

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

Bài tập nhóm Đồ họa máy tính



Tên đề tài

Mô tả một căn phòng khách với bàn ghế , TV , tủ , quạt

CBHD : Th.s Vũ Minh Yến

Lớp : 2020IT601.2

Nhóm số : 1

Sinh viên : 2018602242 – Nguyễn Quý Long

2018603478 – Nguyễn Đức Long

2018603626 – Nguyễn Tiến Mạnh

Hà Nội – 2020

Mục lục

Chương I : Mô tả bài toán.....	3
1.1. Giới thiệu đề tài :	3
1.2. Mô tả mô hình và kịch bản :	3
1.2.1 Mô tả mô hình :	3
1.2.2. Kịch bản :	4
Chương II. Một số kỹ thuật diễn hình phát triển chương trình.....	5
2.1. Phát triển vị trí , hình dạng các đối tượng (model) :	5
2.2. Thiết kế và cài đặt đối tượng : Căn phòng phòng (không gian bao quanh và các thiết bị trong phòng)	6
2.2.1. Ý tưởng :	6
2.2.2. Thiết kế	6
2.3. Thiết kế và cài đặt đối tượng : Không gian phòng (nền nhà và tường).....	6
2.3.1. Ý tưởng :	6
2.3.2. Thiết kế :	7
2.3.2. Cài đặt :	7
2.4 : Thiết kế và cài đặt đối tượng : Bộ bàn ghế (bàn và ghế)	9
2.4.1. Ý tưởng thiết kế chiếc bàn :	9
2.4.2. Thiết kế chiếc bàn:	9
2.4.3.Cài đặt chiếc bàn:	9
2.4.4. Ý tưởng thiết kế chiếc ghế:	10
2.4.5. Thiết kế chiếc ghế :	11
2.4.6. Cài đặt chiếc ghế :	11
2.5. : Thiết kế và cài đặt đối tượng : Quạt bàn	13
2.5.1. Ý tưởng :	13
2.5.2. Thiết kế chiếc quạt	13
2.5.3. Cài đặt quạt :	14
2.6. Thiết kế và cài đặt đối tượng : Tủ và TV	17
2.6.1. Ý tưởng thiết kế tủ:	17
2.6.2. Thiết kế tủ:	17

2.6.3. Cài đặt chiếc tủ :	18
2.6.4. Ý tưởng thiết kế chiếc TV :	18
2.6.5. Thiết kế chiếc TV :	18
2.6.6. Cài đặt chiếc TV :	18
2.7. Cài đặt phép chiếu :	20
2.7.1. Thực hiện phép chiếu phối cảnh :	20
2.7.2. Cài đặt phép chiếu phối cảnh	21
2.7.3 Thực hiện phép chiếu song song	21
2.7.4 Cài đặt phép chiếu song song	22
CHƯƠNG 3 : KẾT QUẢ CHƯƠNG TRÌNH	23
Một số hình ảnh chương trình	24

Chương I : Mô tả bài toán

1.1. Giới thiệu đề tài :

- Bài tập mô tả lại bối cảnh cơ bản của một căn phòng khách với một bộ bàn ghế , một chiếc vô tuyến (TV) được kê trên một chiếc tủ có ngăn kéo và một chiếc quạt bàn có thể chạy

1.2. Mô tả mô hình và kịch bản :

1.2.1 Mô tả mô hình :



Hình 1.1. Hình minh họa mô hình căn phòng khách

1.2.2. Kịch bản :

- Không gian giới hạn căn phòng (nền nhà và tường) là 3 bức tường tạo thành 3 hình lập phương đơn vị được biến đổi . Khi không gian phòng di chuyển thì tất cả các đồ dùng trong phòng đều di chuyển theo.
- Chiếc bàn được tạo nên từ 2 hình lập phương đơn vị được biến đổi (1 hình tạo mặt bàn và 1 hình tạo chân bàn).
- Chiếc quạt được tạo nên từ 5 hình lập phương đơn vị được biến đổi (đế quạt – nút bấm – thân dưới quạt – thân trên quạt – cánh quạt) thiết kế theo mô hình phân cấp. Quạt có thể di chuyển , nâng cao phần thân dưới , quay phần thân trên , xoay cánh quạt.
- Chiếc tủ gồm có khung và các ngăn kéo , nằm trên nó là một chiếc TV gồm đế, giá đỡ , màn hình. Khi tủ chuyển động thì TV sẽ chuyển động theo , chiếc TV có thể gập màn hình .

Chương II. Một số kỹ thuật diễn hình phát triển chương trình

2.1. Phát triển vị trí, hình dạng các đối tượng (model) :

Ý tưởng cài đặt trung :

+ Khai báo biến uniform có kiểu dữ liệu mat4 model trong file vshader.glsl (chương trình xử lý trên từng đỉnh)

+ Sử dụng biến ma trận (mat4 – thư viện Angel.h) instance để khởi tạo việc biến đổi các đối tượng từ khối lập phương đơn vị , biến ma trận (mat4 – thư viện Angel.h) model để biến đổi đối tượng đã được khởi tạo từ ma trận instance. Dữ liệu từ 2 ma trận sẽ được truyền vào biến model trong file vshader.glsl để xử lý.

Các từ viết tắt trong bài :

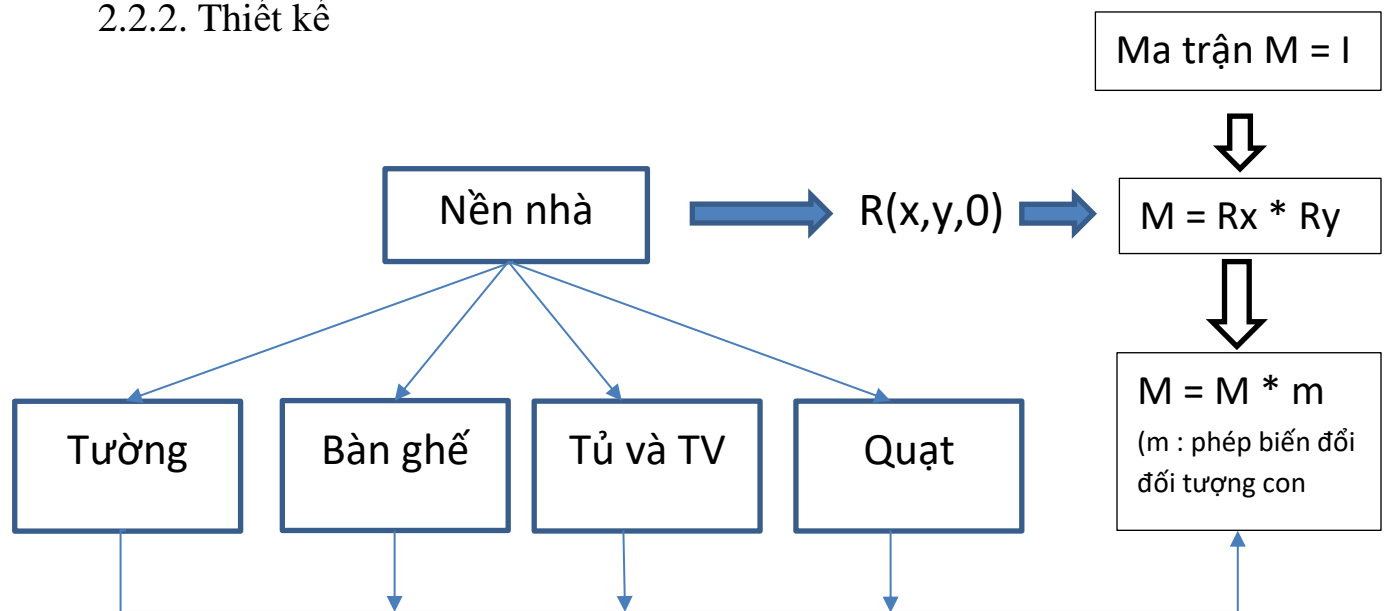
Từ viết tắt	Ý nghĩa
M	Ma trận 4x4 lưu các phép biến đổi
I	Ma trận đơn vị 4x4
T(x,y,z)	Phép tịnh tiến theo trục x, y, z
S(x,y,z)	Phép biến đổi tỷ lệ theo trục x, y, z
Rx,Ry,Rz	Phép xoay tại tâm O theo trục x, y, z

2.2. Thiết kế và cài đặt đối tượng : Căn phòng phòng (không gian bao quanh và các thiết bị trong phòng)

2.2.1. Ý tưởng :

+ Toàn bộ căn phòng là một mô hình phân cấp . Nền nhà là đối tượng cha , tường và các thiết bị trong phòng là đối tượng con . Khi đối tượng cha chuyển động , các đối tượng con sẽ bị chuyển động kéo theo nhưng không ngược lại.

2.2.2. Thiết kế



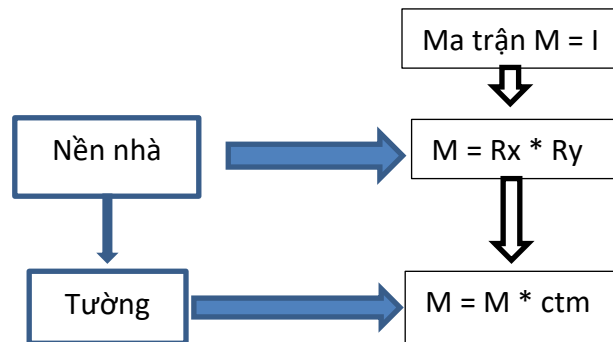
Hình 2.1. Thiết kế phân cấp cho toàn bộ căn phòng

2.3. Thiết kế và cài đặt đối tượng : Không gian phòng (nền nhà và tường)

2.3.1. Ý tưởng :

Nền nhà là một thể hiện của hình lập phương đơn vị , tường là một thể hiện của nền nhà được xoay theo trục x và z , là mô hình con của nền nhà.

2.3.2. Thiết kế :



Hình 2.2. Thiết kế phân cấp cho không gian phòng

2.3.2. Cài đặt :

// Thiết lập kích thước

```
GLfloat surface_width = 0.75, surface_height = 0.1, surface_depth = 0.75;
```

```
GLfloat wall_width = 0.75, wall_height = 0.75, wall_depth = 0.1;
```

```
GLfloat wall_theta[3] = { 0.0, 0.0, 0.0 };
```

// Vẽ nền nhà

```
void surface() {
```

```
    instance = Scale(surface_width * 2, surface_height , surface_depth * 2);
```

```
    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
```

```
    glDrawArrays(GL_TRIANGLES, 0, NumPoints);
```

```
}
```

// Vẽ 1 bức tường

```
void wall() {
```

```
    instance = Scale(wall_width * 2, wall_height , wall_depth) * RotateY(270);
```

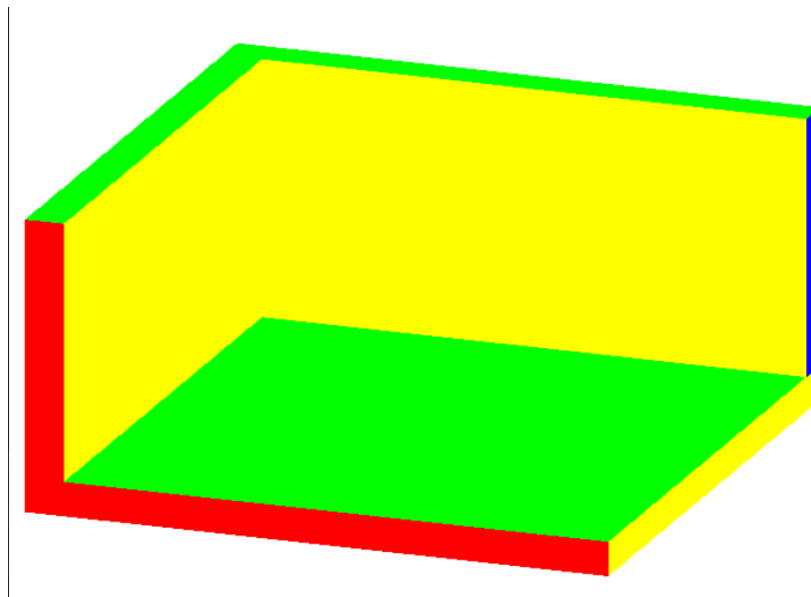


```

        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);
    }

    // Vẽ không gian bao quanh
    void makeSpace() {
        model = model * Translate(0, -wall_height* 0.5 - surface_height
        * 0.5, 0) * RotateX(180) ;
        surface();
        mat4 space = model;
        model = space * Translate(-surface_width + wall_depth * 0.5, -
        wall_height * 0.5 - surface_height * 0.5, 0) * RotateY(90);
        wall();
        model = space * Translate(0, -wall_height * 0.5 -
        surface_height * 0.5, -surface_depth + wall_depth * 0.5);
        wall();
    }
}

```



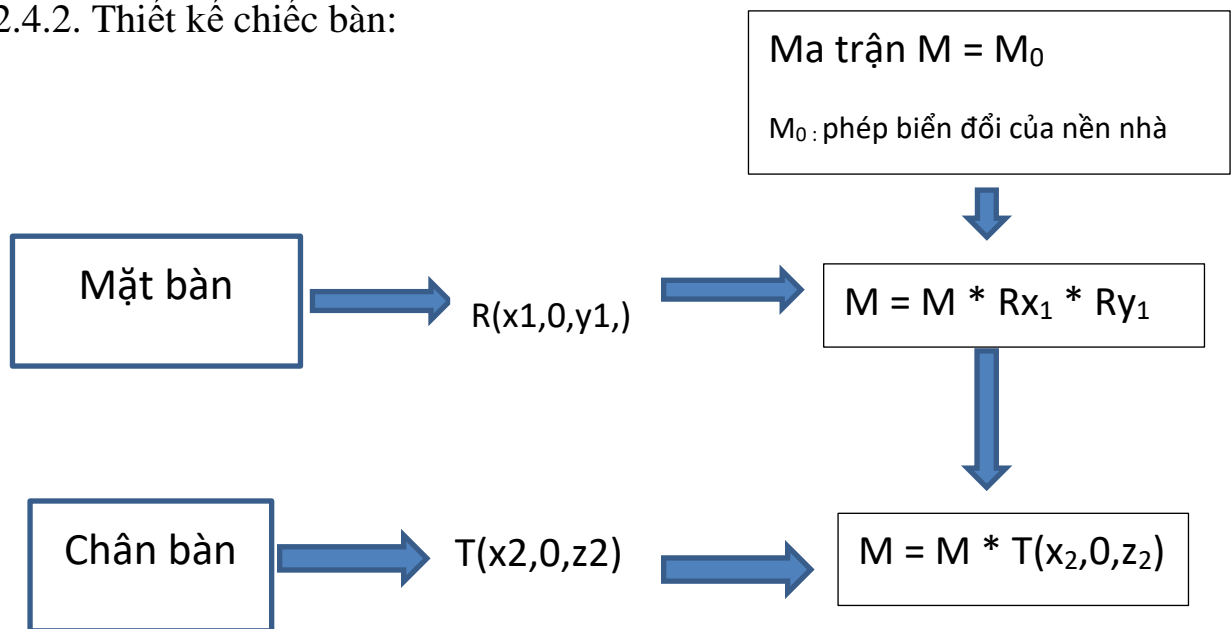
Hình 2.3: Không gian bao quanh sau khi cài đặt

2.4 : Thiết kế và cài đặt đối tượng : Bộ bàn ghế (bàn và ghế)

2.4.1. Ý tưởng thiết kế chiếc bàn :

Bàn bao gồm mặt bàn và một chân bàn được nhân bản làm 4

2.4.2. Thiết kế chiếc bàn:



Hình 2.4. Thiết kế phân cấp cho chiếc bàn

2.4.3.Cài đặt chiếc bàn:

```
void Table() {  
    model = model * Translate(0, -0.25 * (table_post_height +  
table_surface_height), 0) * Scale(0.5, 0.5, 0.5);  
  
    mat4 table = model;  
    table_surface();  
    model = table * Translate(-0.5 * (table_surface_width -  
table_post_width), 0, -0.5 * (table_surface_depth -  
table_post_depth));  
}
```

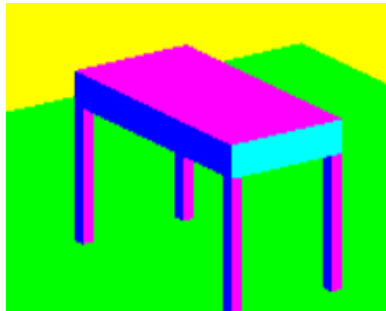
```

    table_post();

    model = table * Translate(0.5 * (table_surface_width -
table_post_width), 0, -0.5 * (table_surface_depth -
table_post_depth)) ;
    table_post();

    model = table * Translate(0.5 * (table_surface_width -
table_post_width), 0, 0.5 * (table_surface_depth - table_post_depth))
;
    table_post();
    model = table * Translate(-0.5 * (table_surface_width -
table_post_width), 0, 0.5 * (table_surface_depth - table_post_depth))
;
    table_post();
}

```

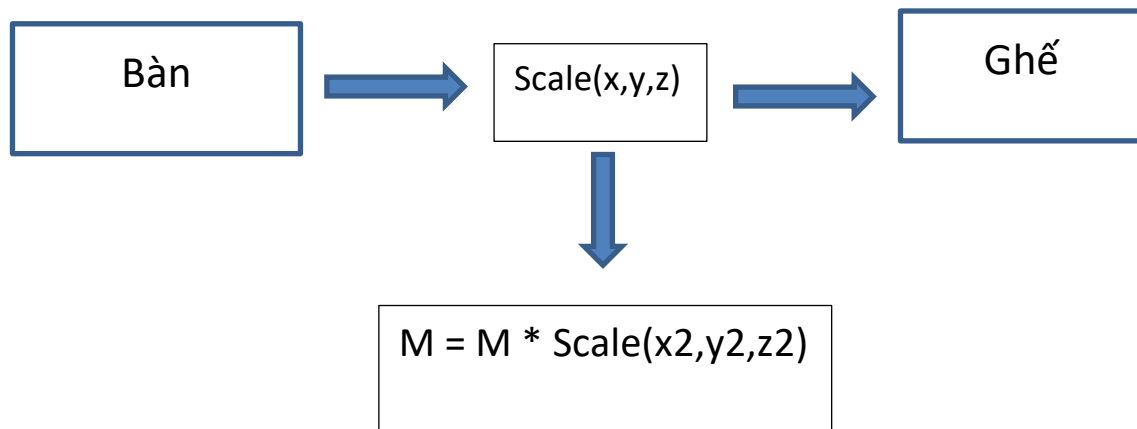


Hình 2.5. Chiếc bàn sau khi cài đặt

2.4.4. Ý tưởng thiết kế chiếc ghế:

Ghế là một chiếc bàn được biến đổi , coi ghế là con của bàn.

2.4.5. Thiết kế chiếc ghế :



Hình 2.6. Quá trình biến đổi một chiếc ghế từ một chiếc bàn

2.4.6. Cài đặt chiếc ghế :

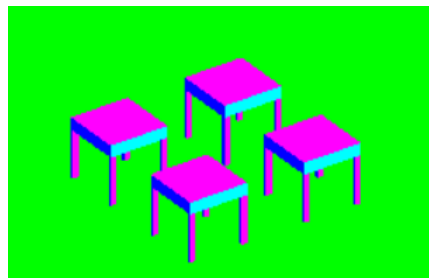
```
void Chair() {  
    model = model * Translate(0, -table_post_height * 0.25-  
table_post_height * 0.125, 0) * Scale(0.25, 0.5, 0.5);  
    Table();  
}  
  
void drawChairs(){  
  
model = ctm * Translate(-table_surface_width * 0.125 ,0,-  
table_surface_depth*0.25) *Translate(table_surface_width * 0.25, 0, -  
surface_depth * 0.25) * RotateY(table_theta) *  
Translate(chair_tranX,0,chair_tranZ);  
    Chair();  
    model = ctm * Translate(table_surface_width * 0.125, 0, -  
table_surface_depth * 0.25) * Translate(table_surface_width * 0.25,  
0, -surface_depth * 0.25) * RotateY(table_theta) * Translate(-  
chair_tranX, 0, chair_tranZ);  
    Chair();  
}
```

```

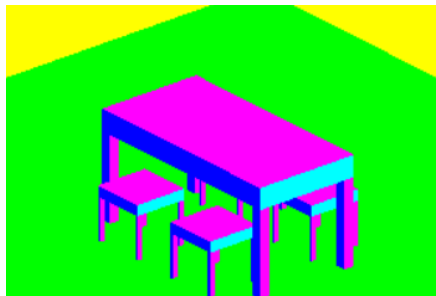
        model = ctm * Translate(-table_surface_width * 0.125, 0,
table_surface_depth * 0.25) * Translate(table_surface_width * 0.25,
0, -surface_depth * 0.25) * RotateY(table_theta) *
Translate(chair_tranX, 0, -chair_tranZ);
        Chair();

        model = ctm * Translate(table_surface_width * 0.125, 0,
table_surface_depth * 0.25) * Translate(table_surface_width * 0.25,
0, -surface_depth * 0.25) * RotateY(table_theta) * Translate(-
chair_tranX, 0, -chair_tranZ);
        Chair();
}

```



Hình 2.7 : Bộ ghế sau khi cài đặt

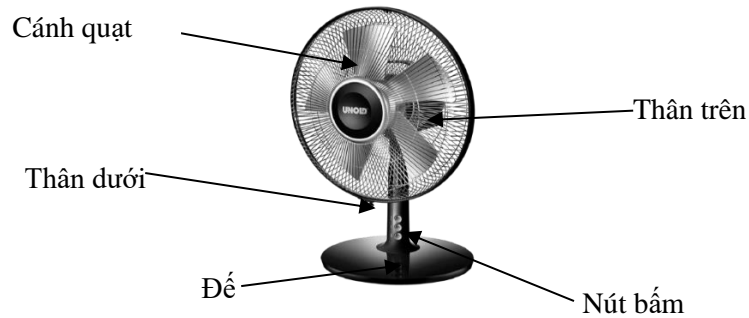


Hình 2.8 : Bộ bàn ghế sau khi cài đặt

2.5. : Thiết kế và cài đặt đối tượng : Quạt bàn

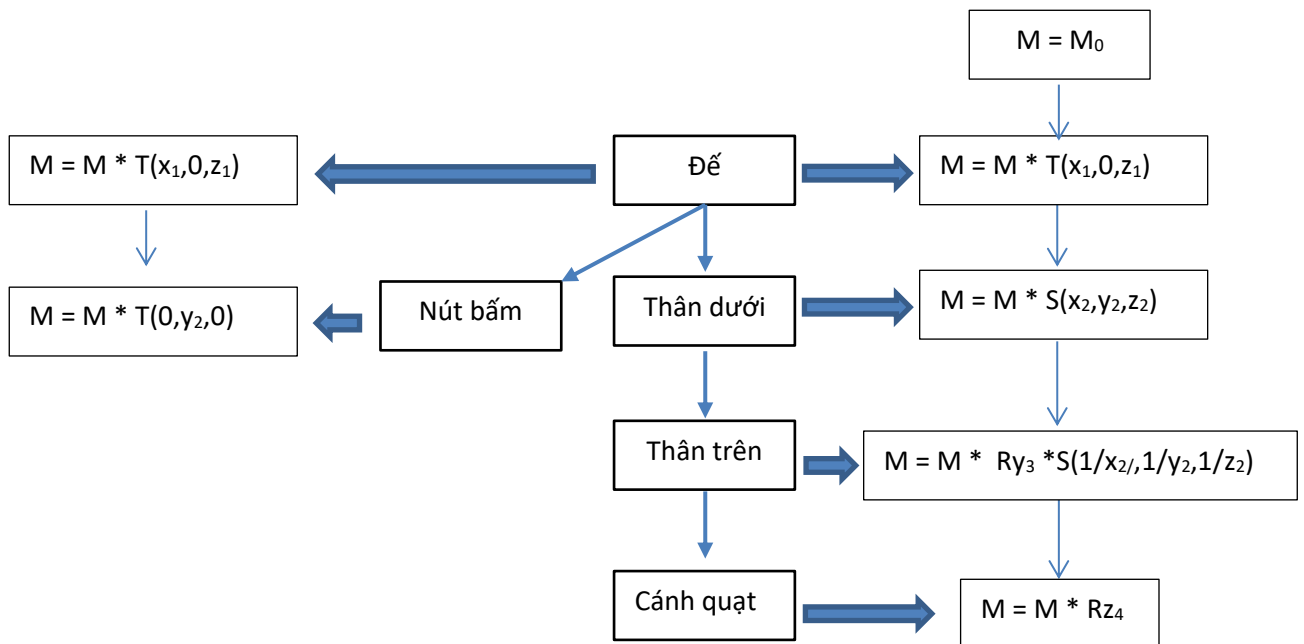
2.5.1. Ý tưởng :

Quạt bàn gồm 5 bộ phận : đế , nút , thân dưới , thân trên , cánh quạt



Hình 2.9. Các bộ phận của một chiếc quạt

2.5.2. Thiết kế chiếc quạt



Hình 2.10. Thiết kế phân cấp cho chiếc quạt

2.5.3. Cài đặt quạt :

```
GLfloat fan_base_width = 0.45, fan_base_height = 0.1, fan_base_depth  
= 0.5;
```

```
GLfloat button_width = 0.05, button_height = 0.1, button_depth =  
0.07;
```

```
GLfloat fan_lower_width = 0.15, fan_lower_height = 0.3,  
fan_lower_depth = 0.1;
```

```
GLfloat fan_upper_width = 0.15, fan_upper_height = 0.1,  
fan_upper_depth = 0.3;
```

```
GLfloat fan_width = 0.5, fan_height = 0.1, fan_depth = 0.025;
```

```
GLfloat fan_theta[3] = { 0.0,0.0,0.0 };
```

```
GLfloat fan_tranX = 0, fan_tranZ = 0;
```

```
GLfloat fan_scale = 1.0;
```

```
GLfloat spin_speed, spin_time;
```

```
GLfloat fan_push[4] = { 0.0,0.0,0.0,0.0 };
```

```
GLboolean isTurn = true;
```

```
GLint fan_state = 0;
```

```
void fan_base() {
```

```
    instance = Translate(0, fan_base_height * 0.5, 0) *
```

```
Scale(fan_base_width, fan_base_height, fan_base_depth);
```

```
    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
```

```
    glDrawArrays(GL_TRIANGLES, 0, NumPoints);
```

```
}
```

```
void button() {
```

```

        instance = Translate(-fan_base_width * 0.4, fan_base_height +
fan_push[0], -fan_base_depth * 0.25) * Scale(button_width,
button_height, button_depth) * RotateY(180);
        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);

        instance = Translate(-fan_base_width * 0.15, fan_base_height +
fan_push[1], -fan_base_depth * 0.25) * Scale(button_width,
button_height, button_depth) * RotateX(-90) * RotateY(90);
        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);

        instance = Translate(fan_base_width * 0.10, fan_base_height +
fan_push[2], -fan_base_depth * 0.25) * Scale(button_width,
button_height, button_depth) * RotateX(-90) * RotateY(90);
        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);

        instance = Translate(fan_base_width * 0.35, fan_base_height +
fan_push[3], -fan_base_depth * 0.25) * Scale(button_width,
button_height, button_depth) * RotateX(-90) * RotateY(90);
        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);
    }
    void fan_lower() {
        instance = Translate(0, fan_lower_height * 0.5, 0) *
Scale(fan_lower_width, fan_lower_height, fan_lower_depth) *
RotateZ(90);
        glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
        glDrawArrays(GL_TRIANGLES, 0, NumPoints);
    }

```



```

}
void fan_upper() {
    instance = Translate(0, fan_upper_height * 0.5, 0) *
Scale(fan_upper_width, fan_upper_height, fan_upper_depth) *
RotateX(90) * RotateY(90);
    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
    glDrawArrays(GL_TRIANGLES, 0, NumPoints);
}
void fan() {
    instance = Translate(0, 0, -fan_upper_depth * 0.5) *
RotateZ(45) * Scale(fan_width, fan_height, fan_depth) * RotateY(180);
    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
    glDrawArrays(GL_TRIANGLES, 0, NumPoints);

    instance = Translate(0, 0, -fan_upper_depth * 0.5) * RotateZ(-
45) * Scale(fan_width, fan_height, fan_depth) * RotateY(180);
    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);
    glDrawArrays(GL_TRIANGLES, 0, NumPoints);
}

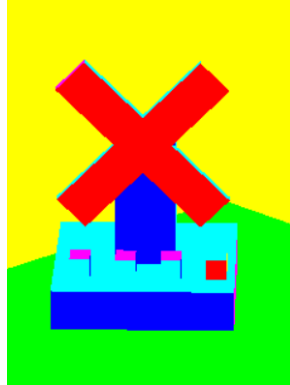
void drawFan() {
    model = model * Translate(fan_tranX, -wall_height*0.5,
fan_tranZ) * Scale(0.4, 0.4, 0.4) * RotateY(-45) ;
    fan_base();
    button();
    model = model * Translate(0, fan_base_height, 0) * Scale(1,
fan_scale, 1);
    fan_lower();
    model = model * Translate(0, fan_lower_height, 0) *
RotateY(fan_theta[1]) * Scale(1, 1 / fan_scale, 1);
    fan_upper();
}

```

```

        model = model * Translate(0, fan_upper_height * 0.5, 0) *
RotateZ(fan_theta[2]*2);
        fan();
    }

```



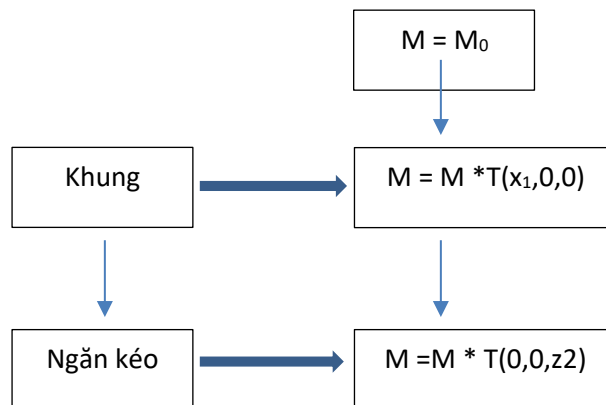
Hình 2.11. Chiếc quạt sau khi được cài đặt

2.6. Thiết kế và cài đặt đối tượng : Tủ và TV

2.6.1. Ý tưởng thiết kế tủ:

Chiếc tủ bao gồm : khung và ngăn kéo , ngăn kéo là con của khung tủ

2.6.2. Thiết kế tủ:



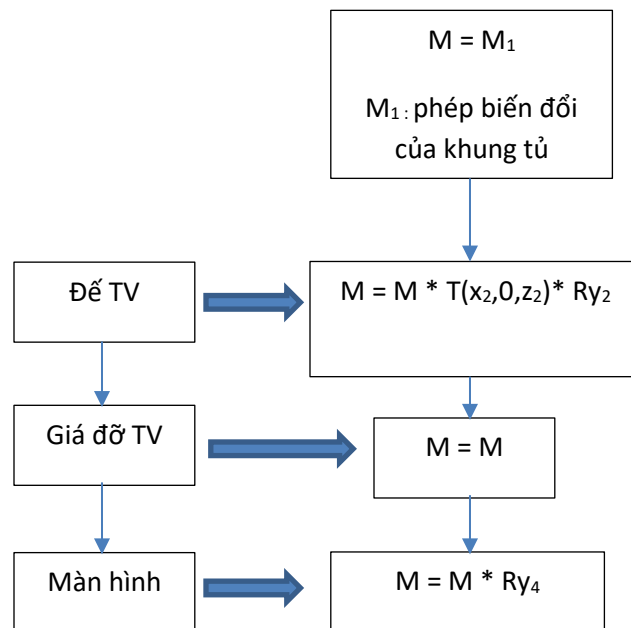
Hình 2.12. Thiết kế phân cấp chiếc tủ

2.6.3. Cài đặt chiếc tủ :

2.6.4. Ý tưởng thiết kế chiếc TV :

Chiếc TV bao gồm : đế , giá đỡ và màn hình được phân cấp theo thứ tự từ dưới lên , là mô hình con của khung tủ.

2.6.5. Thiết kế chiếc TV :



Hình 2.13. Thiết kế phân cấp chiếc TV

2.6.6. Cài đặt chiếc TV :

```
GLfloat base_width = 0.7, base_height = 0.05, base_depth = 0.3;
```

```
GLfloat bar_width = 0.2, bar_height = 0.15, bar_depth = 0.025;
```

```
GLfloat sc_width = 0.9, sc_height = 0.6, sc_depth = 0.025;
```

```
GLfloat monitor_theta = 0;
```

```

void base() {

    instance = Translate(0, base_height * 0.5, 0) *
Scale(base_width, base_height, base_depth) * RotateX(180);

    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);

    glDrawArrays(GL_TRIANGLES, 0, NumPoints);

}

void bar() {

    instance = Translate(0, bar_height * 0.5, 0) * Scale(bar_width,
bar_height, bar_depth) * RotateX(90);

    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);

    glDrawArrays(GL_TRIANGLES, 0, NumPoints);

}

void screen() {

    instance = Translate(0, sc_height * 0.5, 0) * Scale(sc_width *
1.25, sc_height, sc_depth) * RotateX(-90);

    glUniformMatrix4fv(model_loc, 1, GL_TRUE, model * instance);

    glDrawArrays(GL_TRIANGLES, 0, NumPoints);

}

GLfloat tran = 0;

void monitor() {

    model = model * Translate(tran, 0, 0) * Translate(0, -sc_height
* 0.5, 0) ;

    base();

```

```

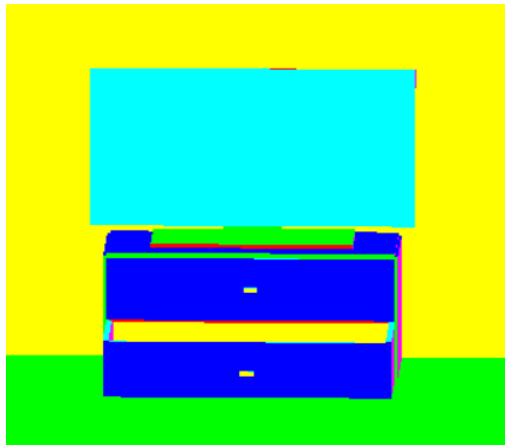
    model = model * Translate(0, base_height, 0) * RotateX(-15);

    bar();

    model = model * Translate(0, bar_height, (bar_depth * -cos(M_PI
/ 18)) - sc_depth) * RotateX(-monitor_theta) * Translate(0, -
bar_height / 2, 0) * RotateX(15);

    screen();
}

```



Hình 2.14. Chiếc tủ và TV sau khi cài đặt

2.7. Cài đặt phép chiếu :

2.7.1. Thực hiện phép chiếu phối cảnh :

- Phép chiếu phối cảnh là phép chiếu mà mọi tia chiếu hội tụ tại 1 điểm
- Các bước thực hiện :

B1 : Thiết lập tọa độ camera (eye)

B2 : Thiết lập điểm nhìn tới

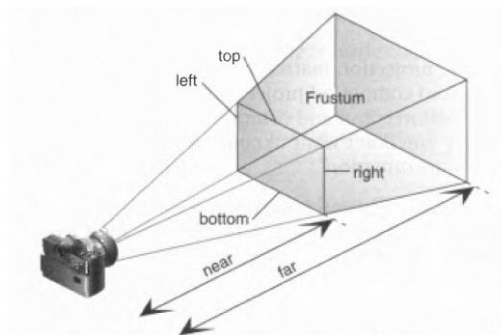
B3 : Thiết lập vector up của camera (vup)

B4 : Chuyển đổi từ hệ tọa độ thế giới thực sang hệ tọa độ camera

B5 : Thiết lập tầm nhìn của phép chiếu phối cảnh

2.7.2. Cài đặt phép chiếu phối cảnh

```
GLfloat eye_x = 2.0 , eye_y = 1.0 , eye_z = -1.5;  
GLfloat l = -0.8, r = 0.8;  
GLfloat bottom = -0.8, top = 0.8;  
GLfloat zNear = 1.9, zFar = 4.0;  
  
point4 eye(eye_x, eye_y, eye_z, 1.0);  
point4 at(0.0, 0.0, 0.0, 1.0);  
vec4 up(0.0, 1.0, 0.0, 0.0);  
view = LookAt(eye, at, up);  
glUniformMatrix4fv(view_loc, 1, GL_TRUE, view);  
mat4 p = Frustum(l, r, bottom, top, zNear, zFar);
```



Hình 2.15. Phép chiếu phối cảnh Frustum

2.7.3 Thực hiện phép chiếu song song

- Phép chiếu phối cảnh là phép chiếu mà các tia chiếu đều song song với nhau

- Các bước thực hiện :

B1->B4 : Tương tự phép chiếu phối cảnh

B5 : Thiết lập tầm nhìn của phép chiếu song song

2.7.4 Cài đặt phép chiếu song song

```
GLfloat eye_x = 2.0 , eye_y = 1.0 , eye_z = -1.5;
```

```
GLfloat l = -1.3, r = 1.3;
```

```
GLfloat bottom = -1.3, top = 1.3;
```

```
GLfloat zNear = 1.9, zFar = 4.0;
```

```
point4 eye(eye_x, eye_y, eye_z, 1.0);
```

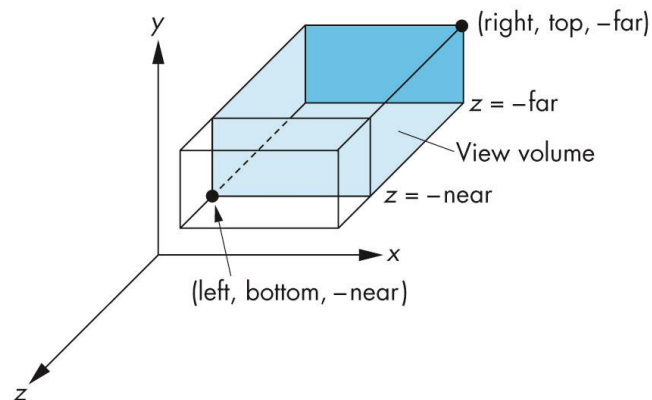
```
point4 at(0.0, 0.0, 0.0, 1.0);
```

```
vec4 up(0.0, 1.0, 0.0, 0.0);
```

```
view = LookAt(eye, at, up);
```

```
glUniformMatrix4fv(view_loc, 1, GL_TRUE, view);
```

```
mat4 p = Ortho(l, r, bottom, top, zNear, zFar);
```



Hình 2.7.4 Phép chiếu song song sử dụng hàm Ortho

CHƯƠNG 3 : KẾT QUẢ CHƯƠNG TRÌNH

Kết quả đạt được :

Chương trình sau khi hoàn thiện giống với ý tưởng và thiết kế ban đầu , tuy nhiên do chưa chuẩn bị kỹ các kiến thức cần có nên chưa thực hiện được các kỹ thuật nâng cao (thiết lập ánh sáng ,lập trình hướng đối tượng , vẽ đường cong) .

Bài học rút ra :

- + Cần nắm chắc các kiến thức nền tảng
- + Cần nghiên cứu thêm các nguồn tài liệu để nâng cao kiến thức
- + Cần lên kế hoạch , bản thiết kế , các quy ước chung khi phát triển chương trình một cách chi tiết

Trong quá trình làm bài , mặc dù đã cố gắng , song khó tránh khỏi thiếu sót.Rất mong nhận được sự đóng góp ý kiến xây dựng từ giảng viên và các bạn để sản phẩm được hoàn thiện hơn.

Một số hình ảnh chương trình

