TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**Môn học: Đồ họa máy tính  
Lớp 60TH4**

**Đề tài: Mô phỏng hoạt động của khối rubik**

**Nhóm sinh viên : Nguyễn Xuân Quý- 1851061516**

**: Nguyễn Văn An – 1851061402**

**Hà nội, 2021**

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc74963411)

[**1. Giới thiệu** 3](#_Toc74963412)

[**2. Công nghệ sử dụng.** 3](#_Toc74963413)

[**I. Nội dung** 4](#_Toc74963414)

[**A. Phát biểu bài toán** 4](#_Toc74963415)

[ Tổng quan Rubik: 4](#_Toc74963416)

[ Yêu cầu bài cần có 5](#_Toc74963417)

[**B. Các bước giải quyết** 5](#_Toc74963418)

[1. Tạo khối rubik 5](#_Toc74963419)

[2. Tạo trục tọa độ 8](#_Toc74963420)

[3. Rubik được chiếu sáng. 9](#_Toc74963421)

[4. Hiển thị ra màn hình 10](#_Toc74963422)

[5. Có thể nhìn được xung quanh khối rubik. 11](#_Toc74963423)

[6. Các mặt rubik có thể xoay được và các mặt được xoay phải hiển thị đúng màu (giống một khối Rubik thật). 12](#_Toc74963424)

[**C. Kết quả thực nghiệm** 16](#_Toc74963425)

[1. Hiển thị khối rubik và 3 trục tọa độ 16](#_Toc74963426)

[2. Xoay khối rubik 16](#_Toc74963427)

[3. Xoay các mặt xung quanh khối rubik 17](#_Toc74963428)

[**II. Kết Luận** 18](#_Toc74963429)

[**III. Tài liệu tham khảo** 19](#_Toc74963430)

# **MỞ ĐẦU**

**1. Giới thiệu**

Trong những năm trở lại đây, với sự phát triển mạnh mẽ của công nghệ thông tin. Với việc hỗ trợ của công nghệ thông tin thì mọi công việc đều có thể giải quyết một cách dễ dạng và thuận tiện hơn.

Đồ họa máy tính là một trong các lĩnh vực mà ngành tin học đang rất quan tâm. Xuất phát từ nhu cầu đồ họa thực tế, nhóm chúng em đã học tập, phân tích, thực hành và làm hoàn thiện chương trình với đề tài: “Mô phỏng hoạt động của khối rubik”.

**2. Công nghệ sử dụng.**

Công cụ lập trình: Visual studio 2019 và Dev c++

Ngôn ngữ lập trình: C++ và thư viện đồ họa OpengGL.

Hệ điều hành: Windown 10.

# **I. Nội dung**

## **A. Phát biểu bài toán**

* Tổng quan Rubik:

+ Một khối Rubik 3x3 có 6 mặt: Trắng, Vàng, Đỏ, Cam, xanh lá, xanh dương; gồm 26 viên ghép lại với nhau trong đó có:

* 6 viên tâm: mỗi viên chỉ có 1 mặt màu, dù bạn quay như thế nào đi nữa thì vị trí của các viên này đều không thay đổi. Như vậy, màu của một viên tâm ở một mặt nào đó cũng chính là màu của cả mặt đó.



* 12 viên cạnh: mỗi viên có 2 mặt màu. Các viên này nằm giữa các cạnh của khối Rubik.



* 8 viên góc: mỗi viên có 3 mặt màu. Các viên nằm ở các góc của khối Rubik.



+Ta luôn có:

* Trắng đối diện với Vàng.
* Cam đối diện với Đỏ.
* Lục đối diện với Lam.
* Yêu cầu bài cần có

1. Tạo một một khối rubik 3x3 có 6 mặt hiển thị ra màn hình, mỗi mặt rubik 1 màu (Trắng, Vàng, Đỏ, Cam, xanh lá, xanh dương)

2. Tạo trục tọa độ

3. Rubik được chiếu sáng.

4. Hiển thị ra màn hình

5. Có thể nhìn được xung quanh khối rubik.

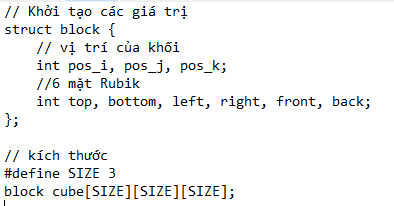
6. Các mặt rubik có thể xoay được và các mặt được xoay phải hiển thị đúng màu (giống một khối Rubik thật).

**B. Các bước giải quyết**

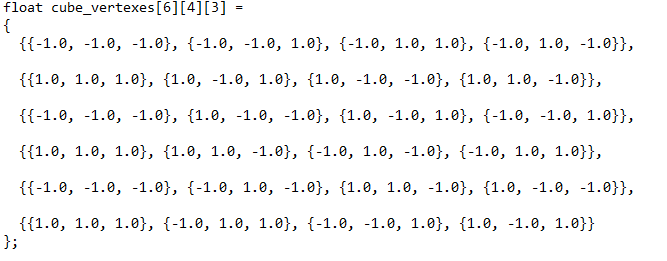
1. Tạo khối rubik

- Để tạo một khối Rubik ta sẽ tạo 27 khối lập phương nhỏ (khối cube)

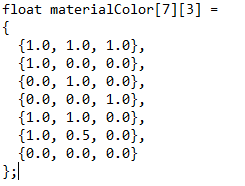
* Khởi các thuộc tính của khối cube và khởi tạo ma trận để chứa các khối cube



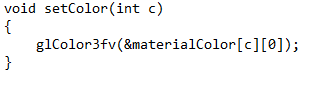
* Đầu tiên tạo 1 khối lập phương thì ta cần tạo 6 mặt mà một mặt có 4 điểm, mỗi điểm sẽ có các vị x,y,z tương ứng. Còn các khối còn ta dùng 3 lần lặp for để tạo và tinh tiến chúng đi. Ta thiết lập ma trận lưu trữ các điểm vẽ của khối cube.

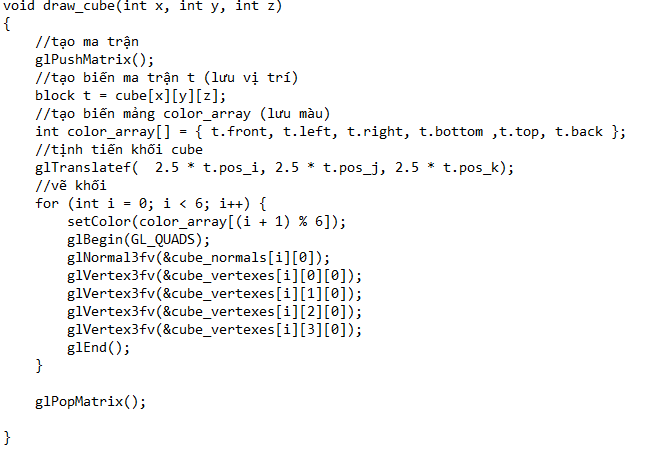


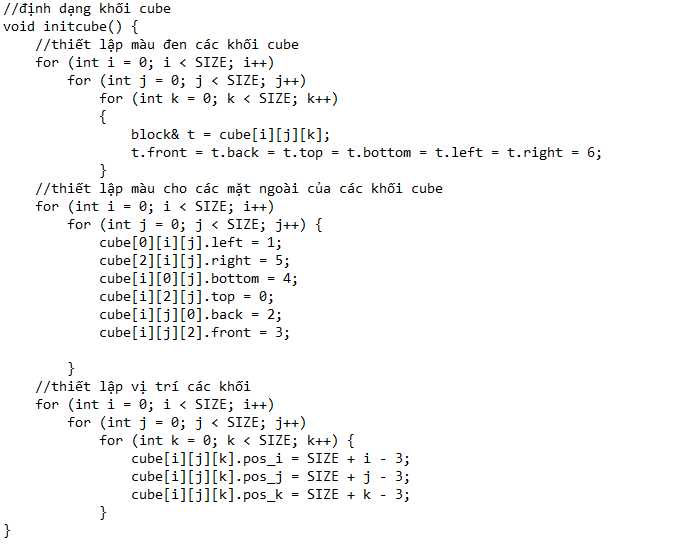
* Thiết lập ma trận lưu trữ các điểm màu vẽ các mặt.



* Thiết lập hàm tô màu bằng hàm lưu trữ các điểm màu vẽ các mặt materialColor

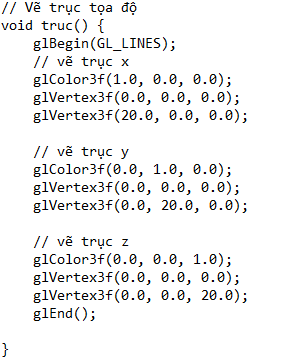


* Vẽ Khối cube bằng cách gọi các điểm vẽ của hàm cube\_vertexes và vẽ màu bằng hàm setColor.
* Định dạng lại các khối cube đã được vẽ



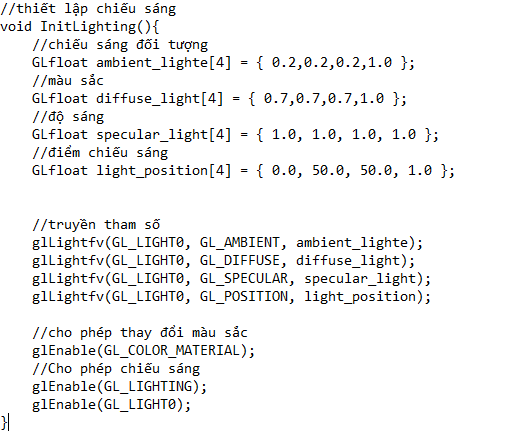
2. Tạo trục tọa độ

- Vẽ trục tọa độ sử dụng chế độ vẽ GL\_LINES, sau đấy thiết lập màu cho đường thẳng bằng cách sử dụng glColor3f, vẽ đường thằng bằng cách tạo 2 điểm dùng glVertex3f



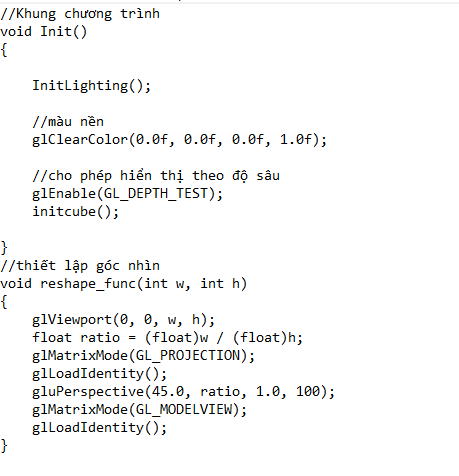
3. Rubik được chiếu sáng.

- Tạo hàm thiết lập chiếu sáng bằng cách dùng glLightfv

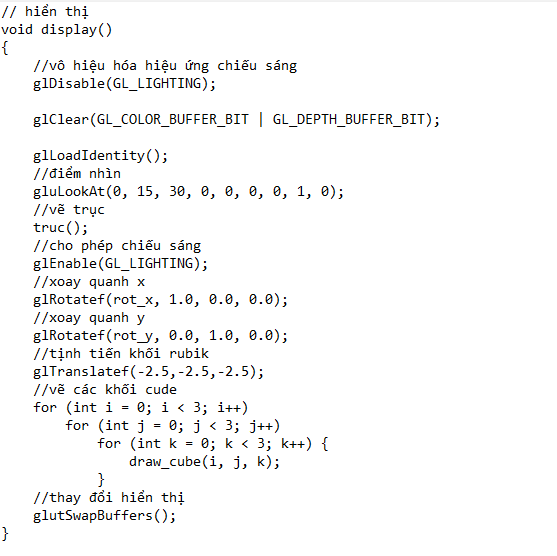


4. Hiển thị ra màn hình

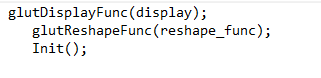
- Ta thiết lập khung chương trình dùng hàm init và góc nhìn dùng hàm reshape\_func



* Tạo hàm display hiển thị đối tượng ra màn hình



* Ở trong hàm main thì gọi ra 3 hàm nêu ở trên ra

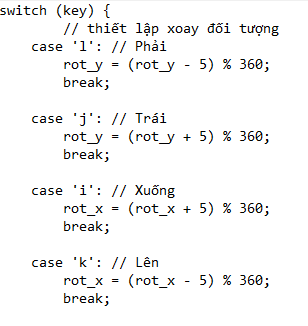


5. Có thể nhìn được xung quanh khối rubik.

- Thiết lập 2 biến xoay, biến rot\_x để xoay rubik quanh x, rot\_y để xoay quanh y



* Thiết lập sự kiên nhập bàn phím bằng cách sử dụng hàm keyboard\_func(unsigned char key, int x, int y) để điểu chỉnh góc xoay

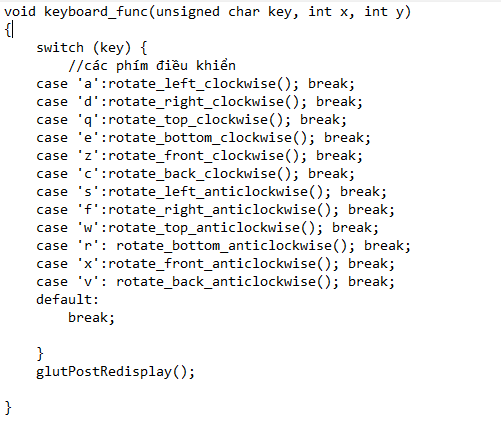


* Ở hàm main thì gọi hàm ra

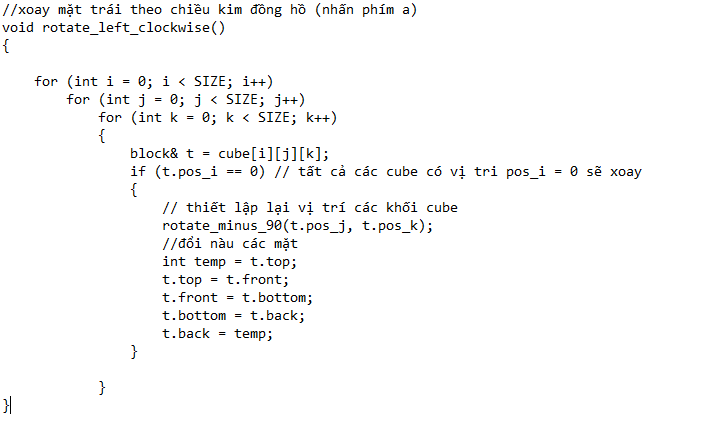
glutKeyboardFunc(keyboard\_func)

6. Các mặt rubik có thể xoay được và các mặt được xoay phải hiển thị đúng màu (giống một khối Rubik thật).

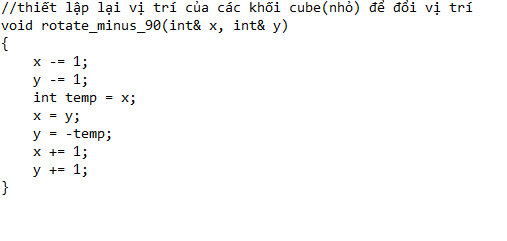
- Thiết lập sự kiện gõ bàn phím để thay đổi vị trí và màu các khối cube của mặt được xét



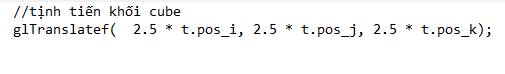
* Mỗi sự kiện các phím nhấn (q, w, e, r, a, s, d, f, z, x, c, v) thì gọi đến các hàm tương ứng. Ví dụ lấy hàm sự kiện nhấn phím a và s. Thì sẽ gọi đến 2 hàm là rotate\_left\_clockwise(); rotate\_left\_anticlockwise();
* Đối với hàm rotate\_left \_clockwise() là quay mặt trái cùng chiều kim đồng hồ



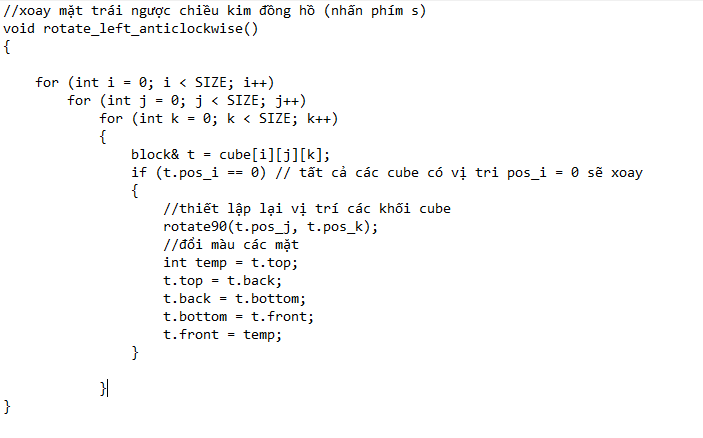
* Sau khi gọi hàm rotate\_left \_clockwise() , thì trong hàm rotate\_minus\_90 sẽ thiết lập lại vị trí các khối cube theo chiều kim đồng hồ với các khối cube có vị trí pos\_i=0



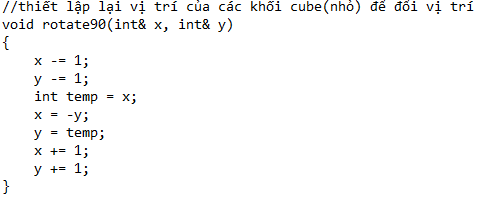
* Ở trong draw\_cube() sẽ tịnh tiến các khối cube với vị trí vừa được thay đổi và vẽ lại các khối cube



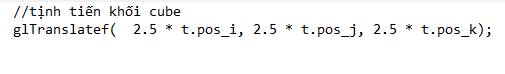
* Đối với hàm rotate\_left \_anticlockwise() là quay mặt trái ngược chiều kim đồng hồ



* Sau khi gọi hàm rotate\_left \_anticlockwise() , thì trong hàm rotate90 sẽ thiết lập lại vị trí các khối cube ngược chiều kim đồng hồ với các khối cube có vị trí pos\_i=0



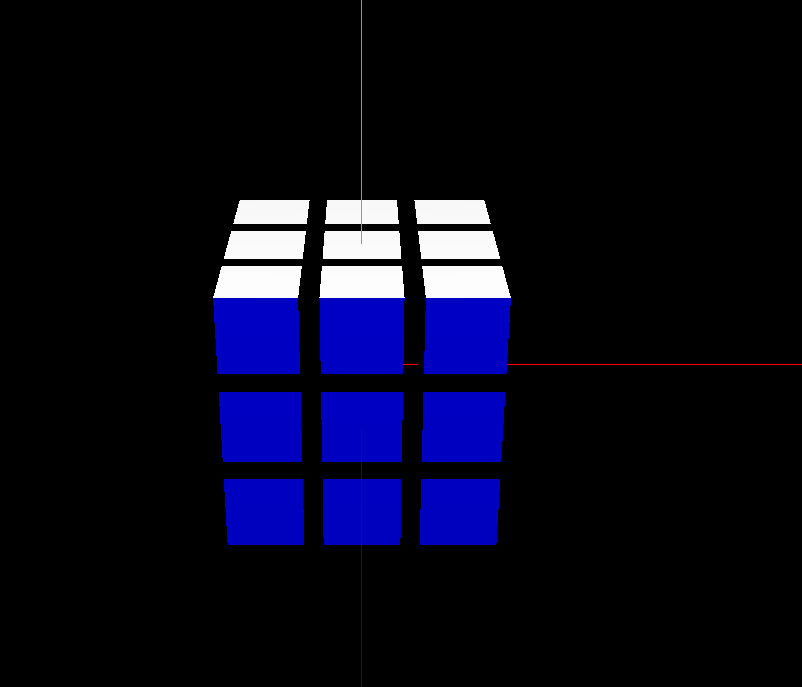
* Ở trong draw\_cube() sẽ tịnh tiến các khối cube với vị trí vừa được thay đổi và vẽ lại các khối cube



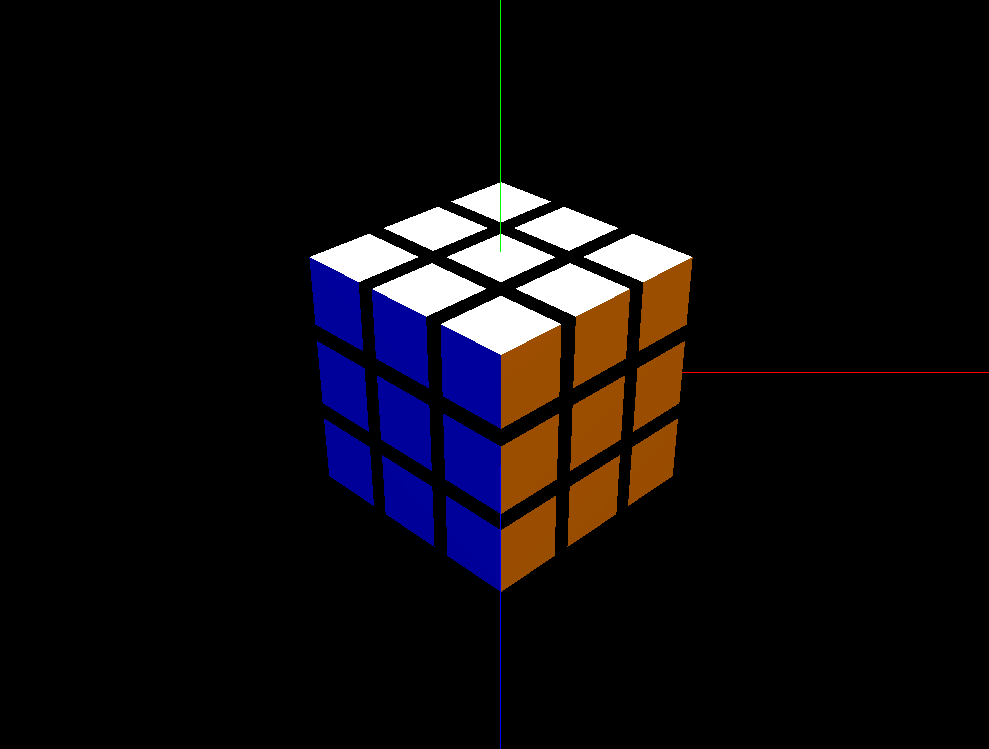
* Tương tự với 10 hàm sự kiện nhấn phìm còn lại thì sẽ quay các mặt khác nhau cùng hoặc ngược chiều kim đồng hồ

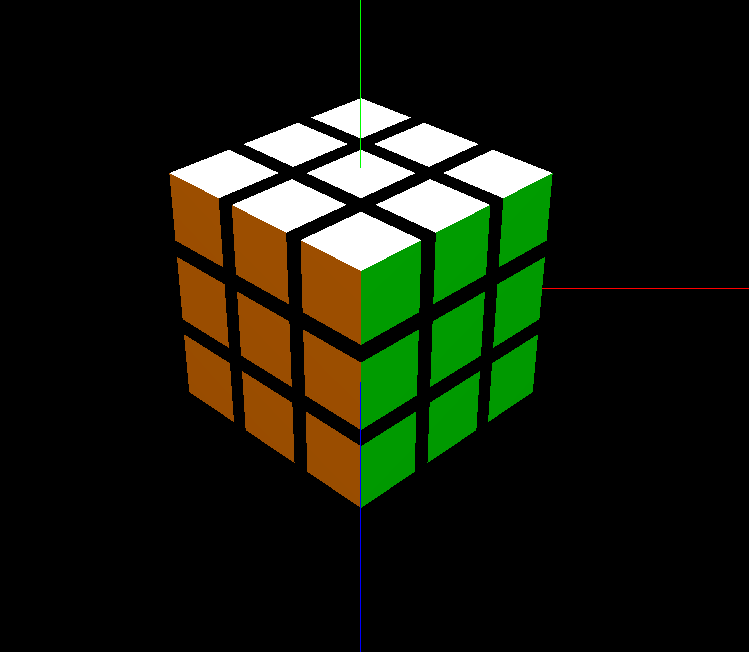
**C. Kết quả thực nghiệm**

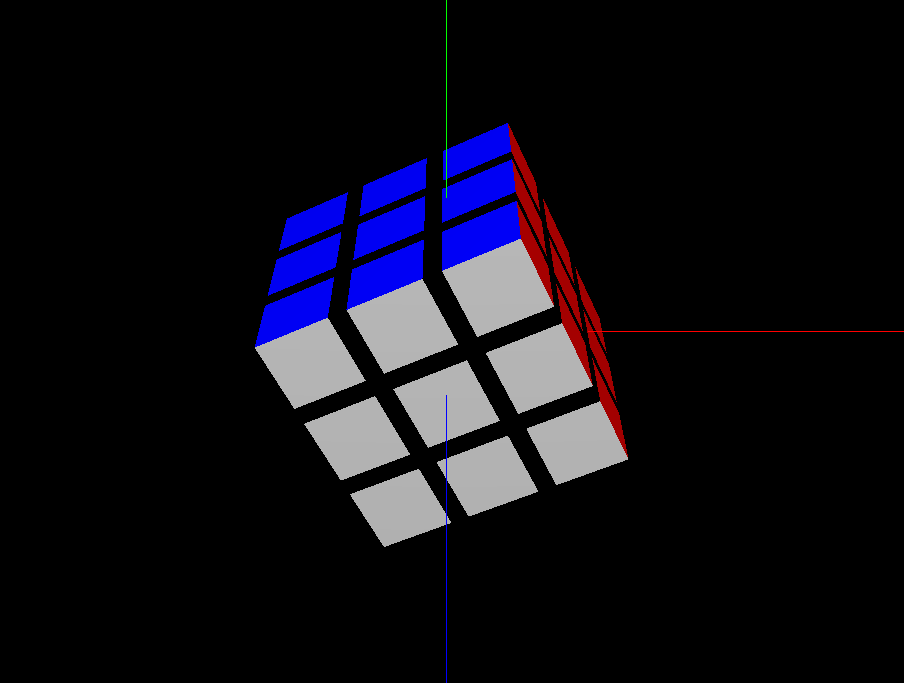
1. Hiển thị khối rubik và 3 trục tọa độ



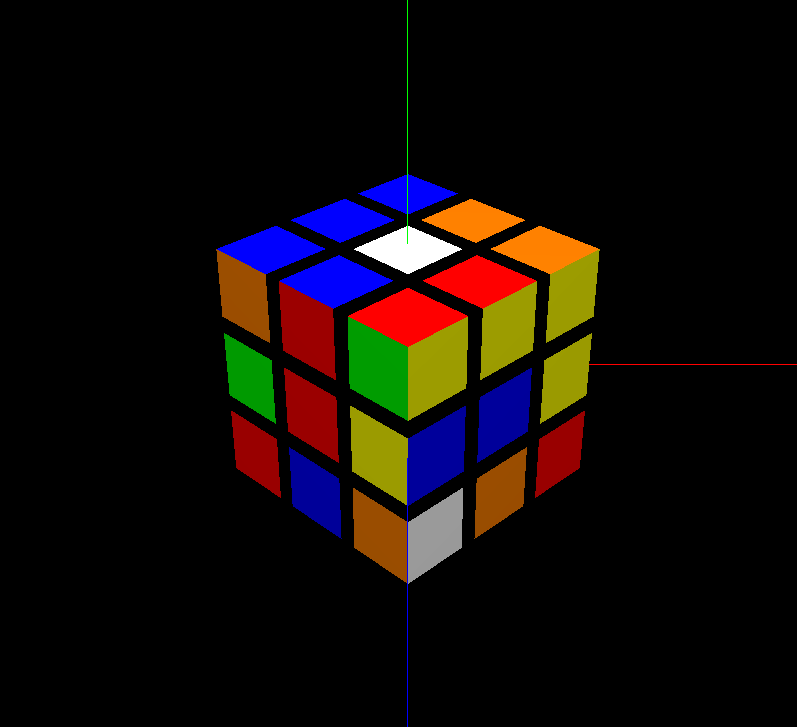
1. Xoay khối rubik







1. Xoay các mặt xung quanh khối rubik



**II. Kết Luận**

- Sau thời gian phân tích và xử lý các yêu cầu của bài toán, nhóm đã hoàn thành đủ các yêu cầu đã đặt ra để thiết lập một khối rubik có thể hoạt động giống như một khối rubik thật.

- Mặc dù đã có nhiều cố gắng trong suốt quá trình làm bài tập lớn, nhưng do thời gian có hạn và thiếu kinh nghiệm về kiến thức nên cũng không thể tránh khỏi những nhược điểm và thiếu sót. Vì vậy, em mong nhận được sự góp ý của thầy để nhóm chúng em có thể phát triển mô hình này hoàn thiện hơn.

***Cảm ơn thầy Ngô Trường Giang đã giúp đỡ bọn em trong thời gian chúng em học môn đồ họa máy tính.***

**III. Tài liệu tham khảo**

Slides môn đồ họa máy tính của thầy Ngô Trường Giang trên classroom

Link học và tìm hiểu them

<https://learnopengl.com/>

Link hướng dẫn học opengl

<https://cuuduongthancong.com/ggdpvf/1294439/huong-dan-lap-trinh-opengl-can-ban.pdf?src=subject>