Rubik’s Cube Solver: Algorithm Overview

Step 1: Initialize the Cube

- Create a dictionary with 6 faces: U, D, L, R, F, B
- Each face is a 3x3 grid filled with its own color
- Represents the solved cube state

Step 2: Display Menu and Wait for User Input

- Prompt user to enter:
    - S to Scramble
    - V to Solve
    - Q to Quit

Step 3: Scramble Logic (if input is 'S')

1. Generate 20 random moves from valid set (e.g., R, F', U2)
2. Reset cube to solved state
3. Apply scramble moves using apply_move() function
4. Visualize the scrambled cube using matplotlib

Step 4: Solve Logic (if input is 'V')

1. Convert current cube state to a 54-character facelet string
2. Use kociemba.solve(facelet_string) to compute solution
3. Apply returned moves to the cube
4. Visualize the solved cube

Step 5: Quit Logic (if input is 'Q')

- Exit the program

Loop

- Return to Step 2 after any operation unless 'Q' is chosen

Error Handling

- For invalid input: print an error message and re-display menu
- For invalid cube state: catch and display kociemba errors, return to menu