n

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

**KHOA: CÔNG NGHỆ THÔNG TIN**

-------------------------------------

****

BÁO CÁO THỰC NGHIỆM THUỘC HỌC PHẦN:

ĐỒ HỌA MÁY TÍNH

**Mô phỏng một phòng học lý thuyết thuộc**

**nhà A9 trường ĐH Công nghiệp Hà Nội**

|  |  |
| --- | --- |
| Giáo viên hướng dẫn: | ThS.Vũ Minh Yến |
| Nhóm: | Nhóm 16 |
| Thành Viên: | Phó Đình Mạnh – 2020604634  Trần Minh Thái – 2020600643  Bùi Khôi Nguyên – 2019601721 |

**Hà Nội - Năm 2024**

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc159956725)

[CHƯƠNG 1. MÔ TẢ BÀI TOÁN 5](#_Toc159956726)

[1.1. Giới thiệu đề tài 5](#_Toc159956727)

[1.2. Mô tả mô hình và kịch bản 5](#_Toc159956728)

[CHƯƠNG 2. MỘT SỐ KỸ THUẬT PHÁT TRIỂN CHƯƠNG TRÌNH 6](#_Toc159956729)

[2.1. Thiết kế và cài đặt (Căn Phòng ) 6](#_Toc159956730)

[2.2. Thiết kế và cài đặt (Bàn ghế) 7](#_Toc159956731)

[2.3. Thiết kế và cài đặt (Quạt Trần) 9](#_Toc159956732)

[2.4. Thiết kế và cài đặt (Cửa ra vào) 9](#_Toc159956733)

[2.5. Thiết kế và cài đặt (Cửa sổ) 10](#_Toc159956734)

[2.6. Thiết kế và cài đặt (Bàn Ghế Giáo Viên) 10](#_Toc159956735)

[2.7. Thiết kế và cài đặt (Điều Hòa) 12](#_Toc159956736)

[2.8. Thiết kế và cài đặt (Đèn) 13](#_Toc159956737)

[2.9. Thiết kế và cài đặt (Chiếu sáng và tô bóng) 14](#_Toc159956738)

[2.10. Thiết lập và điều khiển phép chiếu, camera 14](#_Toc159956739)

[CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH 15](#_Toc159956740)

[Tài liệu tham khảo 18](#_Toc159956741)

[KẾT LUẬN 19](#_Toc159956742)

**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

**DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ**

Hình 1. Mô phỏng phòng học lý thuyết A9 16

Hình 2. Mô phỏng cửa ra vào lớp học 17

Hình 3. Mô phỏng cửa sổ và điều hòa 18

Hình 4. Mô phỏng bục giảng 19

Hình 5. Mô phỏng trạng thái phòng khi tắt đèn 20

### **MỞ ĐẦU**

Như chúng ta đã thấy trong những năm gần đây sự phát triển vượt bậc về công nghệ đồ họa đã mang lại cho con người chúng ta vô vàn tiện nghi, những trải nghiệm mới mẻ, chân thực.

VậyĐồ họa máy tính là gì? Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kĩ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,... và kĩ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vỉ mạch đồ họa...). Hiện nay đồ họa máy tính có rất nhiều ứng dụng. Ước tính đến 80% thông tin xử lý là hình ảnh. Một số ứng dụng tiêu biểu của đồ họa máy tính như là:

* Tạo mô hình, hoạt cảnh (game, giải trí,...)
* Hỗ trợ thiết kế đồ họa
* Mô phỏng hình ảnh, chẩn đoán hình ảnh (trong y tế)
* Huấn luyện đào tạo ảo (quân sự, hàng không,...)

Có nhiều cách tiếp cận trong việc học môn đồ họa, trải rộng từ việc nghiên cứu phần cứng tới việc học để sử dụng đồ họa máy tính chỉ trong một lĩnh vực chuyên biệt nào đó như là thiết kế mạch tích hợp cao (VLSI – very large scale integrated circuit). Ở đây chúng ta tiếp cận từ góc độ của người lập trình ứng dụng, đó là người sử dụng tất cả các hỗ trợ của phần cứng, các công cụ phần mềm để xây dựng nên các ứng dụng.

Tuy nhiên để có thể thiết kế và cài đặt các chương trình ứng dụng đồ họa được tốt, ngoài việc tìm hiểu các khả năng của công cụ lập trình, chúng ta cũng cần phải nắm vững các khái niệm về phần cứng; các vấn đề, các nguyên lí liên quan đến cài đặt phần mềm, các thuật toán, các ứng dụng, …

Là sinh viên khoa Công nghệ thông tin trường Đại học Công Nghiệp Hà Nội. Chúng em cũng được tiếp xúc với môn học đồ họa máy tính. Với những kiến thức đã được học và được sự hướng dẫn của cô Vũ Minh Yến chúng em đã thực hiện đề tài "Mô phỏng một phòng học lý thuyết thuộc toà A9 trường Đại học Công Nghiệp Hà Nội". Trong quá trình thực hiện nghiên cứu chúng em đã sử dụng thư viện đồ họa openGL đang được sử dụng rộng rãi hiện nay.

Có lẽ rằng chương trình và báo cáo của chúng em chưa được chuyên nghiệp, hoàn chỉnh nhất. Vì thế em rất mong thầy cô và các bạn có thể góp ý để nhóm em xây dựng đề đạt kết quả tốt nhất có thể.

# CHƯƠNG 1. MÔ TẢ BÀI TOÁN

* 1. Giới thiệu đề tài

Tên đề tài: Mô phỏng một phòng học lý thuyết thuộc toà A9 trường Đại học Công Nghiệp Hà Nội. Sử dụng máy tính, thư viện OpenGL khả lập trình và Visual C++.

* 1. Mô tả mô hình và kịch bản

*Yêu cầu của chương trình*

* Mô phỏng được một phòng học lý thuyết thuộc toà A9 thành phần: bàn ghế, cửa ra vào, cửa sổ, bảng, quạt trần, điều hòa.
* Có thay đổi góc nhìn.
* Có thể quay quạt, mở đóng cửa sổ và cửa ra vào.
* Bàn giáo viên có thể đóng mở tủ, ghế của giáo viên có thể xê dịch và quay.
* Cánh điều hòa có thể đóng mở.

*Mô tả chức năng của chương trình*

* Phím “S” và “s” để tiến lại gần và ra xa.
* Phím “X” và “x” để sang phải và trái.
* Phím “Y” và “y” để lên và xuống.
* Phím “q” để nhìn xung quanh.
* Phím “c,v” để mở cửa , “C,V” để đóng cửa.
* Phím “b,n,m” để mở cửa sổ, “B,M,N” để đóng cửa sổ.
* Phím “t” để bật tắt đèn.
* Phím “1/2/3/4/5/6” để xoay cánh quạt.
* Phím “o,p” và “O,P” để đóng mở cánh điều hòa
* Phím “l” và “L” để đóng mở tủ bàn giáo viên
* Phím “k” để xoay ghế bàn giáo viên
* Phím “j” và “J” để xê dịch ghế bàn giáo viên

# CHƯƠNG 2. MỘT SỐ KỸ THUẬT PHÁT TRIỂN CHƯƠNG TRÌNH

## 2.1. Thiết kế và cài đặt (Căn Phòng )

mat4 instance\_room;

//tuong

void wall\_room(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

GLfloat xxl, yyl, zzl;

//dung tuong

void wall\_build() {

instance\_room = identity();

//dựng nền căn phòng

instance\_room = Translate(0.0, -2.0,0.0f);

changeColor(225, 239, 148);

wall\_room(8, 0.05, 11);

//trần nhà

instance\_room = RotateZ(360) \* Translate(0, 2.0, 0.0f);

changeColor(92, 172, 238);

wall\_room(8, 0.05, 11);

//tường truoc

instance\_room = Translate(0, 0,5.5f);

changeColor(225, 296, 155);

wall\_room(8.0, 4, 0.05);

//tuong sau

instance\_room = Translate(0, 0, -5.5f);

changeColor(225, 296, 155);

wall\_room(8.0, 4, 0.05);

//buc giang

instance\_room = Translate(-1.0, -1.83, 5.0f);

changeColor(225, 239, 148);

wall\_room(6, 0.3, 1.1);

instance\_room = Translate(-3, -1.83, 4.2f);

changeColor(225, 239, 148);

wall\_room(2, 0.3, 0.5);

//bang

instance\_room = Translate(0, 0.3, 5.45f);

changeColor(0, 200, 128);

wall\_room(5, 1.8, 0.05);

////tường trái 1

instance\_room = Translate(4.0f, -0.75, 0.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 2.5, 6.0);

////tường trái 2

instance\_room = Translate(4.0f, -0.75, 5.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 2.5, 1.0);

////tường trái 3

instance\_room = Translate(4.0f, -0.75, -5.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 2.5, 1.0);

////tường trái 4

instance\_room = Translate(4.0f, 1.25, 0.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.5, 11);

////tường phải1

instance\_room = Translate(-4.0f, 0, 5.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.5, 1.0);

////tường phải2

instance\_room = Translate(-4.0f, 0, 2.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.5, 2.0);

////tường phải3

instance\_room = Translate(-4.0f, 0, -1.5f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.5, 2.0);

////tường phải4

instance\_room = Translate(-4.0f, 0, -4.75f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.5, 1.5);

////tường phải5

instance\_room = Translate(-4.0f, 1.375f, 0.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.25, 11);

////tường phải6

instance\_room = Translate(-4.0f, -1.375f, 0.0f);

changeColor(225, 296, 155);

wall\_room(0.05, 1.25, 11);

}

## 2.2. Thiết kế và cài đặt (Bàn ghế)

mat4 instance\_ban;

void ban(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_ban \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void matBan() {

instance\_ban = Translate(0, 0.3, 0);

changeColor(252.0, 245.0, 76.0);

ban(0.8, 0.02, 0.4);

}

void chanBan() {

//chan 1

instance\_ban = Translate(-0.375, -0.01, -0.2);

changeColor(225, 296, 155);

ban(0.02, 0.6, 0.02);

//chan 2

instance\_ban = Translate(-0.375, -0.01, 0.2);

changeColor(225, 296, 155);

ban(0.02, 0.6, 0.02);

//chan 3

instance\_ban = Translate(0.375, -0.01, -0.2);

changeColor(225, 296, 155);

ban(0.02, 0.6, 0.02);

//chan 4

instance\_ban = Translate(0.375, -0.01, 0.2);

changeColor(225, 296, 155);

ban(0.02, 0.6, 0.02);

//thang doc 1

instance\_ban = Translate(-0.375, -0.24, 0);

changeColor(225, 296, 155);

ban(0.02, 0.02, 0.4);

//thanh doc 2

instance\_ban = Translate(0.375, -0.24, 0);

changeColor(225, 296, 155);

ban(0.02, 0.02, 0.4);

//thanh ngang

instance\_ban = Translate(0, -0.24, 0);

changeColor(225, 296, 155);

ban(0.78, 0.02, 0.02);

}

void matGhe() {

instance\_ban = Translate(0, 0, 0.5);

changeColor(252.0, 245.0, 76.0);

ban(0.8, 0.02, 0.25);

}

void chanGhe() {

//chan 1

instance\_ban = Translate(-0.375, -0.15, 0.4);

changeColor(225, 296, 155);

ban(0.02, 0.3, 0.02);

//chan 2

instance\_ban = Translate(-0.375, -0.15, 0.6);

changeColor(225, 296, 155);

ban(0.02, 0.3, 0.02);

//chan 3

instance\_ban = Translate(0.375, -0.15, 0.4);

changeColor(225, 296, 155);

ban(0.02, 0.3, 0.02);

//chan 4

instance\_ban = Translate(0.375, -0.15, 0.6);

changeColor(225, 296, 155);

ban(0.02, 0.3, 0.02);

//thang doc 1

instance\_ban = Translate(-0.375, -0.24, 0.4);

changeColor(225, 296, 155);

ban(0.02, 0.02, 0.4);

//thanh doc 2

instance\_ban = Translate(0.375, -0.24, 0.4);

changeColor(225, 296, 155);

ban(0.02, 0.02, 0.4);

//thanh ngang

instance\_ban = Translate(0, -0.24, 0.5);

changeColor(225, 296, 155);

ban(0.78, 0.02, 0.02);

}

void caiBan() {

matBan();

chanBan();

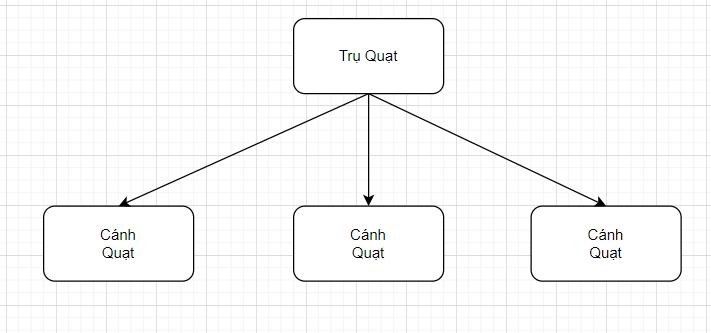
chanGhe();

matGhe();

}

## 2.3. Thiết kế và cài đặt (Quạt Trần)

- Mô hình phân cấp:



- Thiết kế:

mat4 instance\_quat;

GLfloat daiQ = 0.7;

GLfloat dayQ = 0.025;

GLfloat rongQ = 0.1;

GLfloat quayCanh[] = { 0,0,0,0,0,0,0 };

void quat(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_quat \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void tru\_quat() {

instance\_quat = Translate(0, 0.14, 0);

changeColor(0, 200, 128);

quat(0.1, 0.25, 0.1);

}

void canh\_quat1() {

instance\_quat = RotateY(quayCanh[1]) \* Translate(0.5 \* daiQ, 0, 0);

changeColor(0, 200, 128);

quat(daiQ, dayQ, rongQ);

instance\_quat = RotateY(quayCanh[1]) \* RotateY(120) \* Translate(0.5 \* daiQ, 0, 0);

changeColor(0, 200, 128);

quat(daiQ, dayQ, rongQ);

instance\_quat = RotateY(quayCanh[1]) \* RotateY(240) \* Translate(0.5 \* daiQ, 0, 0);

changeColor(0, 200, 128);

quat(daiQ, dayQ, rongQ);

}

void QuatTran1() {

tru\_quat();

canh\_quat1();

}

## 2.4. Thiết kế và cài đặt (Cửa ra vào)

GLfloat mo\_cua[] = { 0,0,0,0 };

mat4 instance\_cuavao;

void canh\_cua(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room\* instance\_cuavao \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void cua() {

instance\_cuavao = identity() \* Translate(4, -0.75, 3.75) \*Translate(0, 0, 0.75) \* RotateY(mo\_cua[0])\* Translate(0,0,-0.375);

changeColor(252.0, 245.0, 76.0);

canh\_cua(0.05, 2.5, 0.75);

instance\_cuavao = Translate(4, -0.75, 3.75) \* Translate(0, 0, -0.75) \* RotateY(-mo\_cua[0]) \* Translate(0, 0, 0.375);

changeColor(252.0, 245.0, 76.0);

canh\_cua(0.05, 2.5, 0.75);

instance\_cuavao = Translate(4, -0.75, -3.75) \* Translate(0, 0, 0.75) \* RotateY(mo\_cua[1]) \* Translate(0, 0, -0.375);

changeColor(252.0, 245.0, 76.0);

canh\_cua(0.05, 2.5, 0.75);

instance\_cuavao = Translate(4, -0.75, -3.75) \* Translate(0, 0, -0.75) \* RotateY(-mo\_cua[1]) \* Translate(0, 0, 0.375);

changeColor(252.0, 245.0, 76.0);

canh\_cua(0.05, 2.5, 0.75);

}

void bo\_cua() {

instance\_room = identity();

cua();

}

## 2.5. Thiết kế và cài đặt (Cửa sổ)

mat4 instance\_cuaso;

GLfloat z\_cs1 =0, z\_cs2 =0, z\_cs3 =0;

void canh\_cua2(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_cuaso \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void cua2() {

instance\_cuaso = identity() \* Translate(-4, 0, 4.125) \* Translate(0,0,z\_cs1);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

instance\_cuaso = Translate(-4.05, 0, 3.375) \* Translate(0, 0, -z\_cs1);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

instance\_cuaso = Translate(-4, 0, 0.625) \* Translate(0, 0, z\_cs2);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

instance\_cuaso = Translate(-4.05, 0, -0.125) \* Translate(0, 0, -z\_cs2);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

instance\_cuaso = Translate(-4, 0, -2.875) \* Translate(0, 0, z\_cs3);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

instance\_cuaso = Translate(-4.05, 0, -3.625) \* Translate(0, 0, -z\_cs3);

changeColor(252.0, 245.0, 76.0);

canh\_cua2(0.05, 1.5, 0.75);

}

void bo\_cuaso() {

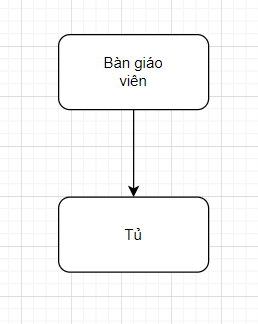
instance\_room = identity();

cua2();

}

## 2.6. Thiết kế và cài đặt (Bàn Ghế Giáo Viên)

- Mô hình phân cấp:



- Thiết kế:

mat4 instance\_bangv;

void bangv(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_bangv \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void matBanGV() {

instance\_bangv = Translate(0, 0, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.8, 0.02, 0.4);

}

void chanBanGV() {

//chan 1

instance\_bangv = Translate(0.39, -0.31, 0.19);

changeColor(225, 296, 155);

bangv(0.02, 0.6, 0.02);

//chan 2

instance\_bangv = Translate(0.39, -0.31, -0.19);

changeColor(225, 296, 155);

bangv(0.02, 0.6, 0.02);

//chan 3

instance\_bangv = Translate(-0.39, -0.31, 0.19);

changeColor(225, 296, 155);

bangv(0.02, 0.6, 0.02);

//chan 4

instance\_bangv = Translate(-0.39, -0.31, -0.19);

changeColor(225, 296, 155);

bangv(0.02, 0.6, 0.02);

}

GLfloat goctu = 0;

void hoctu() {

instance\_bangv = Translate(0, -0.26, 0.19);

changeColor(252.0, 245.0, 76.0);

bangv(0.76, 0.5, 0.02);

instance\_bangv = Translate(0.39, -0.26, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.02, 0.5, 0.36);

instance\_bangv = Translate(-0.39, -0.26, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.02, 0.5, 0.36);

instance\_bangv = Translate(-0.1, -0.26, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.02, 0.5, 0.36);

instance\_bangv = Translate(-0.245, -0.5, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.27, 0.02, 0.36);

instance\_bangv = Translate(-0.39, -0.26, -0.19) \* RotateY(goctu) \* Translate(0.145, 0, 0);

changeColor(252.0, 245.0, 76.0);

bangv(0.29, 0.5, 0.02);

}

void caibanGV()

{

matBanGV();

chanBanGV();

hoctu();

}

void botriBanGV() {

instance\_room = Translate(-3, -1, 4.3) \* RotateY(180);

caibanGV();

}

//ghegv

mat4 instance\_ghegv;

void ghegv(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_ghegv \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void matgheGV() {

instance\_ghegv = Translate(0, 0, 0);

changeColor(252.0, 245.0, 76.0);

ghegv(0.3, 0.02, 0.3);

}

void changheGV() {

//chan 1

instance\_ghegv = Translate(0.14, -0.16, 0.14);

changeColor(225, 296, 155);

ghegv(0.02, 0.3, 0.02);

//chan 2

instance\_ghegv = Translate(0.14, -0.16, -0.14);

changeColor(225, 296, 155);

ghegv(0.02, 0.3, 0.02);

//chan 3

instance\_ghegv = Translate(-0.14, -0.16, 0.14);

changeColor(225, 296, 155);

ghegv(0.02, 0.3, 0.02);

//chan 4

instance\_ghegv = Translate(-0.14, -0.16, -0.14);

changeColor(225, 296, 155);

ghegv(0.02, 0.3, 0.02);

}

void lunggheGV() {

instance\_ghegv = Translate(0, 0.21, 0.14);

changeColor(252.0, 245.0, 76.0);

ghegv(0.3, 0.4, 0.02);

}

void caighegv() {

matgheGV();

changheGV();

lunggheGV();

}

GLfloat xoayghe = 0;

GLfloat zg;

void botrighegv() {

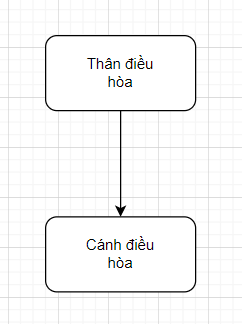
instance\_room = Translate(-3.2, -1.3, 4.5) \* Translate( 0, 0, zg) \* RotateY(xoayghe);

caighegv();

}

## 2.7. Thiết kế và cài đặt (Điều Hòa)

- Mô hình phân cấp :



- Thiết kế ;

mat4 instance\_dieuhoa;

void dh(GLfloat x, GLfloat y, GLfloat z) {

instance = Scale(x, y, z);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_dieuhoa \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void than\_dh() {

// mat trc

instance\_dieuhoa = Translate(0, 0, 0);

dh(1, 0.4, 0.05);

//mat sau

instance\_dieuhoa = Translate(0, -0.1, 0.3);

dh(1, 0.6, 0.05);

//mat tren

instance\_dieuhoa = Translate(0, 0.175, 0.15);

dh(1, 0.05, 0.25);

//mat duoi

instance\_dieuhoa = Translate(0, -0.375, 0.15);

dh(1, 0.05, 0.25);

//mat trai

instance\_dieuhoa = Translate(0.525, -0.1, 0.15);

dh(0.05, 0.6, 0.35);

// mat phai

instance\_dieuhoa = Translate(-0.525, -0.1, 0.15);

dh(0.05, 0.6, 0.35);

}

GLfloat quaydh[] = { 0,0,0,0 };

void canh\_dh1() {

instance\_dieuhoa = Translate(0, -0.2, 0) \* RotateX(quaydh[1]) \* Translate(0, -0.1, 0);

dh(1, 0.2, 0.05);

}

void dieuhoa1(){

than\_dh();

canh\_dh1();

}

## 2.8. Thiết kế và cài đặt (Đèn)

mat4 instance\_den;

bool bat\_den = true;

bool view\_banGV = true;

void den(GLfloat w, GLfloat h, GLfloat t) {

instance = Scale(w, h, t);

glUniformMatrix4fv(Model\_loc, 1, GL\_TRUE, ctm\_phong \* instance\_room \* instance\_den \* instance);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void veDen() {

// tren

changeColor(255, 215, 0);

instance\_den = Translate(0, 0.03, 0);

den(1.0, 0.04, 0.15);

//truoc

instance\_den = Translate(0, 0, -0.075);

den(1.0, 0.06, 0.02);

//sau

instance\_den = Translate(0, 0, 0.075);

den(1.0, 0.06, 0.02);

//trai

instance\_den = Translate(-0.5, 0, 0);

den(0.04, 0.06, 0.15);

//phai

instance\_den = Translate(0.5, 0, 0);

den(0.04, 0.06, 0.15);

// bong den

changeColor(mauBD, mauBD, mauBD);

instance\_den = Translate(0, -0.03, 0);

den(0.9, 0.06, 0.1);

}

void Den\_buil() {

for (int i = 0; i <= 3; i ++) {

instance\_room = identity() \* Translate(-2, 1.85, -3 + i +i);

veDen();

instance\_room = identity() \* Translate(2, 1.85, -3 + i +i);

veDen();

}

}

## 2.9. Thiết kế và cài đặt (Chiếu sáng và tô bóng)

void changeColor(GLfloat a, GLfloat b, GLfloat c) {

/\* Khởi tạo các tham số chiếu sáng - tô bóng\*/

point4 light\_position(0.0, 2.0, 0.0, 1.0);

color4 light\_ambient(0.2, 0.2, 0.2, 1.0);

color4 light\_diffuse(mauAnhSang, mauAnhSang, mauAnhSang, 1.0);

color4 light\_specular(choisang, choisang, choisang, 1.0);

color4 material\_ambient(1.0, 0.0, 1.0, 1.0);

color4 material\_diffuse(a / 255.0, b / 255.0, c / 255.0, 1.0);

color4 material\_specular(255.0 / 255.0, 255.0 / 255.0, 255 / 255.0, 1.0);

float material\_shininess = 100.0;

color4 ambient\_product = light\_ambient \* material\_ambient;

color4 diffuse\_product = light\_diffuse \* material\_diffuse;

color4 specular\_product = light\_specular \* material\_specular;

glUniform4fv(glGetUniformLocation(program, "AmbientProduct"), 1, ambient\_product);

glUniform4fv(glGetUniformLocation(program, "DiffuseProduct"), 1, diffuse\_product);

glUniform4fv(glGetUniformLocation(program, "SpecularProduct"), 1, specular\_product);

glUniform4fv(glGetUniformLocation(program, "LightPosition"), 1, light\_position);

glUniform1f(glGetUniformLocation(program, "Shininess"), material\_shininess);

}

## 2.10. Thiết lập và điều khiển phép chiếu, camera

GLfloat l = -2.5, r = 2.5; // giới hạn của khung nhìn

GLfloat bottom = -2.5, top = 2.5; //giới hạn khung nhìn

GLfloat zNear = 1, zFar = 17; // giới hạn khoảng cách hiển thị được

GLfloat z\_eye = -10.0, y\_eye = 1; //điểm view x,y

GLdouble xx\_p, yy\_ph, zz\_p;

GLfloat x\_phong = 0, y\_phong = 0, z\_phong = -4.4;

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

const vec3 viewer\_pos(0, y\_eye, z\_eye); /\*Trùng với eye của camera\*/

ctm\_phong = Translate(x\_phong, y\_phong, z\_phong) \* RotateY(yy\_ph);

room();

vec4 eye(0, y\_eye, z\_eye, 1); //điểm nhìn

vec4 at(0, 0, 0, 1);

vec4 up(0, 1, 0, 1);

view = LookAt(eye, at, up);

glUniformMatrix4fv(View\_loc, 1, GL\_TRUE, view);

projection = Frustum(l, r, bottom, top, zNear, zFar);

glUniformMatrix4fv(Projection\_loc, 1, GL\_TRUE, projection);

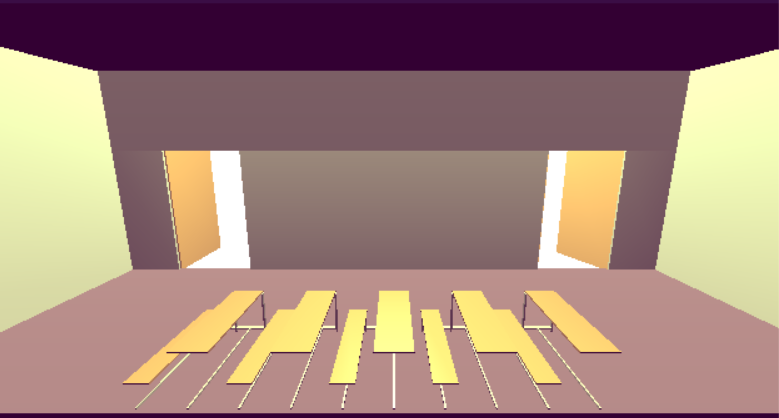
glutSwapBuffers();

}

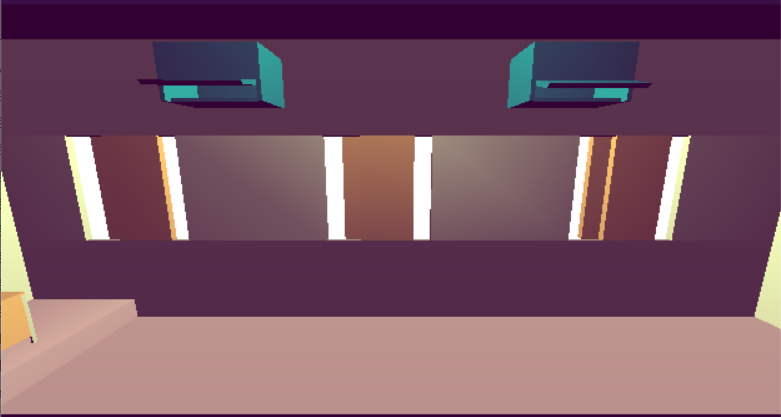
CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH



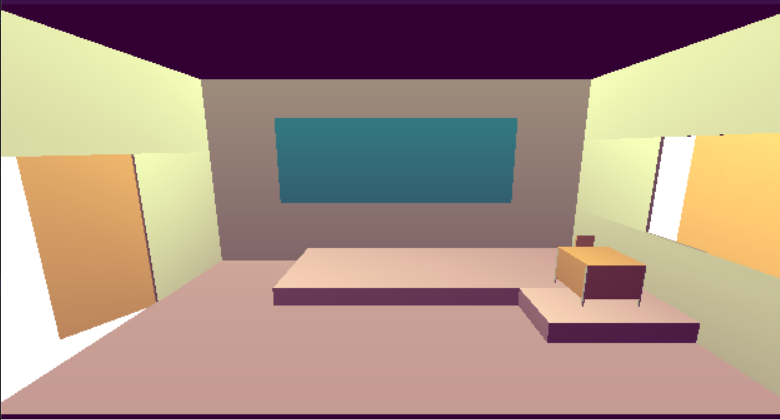
Hình 1. Phòng học lý thuyết A9



Hình 2. Cửa ra vào lớp học



Hình 3. Cửa sổ và điều hòa

Hình 4. Bục giảng



*Hình 5. Phòng khi tắt đèn*

Phụ Lục

## Tài liệu tham khảo

* Slide bài giảng và các tài liệu học tập của ThS. Vũ Minh Yến
* Interative computers graphics- Adward Angel and Dave Shreiner
* OpenGL Programming- Eighth Editison

# KẾT LUẬN

Qua việc thực hiện nghiên cứu đề tài “Mô phỏng một phòng học lý thuyết thuộc toà A9 trường Đại học Công Nghiệp Hà Nội”. Nhóm chúng em đã được biết thêm rất nhiều về môn đồ họa máy tính. Bên cạnh đó, việc làm nghiên cứu giúp chúng em đoàn kết hơn, rèn luyện cho chúng em kỹ năng làm việc nhóm.

Trong quá trình thực hiện đề tài có rất nhiều ý tưởng hay, độc đáo. Nhưng do kiến thức của chúng em hạn hẹp và thời gian không cho phép nên chúng em chưa thể thực hiện được những ý tưởng đó. Tuy nhiên chúng em đã cố gắng để xây dựng một chương trình hoàn chỉnh nhất, đẹp nhất. Vì vậy em rất mong rằng thầy và các bạn cùng góp ý với nhóm em để có thể hòa thành sản phẩm một cách hoàn chỉnh nhất.

Chúng em xin cảm ơn cô Vũ Minh Yến đã tận tình giảng dạy chúng em trong môn Đồ họa máy tính. Giúp đỡ chúng em trong quá trình nghiên cứu đề tài. Chia sẻ những tài liệu hay về thư viện OpenGL cũng như các kĩ năng lập trình.

Một lần nữa chúng em xin chân thành cảm ơn !