

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**~~~~~~\*~~~~~~**



**BÁO CÁO THỰC NGHIỆM**

**ĐỒ HỌA MÁY TÍNH**

**ĐỀ TÀI: MÔ PHỎNG MỘT PHÒNG THỰC HÀNH TẦNG 8 TOÀ A1 TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI.**

**GVHD: ThS. Vũ Minh Yến**

**Nhóm : 02**

**Nhóm sinh viên: Dương Đức Tùng**

**Trần Thế Hưởng**

**Hoàng Quốc Cường**

**Lê Văn Hùng**

**Dương Đình Trung**

**Lớp: 20233IT6010001**

**Hà Nội - 2024**

**MỤC LỤC**

[**CHƯƠNG 1. MÔ TẢ BÀI TOÁN 4**](#_heading=h.gjdgxs)

[**CHƯƠNG 2. MỘT SỐ KỸ THUẬT PHÁT TRIỂN CHƯƠNG TRÌNH 6**](#_heading=h.3znysh7)

[2.1 Kỹ thuật tạo mô hình các bức tường 6](#_heading=h.2et92p0)

[2.2 Kỹ thuật tạo mô hình Cửa chính 7](#_heading=h.tyjcwt)

[2.3 Kỹ thuật tạo mô hình Tủ đựng đồ 8 ngăn 7](#_heading=h.3dy6vkm)

[2.4 Kỹ thuật tạo mô hình bảng và máy chiếu 11](#_heading=h.1t3h5sf)

[2.5 Kỹ thuật tạo mô hình bàn, ghế, máy tính 12](#_heading=h.4d34og8)

[2.6 Kỹ thuật tạo mô hình Cửa sổ 14](#_heading=h.6g5ggk9cjbeo)

[2.7 Kỹ thuật tạo mô hình Laptop 16](#_heading=h.m6d3k6i3ky6g)

[2.8 Kỹ thuật tạo mô hình Cầu dao 17](#_heading=h.va02a4rwd8kf)

[2.9 Kỹ thuật tạo mô hình Căn phòng 19](#_heading=h.2s8eyo1)

[2.9.1 Mô hình phân cấp 19](#_heading=h.17dp8vu)

[2.9.2 Kỹ thuật cài đặt 20](#_heading=h.3rdcrjn)

[2.10 Kỹ thuật thay đổi góc nhìn 21](#_heading=h.26in1rg)

[2.10.1 Đầu vào (Input) 21](#_heading=h.8o56ywmfh5eh)

[2.10.1 Đầu ra (Input) 22](#_heading=h.df29top81xxd)

[2.10.3 Kỹ thuật cài đặt 22](#_heading=h.ftrtk1grfdof)

[2.11 Kỹ thuật điều khiển vật thể 24](#_heading=h.hmkm0xcufebx)

[2.12 Kỹ thuật chiếu sáng 28](#_heading=h.e96a3obj4wpn)

[**CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH. 29**](#_heading=h.lnxbz9)

[**TỔNG KẾT 33**](#_heading=h.35nkun2)

**LỜI NÓI ĐẦU**

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 cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích và hình học họa hình, quang học…… và kỹ thuật máy tính và đặc biệt là chế tạo phần cứng ( các loại màn hình, các thiết bị nhập xuất, các vi mạch đồ họa).

Nghiên cứu các phép biến đổi hình học trong không gian thực hai chiều là một trong những nội dung quan trọng..

Dưới sự hướng dẫn của cô Vũ Minh Yến cùng với những kiến thức mà bọn em đã học được ở trong bộ môn Đồ Họa Máy Tính, chúng em cuối cùng đã thực hiện được đề tài “Mô phỏng phòng thực hành tầng 8 tòa A1 của trường Đại Học Công Nghiệp”

Do quá trình tìm hiểu còn nhiều thiếu sót nên chương trình còn nhiều hạn chế, nhóm thực hiện rất mong nhận được ý kiến đóng góp từ phía cô để phần mềm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn

**Nhóm sinh viên thực hiện!**

# CHƯƠNG 1. MÔ TẢ BÀI TOÁN

* 1. **Giới thiệu đề tài**

Tên đề tài: Mô phỏng phòng thực hành tầng 8 tòa A1 của trường Đại Học Công Nghiệp Hà Nội.

* 1. **Mô tả đối tượng cần thiết kế**

Yêu cầu của chương trình

* Mô phỏng 1 phòng thực hành tầng 8 tòa A1 gồm: cửa, tường, bàn, ghế , máy tính, máy chiếu, tủ, laptop, cây cảnh.
* Mô phỏng được việc chuyển động của các đối tượng cụ thể : cửa chính và cửa sổ có thể đóng vào, mở ra…
* Có thay đổi góc nhìn tùy ý.
* Có thể chỉnh kích thước khung hình.

**1.3 Mô tả bố cục khung cảnh chung**

****

Hình 1:Ảnh chụp mô hình thật định xây dựng theo

**1.4 Mô tả kịch bản**

Bài toán xây dựng một phòng thực hành tầng 8 tòa A1 trường Đại học Công nghiệp Hà Nội gồm

* 4 bức tường, trần nhà và sàn nhà
* 1 màn chiếu có thể kéo vào - ra
* 1 máy chiếu
* 3 cửa sổ có thể mở vào - ra
* 1 cửa ra vào
* 1 bảng
* 1 laptop có thể gập vào mở ra và tắt - mở màn hình
* 1 tủ đồ có thể đóng - mở cửa tủ từng ô
* 17 bộ bàn ghế (bàn, ghế kéo ra - vào, cây máy tính, máy tính )
* 4 điều hòa
* 9 đèn có thể bật tắt
* 2 loa
* 1 hộp cầu dao có thể mở - đóng cửa và bật - tắt cầu dao

Mô tả chức năng của chương trình

* Sau khi chạy chương trình nhấn phím “c” và “C” để đóng mở cửa chính
* Sử dụng phím “v” và “V” để đóng mở cửa sổ
* Sử dụng phím “w” , “a”, “s”, “d” để thay đổi góc nhìn tùy ý.
* Sử dụng “x”, “X”, “y”, “Y”, “z”, “Z” để thay đổi kích thước khung hình.
* Phím “t” và “T” để nhìn xung quanh.
* Phím số từ “1-8” để mở các ô tủ tương ứng.
* Phím “k” điều khiển màn chiếu kéo lên – xuống.
* Phím “h” điều khiển ghế ra vào.
* Phím “l” để gập laptop xuống-lên
* Phím “i” để mở màn hình laptop
* Phím “g” để di chuyển chậu cây
* Phím “n” để mở hộp cầu dao
* Phím “m” để bật - tắt cầu dao và đèn

# 

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

## 2.1 Kỹ thuật tạo mô hình (các bức tường)

//Tường trái

void tuongTrai() {

model = Translate(-3.25, 0.4, -0.35);

matPhang(0.02, 1.5, 6.8, model, mautuong);

model = Translate(-3.25, 0.4, -6.1);

matPhang(0.02, 1.5, 1.7, model, mautuong);

model = Translate(-3.26, 2.9, -1.95);

matPhang(0.02, 3.5, 10, model, mautuong);

}

//Tường phải

void tuongPhai() {

model = Translate(3.25, 3.9, -1.95);

matPhang(0.02, 1.5, 10, model, mautuong);

model = Translate(3.25, 0.45, -1.95);

matPhang(0.02, 1.5, 10, model, mautuong);

model = Translate(3.25, 2.18, -6.4);

matPhang(0.02, 2, 1, model, mautuong);

model = Translate(3.25, 2.18, -1.95);

matPhang(0.02, 2, 1, model, mautuong);

model = Translate(3.25, 2.18, 2.55);

matPhang(0.02, 2, 1, model, mautuong);

}

//Trần

void tranNha() {

model = Translate(0, 4.68, -1.95);

matPhang(6.5, 0.02, 10, model, maucua);

}

//Tường sau

void tuongSau() {

model = Translate(0, 2.2, -6.9);

matPhang(6.5, 5, 0.02, model, maucua);

}

//Sàn

void san() {

model = Translate(0, -0.31, -1.95);

matPhang(6.5, 0.02, 10, model, mautuong);

}

## 2.2 Kỹ thuật tạo mô hình Cửa chính

//Cửa sổ

GLfloat quayCuaSo1 = 0, quayCuaSo2 = 0;

void cuaSo() {

//canh cua 1

model = Translate(3.25, 0, -5.9) \* RotateY(quayCuaSo1) \* Translate(0, 2.18, 0.5 \* 1.75);

matPhang(0.02, 2, 1.75, model, maucua);

//canh cua 2

model = Translate(3.25, 0, -2.45) \* RotateY(180) \* RotateY(quayCuaSo2) \* Translate(0, 2.18, 0.5 \* 1.75);

matPhang(0.02, 2, 1.75, model, maucua);

}

//Cửa chính

GLfloat quayCuaChinh1 = 0, quayCuaChinh2 = 0;

void cuaChinh() {

//canh cua 1

model = Translate(-3.25,0,-5.25) \* RotateY(quayCuaChinh1) \* Translate(0, 0.4, 0.5 \* 0.75);

matPhang(0.02, 1.5, 0.75, model,maucua);

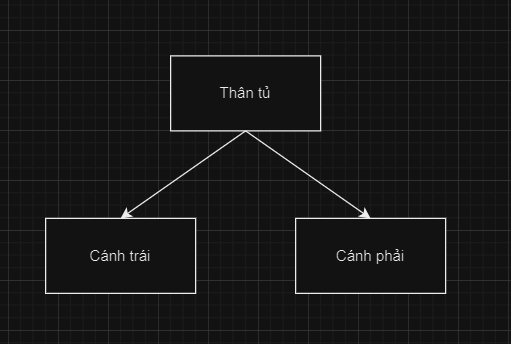
//canh cua 2

model = Translate(-3.25, 0, -3.75)\* RotateY(180) \* RotateY(quayCuaChinh2) \* Translate(0, 0.4, 0.5 \* 0.75);

matPhang(0.02, 1.5, 0.75, model, maucua);

}

## 2.3 Kỹ thuật tạo mô hình Tủ đựng đồ 8 ngăn



//Khung tủ

void khungTu() {

//Mặt trái

model = Translate(-0.5 \* widthTu - 0.5 \* day, 0, 0);

matPhang(day, heighTu, widthTu, model, mautu);

//Mặt phải

model = Translate(0.5 \* widthTu + 0.5 \* day, 0, 0);

matPhang(day, heighTu, widthTu, model, mautu);

//Mặt trên

model = Translate(0, 0.5 \* heighTu - 0.5 \* day, 0);

matPhang(widthTu, day, widthTu, model, mautu);

//Mặt dưới

model = Translate(0, -0.5 \* heighTu + 0.5 \* day, 0);

matPhang(widthTu, day, widthTu, model, mautu);

// mặt sau

model = Translate(0, 0, 0.5 \* widthTu - 0.5 \* day);

matPhang(widthTu, heighTu, day, model, mautu);

// ngăn ngag tủ trên

model = Translate(0, 0.25 \* heighTu, 0);

matPhang(widthTu, day + 0.02, widthTu, model, mautu);

// ngăn ngag tủ giữa

model = Translate(0, 0, 0);

matPhang(widthTu, day + 0.02, widthTu, model, mautu);

// ngăn ngag tủ dưới

model = Translate(0, -0.25 \* heighTu, 0);

matPhang(widthTu, day + 0.02, widthTu, model, mautu);

// ngăn dọc giữa

model = Translate(0, 0, 0);

matPhang(day + 0.02, heighTu, widthTu, model, mautu);

}

GLfloat widthCT = 0.5 \* widthTu;

GLfloat heightCT = 0.25 \* heighTu;

GLfloat MoTu = 0;

int tuState = 0;

GLfloat MoTu1 = 0;

int tuState1 = 0;

GLfloat MoTu2 = 0;

int tuState2 = 0;

void cuaTu() {

// cửa 1 trái

model = Translate(-widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu1) \* Translate(widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(-widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu1) \* Translate(widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(-widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(MoTu1) \* Translate(widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 2 trái

model = Translate(-widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(-widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(-widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(MoTu) \* Translate(widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 3 trái

model = Translate(-widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(-widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(-widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(MoTu) \* Translate(widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 4 trái

model = Translate(-widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(-widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(MoTu) \* Translate(widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(-widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(MoTu) \* Translate(widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 1 phai

model = Translate(widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu2) \* Translate(-widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu2) \* Translate(-widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(widthCT, 1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(-MoTu2) \* Translate(-widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 2 phai

model = Translate(widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(widthCT, 0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 3 phai

model = Translate(widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(widthCT, -0.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

// cửa 4 phai

model = Translate(widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.5, 0, 0);

matPhang(widthCT, heightCT - 0.025, day, model, maucuatu);

model = Translate(widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.65, 0, -0.03 \* 0.5);

matPhang(0.02, 0.1, 0.03, model, mautaykeocuatu);

model = Translate(widthCT, -1.5 \* heightCT, -0.5 \* widthTu + 0.5 \* day - day) \* RotateY(-MoTu) \* Translate(-widthCT \* 0.8, 0, -0.018 \* 0.5) \* RotateX(90);

hinhTru(0.018, 0.02, 0.018, model, mauden);

}

void tu() {

khungTu();

cuaTu();

}

## 2.4 Kỹ thuật tạo mô hình (bảng và máy chiếu)

//Máy chiếu

void truMayChieu() {

model = Translate(0, 0.25, 0);

matPhang(0.2, 0.5, 0.2, model, maumaychieu);

}

void hopMayChieu() {

model = Translate(-0.5 \* 0.025, 0, 0);

matPhang(0.6, 0.05, 0.2, model, maumaychieu);

}

GLfloat wM = 1.5, hM = 2.5, dM = 0.05;

GLfloat wTN = 1.5, hTN = 0.1, dTN = 0.2;

GLfloat keo = 0;

bool checkkeo = false;

//Màn chiếu

void manChieu() {

model = Translate(0, 0.5 \* keo, -6.8 ) \* RotateY(180);

matPhang(wM, hM - keo, dM, model, maumanchieu);

model = Translate(0, 0.5 \* hM + 0.5 \* hTN, -6.8) \* RotateY(180);

matPhang(wTN, hTN, dTN, model, maumanchieu);

}

void mayChieu() {

truMayChieu();

hopMayChieu();

}

//bảng

void bang() {

model = Translate(0, 2.2, -6.85) \* RotateY(180);

matPhang(3, 3, 0.02, model, maubang);

}

## 2.5 Kỹ thuật tạo mô hình (bàn, ghế, máy tính)

//Mặt bàn

void matBan() {

model = Translate(0, 0.3, 0);

matPhang(0.8, 0.02, 0.4, model, maubanghe);

}

//Chân bàn

void chanBan() {

//chan 1

model = Translate(-0.375,-0.01,-0.2);

matPhang(0.02, 0.6, 0.02, model, maubanghe);

//chan 2

model = Translate(-0.375, -0.01, 0.2);

matPhang(0.02, 0.6, 0.02, model, maubanghe);

//chan 3

model = Translate(0.375, -0.01, -0.2);

matPhang(0.02, 0.6, 0.02, model, maubanghe);

//chan 4

model = Translate(0.375, -0.01, 0.2);

matPhang(0.02, 0.6, 0.02, model, maubanghe);

//thang doc 1

model = Translate(-0.375, -0.24, 0);

matPhang(0.02, 0.02, 0.4, model, maubanghe);

//thanh doc 2

model = Translate(0.375, -0.24, 0);

matPhang(0.02, 0.02, 0.4, model, maubanghe);

//thanh ngang

model = Translate(0, -0.24, 0);

matPhang(0.78, 0.02, 0.02, model, maubanghe);

//hop may tinh

model = Translate(-0.2, 0.02, 0.05);

matPhang(0.2, 0.35, 0.4, model, maumaytinh);

}

//Máy tính

void MayTinh() {

//De may tinh

model = Translate(0, 0.3, 0);

matPhang(0.3, 0.03, 0.02, model, maumaytinh);

//Than may tinh

model = Translate(0, 0.45, 0);

matPhang(0.05, 0.25, 0.02, model, maumaytinh);

//Man may tinh

model = Translate(0, 0.7, 0);

matPhang(0.35, 0.4, 0.04, model, maumaytinh);

}

//Ghế

GLfloat keoghe = 0;

bool checkkeoghe = false;

void matGhe() {

//phần ngồi

model = Translate(0, 0, -0.5 - keoghe);

matPhang(0.4, 0.02, 0.25, model, maubanghe);

//phần tựa

model = Translate(0, 0.15, -0.62 - keoghe) \* RotateX(90);

matPhang(0.4, 0.02, 0.25, model, maubanghe);

}

void chanGhe() {

//chan 1

model = Translate(-0.175, -0.15, -0.4 - keoghe);

matPhang(0.02, 0.3, 0.02, model, maubanghe);

//chan 2

model = Translate(-0.175, -0.15, -0.6 - keoghe);

matPhang(0.02, 0.3, 0.02, model, maubanghe);

//chan 3

model = Translate(0.175, -0.15, -0.4 - keoghe);

matPhang(0.02, 0.3, 0.02, model, maubanghe);

//chan 4

model = Translate(0.175, -0.15, -0.6 - keoghe);

matPhang(0.02, 0.3, 0.02, model, maubanghe);

}

void caiGhe() {

chanGhe();

matGhe();

}

void caiBan() {

matBan();

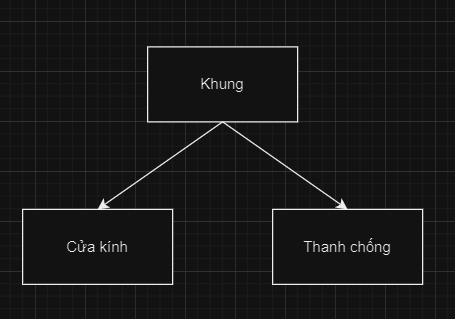
chanBan();

MayTinh();

caiGhe();

}

## 2.6 Kỹ thuật tạo mô hình Cửa sổ



GLfloat quayCuaSo;

int cuaSoState = 0;

void cuaSo() {

//khung doc

model = Translate(0, 0, 0);

matPhang(0.04, 1, 0.03, model, mautu);

model = Translate(0, 0, 0.7);

matPhang(0.04, 1, 0.03, model, mautu);

//khung ngang

model = Translate(0, 0.5, 0.35);

matPhang(0.04, 0.04, 0.72, model, mautu);

model = Translate(0, -0.5, 0.35);

matPhang(0.04, 0.04, 0.72, model, mautu);

//cua kinh

mat4 tmpQuayBase = quayBase;

quayBase \*= Translate(0, 0.5, 0) \* RotateZ(quayCuaSo) \* Translate(0, -0.5, 0) ;

model = Translate(0, 0, 0.04);

matPhang(0.04, 1 - 2 \* 0.04, 0.03, model, maucuatu);

model = Translate(0, 0, 0.66);

matPhang(0.04, 1 - 2 \* 0.04, 0.03, model, maucuatu);

model = Translate(0, 0.5 - 0.04, 0.35);

matPhang(0.04, 0.04, 0.72 - 2\*0.04, model, maucuatu);

model = Translate(0, -0.5 + 0.04, 0.35);

matPhang(0.04, 0.04, 0.72 - 2 \* 0.04, model, maucuatu);

//tay nam cua

model = Translate(-0.08, -0.5 + 0.04, 0.35);

matPhang(0.08, 0.04, 0.04, model, mauden);

model = Translate(-0.08 - 0.08, -0.5 + 0.04, 0.35 + 0.15 \* 0.5 - 0.02);

matPhang(0.08, 0.04, 0.15, model, mauden);

model = Translate(0, 0, 0.35);

matPhang(0.02, 1, 0.7, model, mautrang);

quayBase = tmpQuayBase;

quayBase \*= Translate(0, -0.5, 0) \* RotateZ(-quayCuaSo) \*Translate(0,0.5,0);

model = Translate(0, -0.25, 0.03);

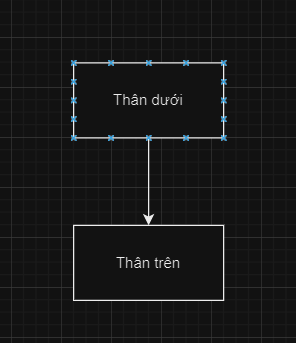
matPhang(0.02, 0.5, 0.01, model, maucuatu);

model = Translate(0, -0.25, 0.67);

matPhang(0.02, 0.5, 0.01, model, maucuatu);

}

## 2.7 Kỹ thuật tạo mô hình Laptop

****

void laptop() {

//than duoi

model = Translate(0, 0, 0);

matPhang(0.6, 0.02, 0.5, model, mautu);

//banphim

model = Translate(0, 0.009, -0.03);

matPhang(0.5, 0.005, 0.25, model, mautrang);

//touchpad

model = Translate(0, 0.009, 0.18);

matPhang(0.25, 0.005, 0.09, model, mautrang);

quayBase \*= Translate(0, 0.02, -0.25) \* RotateX(quayLaptop) \* Translate(0, -0.02, 0.25);

//than tren thanh doc

model = Translate(-0.2925, 0.25, -0.25);

matPhang(0.015, 0.5, 0.015, model, mautu);

model = Translate(0.2925, 0.25, -0.25);

matPhang(0.015, 0.5, 0.015, model, mautu);

//than tren thanh ngang

model = Translate(0, 0.025, -0.25);

matPhang(0.6, 0.05, 0.01, model, mautu);

model = Translate(0, 0.515, -0.25);

matPhang(0.6, 0.03, 0.01, model, mautu);

//man hinh

vec4 mauManHinh = isLaptopOpen ? mautuong : vec4(0, 0, 0, 0);

model = Translate(0, 0.28, -0.25);

matPhang(0.57, 0.45, 0.005, model, mauManHinh);

model = Translate(0, 0.28, -0.255);

matPhang(0.57, 0.455, 0.001, model, mautu);

model = Translate(0, 0.28, -0.255) \* RotateX(90);

hinhTru(0.1, 0.003, 0.1, model, mauden);

}

## 2.8 Kỹ thuật tạo mô hình Cầu dao

GLfloat heighCauDao = 0.4;

GLfloat widthCauDao = 0.3;

GLfloat Doday\_CauDao = 0.02;

GLfloat MoCD = 0;

int CDState = 0;

GLfloat MoHCD = 0;

int HCDState = 0;

void KhungCauDao() {

//Mặt trái Cầu dao

model = Translate(-0.5 \* widthCauDao - 0.5 \* Doday\_CauDao, 0, 0);

matPhang(Doday\_CauDao, heighCauDao, widthCauDao, model, maudieuhoa);

//Mặt phải Cầu dao

model = Translate(0.5 \* widthCauDao + 0.5 \* Doday\_CauDao, 0, 0);

matPhang(Doday\_CauDao, heighCauDao, widthCauDao, model, maudieuhoa);

//Mặt trên Cầu dao

model = Translate(0, 0.5 \* heighCauDao - 0.5 \* Doday\_CauDao, 0);

matPhang(widthCauDao, Doday\_CauDao, widthCauDao, model, maudieuhoa);

//Mặt dưới Cầu dao

model = Translate(0, -0.5 \* heighCauDao + 0.5 \* Doday\_CauDao, 0);

matPhang(widthCauDao, Doday\_CauDao, widthCauDao, model, maudieuhoa);

// mặt sau Cầu dao

model = Translate(0, 0, 0.5 \* widthCauDao - 0.5 \* Doday\_CauDao);

matPhang(widthCauDao, heighCauDao, Doday\_CauDao, model, maudieuhoa);

}

void CuaCauDao() {

model = Translate(widthCauDao\*0.5, 0, -0.5 \*widthCauDao + 0.5 \* Doday\_CauDao) \* RotateY(-MoHCD) \* Translate(-widthCauDao \* 0.5, 0, 0);

matPhang(widthCauDao, heighCauDao, Doday\_CauDao, model, maudieuhoa);

model = Translate(widthCauDao\*0.5, 0, -0.5 \* widthCauDao + 0.5 \* Doday\_CauDao - Doday\_CauDao) \* RotateY(-MoHCD) \* Translate(-widthCauDao \* 0.5, 0,0);

matPhang(0.03, 0.1, 0.01, model, maumanchieu);

}

void ApToMat() {

model = Translate(0, 0, 0.5 \* widthCauDao - 0.5 \* Doday\_CauDao);

matPhang(0.2, 0.15, 0.2, model, maumanchieu);

model = Translate(0, 0, 0.5 \* widthCauDao - 0.5 \* Doday\_CauDao);

matPhang(0.02, 0.4, 0.1, model, maumaychieu);

model = Translate(-0.035, 0, 0.5 \* widthCauDao - 0.5 \* Doday\_CauDao);

matPhang(0.02, 0.4, 0.1, model, maubanghe);

model = Translate(0.035, 0, 0.5 \* widthCauDao - 0.5 \* Doday\_CauDao );

matPhang(0.02, 0.4, 0.1, model, maucua);

model = Translate(0, 0, widthCauDao\*0.5\*0.5) \* RotateX(MoCD) \* Translate(0, 0,-widthCauDao\*0.5\*0.5);

matPhang(0.2, 0.02, 0.05, model, mauden);

}

void CauDao() {

KhungCauDao();

CuaCauDao();

ApToMat();

}

## 2.9 Kỹ thuật tạo mô hình (Căn phòng)

### 2.9.1 Mô hình phân cấp

Căn phòng gồm 17 bộ bàn ghế (1 bàn giáo viên, 16 bàn sinh viên), 1 máy chiếu, 1 màn chiếu, 1 bảng, 1 cửa ra vào, 2 cửa sổ.

Diagram

Description automatically generated

Mô hình phân cấp phòng thực hành

### 2.9.2 Kỹ thuật cài đặt

void canPhong() {

quayBase = Translate(0, 0, z) \* Translate(-0.23, 0, 0) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(0.23, 0, 0) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(0.23, 0, -1.2) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-0.23, 0, -1.2) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-0.23, 0, -2.4) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(0.23, 0, -2.4) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-0.23, 0, -3.6) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(0.23, 0, -3.6) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-1.8, 0, 0) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(1.8, 0, 0) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(1.8, 0, -1.2) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-1.8, 0, -1.2) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-1.8, 0, -2.4) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(1.8, 0, -2.4) \* RotateY(90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(-1.8, 0, -3.6) \* RotateY(-90);

caiBan();

quayBase = Translate(0, 0, z) \* Translate(1.8, 0, -3.6) \* RotateY(90);

caiBan();

//tu

quayBase = Translate(0,0,z) \* Translate(-2.5, 0.6, -6) \* RotateY(180);

tu();

//tuong

quayBase = Translate(0, 0, z);

tuongTrai();

quayBase = Translate(0, 0, z);

tuongPhai();

//san

quayBase = Translate(0, 0, z) ;

san();

//tran nha

quayBase = Translate(0, 0, z) ;

tranNha();

//tuong sau

quayBase = Translate(0, 0, z) ;

tuongSau();

quayBase = Translate(0, 0, z) \* Translate(0,0,9.9);

tuongSau();

//bang

quayBase = Translate(0, 0, z);

bang();

//Man chieu

quayBase = Translate(0, 0, z) \* Translate(0, 2.45 ,0);

manChieu();

//ban giao vien

quayBase = Translate(0, 0, z) \* Translate(1.8, 0, -4.8);

caiBan();

//maychieu

quayBase = Translate(0, 0, z) \* Translate(0.28, 4, -2.25);

mayChieu();

//cua chinh

quayBase = Translate(0, 0, z);

cuaChinh();

//cua so

quayBase = Translate(0, 0, z);

cuaSo();

quayBase = Translate(0, 0, z) \* Translate(0,0,4.48);

cuaSo();

}

## 2.10 Kỹ thuật thay đổi góc nhìn

### 2.10.1 Mô tả kỹ thuật

Lưu ý: các phép biến đổi dưới đây chỉ áp dụng cho ma trận Cột

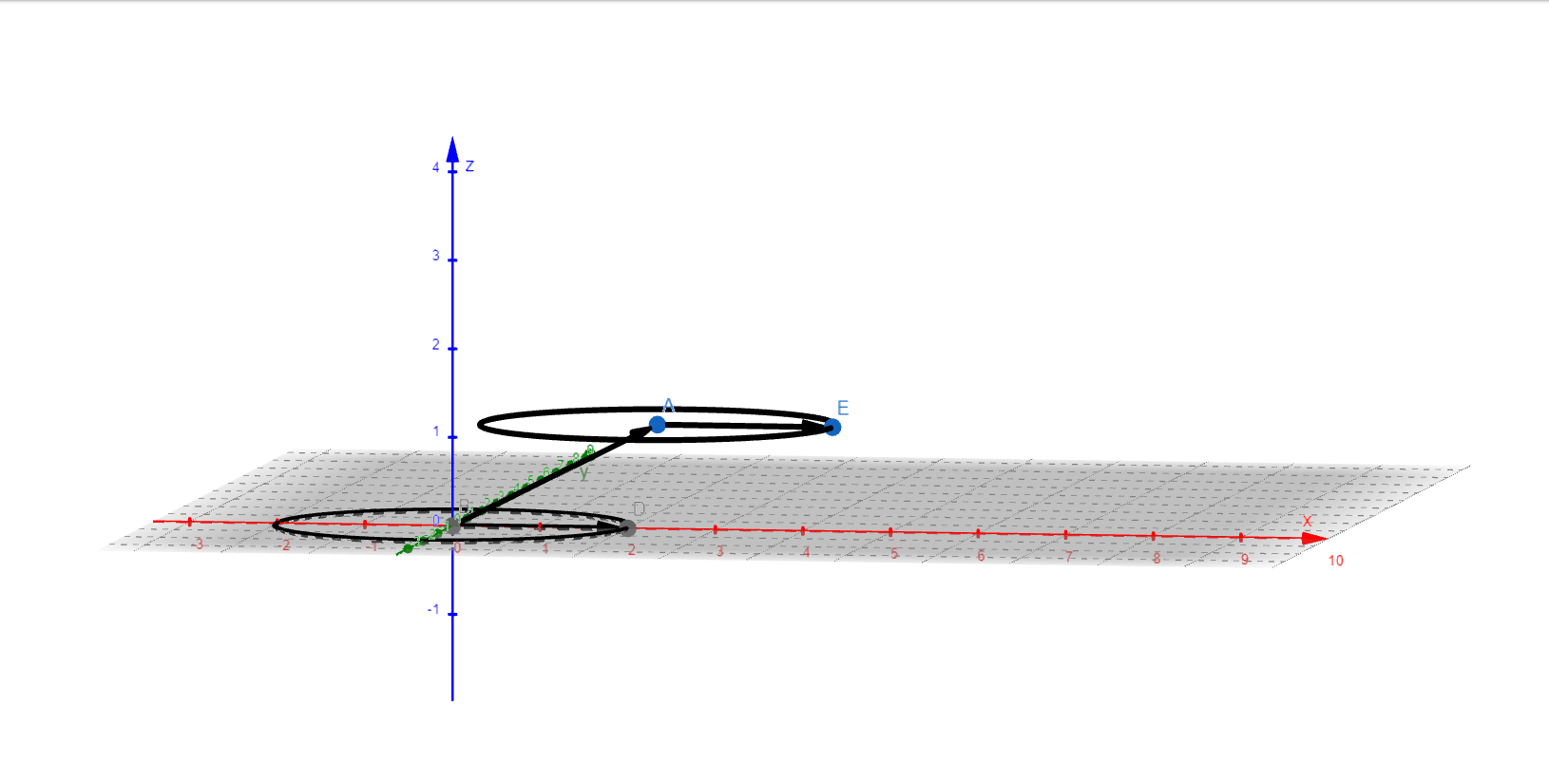
Giả sử trong không gian tọa độ Oxyz có ma trận điểm đặt mắt eye = [x1,y1,z1,1.0] và ma trận điểm nhìn at = [x1,y1,z1,1.0]

Ta có vecto hướng nhìn eye\_at1(eye\_at1x, eye\_at1y, eye\_at1z) = eye - at;

Mặt khác, bằng cách áp dụng một số phép biến đổi, ta có thể thực điều khiển camera bằng các cách sau:

* Để thực hiện di chuyển camera trong hệ tọa độ thực, ta thực hiện phép tịnh tiến Translate(x3,y3,z3) cho vector eye\_at1, ta được vector eye\_at2(eye\_at2x, eye\_at2y, eye\_at2z) = Translate(x3,y3,z3) \* eye\_at1
* Để thực hiện xoay hướng nhìn của camera trong hệ tọa độ thực, ta thực hiện phép quay( theo trục X, Y, Z) Rotate(phi) cho vector eye\_at1, ta được vector eye\_at3(eye\_at3x, eye\_at3y, eye\_at3z) = Rotate(phi) \* eye\_at1

=> Kết hợp 2 cách điều khiển camera trên, ta có thể thực hiện di chuyển camera theo hướng nhìn trong hệ tọa độ thực bằng cách thực hiện phép Rotate(phi) cho vector eye\_at sau đó tịnh tiến vector đến điểm mắt đã tịnh tiến trước đó Translate(eye), ta được vector eye\_at = Translate(eye) \* Rotate(phi) \* eye\_at1



Hình 2: Minh họa cho cách điều khiển camera

### 2.10.2 Kỹ thuật cài đặt

GLfloat l = -1.0, r = 1.0, t = 1.0, b = -1.0, zNear = 0.5, zFar = 6;

GLfloat radius = 1, thetal = 180, phi = 0;

GLfloat dr = 5 \* M\_PI / 180;

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

point4 eye = point4(0, 1, 0, 1.0);

GLfloat xcam, ycam , zcam;

GLfloat xat=0, yat = 180, zat ;

void khungNhin() {

point4 at = point4(0, 0, 1, 1.0);

eye = Translate(eye) \* RotateY(yat) \* point4(xcam, 0, zcam, 1.0);

at = Translate(eye) \* RotateY(yat) \* RotateX(xat) \* at;

xcam = ycam = zcam = 0;

vec4 up(0, 1, 0, 1.0);

mat4 v = LookAt(eye, at, up);

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

mat4 p = Frustum(l, r, b, t, zNear, zFar);

glUniformMatrix4fv(loc\_projection, 1, GL\_TRUE, p);

}

Vshader.glsl

#version 400

in vec4 vPosition;

in vec4 vColor;

in vec3 vNormal;

out vec4 color;

uniform vec4 AmbientProduct, DiffuseProduct, SpecularProduct;

uniform vec4 LightPosition;

uniform float Shininess;

uniform mat4 modelMatrix;

uniform mat4 View;

uniform mat4 Projection;

void main()

{

//Mo hinh chieu sang Blinn - Phong (Phong sua doi)

// Transform vertex position into eye coordinates

vec3 pos = (View \*modelMatrix\* vPosition).xyz;

vec3 L = normalize( LightPosition.xyz - pos );

vec3 E = normalize( -pos );

vec3 H = normalize(L+E);

// Transform vertex normal into eye coordinates

vec3 N = normalize( View \* modelMatrix\* vec4(vNormal, 0.0) ).xyz;

// Compute terms in the illumination equation

vec4 ambient = AmbientProduct;

float Kd = max( dot(L, N), 0.0 );

vec4 diffuse = Kd\*DiffuseProduct;

float Ks = pow( max(dot(N, H), 0.0), Shininess );

vec4 specular = Ks \* SpecularProduct;

if( dot(L, N) < 0.0 ) specular = vec4(0.0, 0.0, 0.0, 1.0);

color = ambient + diffuse + specular;

color.a = 1.0;

gl\_Position = Projection \* View \* modelMatrix \* vPosition/vPosition.w;

}//

## 2.11 Kỹ thuật điều khiển vật thể

void camera\_movement(unsigned char key) {

switch (key) {

case 'd':

xcam -= 0.1;

break;

case 'a':

xcam += 0.1;

break;

case 'w':

zcam += 0.1;

break;

case 's':

zcam -= 0.1;

break;

}

}

void camera\_direction(int key, int a, int b) {

switch (key) {

case GLUT\_KEY\_LEFT:

yat += 2;

xcam = 0;

zcam = 0;

break;

case GLUT\_KEY\_RIGHT:

yat -= 2;

xcam = 0;

zcam = 0;

break;

case GLUT\_KEY\_UP:

xat += 2;

xcam = 0;

zcam = 0;

break;

case GLUT\_KEY\_DOWN:

xat -= 2;

xcam = 0;

zcam = 0;

break;

}

glutPostRedisplay();

}

void keyboard(unsigned char key, int x, int y)

{

// keyboard handler

switch (key) {

//quay cửa - c

case 'c':

if (quayCuaChinh1 < 90)

quayCuaChinh1 += 5;

if (quayCuaChinh2 > -90)

quayCuaChinh2 -= 5;

break;

case 'C':

if (quayCuaChinh1 > 0)

quayCuaChinh1 -= 5;

if (quayCuaChinh2 < 0)

quayCuaChinh2 += 5;

break;

//mo ap to mat - m

case 'm':

moCD(CDState);

isDenOpen = !isDenOpen;

break;

//mo cau dao - n

case 'n':

moHCD(HCDState);

break;

//quay may tinh

case 'j':

anphal -= 5;

if (anphal < -90) {

anphal = -90;

}

break;

case 'g': {

if (!checkkeochau) {

keochau -= 0.02;

if (keochau <= -1.0) {

keochau = -1.0;

checkkeochau = true;

}

}

else {

keochau += 0.02;

if (keochau >= 0) {

keochau = 0;

checkkeochau = false;

}

}

break;

}

case 'J':

anphal += 5;

if (anphal > 0) {

anphal = 0;

}

break;

//mo cua so - v

case 'v':

moCuaso(cuaSoState);

break;

//mo tu - o

case 'o':

motu(tuState);

break;

case '1':

motu1(tuState1);

break;

case '2':

motu2(tuState2);

break;

//gap laptop - l

case 'l':

molaptop(laptopState);

break;

//mo laptop - p

case 'i':

isLaptopOpen = !isLaptopOpen;

break;

//kéo màn chiếu - k

case 'k': {

if (!checkkeo) {

keo += 0.02;

if (keo >= hM) {

keo = hM;

checkkeo = true;

}

}

else {

keo -= 0.02;

if (keo <= 0) {

keo = 0;

checkkeo = false;

}

}

break;

}

//kéo ghế - h

case 'h': {

keoGhe(0);

break;

}

case 'x': l \*= 1.1; r \*= 1.1; break;

case 'X': l \*= 0.9; r \*= 0.9; break;

case 'y': b \*= 1.1; t \*= 1.1; break;

case 'Y': b \*= 0.9; t \*= 0.9; break;

case 'z': zNear \*= 1.1; zFar \*= 1.1; break;

case 'Z': zNear \*= 0.9; zFar \*= 0.9; break;

case 'r': radius \*= 2.0; break;

case 'R': radius \*= 0.5; break;

case 't': thetal += dr; break;

case 'T': thetal -= dr; break;

case 'p': phi += dr; break;

case 'P': phi -= dr; break;

case ' ': // reset values to their defaults

l = -1.0;

r = 1.0;

b = -1.0;

t = 1.0;

zNear = 0.5;

zFar = 6.0;

radius = 1.0;

thetal = 0.0;

phi = 0.0;

//theta[0] = 0;

//theta[1] = 0;

//theta[2] = 0;

break;

}

camera\_movement(key);

glutPostRedisplay();

}

## 2.12 Kỹ thuật chiếu sáng

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

color4 material\_diffuse(1.0, 0.8, 0.0, 1.0);

color4 material\_specular(1.0, 0.8, 0.0, 1.0);

float material\_shininess = 100.0;

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

const int Light\_number = 9;

point4 light\_position[Light\_number];

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

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

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

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

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

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

void shaderSetup(void)

{

// Nạp các shader và sử dụng chương trình shader

program = InitShader("vshader1.glsl", "fshader1.glsl"); // hàm InitShader khai báo trong Angel.h

glUseProgram(program);

// Khởi tạo thuộc tính vị trí đỉnh từ vertex shader

GLuint loc\_vPosition = glGetAttribLocation(program, "vPosition");

glEnableVertexAttribArray(loc\_vPosition);

glVertexAttribPointer(loc\_vPosition, 4, GL\_FLOAT, GL\_FALSE, 0, BUFFER\_OFFSET(0));

GLuint loc\_vColor = glGetAttribLocation(program, "vColor");

glEnableVertexAttribArray(loc\_vColor);

glVertexAttribPointer(loc\_vColor, 4, GL\_FLOAT, GL\_FALSE, 0, BUFFER\_OFFSET(sizeof(points)));

GLuint loc\_vNormal = glGetAttribLocation(program, "vNormal");

glEnableVertexAttribArray(loc\_vNormal);

glVertexAttribPointer(loc\_vNormal, 3, GL\_FLOAT, GL\_FALSE, 0, BUFFER\_OFFSET(sizeof(points) + sizeof(colors)));

loc\_Ambient = glGetUniformLocation(program, "AmbientProduct");

loc\_Diffuse = glGetUniformLocation(program, "DiffuseProduct");

loc\_Specular = glGetUniformLocation(program, "SpecularProduct");

loc\_LightPosition = glGetUniformLocation(program, "LightPosition");

loc\_Shininess = glGetUniformLocation(program, "Shininess");

loc\_modelMatrix = glGetUniformLocation(program, "modelMatrix");

loc\_projection = glGetUniformLocation(program, "Projection");

loc\_view = glGetUniformLocation(program, "View");

glEnable(GL\_DEPTH\_TEST);

glClearColor(1.0, 1.0, 1.0, 1.0); /\* Thiết lập màu trắng là màu xóa màn hình\*/

}

GLfloat z, x;

vec4 mautu = vec4(0.9, 0.9, 0.9, 1.0);

vec4 maucuatu = vec4(0.8, 0.8, 0.8, 1.0);

vec4 mautaykeocuatu = vec4(0.5, 0.5, 0.4, 1.0);

vec4 maumaychieu = vec4(1.0, 0.0, 1.0, 1.0);

vec4 maudieuhoa = vec4(0.6, 0.7, 0.5, 1.0);

vec4 maubang = vec4(0.0, 0.0, 0.0, 1.0);

vec4 maumanchieu = vec4(1.0, 1.0, 1.0, 1.0);

vec4 maucua = vec4(1.0, 1.0, 0.0, 1.0);

vec4 maubanghe = vec4(1.0, 0.0, 0.0, 1.0);

vec4 maumaytinh = vec4(0.9, 0.9, 0.9, 1.0);

vec4 mautuong = vec4(0.0, 1.0, 1.0, 1.0);

vec4 mautrang = vec4(1.0, 1.0, 1.0, 1.0);

vec4 mauden = vec4(0, 0, 0.0, 1.0);

vec4 maulacay = vec4(0.0, 1.0, 0.0, 1.0);

vec4 mauthancay = vec4(0.5, 0.2, 0.0, 1.0);

void matPhang(GLfloat x, GLfloat y, GLfloat z, mat4 mt, vec4 colorCode) {

material\_diffuse = colorCode;

light\_position[0] = point4(0.0, 4.0, 4.0, 1.0);

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

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

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

glUniform4fv(loc\_Ambient, 1, ambient\_product);

glUniform4fv(loc\_Specular, 1, specular\_product);

glUniform4fv(loc\_Diffuse, 1, diffuse\_product);

glUniform4fv(loc\_LightPosition, 1, light\_position[0]);

glUniform1f(loc\_Shininess, material\_shininess);

mat4 ins = Scale(x, y, z);

glUniformMatrix4fv(loc\_modelMatrix, 1, GL\_TRUE, quayBase \* mt \* ins);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

// Khởi tạo khối hộp

void cube(point3 model , mat4 mt, color4 material\_ambient, color4 material\_diffuse, color4 material\_specular, float material\_shininess, point4 light\_position) {

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

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

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

glUniform4fv(loc\_Ambient, 1, ambient\_product);

glUniform4fv(loc\_Specular, 1, specular\_product);

glUniform4fv(loc\_Diffuse, 1, diffuse\_product);

glUniform4fv(loc\_LightPosition, 1, light\_position);

glUniform1f(loc\_Shininess, material\_shininess);

mat4 ins = Scale(model);

glUniformMatrix4fv(loc\_modelMatrix, 1, GL\_TRUE, quayBase \* mt \* ins);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

// Khởi tạo khối hộp

bool light\_state = true;

void cube(GLfloat x, GLfloat y, GLfloat z, mat4 mt, color4 material\_ambient, color4 material\_diffuse, color4 material\_specular, float material\_shininess, point4\* light\_position) {

if (light\_state) {

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

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

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

}

else {

light\_ambient = color4(0, 0, 0, 1.0);

light\_diffuse = color4(0, 0, 0, 1.0);

light\_specular = color4(0, 0, 0, 1.0);

}

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

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

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

glUniform4fv(loc\_Ambient, 1, ambient\_product);

glUniform4fv(loc\_Specular, 1, specular\_product);

glUniform4fv(loc\_Diffuse, 1, diffuse\_product);

for (int i = 0; i < sizeof(light\_position); i++) {

glUniform4fv(loc\_LightPosition, i, light\_position[i]);

}

glUniform1f(loc\_Shininess, material\_shininess);

mat4 ins = Scale(x, y, z);

glUniformMatrix4fv(loc\_modelMatrix, 1, GL\_TRUE, quayBase \* mt \* ins);

glDrawArrays(GL\_TRIANGLES, 0, NumPoints);

}

void hinhTru(GLfloat x, GLfloat y, GLfloat z, mat4 mt, vec4 colorCode) {

material\_diffuse = colorCode;

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

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

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

glUniform4fv(loc\_Ambient, 1, ambient\_product);

glUniform4fv(loc\_Specular, 1, specular\_product);

glUniform4fv(loc\_Diffuse, 1, diffuse\_product);

glUniform4fv(loc\_LightPosition, 1, light\_position[0]);

glUniform1f(loc\_Shininess, material\_shininess);

mat4 ins = Scale(x, y, z);

glUniformMatrix4fv(loc\_modelMatrix, 1, GL\_TRUE, quayBase \* mt \* ins);

glDrawArrays(GL\_TRIANGLES, NumPoints, diemHinhTru);

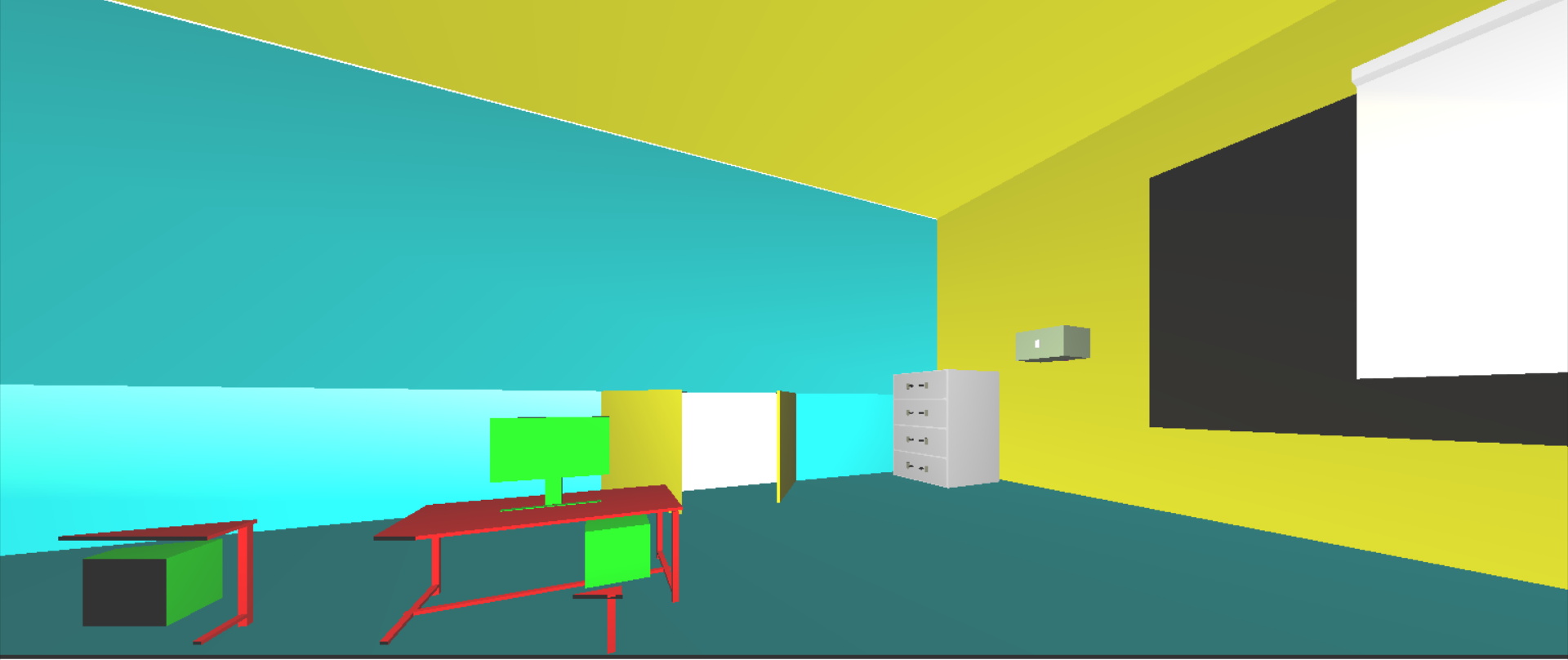
}

# 

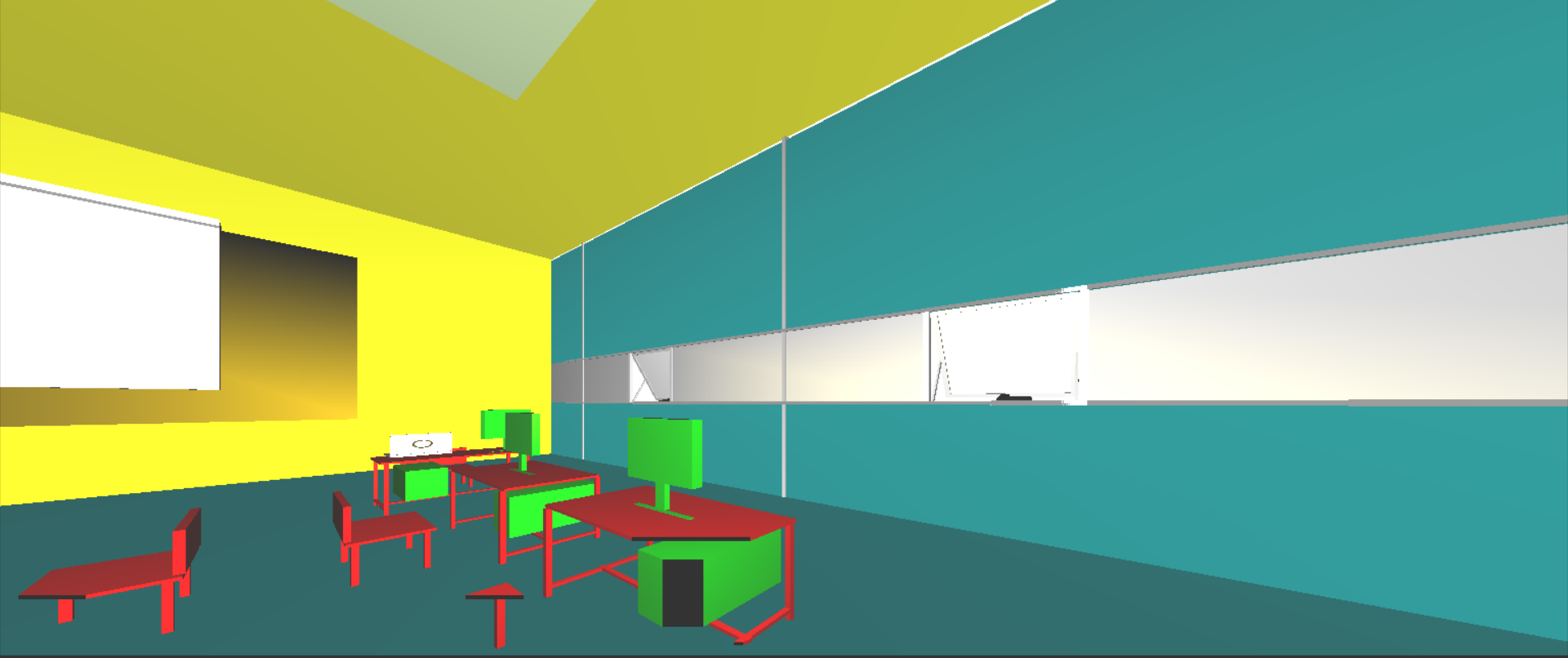
# CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH.



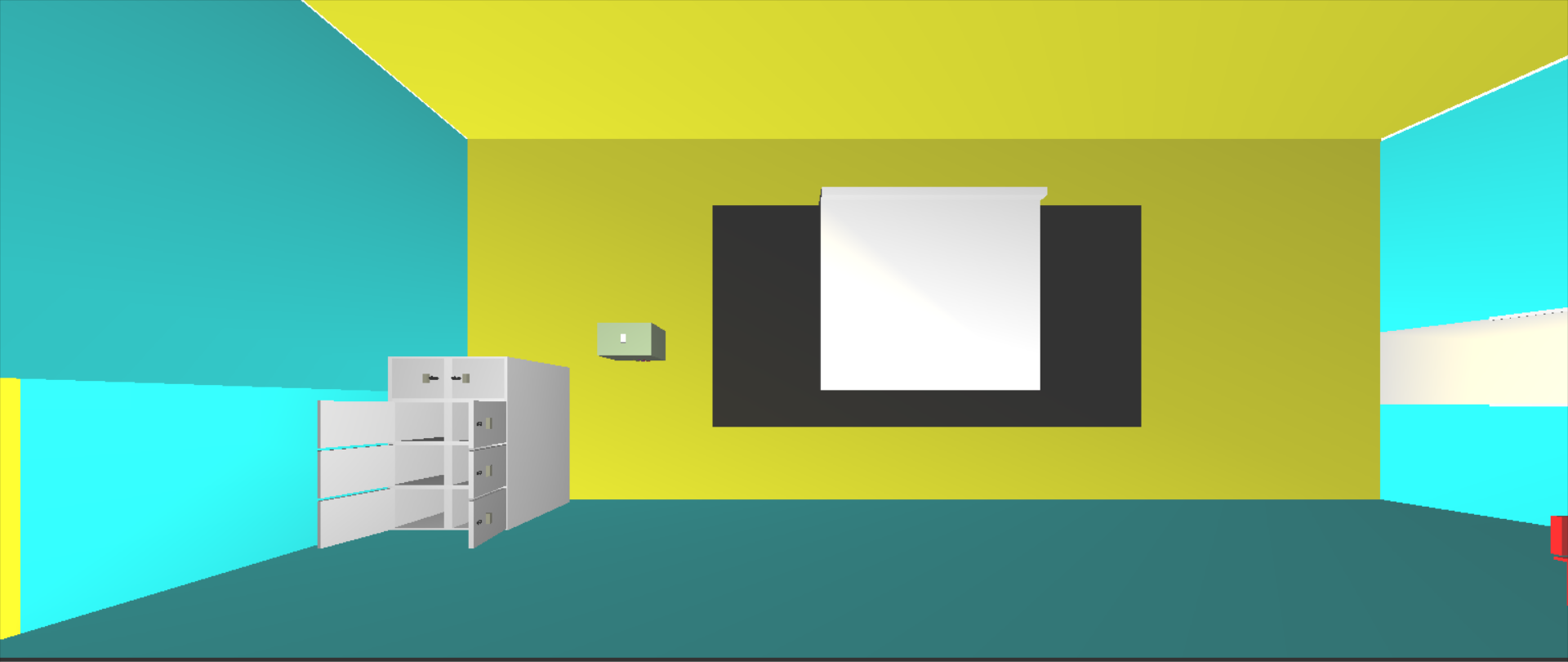
Hình 1: Phòng thực hành



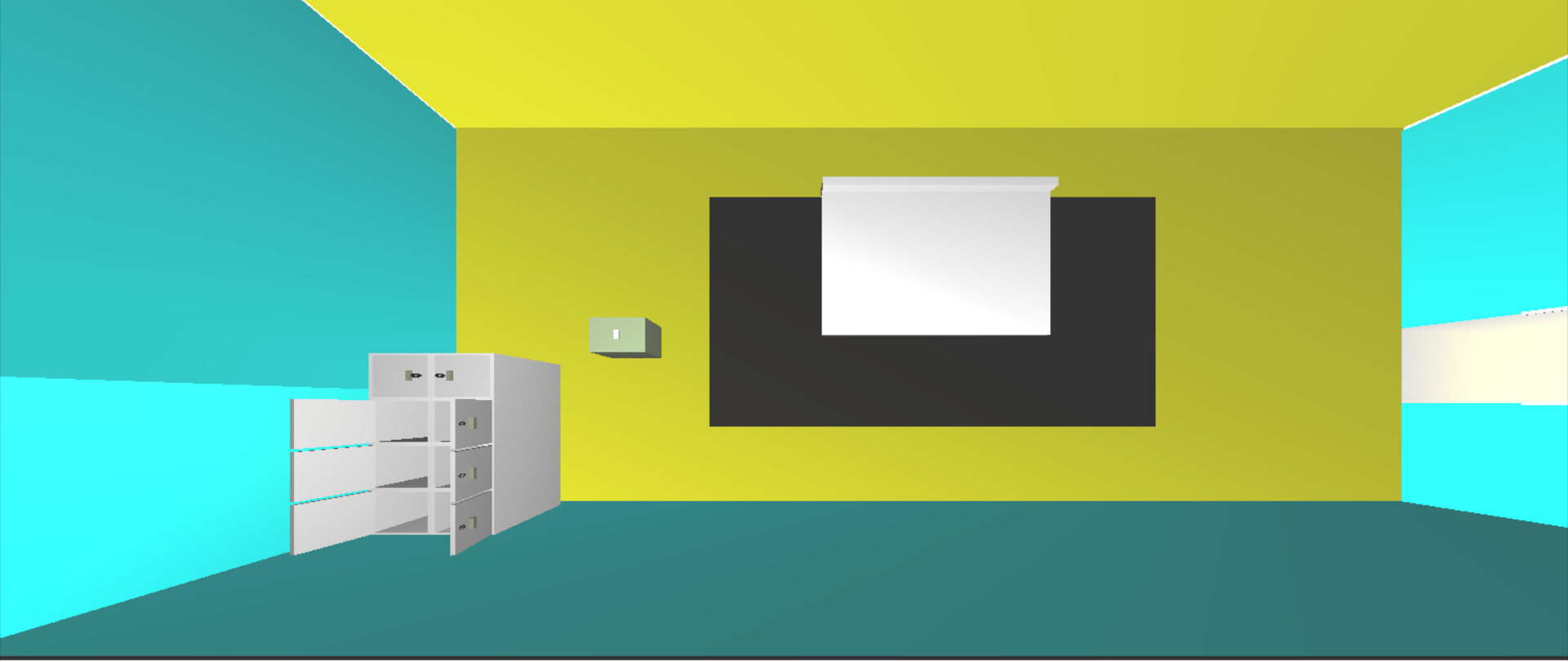
Hình 2. Mô phỏng đóng mở cửa chính



Hình 3. Mô phỏng đóng mở cửa sổ



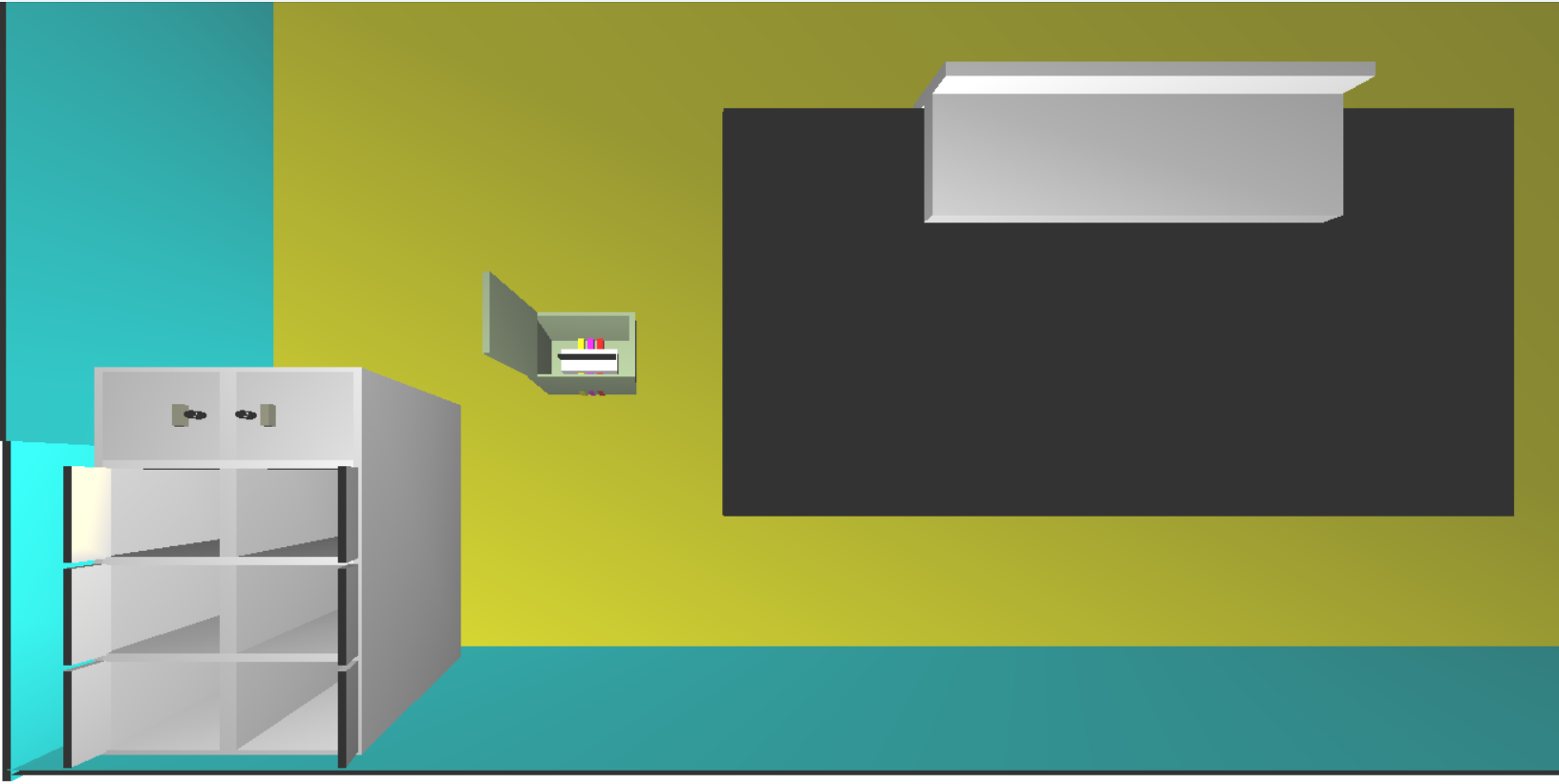
Hình 4. Mô phỏng đóng mở cửa tủ



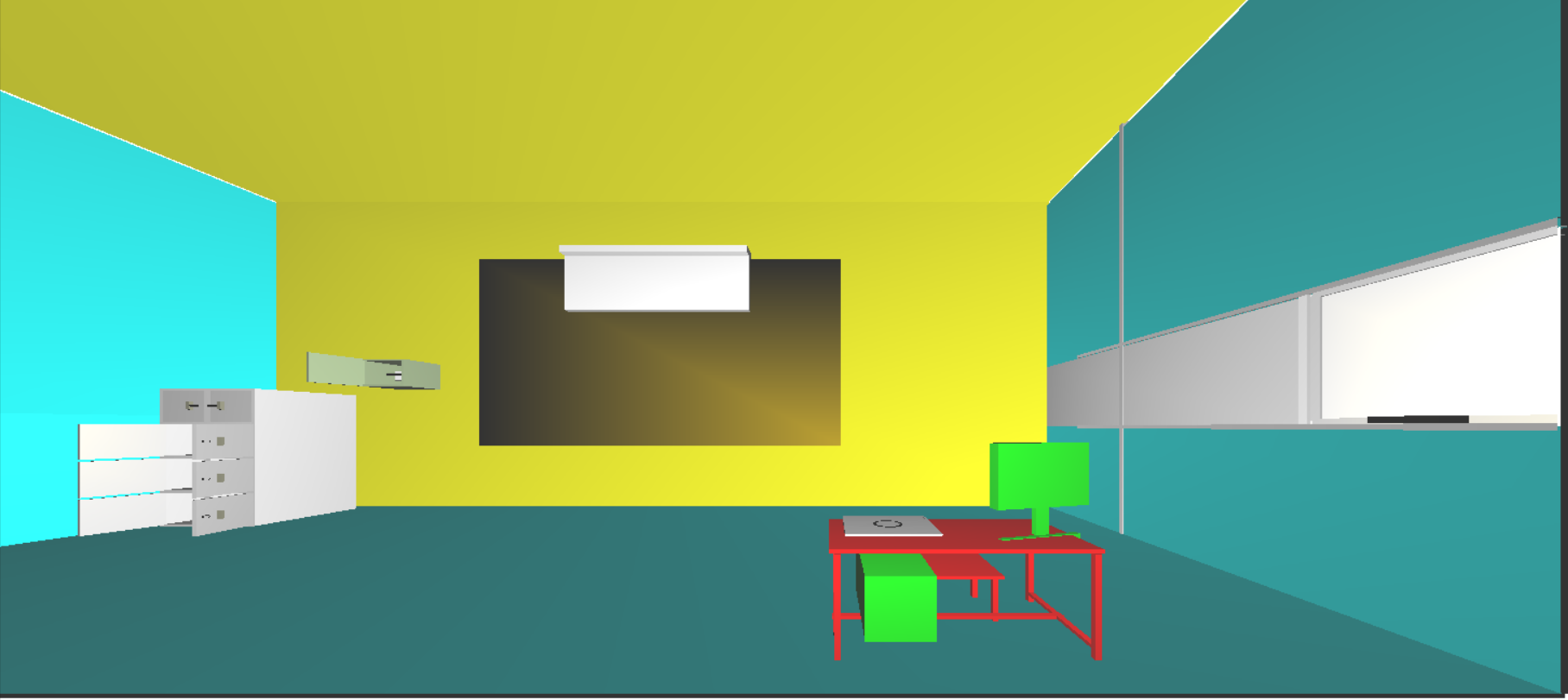
Hình 5. Mô phỏng kéo màn chiếu



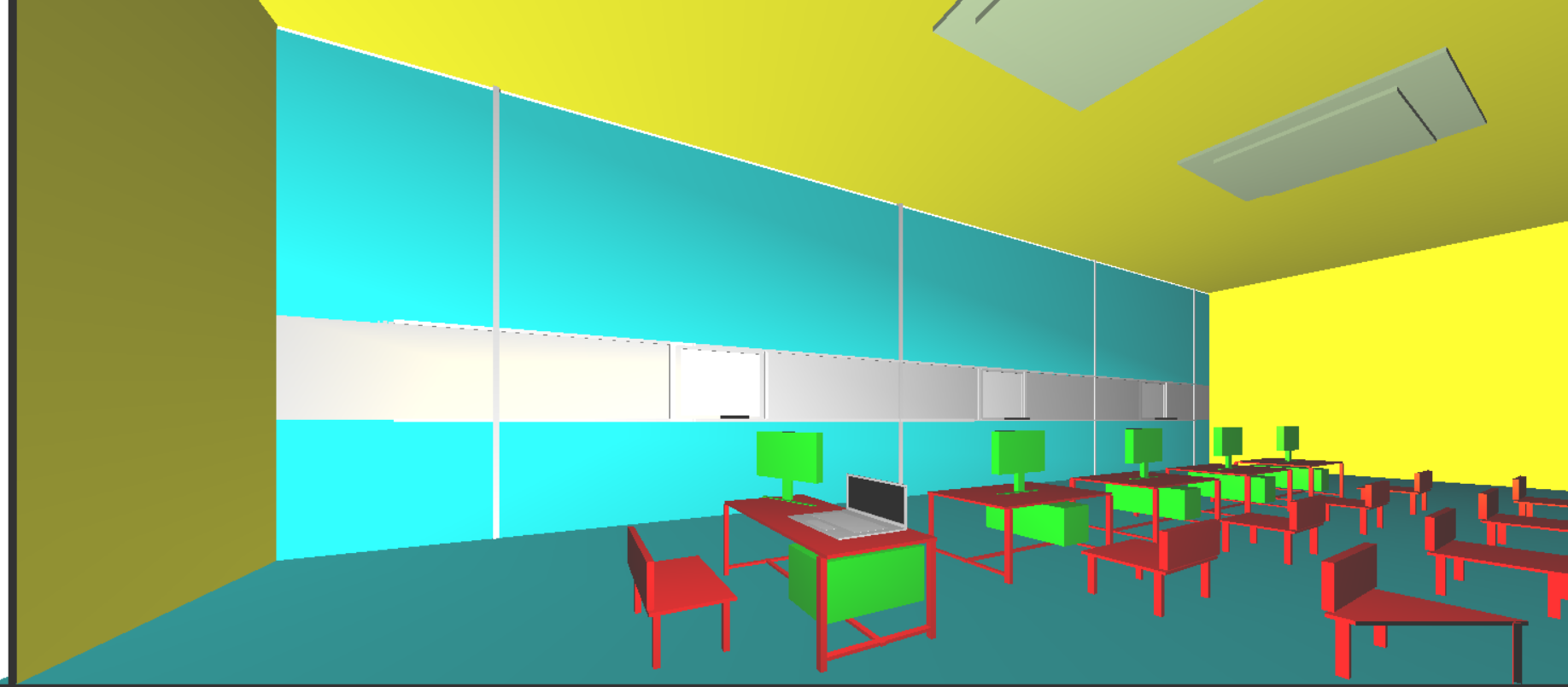
Hình 7. Mô phỏng kéo ghế ra vào.



Hình 8. Mô phỏng cầu dao điện.



Hình 9:Mô phỏng đóng/mở laptop



Hình 10:Mô phỏng bật/tắt màn hình

# 

# TỔNG KẾT

Qua việc thực hiện nghiên cứu đề tài “Mô phỏng phòng thực hành tầng 8 toà A1 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. Trong quá trình xây dựng chương trình nhóm chúng em khó tránh khỏi những sai sót. 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 kỹ năng lập trình.

Một lần nữa chúng em xin chân thành cảm ơn !

**Tài liệu tham khảo**

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