

Rubik's Cube

Alessandro Paglialunga 1946185

September 25, 2022

Abstract

Final project for Interactive Graphics course
held by Prof. Marco Schaerf.

1 Introduction

In this project digital version of Rubik's Cube has been created using: HyperText Markup Language (or HTML), Cascading Style Sheets (or CSS), JavaScript, OpenGL Shading Language (or GLSL) and WebGL libraries.

User can interact with web page through mouse to change: Cube orientation, Cube configuration, difficulty, velocity and music volume.

No animation or hierarchical models have been imported. All downloaded elements have been highlight with their credits. In the following report I will explain software design.

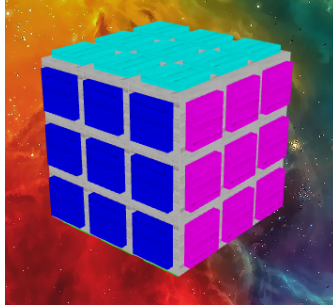


Figure 1: Rubik's Cube

2 How to read code

It has been written using Java style: all code which does not affect the markup or shader part is inside classes.

RubikGame is main class from which all objects are instantiated such as *RubikCube*, *Handler*, *Camera*, carry on game time and manages HTML elements.

From *RubikGame*'s constructor initialization flow can be followed:

1. Initialize canvas environment variables and vectors that will be passed to shaders;
2. Create the *RubikCube* object which is made up of 27 sub-cubes;
3. Set camera, which will be used to give user the possibility to control entire Cube structure;
4. Operate on the style and functionality of HTML elements;
5. Create an event handler that can interact with user interactions;

3 Rubik's Cube class

This class, along with *Subcube*, is the main part of the project.

The first class deals with data structures that manage cube structure from a virtual point of view (such as the sub cubes face that will be visible by user) and from the point of view of rendering or the position of the various sub-cubes.

As for face cube rotation, each sub-cube has the characteristic of belonging to 2 or 3 faces, making it both independent and dependent on the structure. For this reason, the management of face rotation is left to the *RubikCube* class while the actual displacement of the sub-cube and the consecutive modification of sub-cube orientation are left to the *Subecube* class.

4 Camera

The camera can be managed using mouse. Keeping the left button pressed it is possible to rotate the cube around the Y axis and the X axis. However, the upper and lower faces of the cube never exchange.

5 Handler

This event handler allows user interactions, ensuring continuity in the game.

User can:

1. Modify the difficulty of the game or increase the number of initial random rotations;
2. Modify the rotation speed cube faces;

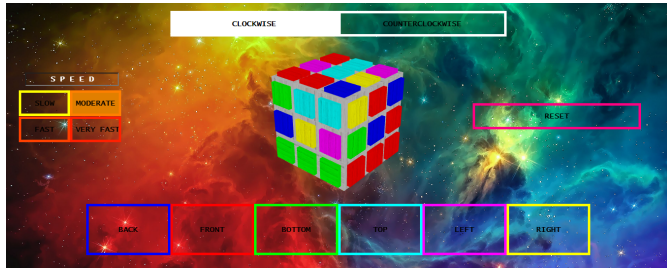


Figure 2: Playground.

3. which face of the cube to rotate according to the color of the central sub-cube;
4. Vary rotation direction of cube faces;
5. Start or reset game;
6. Activate or deactivate background music.