

2-2019

# Rubik's Cube: A Visual and Tactile Learning of Algorithms and Patterns

Lawrence Muller

*CUNY La Guardia Community College*

## How does access to this work benefit you? Let us know!

Follow this and additional works at: [https://academicworks.cuny.edu/lg\\_oers](https://academicworks.cuny.edu/lg_oers)

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

---

### Recommended Citation

Muller, Lawrence, "Rubik's Cube: A Visual and Tactile Learning of Algorithms and Patterns" (2019). *CUNY Academic Works*.  
[https://academicworks.cuny.edu/lg\\_oers/70](https://academicworks.cuny.edu/lg_oers/70)

This Activity or Lab is brought to you for free and open access by the LaGuardia Community College at CUNY Academic Works. It has been accepted for inclusion in Open Educational Resources by an authorized administrator of CUNY Academic Works. For more information, please contact [AcademicWorks@cuny.edu](mailto:AcademicWorks@cuny.edu).

## Abstract

This is a classroom activity report on teaching algorithms as part of a second course in computer programming. Teaching an algorithm in an introductory level programming class is often a dry task for the instructor and the rewards for the student are abstract. To make the learning of algorithms and software more rewarding, this assignment employs a Rubik's cube [1] to: (a) provide immediate and concrete results for all the work, (b) build soft skills (e.g., team work), (c) support "learning how to Learn hard subjects' [2].

## Introduction

Traditional programming is taught in an environment with no graphical or physical computing component (e.g., no robotics) and software or algorithms cannot be seen or felt. The wider acceptance of robotics, gaming, mobile-apps, and webcams into introductory programming and engineering classes has provided more exciting lab material for students and faculty. Like any craft, proficiency and confidence are developed by repetition, correction, and encouragement; to that end, students, as always, are given small programming assignments that advance features of a programming language to solve problems.

In a world such as the fine arts, end results are often tangible exhibits that can be critiqued and admired by a non-artist. This is in stark contrast to many introductory programming assignments which are often [too] small in scope and, to a wide extent, uninteresting to all but a few; and the end work, a program, is a non-tangible abstraction with little, if any, esthetic value; thus, when someone says, I program, that means I work on finding patterns and then try to exploit those patterns as an algorithm expressed in a precise machine understandable language: not very engaging.

Educators have been experimenting with hi-tech "toys" [3], and normally these toys cannot leave the class. This project employs the Rubik's cube as a means to explore learning an algorithm to solve a popular and difficult problem. Students are encouraged to learn the algorithm outside of the formal class. It should be noted that this learning modality follows the flipped classroom, and more closely models the real-world; that is, there is no instructor to help them and they must learn to learn on their own. It is very difficult to solve in one sitting, even when following the algorithm. There are a number of algorithms that can be used as a guide to solve the Rubik's cube. Students are given a starting point in [learn how to solve a 3x3 Rubric's cube](#)[9]. The cube also has a social component [4,5,6,7] and an esthetic component----it's structure made of smaller cubes of six (6) different colors that can rotate about the shape.[11]

### Procedure

The Instructor can lead by demonstrating the solution to the Rubik's cube [7]. The instructor does not need to learn to solve other cubes but it is helpful. I recommend looking at the 2x2 cube[10] because its solution is a subset of the 3x3 cube[9]. If, however, the instructor does not know the solution, it would be interesting to learn alongside the students. This could provide insight into their progress and setbacks, as well as remind the instructor about what it was like as a student.

### Activity

1. Assign the Rubik's cube project, demonstrate a solution[8], and advise students of the various resources that are available [1, 9, 13];
2. Advise the students that they are to bring their cube to class to practice and [attempt] to meet milestones and make improvements. Let them know that active participation [2, 12, 13, 13, 14, 15, 16] will likely help them on the final grading of the cube project.
3. Provide motivation by discussing the need for: team work, setting goals, accepting setbacks, reflection, avoiding negative thinking (e.g., old saying: "as a man thinketh in his heart, so is he" [17]) and getting rest from the problem, for example [2,13,14,15,16];
4. Encourage students to:
  - a. Devise and set-up sub goals---instead of memorizing a sequence of 60 moves and patterns in a single attempt, learn in [personalized] steps [2, 12, 14];
  - b. Keep notes, for example: list videos and websites that they found helpful;
  - c. Keep a brief journal where they reflect on the current status of the work, e.g.:
    - i. Frustrated, I took the cube apart and reassembled it;
    - ii. Stuck on step 3, will take a break for a while
    - iii. Looked easy will clearly need more time....
    - iv. I got, now to practice (e.g., repeat).
    - v. This website description is the best one for step 4.
  - d. Work and share with other students on learning the solution (collaboration and community building) [15,16];
5. As a prelude to in-class work on the cube, consider starting a brief discussion or show a video during the semester about the cube, such as:
  - a. History [2]
  - b. Speedcubers, Robotic solvers, Magic tricks [4, 5, 6, 7]
  - c. The notational language and pattern recognition [1, 9]
  - d. Mathematical permutations (any feeling for 43 quintillion ?) [2]
  - e. Brute force solution

**Grading**

The project was graded on a pre-determined date for a timed test. The objective was for each student to show they can solve a 3x3x3 Rubik's cube that has been scrambled (by the instructor) in under 5 minutes.

**Results**

I have run an advanced C++ class twice using a 3x3 Rubik's cube project. The results are tabled below. A few students in each class solved the cube in less than a minute, many in less than 3 minutes of those who succeeded on the first attempt. There were those one-two students per class who solved the cube in High School and were delighted that their developed skill finally paid off. In both classes there were students that did not complete the assignment in under 5 minutes, some took as long as 10 minutes while getting help (still counted as a success).

C++ (given a free cube)	
Under 5 minutes	Condition(s)
70%	First attempt
20%	2-3 attempts
6%	4 or more attempts
Over 5 minutes	
4%	needed help.
Student's own assessment of project	
70%	Useful-encouraging
30%	Waste of time

C++ (purchased their own cube)	
Under 5 minutes	Condition(s)
80%	First attempt
15%	2-3 attempts
4%	4 or more attempts
Over 5 minutes	
1%	needed help.
Student's own assessment of project	
90%	Useful-encouraging
10%	Waste of time

**Conclusions**

While the sample is small it indicates:

- Students should likely purchase their own cubes;
- the majority found it useful and/or encouraging;

**Problems**

Some students purchased very poor-quality cubes (e.g., very hard to rotate) making learning physically annoying (in my own words); I recommend students be forewarned.

Rubik's Cube: A Visual and Tactile Learning of algorithms and patterns  
Lawrence Muller, Ph.D. LAGCC MEC Dept.

**Future** considerations/comments:

Have students make this into a software research project, where, aside from solving the cube by hand they need to produce:

1. The History of 3x3 Rubik's Cube;
2. What is the descriptive language that is employed to work with a Cube;
3. Discuss the mathematics involved and a programmatic solution based on brute force computation (link with discrete math?);
4. Explore how algorithms are programmed to manipulate a cube toward a solution.

The aforementioned list maybe too much for the MAC-125 class, but maybe as group projects? Also, consider a tiered reward system for learning the algorithm for the cube.

**References**

1. The Rubik's cube, [https://en.wikipedia.org/wiki/Rubik%27s\\_Cube](https://en.wikipedia.org/wiki/Rubik%27s_Cube)
2. Learn How to Learn, <https://www.coursera.org/learn/learning-how-to-learn>
3. S. McKnight, G. Tadmor, M. Ruane, R. Rodriguez-Solis and G. Saulnier, "Creating "High-Tech Tools and Toys Lab," 32nd Annual Frontiers in Education, 2002, pp. S3E-7 vol.3
4. Brundage Magic, [https://www.youtube.com/watch?v=Qph0\\_pBUer8](https://www.youtube.com/watch?v=Qph0_pBUer8)
5. America's Got Talent, [https://www.youtube.com/watch?v=zgK\\_fg1cXvM](https://www.youtube.com/watch?v=zgK_fg1cXvM)
6. Rubik's Cube solving Robot, <https://www.youtube.com/watch?v=ixTddQQ2Hs4>
7. Mr. Lucas Etter's world record [https://www.youtube.com/watch?v=msb4Jzl\\_6Jo](https://www.youtube.com/watch?v=msb4Jzl_6Jo)
8. N.b.: I have not set the assignment up with the instructor (me) not being able to solve the cube, nor attempting to lock step the learning process of the class ...
9. Official Solve 3x3 cube, <https://www.rubiks.com/solve-it/3x3>
10. Official Solve 2x2 cube, <https://www.rubiks.com/solve-it/2x2/>
11. Inside a Rubik's Cube, <https://www.youtube.com/watch?v=bgcScY7CiMs>
12. MOOC Problem Solving course <https://www.coursera.org/learn/problem-solving>
13. Educator's Cube Website <https://www.youcandothecube.com/>
14. George Polya, "How to Solve It" Princeton University Press.
15. J. Bevan, L. Werner and C. McDowell, "Guidelines for the use of pair programming in a freshman programming class," *Proceedings 15th Conference on Software Engineering Education and Training (CSEE&T 2002)*, Covington, KY, 2002, pp. 100-107.
16. Paired programming, <http://wiki.c2.com/?PairProgrammingBenefits>
17. King Solomon, ca 920 BCE, Sayings of the wise, proverbs 23:7, KJV.

Note: The use of the cube is not new[13], except that I am reporting its use as part of my classroom activity. This was run again as required projects in an advanced C++ and a course on Discrete Structures, both produced similar results. The initial inspiration came when spending time with my nephew, Steven Brundage[5], on a flight to Peru. Along the route, he constantly played with a 3x3 cube... solving it in seconds. I elected to learn the algorithm(s).