and others in the field, they convinced me that hosting on an HTML site would be helpful, but necessary. The project is currently available on GitHub with a quick explanation of each of the concepts represented in the project.

Functionality

The cube is fully functional, so each side of the cube can be moved in any direction and those keypresses are listed on the GitHub page. There is also a keypress to reset the cube to its original state from any position.

Notation:

R: Right face moves clockwise
R': Right face moves counterclockwise
U: Up face moves clockwise
U': Up face moves counterclockwise
F: Front face moves clockwise
F': Front face moves counterclockwise
B: Back face moves clockwise
B': Back face moves counterclockwise
L: Left face moves clockwise
L': Left face moves counterclockwise
D: Down face moves clockwise
D': Down face moves counterclockwise

Some of the direction of this is confusing, so I have included a graphic to give further understanding to the notation.

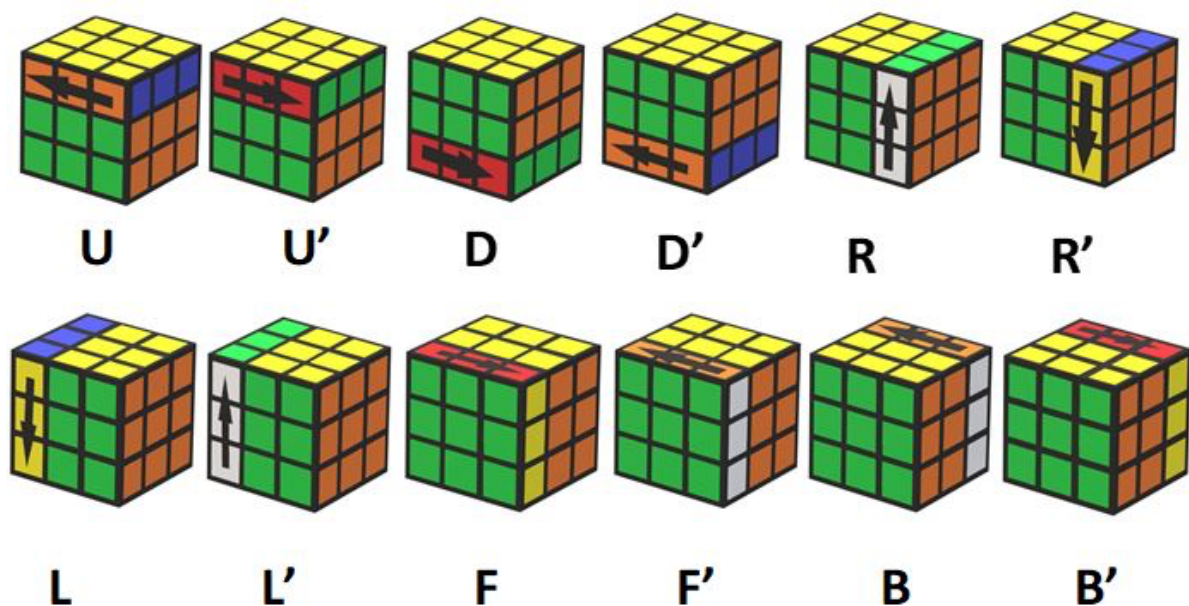


Figure 3: Graphic of the notation used in algorithms for the Rubik's Cube

My adviser and I settled on four concepts to focus on investigating: Identity, Associativity, Order, and Inverses.

In all cases of definitions, "we assume that A is any set and there is an operation on A which we designate with the symbol $*$ (Pinter, 21)"

Identity

According to A Book of Abstract Algebra by Charles Pinter,

“There exists an element e in A such that

$$e * a = a \text{ and } a * e = a \text{ for every } a \text{ in } A$$

If such an element e exists in A , we call it an identity element for the operation $$. An identity element is sometimes called a “neutral” element, for it may be combined with any element a without altering a . For example, 0 is an identity element for addition, and 1 is an identity element for multiplication. (Pinter, 22)”*

For this project, the way it was implemented is by randomly choosing a side of the cube and turning it four times. This may seem like a very simple demonstration, but truly, the cube doing nothing is a way of expressing its identity. Most know what identity is through basic addition and multiplication but may not understand the concept behind it. In addition, the identity element is 0 because when you add 0 to anything, the sum is the original number; the same goes for subtraction. For multiplication and division, the identity element is 1 . Multiplying or dividing anything by one gives the original number. In terms of this cube, when these four moves are done on one side (or nothing is done), the cube is put back to the same position that it was previously. There are many instances of identity within the group that is the Rubik’s cube. In fact, there are many sequences of moves that end with every piece where they began with. All of these sequences are considered identity elements.

Associative Property

According to A Book of Abstract Algebra by Charles Pinter,

$$“a * (b * c) = (a * b) * c \text{ (2)}”$$

If Equation (2) is true for any three elements a , b , and c in A , we say the operation $$ is associative.*

*Remember that an operation is a rule for combining any two elements, so if we want to combine three elements, we can do so in different ways. If we want to combine a , b , and c without changing their order, we may either combine a with the result of combining b and c , which produces $a * (b * c)$; or we may first combine a with b , and then combine the result with c , producing $(a * b) * c$. The associative law asserts that these two possible ways of combining three elements (without changing their order) yield the same result. (Pinter 21, 22)”*

In order for the Rubik’s Cube to be considered a group, it must have an associative property. The grouping of turns can be changed, and the final result will be the same. Within this cube, I have implemented a short series of moves: $R U R'$. Though this may not seem like much, due to the Rubik’s cube being a group, this means that the associative property is evident in these three moves. These moves can be grouped as $(R U) R'$ or $R (U R')$ and still have the same result. The order of the turns is not being changed, just which moves are being grouped together.

Order

According to A Book of Abstract Algebra by Charles Pinter,

“If G is a finite group, the number of elements in G is called the order of G . It is customary to denote the order of G by the symbol $|G|$ (Pinter, 43) “

To demonstrate this concept through an equation, for operation $*$, if we are considering the order of group element a , which is the smallest positive integer n , such that

$$a * a * a * \dots * a = a^n = e$$

where e is the identity element.

The easiest way to understand order is attributing it with size. When a group is generated, it is important to know whether or not that group has an order and what it is. Within this project, I have created a set of moves, $R U R' U'$. Each press of the specific key, the cube will do this set of moves. If the user pressed the button 6 times, the cube will eventually be put back to its original position. This means that this group of moves has an order of 6. The identity always has an order of one, as putting the cube back to its original state can be seen as one move as a whole, but the individual on-sided rotation that is used to demonstrate identity in this project has an order of 4.

Inverse

According to A Book of Abstract Algebra by Charles Pinter,

“For every element a in A , there is an element a^{-1} (“ a inverse”) in A such that

$$a * a^{-1} = e \text{ and } a^{-1} * a = e \text{ (4)}$$

If statement (4) is true in a system of algebra, we say that every element has an inverse with respect to the operation $$. The meaning of the inverse should be clear: the combination of any element with its inverse produces the neutral element (one might roughly say that the inverse of a “neutralizes” a). For example, if A is a set of numbers and the operation is addition, then the inverse of any number a is $(-a)$; if the operation is multiplication, the inverse of any $a \neq 0$ is $1/a$. (Pinter, 22)”*

An inverse is the move that undoes another. In multiplication the inverse of x is $\frac{1}{x}$ as this undoes the initial multiplication. In terms of the Rubik’s cube, each individual move has an inverse. For example, R has R' , U has U' and so on.

For this implementation, I give an example of why inverses, when expanded, are put into a certain order. As mentioned before, the “sock and shoe” example works best for a real-world application. To reiterate, when seeing an equation like the following,

$$\text{Eq. 1 } (xy)^{-1} = y^{-1}x^{-1}$$

students may just memorize that when expanding inverses, the expression puts the variables in reverse order. A real-world example to remember this concept is the shoes and socks explanation. When putting on shoes, you put your socks on first, then your shoes. When taking your shoes off and creating an inverse process, you take your shoes off first, then your socks. For this project, I have the regular functionality (any move and its prime) that shows inverse in the basic sense, but I have also implemented examples of the expansion of inverses. There is a set of keypresses that result in the incorrect order. In terms of the equation above, it would show,

$$\text{Eq. 2 } (xy)^{-1} = x^{-1}y^{-1}$$

which is incorrect. Another set of keypresses will show the correct equation (eq. 1), and the user will be able to tell whether they are doing the correct moves if the resulting cube is fully solved (or in its original position).

Future Expansion and Use

I believe that this project could be helpful to a handful of students who either have some experience in how a Rubik's Cube works and also to those who are open to a unique visualization technique. I would like to add more concepts to the cube along with more solving capabilities such as a scrambler and solver. As mentioned in the method, I would like to use a language such as JavaScript in order to embed the code into an HTML page. The GitHub repository that I set up has instructions on how to run the current code. The page also has instructions on each concept and how the moves of the cube can be interpreted.

As I reflect on the time I spent on this project, I was faced with problems that I had never had in my prior courses. I had a motivation that was something other than getting a good grade. I wanted this piece of software, although small, to have some sort of use for others. This was the perfect opportunity to bring together two parts of my life that had previously only met once. As I finish out my senior year at Portland State, this project reminds me of why I love Computer Science. In a field that seems so concrete, there are ways to incorporate creativity and passion.

References

“Image of Rubik's Cube Solved from Rubiks.com.” *Rubik's Cube Solved 3x3*,
<https://www.rubiks.com/en-us/rubik-s-cube-3x3.html>.

Image of solved Rubik's Cube for demonstrative purposes.

“Image of Rubik's Cube Mixed up.” *On Learning How To Solve a Rubik's Cube*,
<https://classteaching.wordpress.com/2019/09/26/on-learning-how-to-solve-the-rubiks-cube/>.

Image of mixed Rubik's Cube for demonstrative purposes.

“Image of Rubik's Cube notation.” *How to Solver the Rubik's Cube*,
<https://thatspeedcubergirl.home.blog/2019/08/06/how-to-solve-the-rubiks-cube/>.

Image of Rubik's Cube notation for demonstrative purposes.

Joyner, David. *Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys*. Baltimore: Johns Hopkins University Press, 2008.

<http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3318460>.

This book was created by a math professor who wanted to teach group theory and discrete mathematics to students in a way that was fun and "irresistible," as he puts it. He saw the Rubik's Cube as a great way to get students interested in the mathematics behind these sorts of games and toys. The book has three chapters of mathematical background on discrete math and group theory, which includes topics such as sets, cardinality, symmetry, functions, and much more. It also gives standard notation for studying the mathematics of the Rubik's Cube. Though this book has a great background in math, that will not be the main use of the information presented. The project I am creating is a visual teaching tool for the same topics in the book, so this book is a great pedagogical source. The author outlines which chapters are less advanced than others and states that some of the concepts can be understood by high school students.

Pinter, Charles C. *Book of Abstract Algebra*. Place of publication not identified: McGraw-Hill, 2003.

This book is a commonly used textbook in Group Theory classes and is the textbook I used in my MTH344 class last term. This book gives the mathematical background information that I will need to add concepts to my project. When curious about a certain topic or when a concept is not clear, this book provides that clarity. Most Group Theory definitions come from this book.

Reas, Casey. *Getting Started with Processing: A Hands-on Introduction to Making Interactive Graphics*. Place of publication not identified: O'Reilly Media, 2015.

The language and environment that I will be using for this project is Processing and Processing-Java. It is a sketchbook software and language for coding within the context of visual arts. I don't have a lot of experience in visual arts, but I have used Processing previously, so it was the best option. This will be used mostly for the visual effects and features included in this language that is not universal to other languages.

Rokicki, Tomas, Herbert Kociemba, Morley Davidson, and John Dethridge. “The Diameter of the Rubik’s Cube Group Is Twenty.” *SIAM Journal on Discrete Mathematics* 27, no. 2 (January 2013): 1082–1105. <https://doi.org/10.1137/120867366>.

The title of this article refers to the fact that a Rubik's Cube can be solved in 20 moves from any position. The paper gives evidence and source code as to how they proved this fact and their creation of an online solver. This would be a great source if, in the future, I am able to impalement a solver. The paper gives a lot of information and statistics for the math and testing of the solver. Their testing methods can potentially be used to test my solver and I can compare results to evaluate the efficiency and method.

Shiffman, Daniel. “Rubik's Cube Part 1-3.” *The Coding Train*, 18 Apr. 2019, thecodingtrain.com/CodingChallenges/142.3-rubiks-cube.html.

This is the series of videos I used to create the base of the Rubik’s Cube. Shiffman goes into great detail in an informal challenge type video on how to create a cube in the Processing-Java environment. There are three videos in the series each going through their own problems as he figures out the best way to create this simulation. This was very beneficial to watch and learn from as my code does something very similar to his. My code and his code are not identical, though mine is very much inspired by his.