

Bài tập 4

Học kỳ 2 – Năm học 2017-2018

Lưu ý: Tất cả các bài tập đều phải VIẾT TAY

Câu 1:

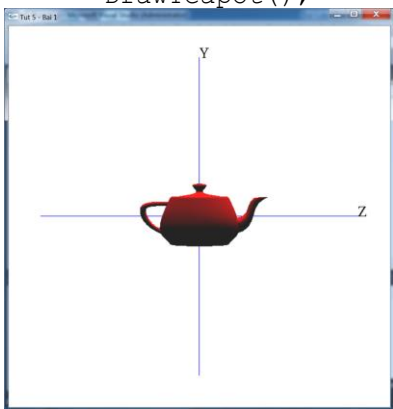
Biết rằng thể tích nhìn của camera được thiết lập bằng hàm:

```
glOrtho(-1.2, 1.2, -1.2, 1.2, 0.1, 100);
```

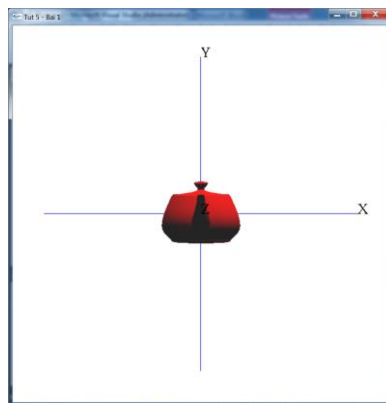
Giả sử hàm **DrawTeapot()** vẽ ấm trà có trọng tâm trùng với gốc tọa độ, vòi ấm nằm trên trục z dương, quai ấm nằm trên trục z âm.

Với mỗi hình vẽ dưới đây, hãy chọn tham số thích hợp cho hàm **gluLookAt(...)**

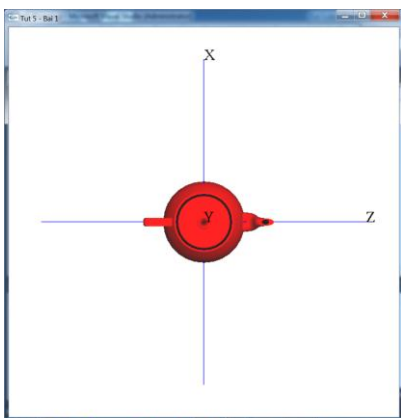
```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
gluLookAt(...);  
DrawTeapot();
```



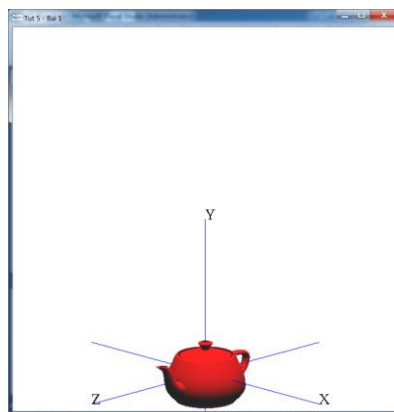
Hình 1



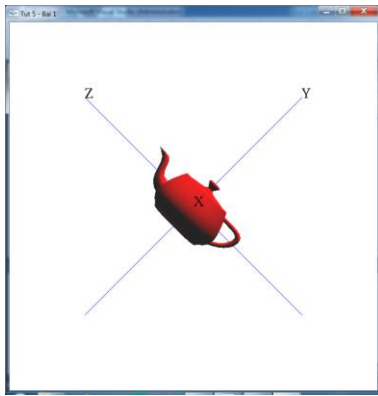
Hình 2



Hình 3



Hình 4



Hình 5

Câu 2:

Đọc trang 209 - 214 trong cuốn sách “Interactive Computer Graphics – A Top-Down Approach Using OpenGL”, phần 4.3.2.

Giả sử ta có lệnh thiết lập camera như sau:

`gluLookAt(2, 4, 4, 0, 0, 0, 0, -1, 0);`

- Tìm các vector u , v , n của hệ tọa độ camera.
- Tính ma trận biến đổi tọa độ từ hệ tọa độ thế giới sang hệ tọa độ camera.

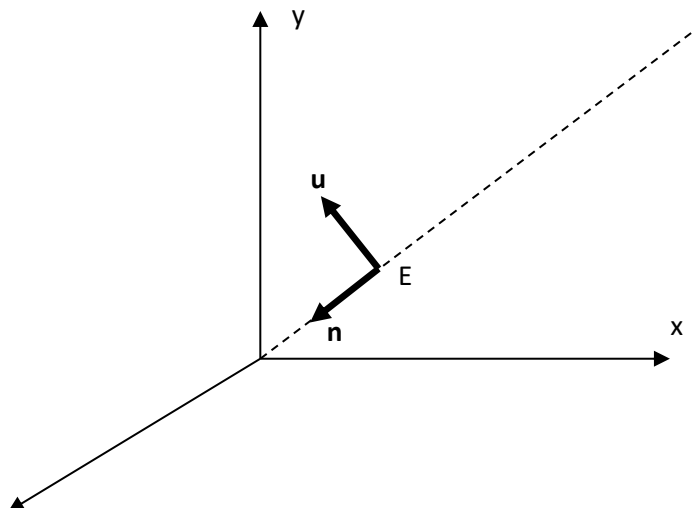
Câu 3:

Một Camera được đặt ở vị trí $E(1,2,3)$ với vector $up(0,1,0)$ và điểm nhìn $O(0,0,0)$

- Tìm Ma trận View
- Tìm tọa độ vị trí tương ứng trong không gian Camera (camera space) của điểm $P(5,4,3)$ trong không gian thế giới

Câu 4:

Cho một camera có vị trí mắt nhìn (view reference point) ở $E = (4, 4, 4)$, vector đơn vị \mathbf{n} nằm trên đường thẳng $x = y = z$, vector đơn vị \mathbf{u} nằm trong mặt phẳng $x = z$, mặt phẳng chiếu (view plan) vuông góc với đường thẳng $x = y = z$ và đi qua điểm $(8, 8, 8)$ (xem hình vẽ).



z

- a) Tìm ba vector \mathbf{u} , \mathbf{v} , \mathbf{n} của hệ tọa độ camera.
- b) Tìm ma trận V biến đổi từ tọa độ thế giới sang tọa độ camera.
- c) Tìm ma trận M biến đổi từ tọa độ thế giới sang tọa độ trên mặt phẳng chiếu của camera. Lưu ý, cần phải chia làm hai trường hợp : phép chiếu trực giao và phép chiếu phối cảnh.

Câu 5:

Đọc trang 232 - 237 trong cuốn sách “Interactive Computer Graphics – A Top-Down Approach Using OpenGL”, phần 4.7.1.

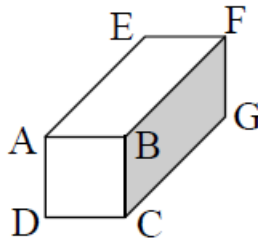
Giả sử ta dùng những câu lệnh sau để thiết lập thể tích nhìn:

```
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
glOrtho(-2.5, 1.5, -2.3, 2.7, 0.3, 300);
```

- a) Hãy tìm ma trận chiếu (projection matrix) để biến đổi thể tích nhìn thành thể tích nhìn chuẩn.
- b) Hãy nêu ra 2 cách làm khác (yêu cầu viết câu lệnh), để thiết lập được thể tích nhìn giống như thể tích nhìn trên.

Câu 6:

Đọc trang 220 - 224 trong cuốn sách “Interactive Computer Graphics – A Top-Down Approach Using OpenGL”, phần 4.4.5.



Giả sử ta sử dụng phép chiếu xiên với các góc $\theta=135^0$, $\phi=45^0$. Mặt phẳng gần nằm cách mắt nhìn 1, mặt phẳng xa nằm cách mắt nhìn 101. Mặt phẳng chiếu trùng với mặt phẳng gần. Thể tích nhìn là một hình hộp xiên ABCDEFGH, với mặt ABCD nằm trên mặt phẳng chiếu, với các tọa độ như sau (trong hệ tọa độ camera): $A = (-2.5, 1.5, -1)$, $B = (1.5, 1.5, -1)$, $C = (1.5, -2.5, -1)$, $D = (-2.5, -2.5, -1)$.

- a) Tính tọa độ các đỉnh E, F, G, H
- b) Tìm ma trận để biến đổi thể tích nhìn ABCDEFGH (hình hộp xiên), thành thể tích nhìn là hình hộp đứng
- c) Tìm ma trận chiếu để biến đổi thể tích nhìn ABCDEFGH (hình hộp xiên), thành thể tích nhìn chuẩn.

Câu 7:

Cho tứ diện ABCD trong không gian đối tượng (Object Space) như sau:

A (1, 0, 0), B (-1, -1, 1), C (-1, 1, 1), D (-1, 0, -1)

Các đỉnh của tứ diện được biến đổi vào không gian thế giới (World Space) bằng cách thực hiện lần lượt các phép biến đổi sau:

- Quay quanh trục Y một góc 90 độ ngược chiều kim đồng hồ.
- Phóng to hai lần theo trục X và Z (với tâm của phép biến đổi là gốc tọa độ): S(2, 1, 2)
- Tịnh tiến theo trục X : T(0, 4, 0)

Sử dụng hàm `gluLookAt()` để thiết lập Camera với tham số như sau:

```
gluLookAt(-10, 0, 10, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
```

Thiết lập phép chiếu phối cảnh (Perspective Viewing) bằng hàm `glFrustum()` với các tham số như sau:

```
glFrustum(-5, 5, -5, 5, 5, 15);
```

- 1) Tính ma trận Model (Model Matrix) để chuyển các đỉnh từ không gian đối tượng vào không gian thế giới
- 2) Tính ma trận View (View Matrix) để chuyển các đỉnh từ không gian thế giới vào không gian Camera
- 3) Tính ma trận Model-View (Model-View Matrix) để chuyển từ không gian đối tượng vào không gian Camera.
- 4) Tính ma trận chiếu (Projection Matrix) để đưa các đỉnh vào không gian cắt
- 5) Tính tọa độ trong không gian thế giới, không gian Camera và không gian cắt của hai đỉnh B và D.