

Summary: Create a solvable Rubik's cube.

Components:

- **Cube:** A standard, six-color, three-by-three-by-three Rubik's cube, modeled in three dimensions.
- **Controller:**
 - **Cube Orientation:** The numpad keys control the orientation of the cube. The cube will be placed in an isometric view, with one face clearly facing the user (the "main face"). Indicators (T for "top," B for "bottom," L for "left," R for "right,") will be placed on the appropriate squares, so that the user quickly understands the orientation direction. Numpad left ("4") rotates the cube so that the left face of the cube (from the previous orientation's perspective) is now the main face of the new orientation. Similar functionality for numpad up ("8"), down ("2") and right ("6"). Numpad CW ("7") rotates the cube clockwise without changing the main face. Numpad CCW ("9") rotates the cube counterclockwise without changing the main face.
 - **Square Selection:** On mouse click release, if the mouse is visually above a square on the main face of the cube, that square will be highlighted. The reasoning for this will be briefly discussed in the "reach goals" section of the proposal.
 - **Row or Column Movement:** If a square on the main face is selected, the arrow keys (or a similarly-structured set of keys, such as WASD) will rotate either the row or the column of the selected square. For example, W will rotate the column of the selected square up, S will rotate the column of the selected square down, A will rotate the row of the selected square left, and D will rotate the row of the selected square right.

Mathematics:

- **3D Rotations:** The cube will need to keep track of the positions of 26 smaller cubes, all of which rotate around a center point, most likely the origin. "Corner" cubes will rotate differently than "side" cubes and "center" cubes will rotate differently than both. Principally this will be trigonometry.

Language(s):

- **VPython:** I intend to create the first iteration of this project in VPython. I believe this will allow me to develop a conceptual understanding of the functionality of a Rubik's cube without needing to delve into graphics at first.
- **OpenGL:** The final version of this project will most likely be done in OpenGL. This should allow me to develop more advanced features such as orientation shifting and mouse controls, as well as give me the opportunity to actually program the 3D graphics for the cube.

Learning Goals:

- **3D Geometry Mathematics:** I have never coded a 3D graphical application before, and learning to apply the appropriate mathematics to accurately represent 3D movement will be new but very doable.
- **VPython:** I have not worked in VPython before but I know that it is a very beginner-friendly language.
- **OpenGL:** I am very interested in learning about more powerful graphics libraries, and OpenGL seems to be a good place to start.
- **Program Structure:** I have limited experience designing architecture for programs with several functional components. I hope that this project will allow me to learn how best to separate components like controllers from 3D models and 2D projections, and furthermore how best to break each of those components into subclasses and/or subroutines.

Reach Goals: If I have the time and skill, I would like to implement...

- **Mouse-Controlled Orientation Shifting:** Allowing the user to click-and-drag in order to change the orientation of the Rubik's cube. This would add complexity to the mouse controller and more mathematics in determining the most appropriate orientation for the cube, given some odd angle. If the mouse-controlled square selection is done using on-mouse-click-release, click-and-drag functionality becomes less cumbersome when considering a user starting their "drag" somewhere visually on the cube.
- **An Automatic Solver:** An algorithm that would simply find and execute an at least moderately efficient solution to the Rubik's cube given its current state.
- **4D:** Writing the mathematics for a 4D Rubik's cube, and projecting it onto the screen seems very difficult and very complicated, but nevertheless something that may be very interesting, especially if time and skill allow.