You are going to *create an interactive Rubik's cube application*. Rubik's cube is a physical, three-dimensional puzzle invented by Ernő Rubik, a Hungarian inventor, in 1974. The cube can be thought of as consisting of 27 sub-cubes, arranged in a 3x3x3 fashion to form a larger cube. You never see the center cube at all, and you typically only see one to three faces of the other cubes. It is possible to rotate any of nine planes consisting of nine cubes each, either clockwise or counter clockwise. Mid-rotation, you see portions of the inner faces of the outer cubes. The cube starts off such that all visible faces that face the same direction have the same color, and each side of the cube has a different color. Typically, one person mixes up the cube arbitrarily, and then another person tries to solve the puzzle by making a series of rotations to return to the cube to its original state. Your application will pseudo-randomly permute the cube on its own, and then allow a user to try to solve the puzzle. The user should be able to specify any of the nine planes and a direction; the specified plane should rotate 90 degrees in the specified direction relative to the rest of the cube. The user should also be able to rotate the entire cube around its own center so that they can view the cube from any desired angle. The two pictures above, found using Google's Image Search, represent the cube in a mixed up position (left) and a cube that is possibly solved (right), although we can't be sure without seeing the rest of it.

Constraints on your programs include:

* At the start of the program, the user should be able to specify an arbitrary number of turns, and the program should randomly make that many moves, starting from the original position.
* The user should be able to rotate the cube as a whole to any desired angle.
* The user should also be able to specify any of the nine planes and a direction, in an intuitive manner, after which the specified plane should rotate 90 degrees in the specified direction. Note that you can decide how you allow the user to interact with the cube. That is, you will decide if the mouse and/or keyboard will be used to specify rotations, and you will decide the details of how the interface works.
* Rotations of the specified plane should be animated. Ideally, these rotations should take approximately the same amount of time, regardless of the speed of the computer. Alternatively, you could make the speed of rotation configurable by the user.
* All rotations should be handled by the vertex shader.
* The program should know when the cube is solved and inform the user.
* The user should be able to save the current state in a specified text file and reload it later. (You may find and use a library to help with this feature.)

Grading will consider the correctness of the implementation, adherence to all of the expressed constraints, the smoothness of the animation, and the ease of use of the interface. You are welcome to add any additional features that you wish, but you will not get extra credit. Also, note that *I am not asking you to write a program that can solve the Rubik's cube*!

There is a lot of flexibility in the assignment, as long as you implement your program by yourself using JavaScript and WebGL. You may not use three.js or any other libraries built on top of WebGL that simplify your implementations. I should be able to run the program using at least one of Firefox, Chrome, or IE on a Windows machine or an Ubuntu virtual machine. When you are finished, you should e-mail me (CarlSable.Cooper@gmail.com) your code. The program is due before midnight on the night of Friday, March 31. Get started early and have fun!