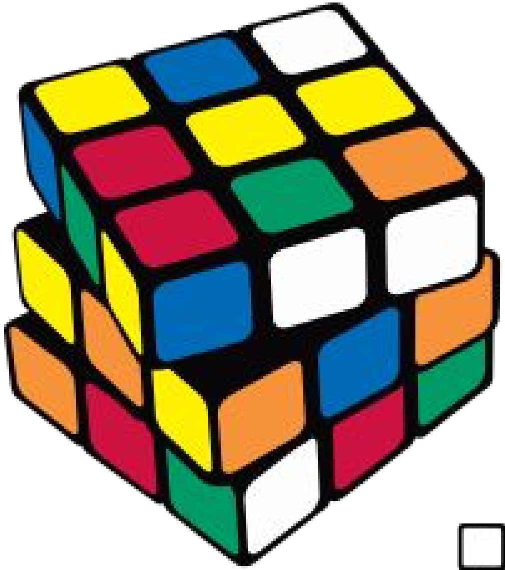
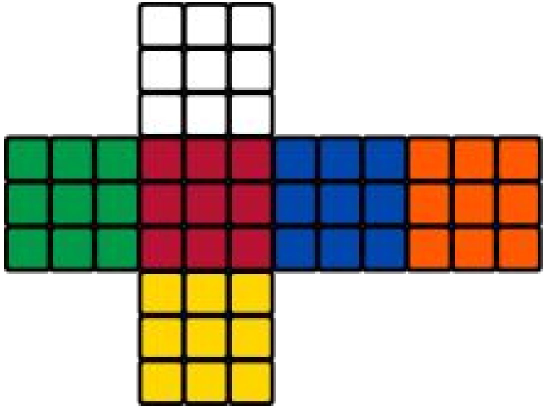
Rubik’s Cube Solving Robot

Nathaniel Knopf
Jacob Swiezy

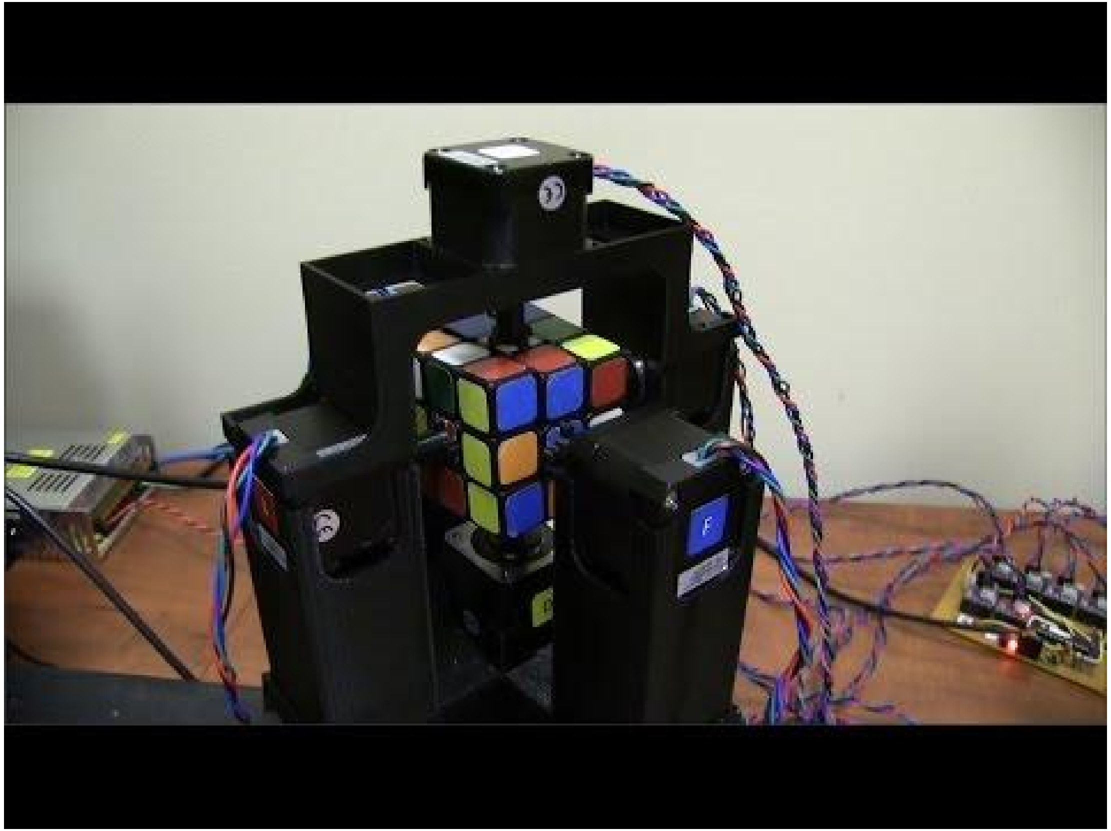
Rubik’s Cube

- 4.3×10^{19} permutations
- World record time: 4.59 seconds
- 3 types of pieces
    - Centers (fixed)
    - Edges
    - Corners

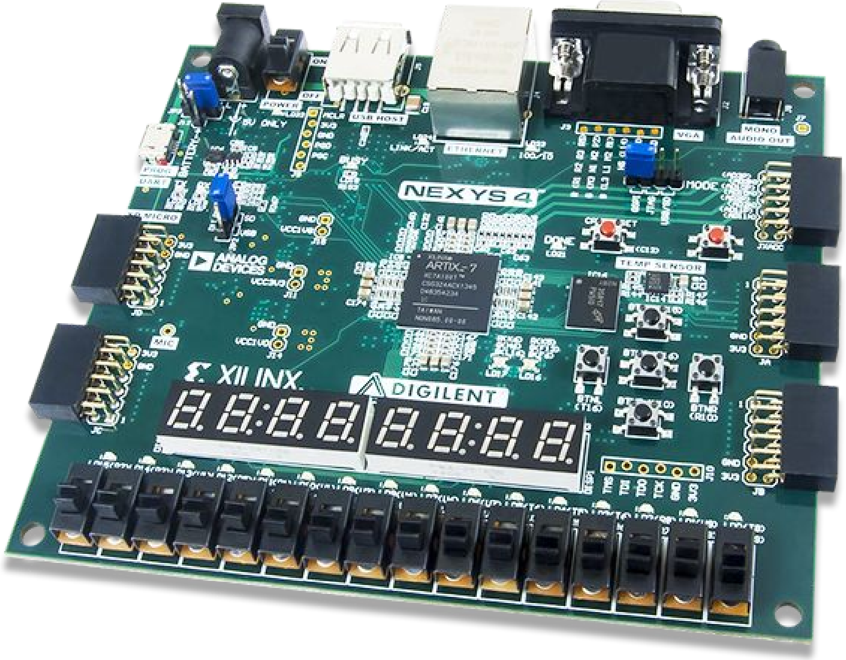




Robots!



Motivation



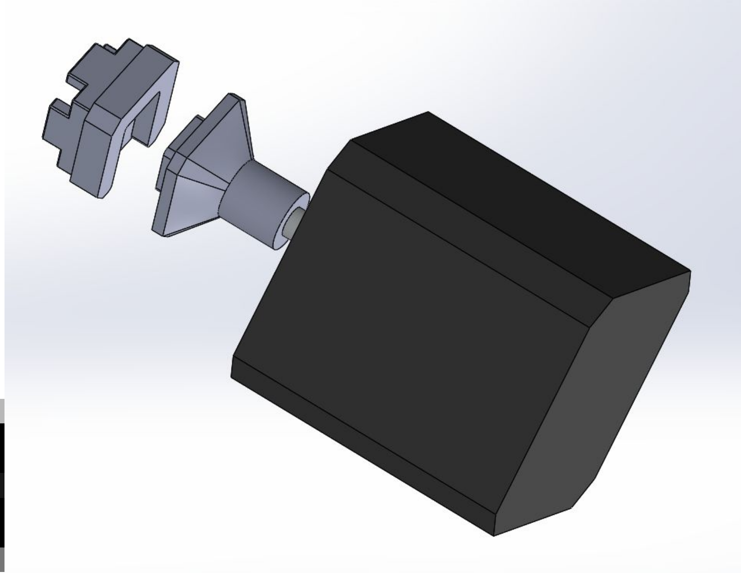
- Robot in previous video: C++ ➔ Arduino ➔ Robot
- Some others implemented entire processor on FPGA, wrote C code
- We'll actually implement the solution on an FPGA

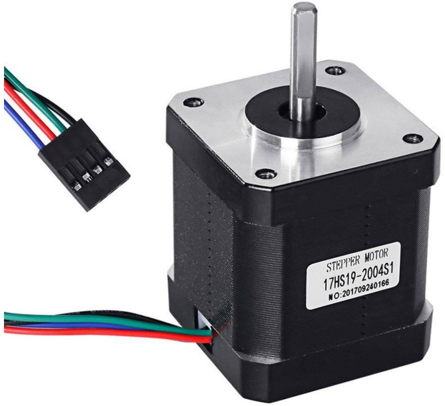
Hardware Components

- Nexys 4
- 6 stepper motors
- 6 stepper motor drivers
- 6 3D printed shaft-cube couplers
- 2 RGB color sensors
- Laser cut acrylic frame
- Rubik’s Cube

Stepper Motors

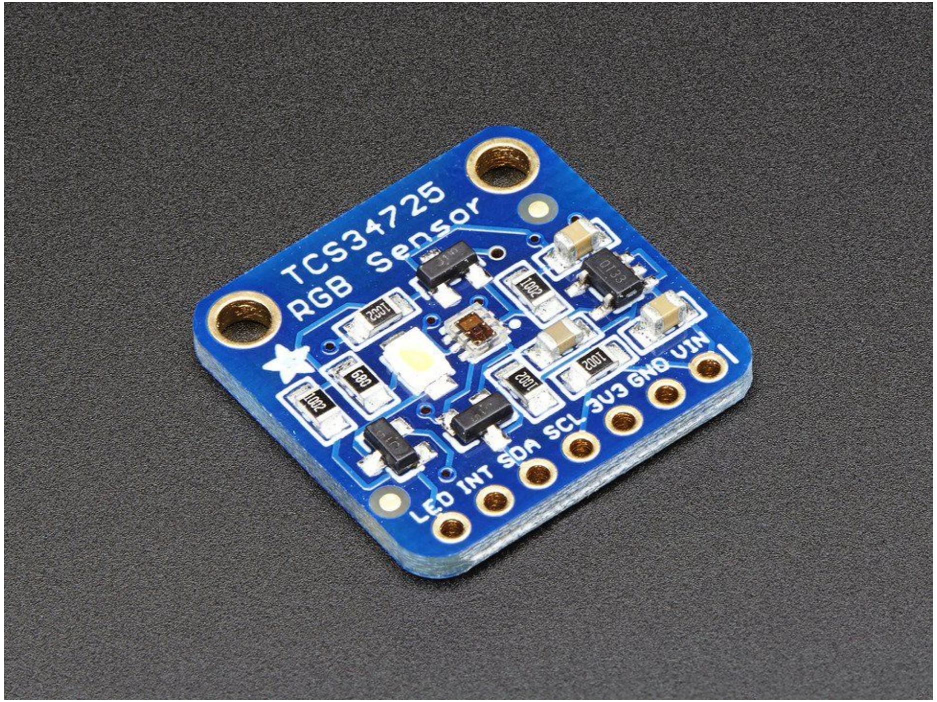
- NEMA 17 Bipolar Stepper Motor
- A4988 Stepper Motor Driver
- 2 Part 3D Printed Shaft-Cube Coupler





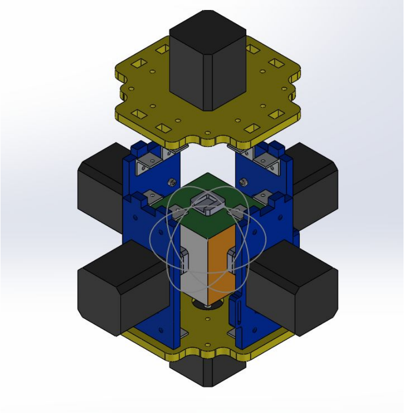
Color Sensor

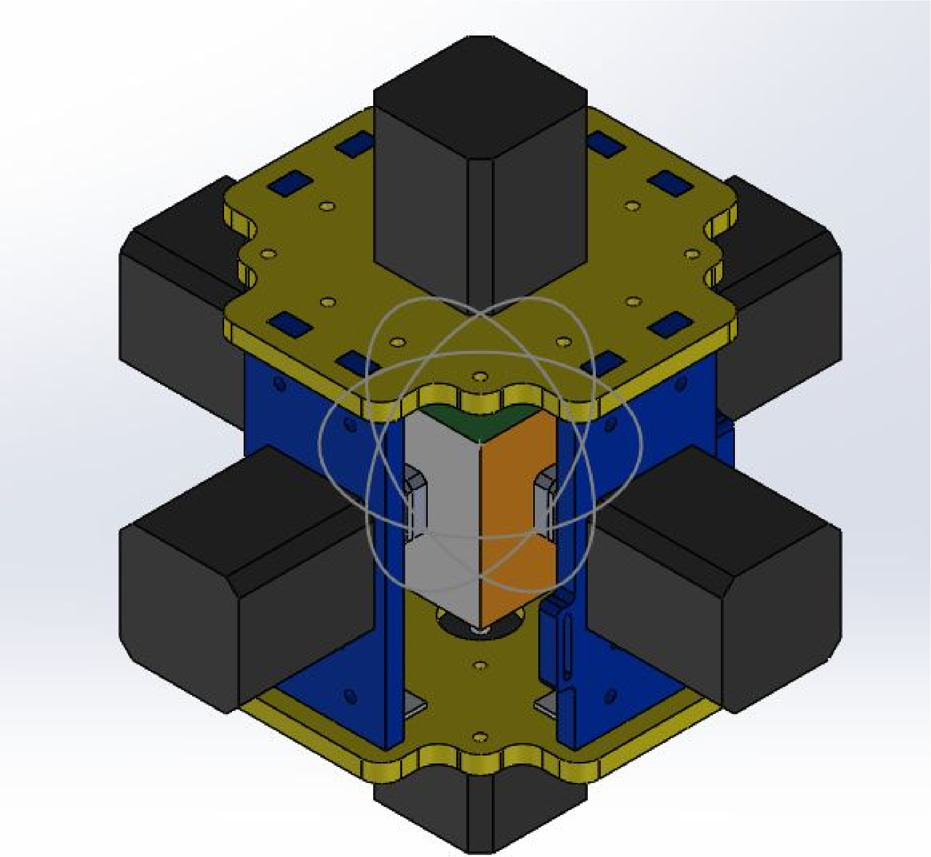
- Communicates over I2C
- Returns 16-bit RGB values
- On-board bright LED for consistent lighting conditions



The Frame

- Laser cut from ¼” acrylic
- Rigidly bolted together
- Removable top allows for cube to be removed and scrambled

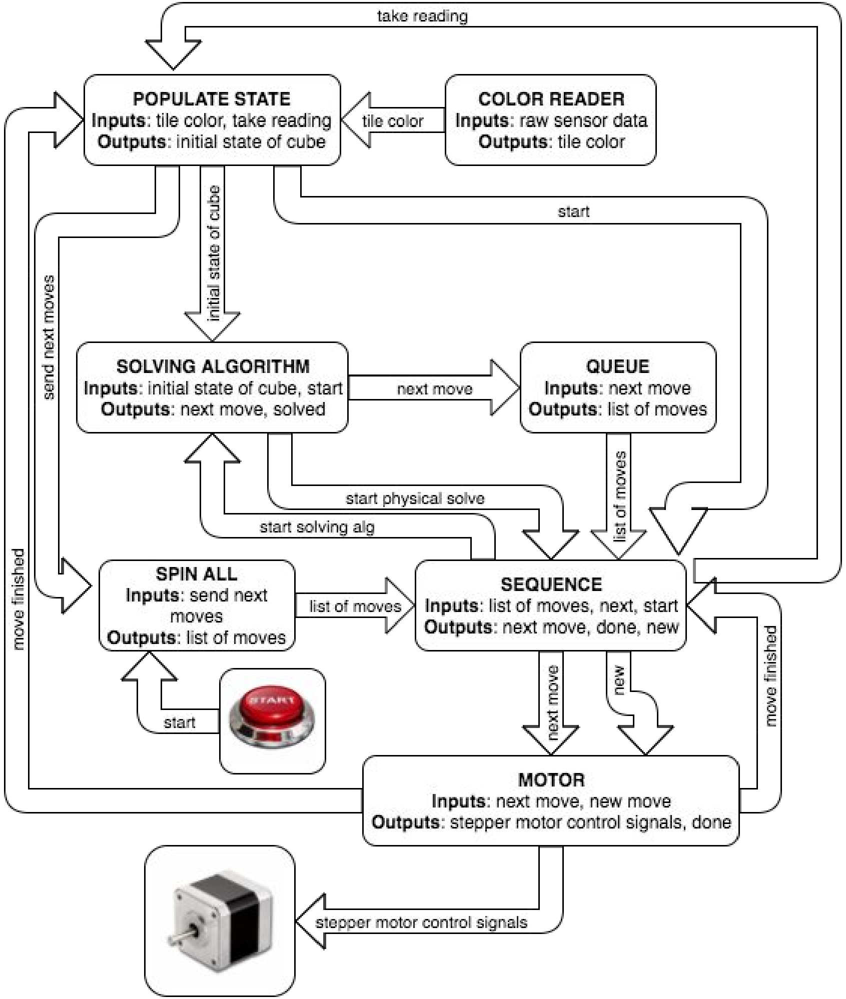




Modules

3 stages

- Determining state of Rubik's Cube
- Planning a solution
- Executing planned solution



POPULATE STATE
Inputs: tile color, take reading
Outputs: initial state of cube

COLOR READER
Inputs: raw sensor data
Outputs: tile color

SOLVING ALGORITHM
Inputs: initial state of cube, start
Outputs: next move, solved

QUEUE
Inputs: next move
Outputs: list of moves

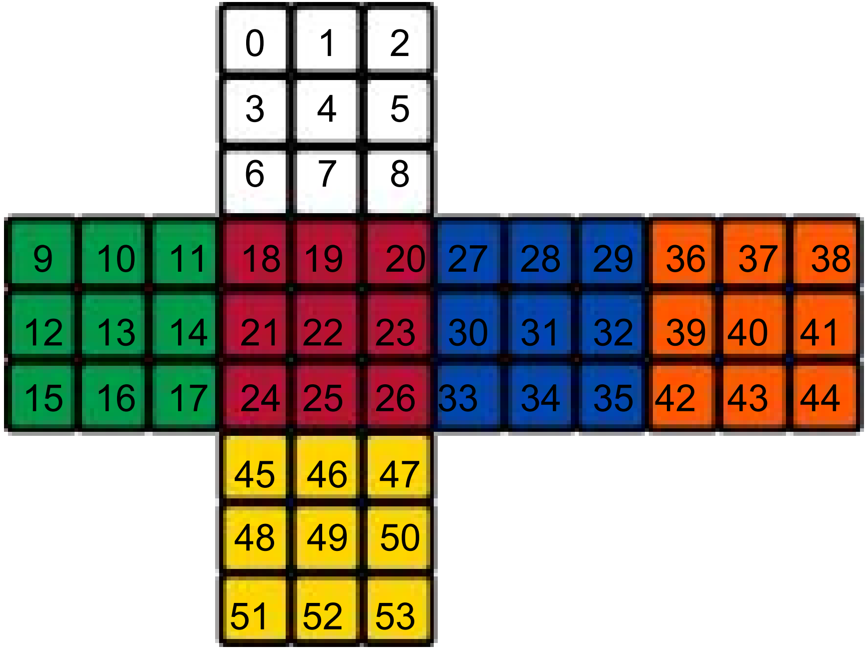
SPIN ALL
Inputs: send next moves
Outputs: list of moves

SEQUENCE
Inputs: list of moves, next, start
Outputs: next move, done, new

MOTOR
Inputs: next move, new move
Outputs: stepper motor control signals, done

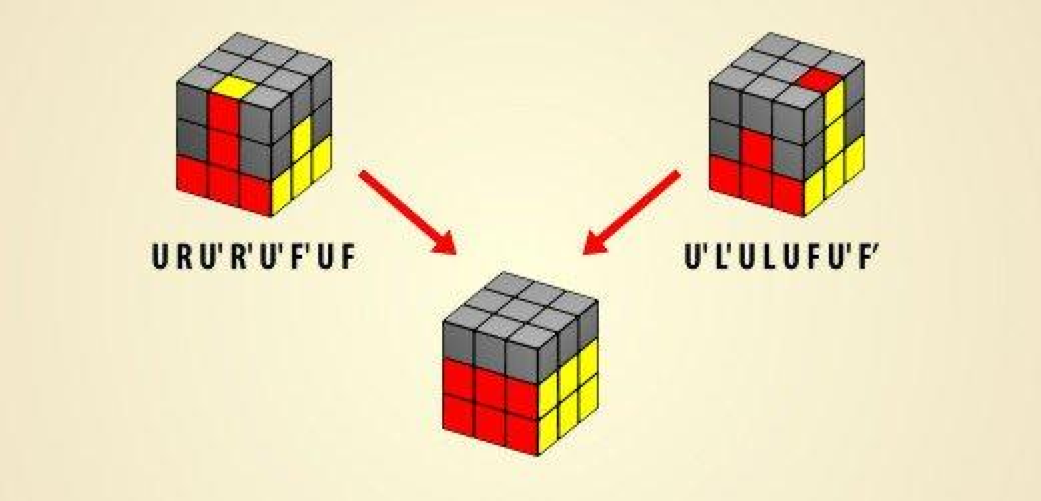
Stage One: Determining State of Rubik’s Cube

- State: 162 bit register
  - 54 pieces * 3 bits for colors (WOGRBY)
- Color Reader module
  - Two color sensors - one for corners, one for edges
  - Color Reader module outputs color of piece it sees
- Spin All module
  - Contains sequences of moves for viewing all pieces
- Populate State module
  - Manages applying moves from Spin All
  - Stores values from Color Reader module in cube state register



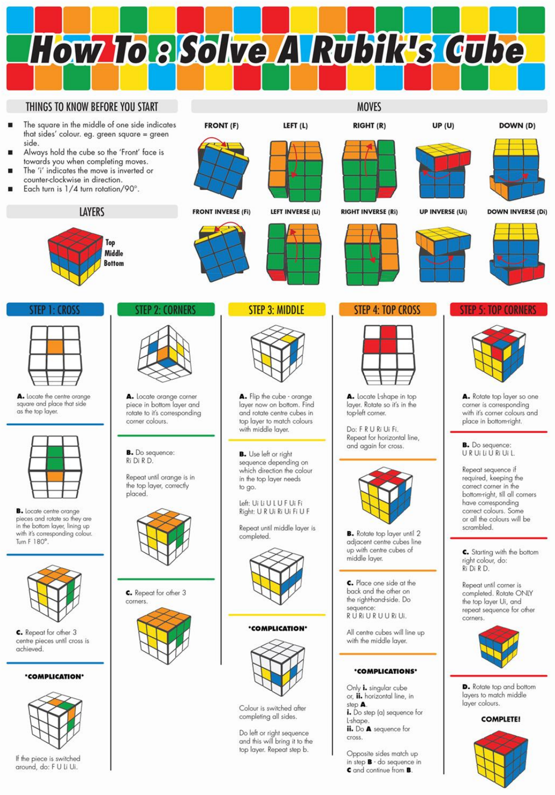
Solving a Rubik's Cube

- Algorithms
  - A sequence of moves to exchange specific pieces of the Rubik's Cube
- Methods
  - Set of algorithms to solve the Rubik's Cube
  - "Beginners Method"



Stage Two: Planning a Solution

- Solving Algorithm module
  - Input: cube state register
  - Output: next move in solving the Rubik's Cube
  - FSM - each state is a step of the method chosen
- Queue module
  - Collects moves output from the Solving Algorithm module
  - Outputs moves to sequence module in combined list
- Sequence module
  - Stores the moves to be executed so Motor module can access them.



How To : Solve A Rubik's Cube

THINGS TO KNOW BEFORE YOU START

- The square in the middle of one side indicates that sides' colour. eg. green square = green side.
- Always hold the cube so the 'Front' face is towards you when completing moves.
- The 'i' indicates the move is inverted or counter-clockwise in direction.
- Each turn is 1/4 turn rotation/90°.

LAYERS

MOVES

STEP 1: CROSS

A. Locate the centre orange square and place that side as the top layer.

B. Locate centre orange pieces and rotate so they are in the bottom layer, lining up with it's corresponding colour. Turn F 180°.

C. Repeat for other 3 centre pieces until cross is achieved.

COMPLICATION

If the piece is switched around, do: F U Li Ui.

STEP 2: CORNERS

A. Locate orange corner piece in bottom layer and rotate to it's corresponding corner colours.

B. Do sequence:
Ri Di R D.

Repeat until orange is in the top layer, correctly placed.

C. Repeat for other 3 corners.

STEP 3: MIDDLE

A. Flip the cube - orange layer now on bottom. Find and rotate centre cubes in top layer to match colours with middle layer.

B. Use left or right sequence depending on which direction the colour in the top layer needs to go.

Left: Ui Li U L U F Ui Fi
Right: U R Ui Ri Ui Fi U F

Repeat until middle layer is completed.

COMPLICATION

Colour is switched after completing all sides.

Do left or right sequence and this will bring it to the top layer. Repeat step b.

STEP 4: TOP CROSS

A. Locate L-shape in top layer. Rotate so it's in the top-left corner.

Do: F R U Ri Ui Fi.
Repeat for horizontal line, and again for cross.

B. Rotate top layer until 2 adjacent centre cubes line up with centre cubes of middle layer.

C. Place one side at the back and the other on the right-hand-side. Do sequence:
R U Ri U R U U Ri Ui.

All centre cubes will line up with the middle layer.

COMPLICATIONS

Only **i.** singular cube or, **ii.** horizontal line, in step **A**.
i. Do step (a) sequence for L-shape.
ii. Do **A** sequence for cross.

Opposite sides match up in step **B** - do sequence in **C** and continue from **B**.

STEP 5: TOP CORNERS

A. Rotate top layer so one corner is corresponding with it's corner colours and place in bottom-right.

B. Do sequence:
U R Ui Li U Ri Ui L.

Repeat sequence if required, keeping the correct corner in the bottom-right, till all corners have corresponding correct colours. Some or all the colours will be scrambled.

C. Starting with the bottom right colour, do:
Ri Di R D.

Repeat until corner is completed. Rotate ONLY the top layer Ui, and repeat sequence for other corners.

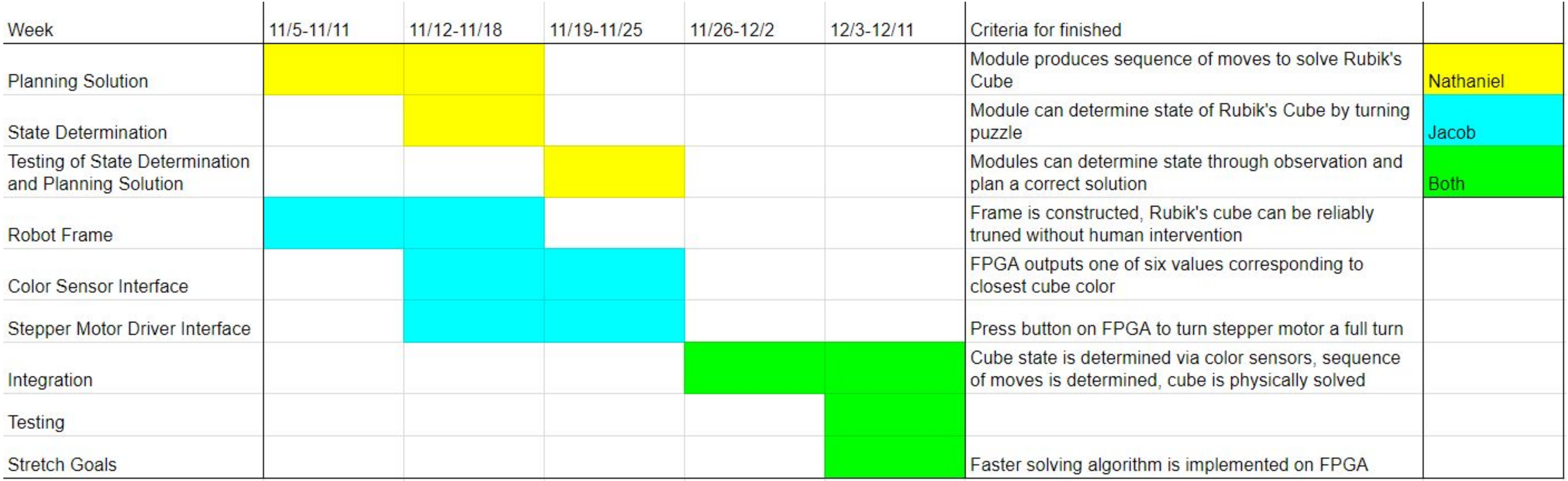
D. Rotate top and bottom layers to match middle layer colours.

COMPLETE!

Stage Three: Executing Planned Solution

- Motor module
  - Pulls moves one-by-one from sequence module
  - 18 possible moves
    - CCW rotation of each face
    - CW rotation of each face
    - 180° rotation of each face
  - Next move translated into appropriate hardware signals for stepper motor drivers
  - Move executed by driving stepper motor driver

Timeline



Week	11/5-11/11	11/12-11/18	11/19-11/25	11/26-12/2	12/3-12/11	Criteria for finished	
Planning Solution	🟨	🟨				Module produces sequence of moves to solve Rubik's Cube	🟨 Nathaniel
State Determination		🟨				Module can determine state of Rubik's Cube by turning puzzle	🟦 Jacob
Testing of State Determination and Planning Solution			🟨			Modules can determine state through observation and plan a correct solution	🟩 Both
Robot Frame	🟦	🟦				Frame is constructed, Rubik's cube can be reliably truned without human intervention	
Color Sensor Interface		🟦	🟦			FPGA outputs one of six values corresponding to closest cube color	
Stepper Motor Driver Interface		🟦	🟦			Press button on FPGA to turn stepper motor a full turn	
Integration				🟩	🟩	Cube state is determined via color sensors, sequence of moves is determined, cube is physically solved	
Testing					🟩		
Stretch Goals					🟩	Faster solving algorithm is implemented on FPGA	

Questions?