

Bài thực hành 2: Phép biến đổi điểm nhìn và biến đổi mô hình

Mục đích:

- Hiểu cách thực hiện lệnh biến đổi điểm nhìn `gluLookAt()` và các lệnh biến đổi mô hình `glTranslate*()`, `glRotate*()`, `glScale*()`
- Cách thay thế lệnh biến đổi điểm nhìn bởi các lệnh biến đổi mô hình

Yêu cầu:

1. Lệnh biến đổi điểm nhìn **`gluLookAt(eyex, eyey, eyez, centerx, centery, centerz, upx, upy, upz)`**

Chạy chương trình **`cube.c`**

- 1.1 Thay lệnh `gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);`

- Bởi lệnh `gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, -100.0, 0.0, 1.0, 0.0);` kết quả hình ảnh không bị thay đổi, tại sao?

- Bởi lệnh `gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 100.0, 0.0, 1.0, 0.0);` kết quả không nhìn thấy hình ảnh, tại sao?

- 1.2 Khôi phục lại chương trình gốc `cube.c`, thay lệnh `glutWireCube()` bởi lệnh `glutWireTeapot(1.5);` chạy lại chương trình

Thay lệnh `gluLookAt()` bởi lần lượt các lệnh sau và giải thích kết quả thu được:

`gluLookAt (10.0, 0.0, 10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);`

`gluLookAt (10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 0.0);`

`gluLookAt (0.0, -5.0, 0.0, 0.0, 0.0, 0.0, 3.0, 0.0, -3.0);`

`gluLookAt (0.0, 0.0, 5.0, -10.0, 0.0, 0.0, 0.0, 1.0, 0.0);`

- 1.3 Khôi phục lại chương trình gốc **`cube.c`**, thay lệnh `gluLookAt()` bởi lần lượt các lệnh sau

`gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, -10.0);`

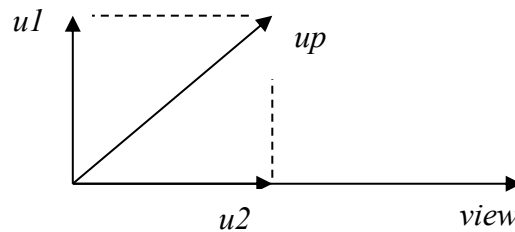
`gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 12.0, 30.0);`

`gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 3.7, 11.6);`

Ta thấy rằng các lệnh trên chỉ khác nhau ở hướng đỉnh của camera ta sẽ gọi đó là véc tơ *`up(upx, upy, upz)`*. Trong cả ba câu lệnh trên hướng đỉnh của camera không vuông góc với hướng nhìn. Kết quả thu được của các câu lệnh đó là như nhau, tại sao vậy?

Ta quay lại lệnh: **`gluLookAt(eyex, eyey, eyez, centerx, centery, centerz, upx, upy, upz)`**

Bình thường ta luôn chọn hướng đỉnh vuông góc với hướng nhìn của camera ta gọi đó là véc tơ $view(centerx-eyex, centery-eyey, centerz-eyez)$. Trong trường hợp nếu véc tơ up không vuông góc và không cộng tuyến với véc tơ $view$ ta luôn có cách phân tích véc tơ up theo cách duy nhất như sau: $up = u1 + u2$ (trong đó $u1$ là véc tơ vuông góc với $view$ còn $u2$ là một véc tơ cộng tuyến với $view$). Khi đó OpenGL sẽ chọn véc tơ $u1$ là hướng đỉnh của camera thay cho véc tơ up .



Ví dụ trong lệnh `gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, -10.0);` ta có

$$up(0.0, 1.0, -10.0) = u1(0.0, 1.0, 0.0) + 2 * view(0.0, 0.0, -5.0)$$

Do đó OpenGL sẽ chọn hướng đỉnh của camera là $u1(0.0, 1.0, 0.0)$, đó là lí do vì sao hình ảnh hiển thị không bị thay đổi.

2. Các lệnh

Khôi phục lại chương trình gốc **cube.c**

2.1 Chèn vào phía trước của lệnh `glutWireCube()` lần lượt các câu lệnh sau:

```
glTranslatef(2.0, 0.0, 0.0);
glTranslatef(2.0, 0.0, -3.0);
glRotatef(30, 0.0, 0.0, 1.0);
glRotatef(-60, 1.0, 0.0, 1.0);
glScalef(1.0, 1.0, 1.0);
glScalef(2.0, 1.5, 3.0);
```

2.2 Khôi phục lại chương trình gốc **cube.c** và chèn vào phía trước của lệnh `glutWireCube()` lần lượt các cụm câu lệnh sau:

Cụm lệnh 1:

```
glTranslatef(2.0, 0.0, 0.0);
glRotatef(45, 0.0, 0.0, 1.0);
```

Cụm lệnh 2:

```
glRotatef(45, 0.0, 0.0, 1.0);
```

```
glTranslatef(2.0, 0.0, 0.0);
```

Kết quả thu được khác nhau, tại sao?

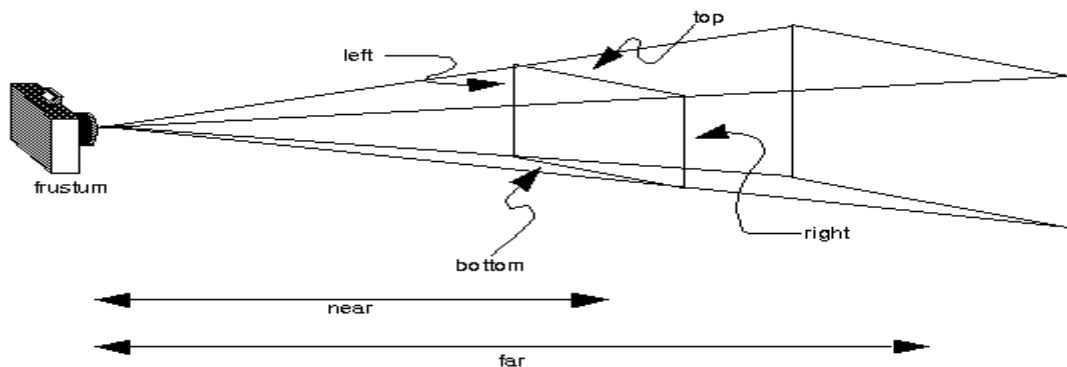
3. Thay thế lệnh biến đổi điểm nhìn bởi các lệnh biến đổi mô hình

Khôi phục lại chương trình gốc `cube.c`

3.1 Loại bỏ lệnh `gluLookAt()`, chạy chương trình ta sẽ không nhìn thấy hình ảnh nữa, tại sao?

Ta xét lệnh thiết lập viewing volume trong chương trình:

```
glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);
```



Chỉ có những phần của vật thể thuộc viewing volume (hình chóp cụt) mới được chiếu lên khung nhìn (khung nhìn thuộc mặt phẳng chiếu $z=-near=-1.5$).

Nếu ta loại bỏ lệnh `gluLookAt()` thì vật thể được vẽ mặc định tại gốc tọa độ do đó nó nằm ngoài viewing volume đó là lý do ta không nhìn thấy hình ảnh.

Bây giờ ta thêm lệnh `glTranslatef(0.0, 0.0, -5.0);` vào phía trước lệnh `glutWireCube()` Ta thấy hình ảnh lại hiển thị giống như khi có lệnh `gluLookAt()` chứng tỏ 2 lệnh này có hiệu lực giống nhau. Lệnh `gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);` nghĩa là kéo camera 5 đơn vị độ dài từ gốc tọa độ dọc theo hướng dương trục oz còn lệnh `glTranslatef(0.0, 0.0, 5.0);` nghĩa là tịnh tiến vật thể 5 đơn vị độ dài dọc theo hướng âm trục oz trong khi camera đặt tại vị trí mặc định là gốc tọa độ.

3.2 Khôi phục lại chương trình gốc `cube.c`

Sử dụng lệnh `gluLookAt (-4.5, 0.0, 4.5, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);` sau đó thay lệnh này bởi 2 câu lệnh biến đổi mô hình sau:

```
glRotatef(45,0.0,1.0,0.0);
```

```
glTranslatef(4.5,0.0,-4.5);
```

So sánh 2 kết quả thu được

3.3 Thay thế lệnh biến đổi điểm nhìn sau bằng các lệnh biến đổi mô hình tương đương

```
gluLookAt (0.0, 7.6, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0);
```

```
gluLookAt (0.0, .0, 4.5, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0);
```

```
gluLookAt (5.5, 0.0, 5.5, 0.0, 0.0, 0.0, -1.0, 1.0, 0.0);
```

3.4 Khôi phục lại chương trình gốc cube.c, thay lệnh glutWireCube() bởi lệnh glutWireTeapot(1.5); chạy lại chương trình

Thay lệnh thiết lập điểm nhìn thành gluLookAt(0.0, 0.0, -5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

Ta nhìn thấy Teapot theo chiều ngược lại, tại sao? Thay thế lệnh gluLookAt() bởi các lệnh biến đổi mô hình tương đương.