Nicholas Larsen, Steven Larsen

Learned Rubik's Proposal

Final Requirements

To satisfy the final requirements we are going to make a virtual Rubik's cube that is fully functioning. It will have:

- A SCNScene to display the Cube
    - With 27 cube nodes
- It will listen to motion. Flicks of the phone will turn the entire cube about a certain axis
- We will use Machine Learning as a service (or deploy a model) to recognize each of the 12 flicks. This was done in our Lab 5

Base Features

- A screen used to train the flicks, like in Lab 5. But the user can use our loaded or server models as well
- A fully functional easy to manipulate virtual Rubik's Cube.
    - Ironed out Machine Learning Algorithm that detects one of twelve flicks of the phone for the cube rotations
    - Turn any layer using swipe gestures
    - A scramble button in this view
- A view to input any state of a Rubik's cube for solving
- An algorithm that solves a rubiks and walks the user through each turn

Stretch Features

We plan to at least meet one of these requirements but will try for both

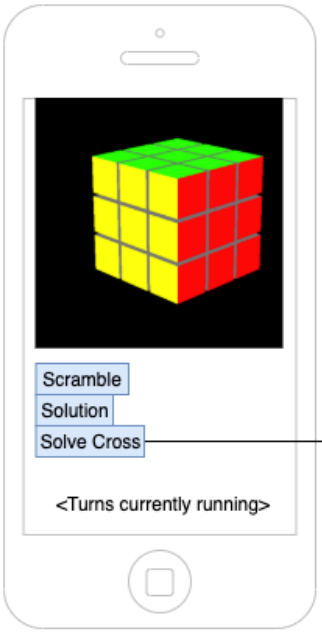
- Allow the user to set the starting point of the cube by taking two pictures of the cube
- Teach a user how to solve a Rubik's cube by walking through each of the "Beginner Phases" to solve it: [CubeSkills Beginner's Method Instructions - Gans]

Nicholas Larsen, Steven Larsen

Screens

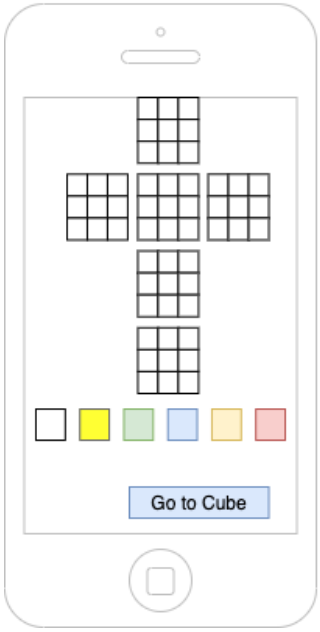
Training Screen

Virtual Cube

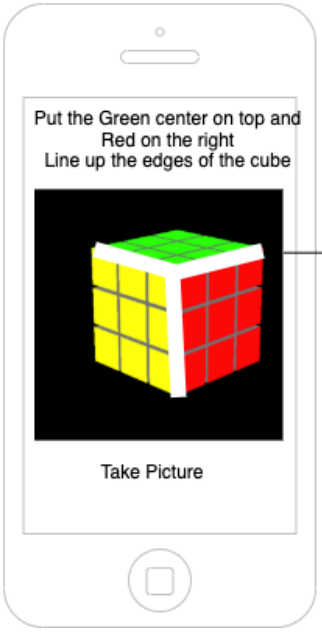


{Solve Cross, Solve Corners, Solve Middle, OLL/PLL }
We are debating between
OLL/PLL and beginner
for the last layer

Manul Input



Camera Input



From the camera