

Bài thực hành 3: Các phép chiếu, thiết lập cửa sổ và cổng nhìn

Mục đích:

- Hiểu cách thiết lập viewing volume của các phép chiếu phối cảnh và trực giao bằng cách sử dụng các lệnh `glFrustum()`, `gluPerspective()`, `glOrtho()`, `gluOrtho2D()`. Ảnh hưởng của các tham số trong các lệnh đó tới hình ảnh hiển thị của vật thể.

- Thiết lập cửa sổ và cổng nhìn hiển thị ảnh trên màn hình

Chú ý quan trọng: Các lệnh thiết lập viewing volume không ảnh hưởng tới vị trí của camera, nghĩa là các phép chiếu luôn coi camera (mắt nhìn) tại vị trí mặc định là gốc tọa độ.

Yêu cầu:

1. Phép chiếu phối cảnh

1.1 Khôi phục chương trình gốc **cube.c**, thay lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);` (*left=-1, right=1, top=1, bottom=-1, near=1.5, far=20*)

- Bởi lệnh: `glFrustum (-2.0, 2.0, -1.0, 1.0, 1.5, 20.0);` ta thấy chiều ngang của hình hộp kích thước thu nhỏ bằng 1/2 kích thước ban đầu, tại sao?

- Bởi lệnh: `glFrustum (-2.0, 2.0, -2.0, 2.0, 1.5, 20.0);` ta thấy cả chiều ngang và chiều dọc của hình hộp kích thước thu nhỏ bằng 1/2 kích thước ban đầu, tại sao?

(Gợi ý: Bản chất vấn đề nằm ở quá trình ánh xạ ảnh từ khung nhìn vào cổng nhìn)

- Bởi lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 30.0);` tiếp đó là `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 50.0);` Ta thấy hình ảnh không thay đổi chứng tỏ tham số *far* không ảnh hưởng tới ảnh, tại sao?

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

- Loại bỏ lệnh `glLookAt()` ta không còn nhìn thấy ảnh nữa (lí do đã được giải thích trong bài thực hành số 2), tiếp đó ta thay lệnh `glutWireCube (1.0);` thành lệnh `glutWireCube (3.0);` (tăng kích thước hình hộp từ 1 lên 3) ta vẫn không nhìn thấy ảnh trên màn hình. Tiếp tục thay lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);` bởi lệnh `glFrustum (-2.0, 2.0, -2.0, 2.0, 1.5, 20.0);` ta nhìn thấy một hình vuông trên màn hình, giải thích tại sao?

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

Thay lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);` bởi lần lượt các lệnh sau:

`glFrustum (-1.0, 1.0, -1.0, 1.0, 2.5, 20.0);` ta thấy ảnh lớn hơn

`glFrustum (-1.0, 1.0, -1.0, 1.0, 3.5, 20.0);` ta thấy ảnh lớn hơn nữa

`glFrustum (-1.0, 1.0, -1.0, 1.0, 5.5, 20.0)`; ta chỉ thấy một hình vuông, tại sao?

`glFrustum (-1.0, 1.0, -1.0, 1.0, 5.6, 20.0)`; ta không nhìn thấy ảnh, tại sao?

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

Thay lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0)`; bởi lần lượt các lệnh sau:

`gluPerspective(60, 1.0, 1.5, 20.0)`; chạy chương trình và nhận xét

`gluPerspective(60, 2.0, 1.5, 20.0)`; ta thấy chiều ngang của hình hộp kích thước thu nhỏ bằng 1/2 kích thước ban đầu, tại sao?

`gluPerspective(60, 0.5, 1.5, 20.0)`; ta thấy chiều ngang của hình hộp kích thước phóng to bằng 2 lần kích thước ban đầu, tại sao?

`gluPerspective(45, 1.0, 1.5, 20.0)`; hình hộp được phóng to

`gluPerspective(30, 1.0, 1.5, 20.0)`; hình hộp được phóng to hơn nữa, tại sao?

`gluPerspective(30, 1.0, 2.5, 20.0)`; kích thước hình hộp không đổi

`gluPerspective(30, 1.0, 3.5, 20.0)`; kích thước hình hộp không đổi

Chú ý: Ở hai trường hợp cuối ta đã thay đổi tham số *near* và hình ảnh là không đổi, điều này trái ngược với khi ta thay đổi tham số *near* trong lệnh `glFrustum()`, tại sao vậy?

2. Phép chiếu trực giao

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

Thay lệnh `glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0)`; bởi lệnh sau:

`glOrtho (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0)`; sau đó thay đổi các tham số của lệnh này và nhận xét kết quả thu được.

3. Thiết lập cửa sổ và cổng nhìn

Trong chương trình **cube.c** thì các lệnh:

`glutInitWindowSize (500, 500)`; định nghĩa hai kích thước của cửa sổ

`glutInitWindowPosition (100, 100)`; định nghĩa vị trí góc trái trên của cửa sổ

Thay đổi tham số của 2 lệnh trên và nhận xét kết quả

Khôi phục chương trình gốc **cube.c**

Lệnh `glViewport (0, 0, (GLsizei) w, (GLsizei) h)`; định nghĩa khung nhìn có kích thước đúng bằng kích thước của cửa sổ hiển thị. Thay lệnh này bằng lần lượt các lệnh sau và giải thích kết quả thu được:

`glViewport (0, 0, 500, 500);` hình ảnh không thay đổi

`glViewport (0, 0, 250, 500);` hình bị bóp méo

`glViewport (0, 0, 500, 250);` hình bị bóp méo

`glViewport (0, 0, 250, 250);` hình bị thu nhỏ