

Bài 6

BIẾN ĐỔI GÓC NHÌN



NỘI DUNG

1. Biến đổi góc nhìn 2D
2. Biến đổi trục tọa độ
3. Biến đổi góc nhìn 3D
4. Di chuyển Camera



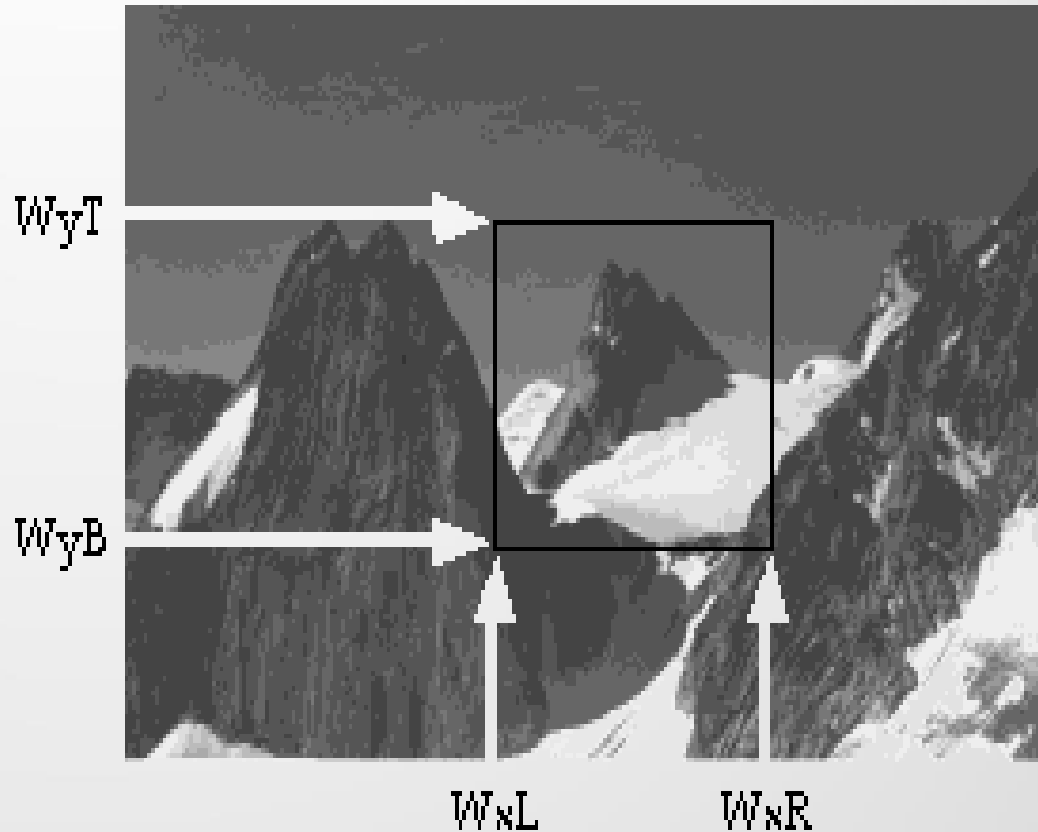
1

BIẾN ĐỔI GÓC NHÌN 2 CHIỀU

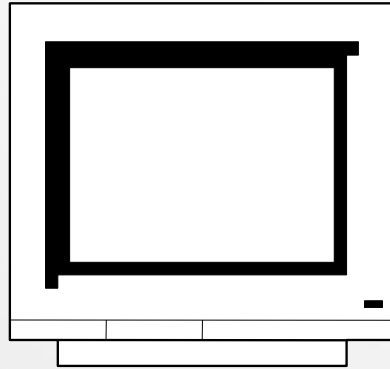
Hệ tọa độ thực

Hệ tọa độ thực – World Coordinate System (WCS)

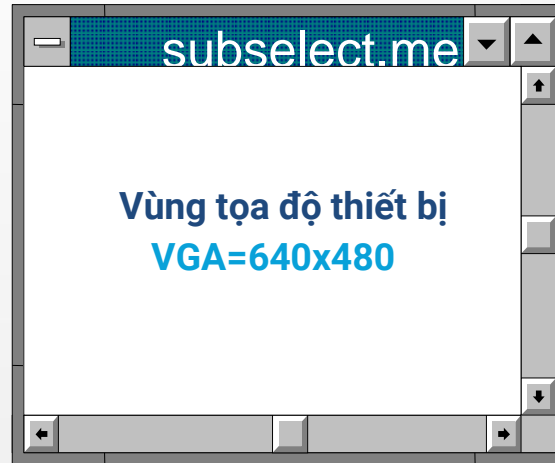
- Là hệ tọa độ của đối tượng được các chương trình ứng dụng sử dụng để mô tả tọa độ của các đối tượng trong thế giới thực.
- Đơn vị trong hệ thống tọa độ phụ thuộc vào không gian và kích thước của đối tượng được mô tả, có thể từ Å, nm, mm ... đến m, km ...
- Modeling Coordinate



Hệ tọa độ thiết bị



Thiết bị hiển thị

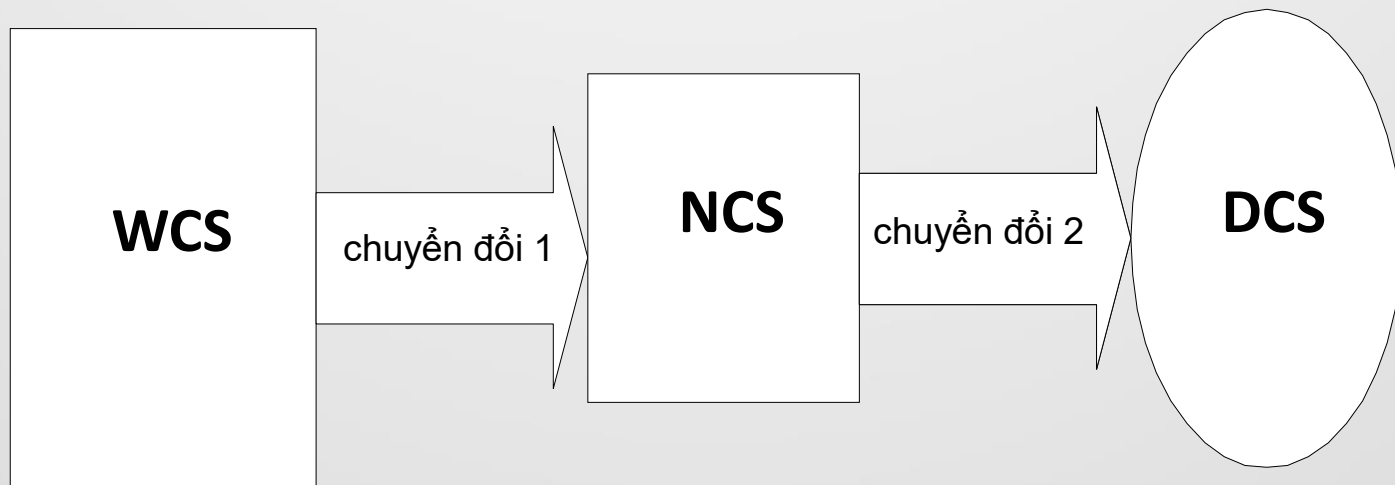


Hệ tọa độ thiết bị (Device Coordinate System)

- Là hệ thống tọa độ của thiết bị nơi hiển thị hình ảnh và không gian của đối tượng mà ứng dụng mô tả.

Hệ tọa độ chuẩn

- Chuyển đổi hệ tọa độ -> Sử dụng hệ tọa độ chuẩn – Normalized Coordinate System (NCS)
- Giải quyết vấn đề khi ứng dụng chạy trên các thiết bị khác nhau
- Có kích thước 1x1

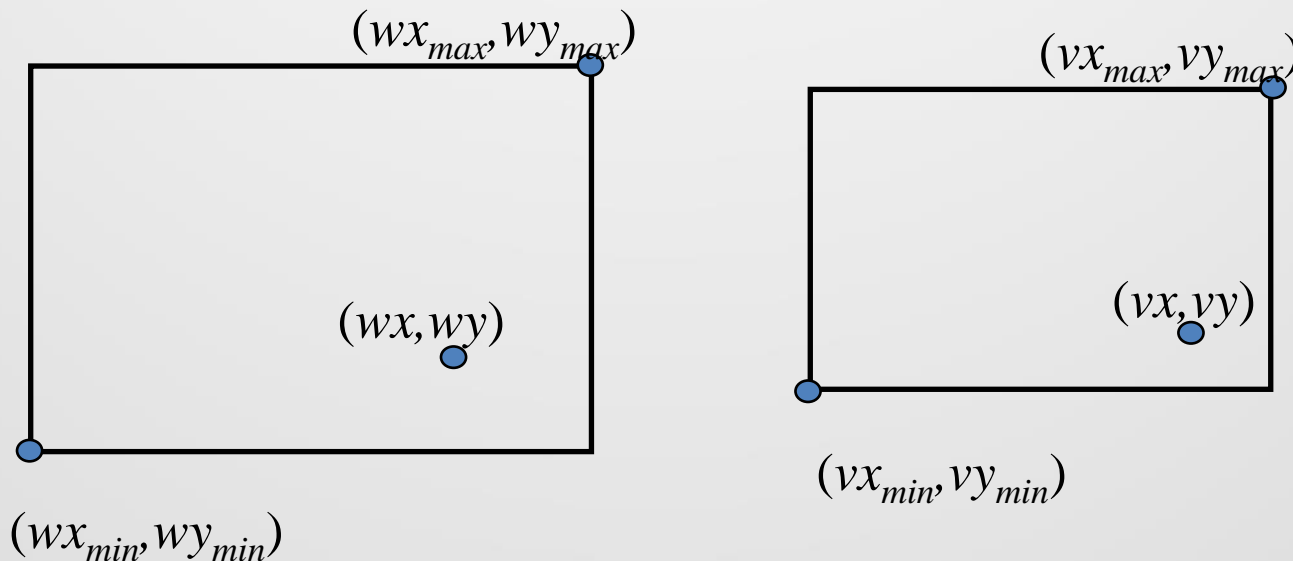


Phép biến đổi góc nhìn cơ bản

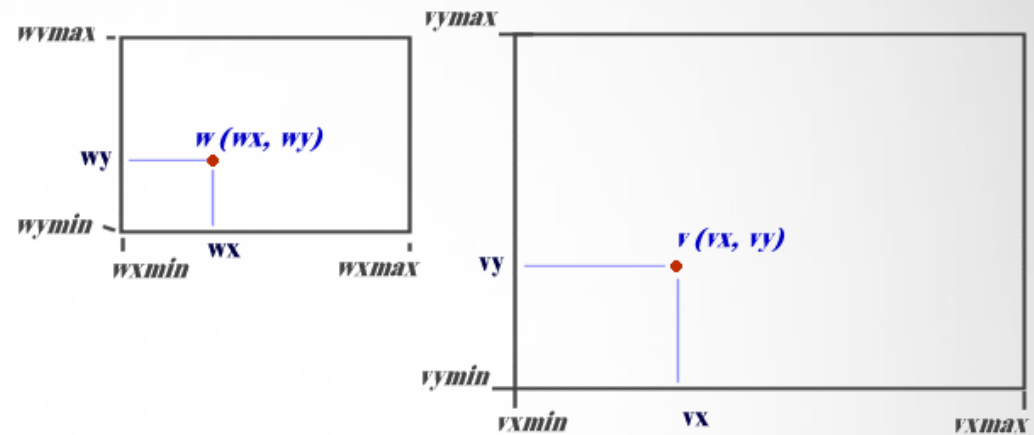
- Phép biến đổi góc nhìn cơ bản là chuyển đổi từ hệ tọa độ mô hình sang hệ tọa độ màn hình (Modelling co-ordinates to Screen Coordinates)
- Vấn đề:
 - Thay đổi tỷ lệ hình ảnh
 - Hiệu ứng về tỉ lệ khi Zoom hình ảnh
 - Phần nào của hình vẽ sẽ hiển thị ?
 - Hiển thị tại đâu trên màn hình ?
 - Chuyển đổi sẽ được thực hiện ra sao?

Trong hệ tọa độ 2 chiều

- Window : Cửa sổ, Phần hình ảnh của thế giới thực sẽ được hiển thị
- Viewport : Cổng nhìn, vùng màn hình nơi hình ảnh thế giới thực sẽ hiển thị



PHÉP CHUYỂN ĐỔI



$$v_x = V_{x_{min}} + \frac{(V_{x_{max}} - V_{x_{min}})(w_x - W_{x_{min}})}{W_{x_{max}} - W_{x_{min}}}$$

$$v_y = V_{y_{min}} + \frac{(V_{y_{max}} - V_{y_{min}})(w_y - W_{y_{min}})}{W_{y_{max}} - W_{y_{min}}}$$

PHÉP BIẾN ĐỔI THEO MA TRẬN

- Ma trận chuyển vị theo Window

$$[T1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -X_w & -Y_w & 1 \end{bmatrix}$$

- Ma trận biến đổi tỉ lệ

$$[S1] = \begin{bmatrix} \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}} & 0 & 0 \\ 0 & \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Ma trận chuyển vị theo tọa độ viewport

$$[T2] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X_v & Y_v & 1 \end{bmatrix}$$

Ma trận biến đổi tổng hợp

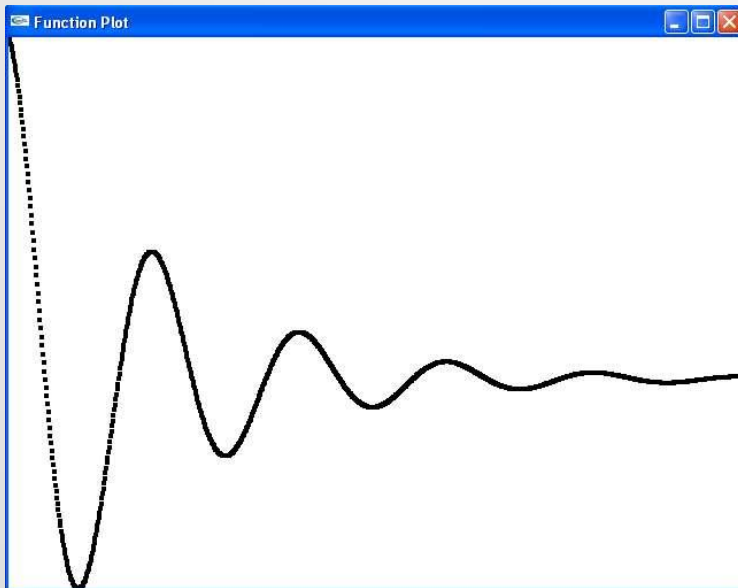
$$[T] = [T1] \times [S1] \times [T2]$$

$$[T] = \begin{bmatrix} \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}} & 0 & 0 \\ 0 & \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}} & 0 \\ X_{vmin} - X_{wmin} \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}} & Y_{vmin} - Y_{wmin} \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}} & 1 \end{bmatrix}$$

```
//set the viewing coordinates
setWindow(xmin, xmax, ymin, ymax);
setViewport(0,640,0,480);
glBegin(GL_POINTS);
for(GLdouble x = xmin; x < xmax; x+=0.005 )
{
    glVertex2d(x, pow(2.7183,-x)
    *cos(2*3.14*x));
}
glEnd();
```

```
//----- setWindow -----
void setWindow(GLdouble left, GLdouble
right, GLdouble bottom, GLdouble top)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(left, right, bottom, top);
}

//----- setViewport -----
void setViewport(GLdouble left, GLdouble
right, GLdouble bottom, GLdouble top)
{
    glViewport(left, bottom, right - left,
top - bottom);
}
```



Ví dụ trong OpenGL

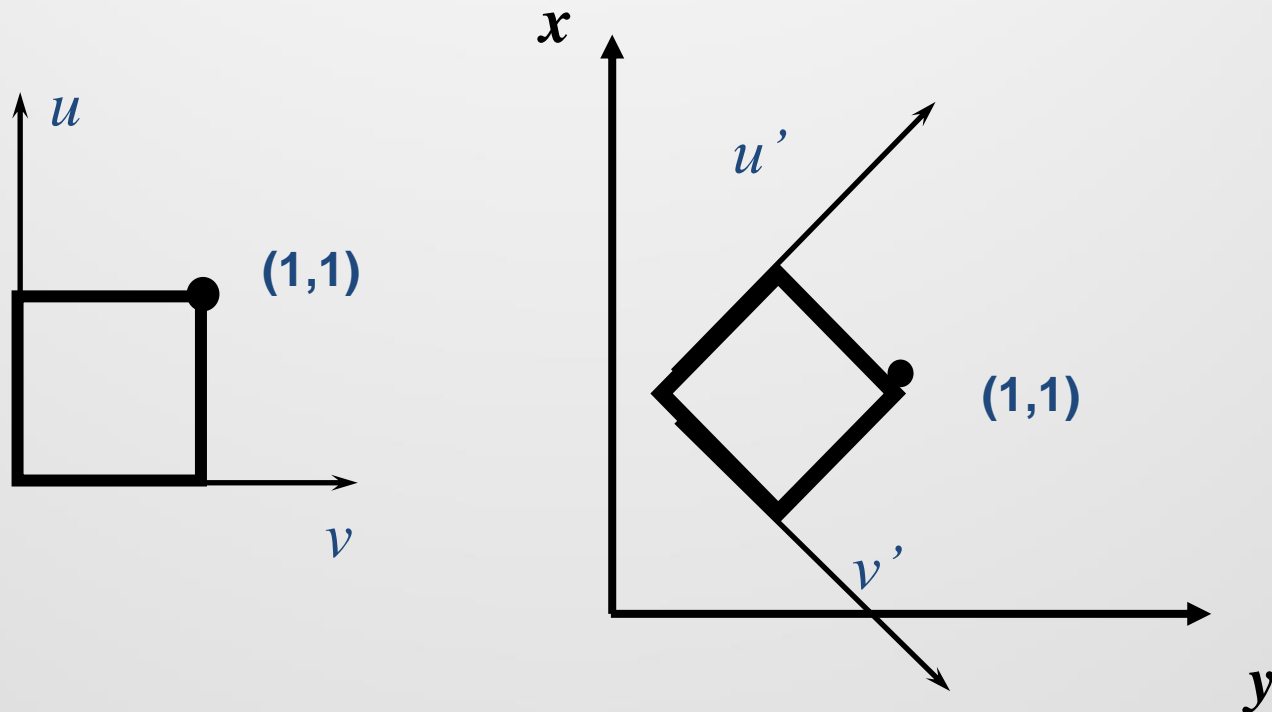


2

BIẾN ĐỔI HỆ TỌA ĐỘ

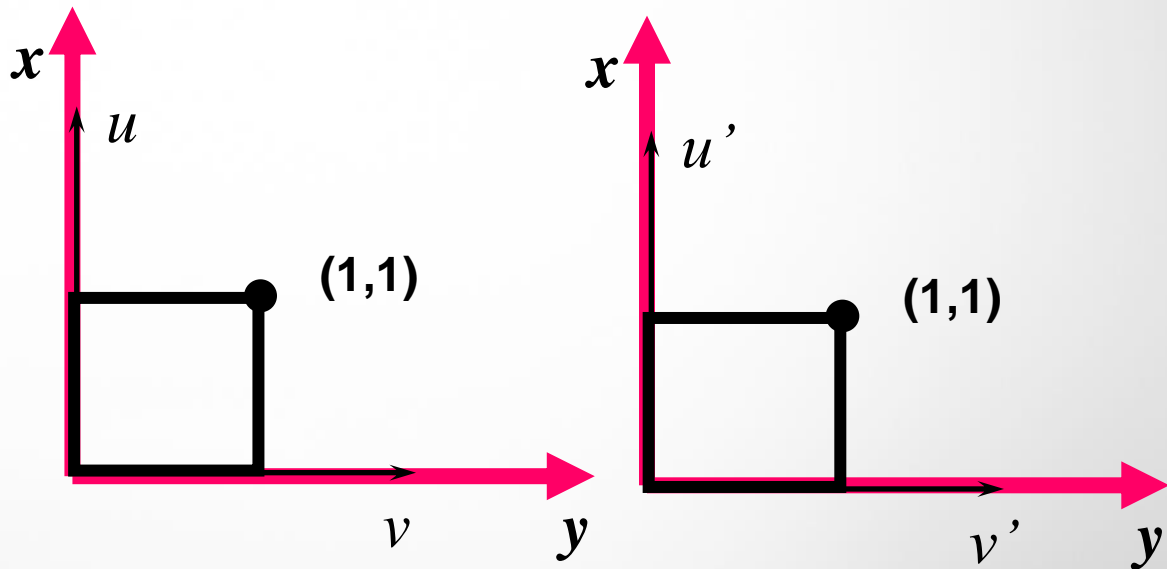
Biến đổi hệ tọa độ

- Là phép biến đổi trên hệ tọa độ của đối tượng. Việc thay đổi hệ tọa độ của đối tượng với hệ thống tọa độ chung tạo nên sự thay đổi toàn cảnh



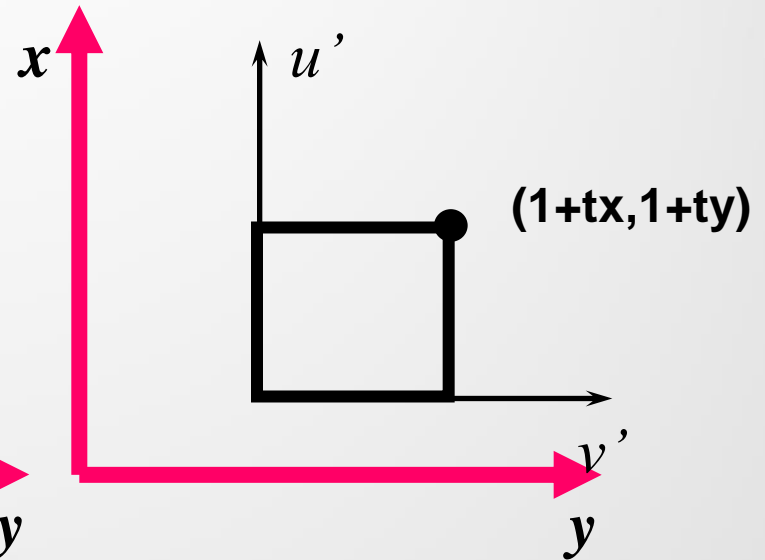
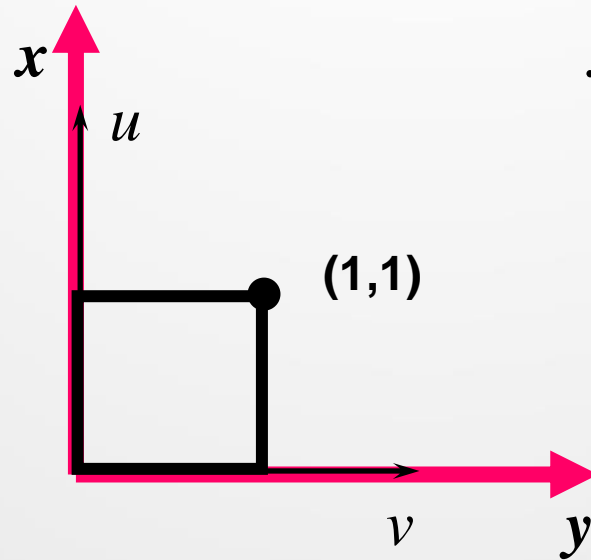
CÁC PHÉP BIẾN ĐỔI

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P$$



Chuyển vị

$$Q = \begin{bmatrix} \mathbf{1} & \mathbf{0} & tx \\ \mathbf{0} & \mathbf{1} & ty \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} P$$



$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & tx \\ \mathbf{0} & \mathbf{1} & ty \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} tx \\ ty \\ \mathbf{1} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & tx \\ \mathbf{0} & \mathbf{1} & ty \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} + tx \\ ty \\ \mathbf{1} \end{bmatrix}$$

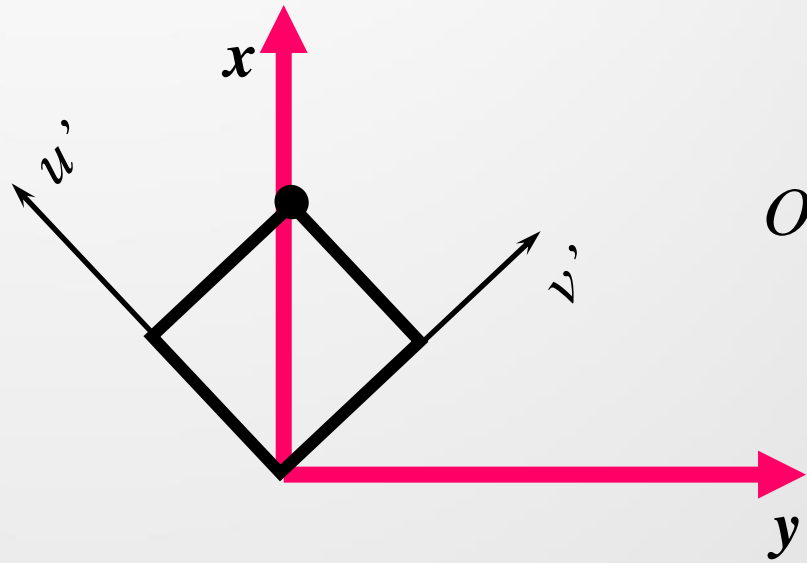
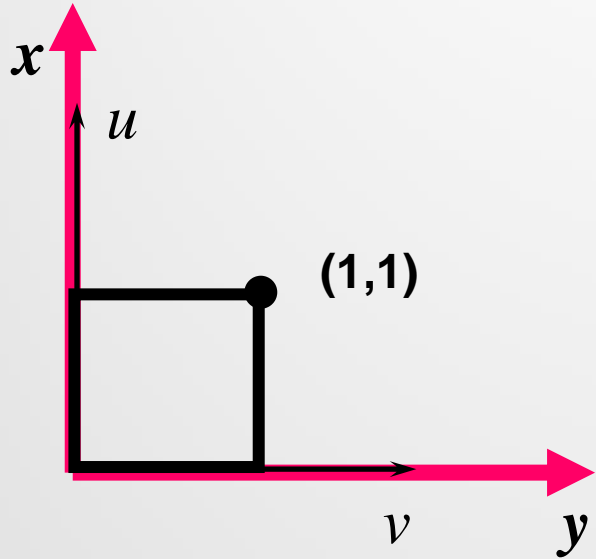
$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & tx \\ \mathbf{0} & \mathbf{1} & ty \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} tx \\ \mathbf{1} + ty \\ \mathbf{1} \end{bmatrix}$$

$$O = \begin{bmatrix} tx \\ ty \\ \mathbf{1} \end{bmatrix} \text{ origin}$$

$$v (1, 0, 0)$$

$$u (0, 1, 0)$$

Xoay



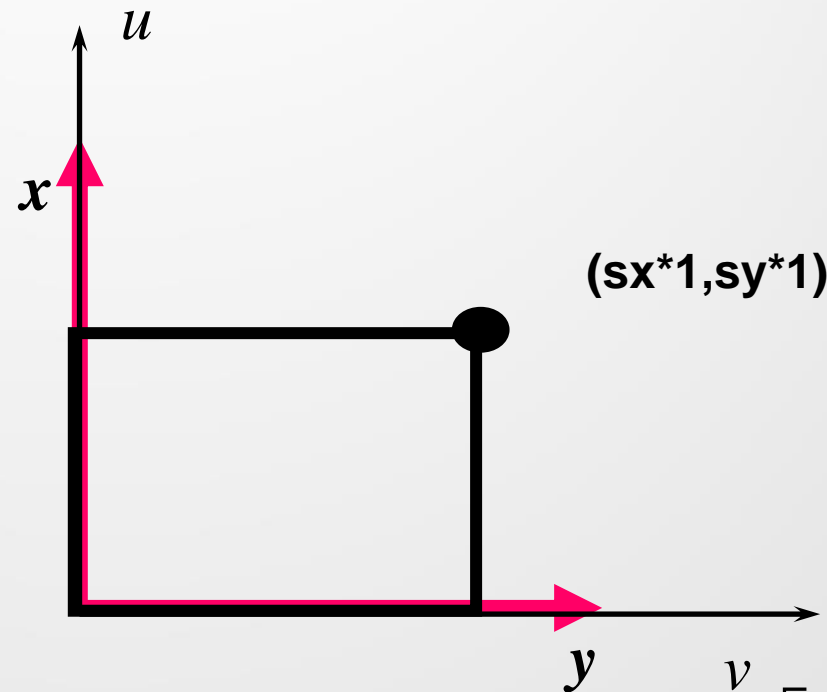
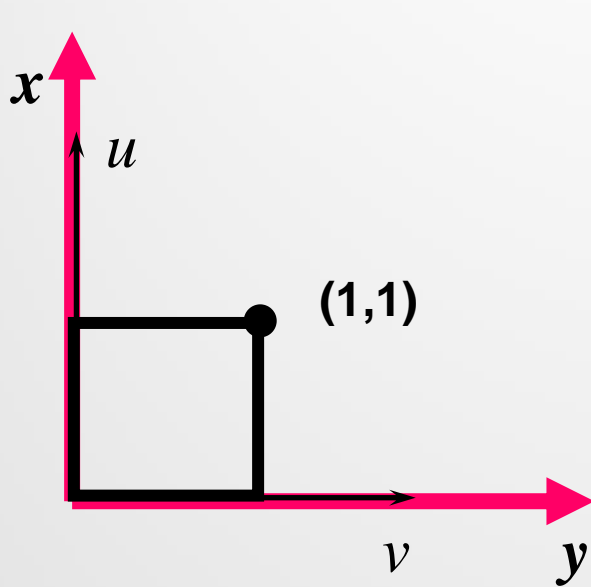
$$O = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} \cos \theta & -\sin \theta & \mathbf{0} \\ \sin \theta & \cos \theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{P}$$

$$v = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 1 \end{bmatrix}$$

$$u = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 1 \end{bmatrix}$$

Tỉ lệ



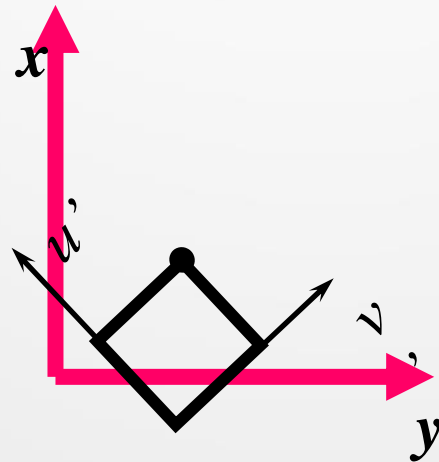
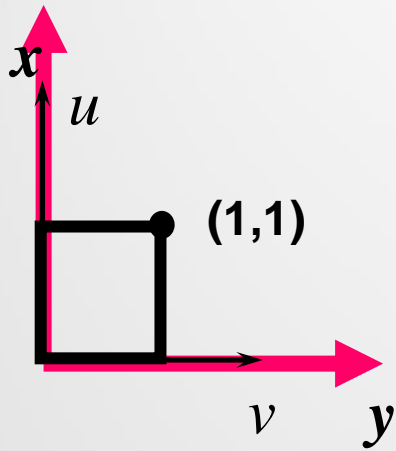
$$\mathbf{Q} = \begin{bmatrix} sx & 0 & \mathbf{0} \\ 0 & sy & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{P}$$

$$\mathbf{O} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} sx \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} 0 \\ sy \\ 1 \end{bmatrix}$$

Phép biến đổi tổng hợp



$$M = \begin{pmatrix} \cos \theta & -\sin \theta & x_1(1 - \cos \theta) + y_1 \sin \theta \\ \sin \theta & \cos \theta & y_1(1 - \cos \theta) - x_1 \sin \theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$O = \begin{bmatrix} x_1(1 - \cos \theta) + y_1 \sin \theta \\ y_1(1 - \cos \theta) - x_1 \sin \theta \\ 1 \end{bmatrix}$$

$$v = \begin{bmatrix} \cos \theta + x_1(1 - \cos \theta) + y_1 \sin \theta \\ \sin \theta + y_1(1 - \cos \theta) - x_1 \sin \theta \\ 1 \end{bmatrix}$$

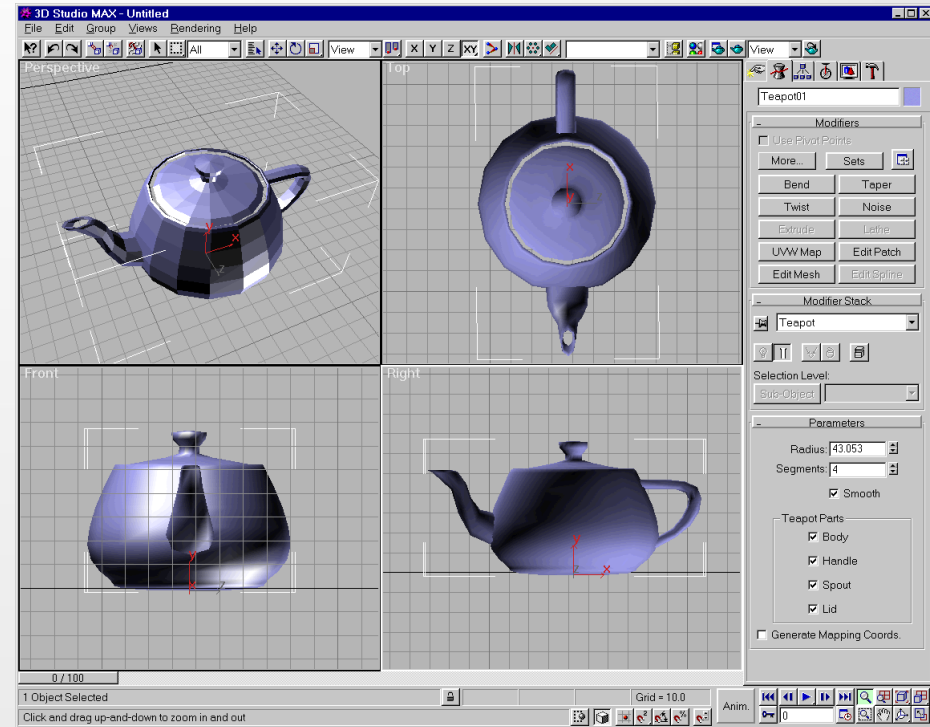
$$u = \begin{bmatrix} -\sin \theta + x_1(1 - \cos \theta) + y_1 \sin \theta \\ \cos \theta + y_1(1 - \cos \theta) - x_1 \sin \theta \\ 1 \end{bmatrix}$$

3

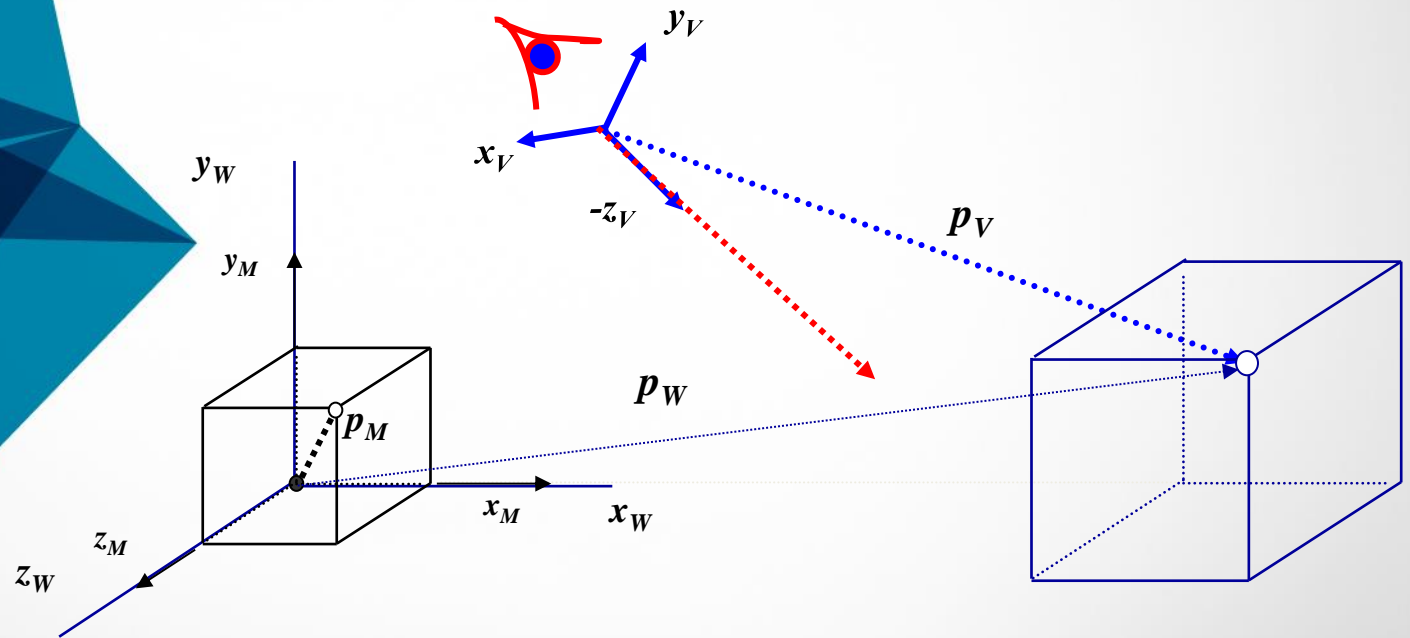
BIẾN ĐỔI GÓC NHÌN 3 CHIỀU

Quan sát 3D - 3D Viewing

- Hiển thị hình ảnh 3D trên các thiết bị 2D
- Mỗi 1 quan sát 3D trên cùng 1 đối tượng chỉ ra trạng thái chiều của điểm quan sát đó.
 - Hình ảnh chiếu 2D của đối tượng 3D phụ thuộc vào người quan sát
- Các quan sát khác nhau được phân biệt bởi thông số: vị trí - position, hướng - orientation, và vùng nhìn - field of view



GÓC NHÌN



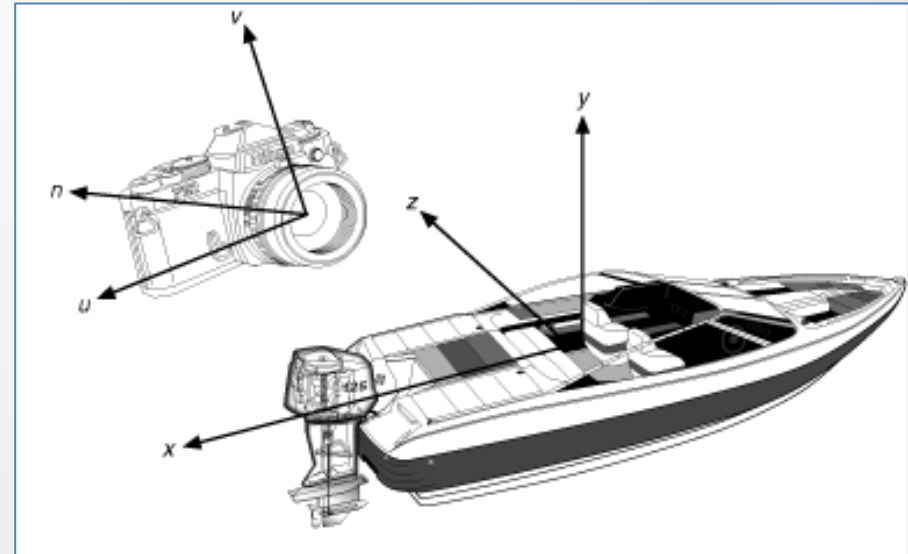
Tiến trình quan sát

- Để tạo 1 cảnh nhìn gồm các bước
 - Mô tả biểu diễn hình học
 - Định nghĩa điểm nhìn – Camera
- Chuyển đổi từ đối tượng 3D sang 2D
- Xác định vùng hiển thị của đối tượng
- Xác định camera để có thể tạo thành phép chiếu từ 3D thành 2D.

Khái quát về camera

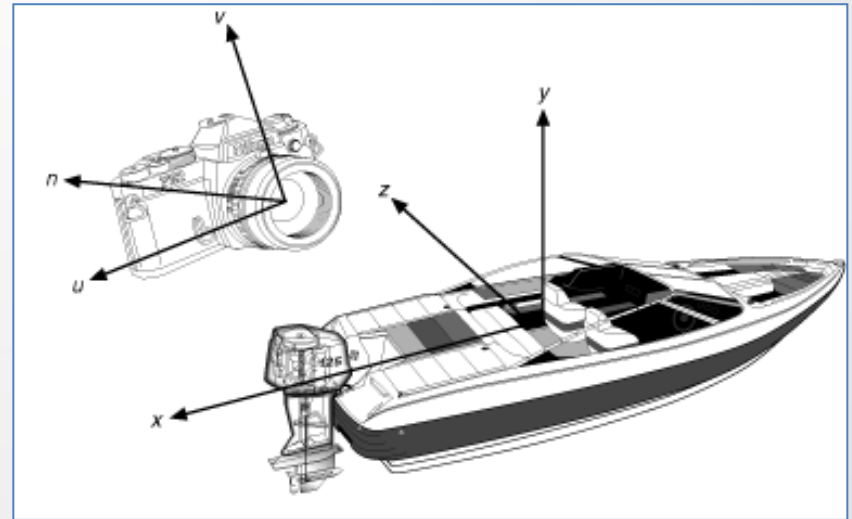
Khái niệm

- camera location: Vị trí trong thế giới thực
- viewing direction: Hướng chiếu của Camera.
- up vector: Hướng phía trên của Camera
- aperture size: Vùng nhìn thấy của Camera



Tọa độ quan sát

- Có thể sử dụng mô hình hệ thống tọa độ quan sát trên các thông số gồm:
 - Điểm nhìn (VRP View Reference Point): điểm gốc quan sát. Vị trí của Camera)
 - Vector mặt phẳng quan sát (VPN - View Plane Normal vector): viewing direction
 - Vector hướng quan sát (VUP - View UP vector): xác định góc nghiêng của camera



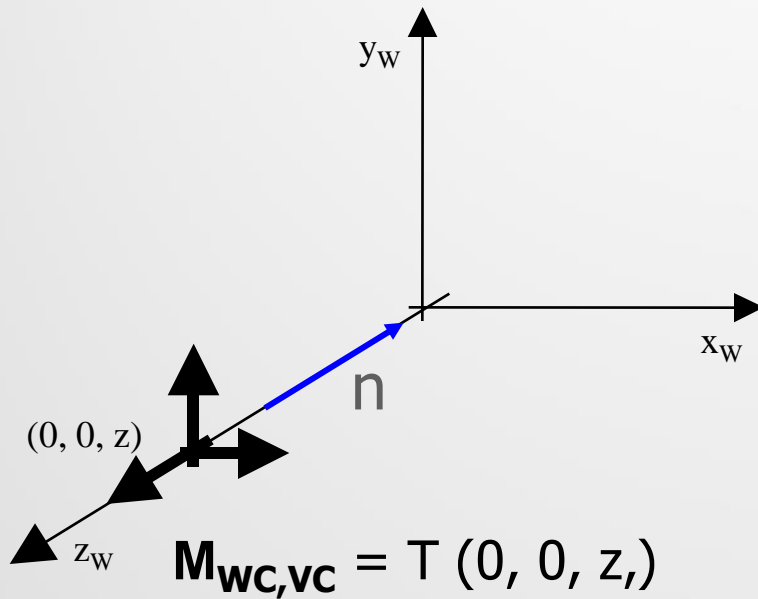
Các véctơ cơ bản:

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_1, n_2, n_3)$$

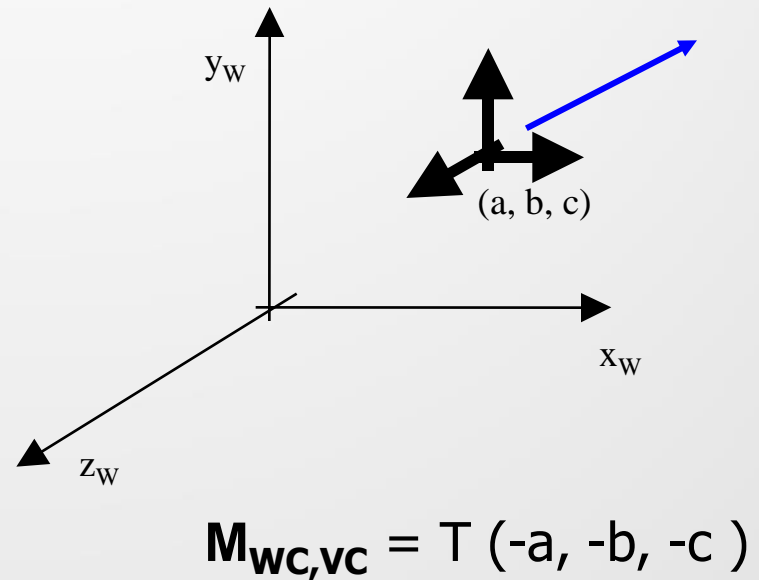
$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_1, u_2, u_3)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_1, v_2, v_3)$$

Điểm nhìn trên trục Z
 z_W -axis



Nhìn ở vị trí (a, b, c) theo trục Z và ngược hướng z_W -axis



Camera luôn nhìn về trục âm của trục tọa độ z

- (x_w, y_w, z_w) and (u, v, n) 2 hệ tọa độ với cùng 1 gốc.
- Vector đơn vị (u, v, n) được định nghĩa theo WCS

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \text{và} \quad \mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}.$$

- Ma trận biến đổi tọa độ thế giới sang tọa độ quan sát
- $\text{MWC,VC} = \mathbf{R} \mathbf{T}$

3D Viewing Transformation

- Với điểm quan sát VRPoint (x_0, y_0, z_0) các phép biến đổi góc nhìn có thể được xác định bằng tổ hợp của hai biến đổi

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

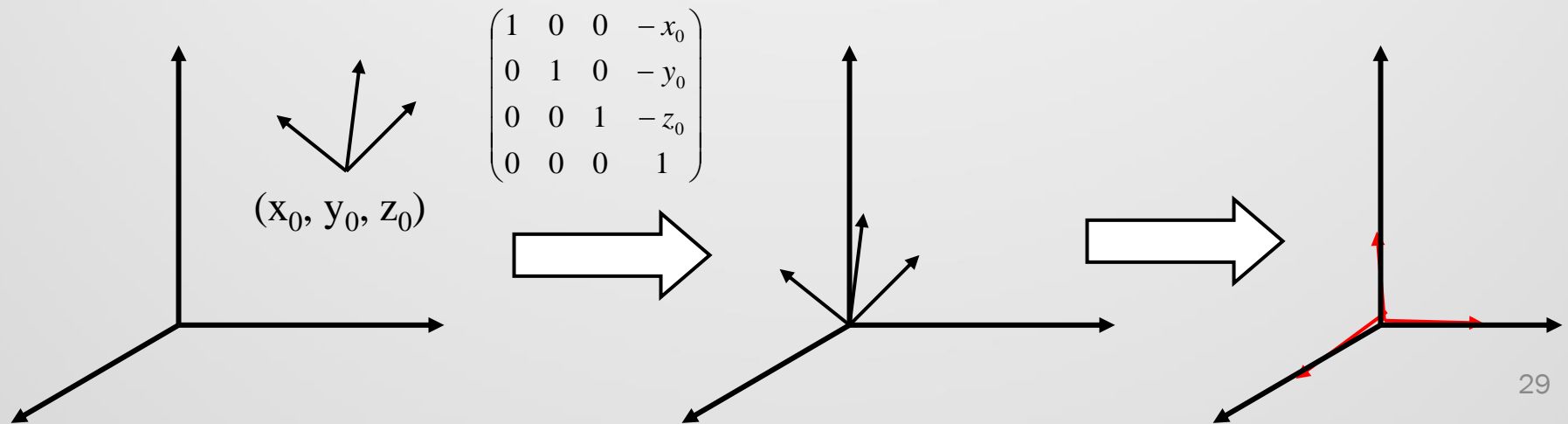
- Phép tính này được sử dụng để xác định vị trí của từng điểm trong hệ tọa độ của camera

$$\mathbf{M}_{wc,vc} = \mathbf{R} \cdot \mathbf{T}$$

$$\vec{x}' = \mathbf{M}_{wc,vc} \vec{x}$$

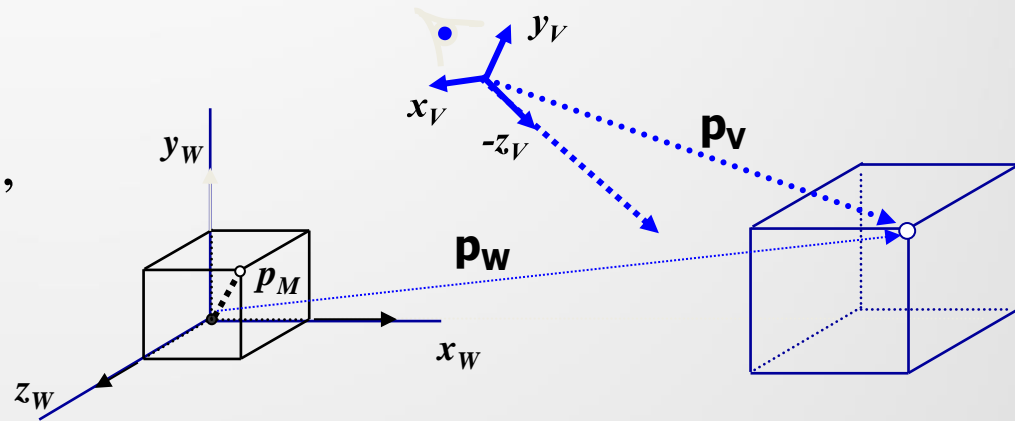
Viewing Transformation

- Chúng ta cũng có thể thực hiện được việc này qua hai bước
 - Dịch chuyển điểm nhìn về tâm tọa độ của hệ trục tọa độ môi trường
 - Thực hiện việc xoay để lần lượt khớp các trục xv , yv và zv



where

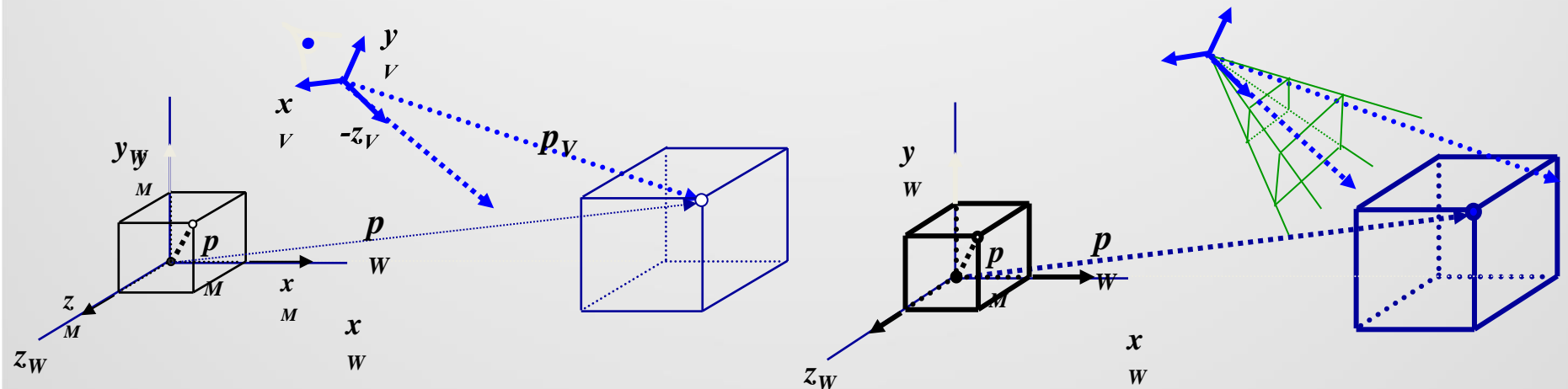
$$\mathbf{R} = \begin{pmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$



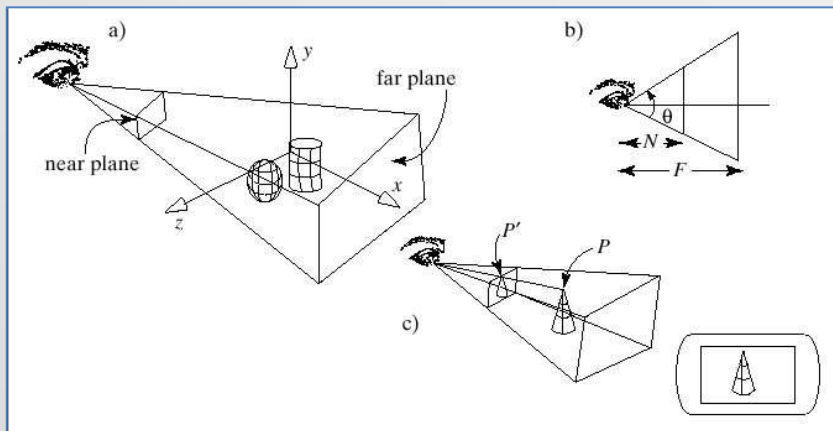
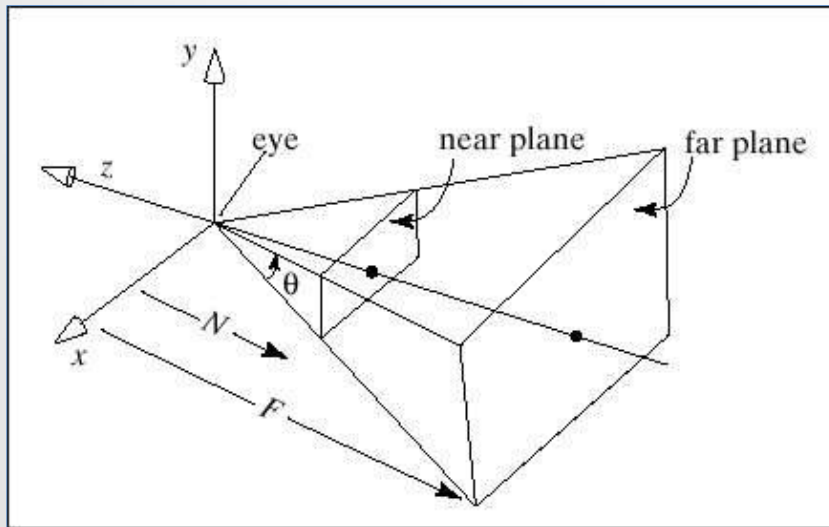
Verification: $\mathbf{R} \mathbf{u} = \mathbf{R} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ 1 \end{bmatrix} = ?$, etc

$$\mathbf{p}_v = \mathbf{M}_{wc,vc} \mathbf{p}_w$$

- Recall we've re-expressed our world geometry in eye's frame reference.
- To do projection, we use a synthetic camera by introducing a viewing frustum after the viewing transformation.



Camera trong OpenGL

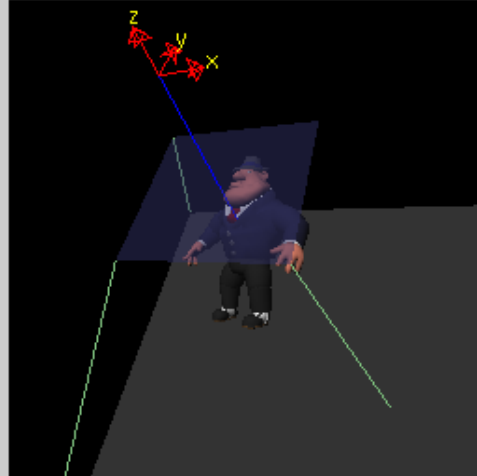


- Các hàm cơ bản
 - `glOrtho()` and `gluLookAt()`
 - sử dụng để tạo các phép chiếu song song
- `gluPerspective`
 - `viewAngle`
 - `aspectRatio (W/H)`
 - `near plane`
 - `far plane`

CAMERA TRONG OPENGL

```
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
gluPerspective(viewAngle,  
    aspectRatio, N, F);
```

World-space view



Screen-space view



Command manipulation window

```
fovy aspect zNear zFar  
gluPerspective( 68.0 , 1.01 , 1.4 , 10.0 );  
gluLookAt( 0.70 , 1.30 , 1.70 , <- eye  
          0.00 , 0.00 , 0.00 , <- center  
          0.00 , 1.00 , 0.00 ); <- up
```

Click on the arguments and move the mouse to modify values.



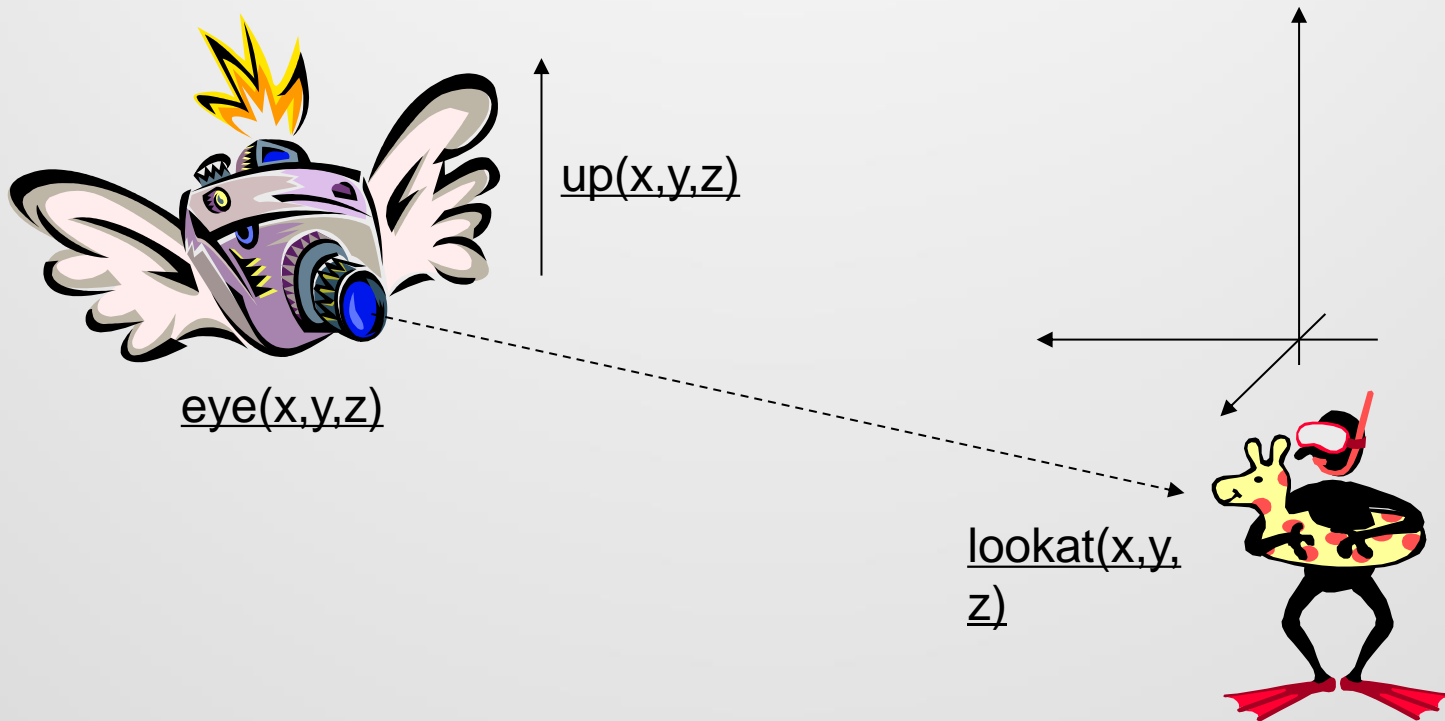
4

DI CHUYỂN CAMERA

Camera

- Đặt vị trí Camera

- Camera được đặt sử dụng kết hợp dịch chuyển và xoay
- Camera được đặt cùng vị trí với mắt của người nhìn

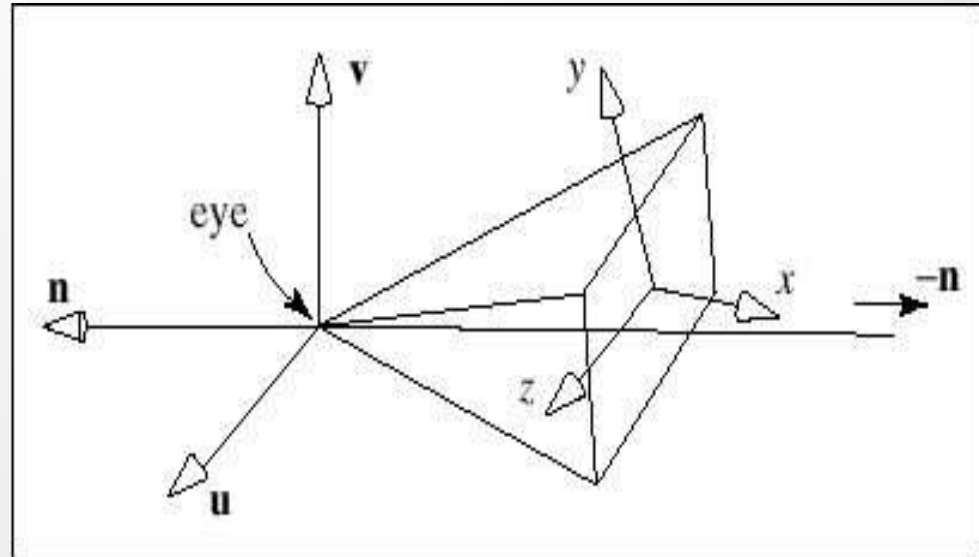


CAMERA TRONG OPENGL

```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
gluLookAt(eye.x, eye.y, eye.z,  
          look.x, look.y, look.z,  
          up.x, up.y, up.z);
```

Camera

- Camera có thể có các hướng và vị trí tùy ý
- Do đó chúng ta có thể coi như camera có các trục tọa độ riêng của nó



Chuyển động của camera

- Một camera có thể có 6 hướng di chuyển
 - "trượt" theo 3 hướng
 - "xoay" theo 3 hướng
- Chuyển động của camera
 - Camera có thể dịch chuyển tịnh tiến theo 3 hướng
 - Việc này được gọi là "trượt" (slide) camera

Camera

- Chuyển động của camera

- Để dịch chuyển camera theo trục u đơn giản là ta thực hiện

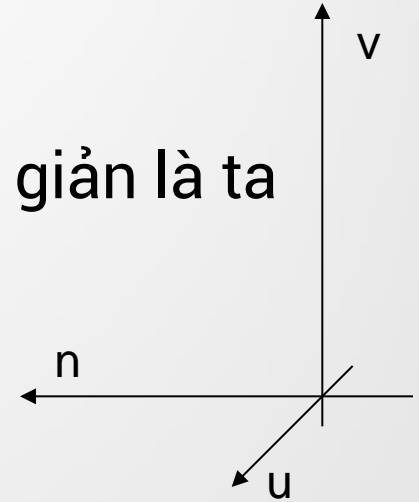
- $eye = eye + Du$

- Di chuyển camera theo trục bất kỳ

- $eye.x = dU*u.x + dV*v.x + dN*n.x$

- $eye.y = dU*u.y + dV*v.y + dN*n.y$

- $eye.z = dU*u.z + dV*v.z + dN*n.z$

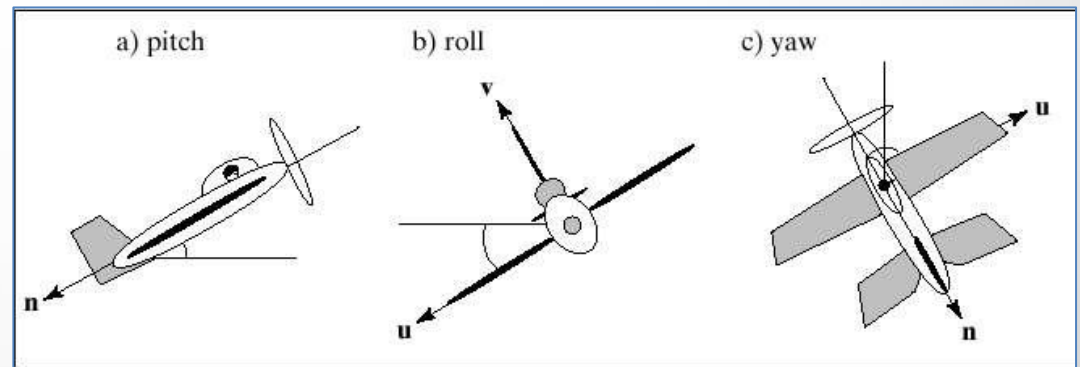


- Chuyển động của camera

- Bên cạnh việc di chuyển camera sang các vị trí khác, camera có thể được xoay nghiêng theo các hướng khác nhau để nhìn các vùng khác nhau của khung cảnh

CAMERA MOVEMENT

- Sử dụng nguyên lý lái máy bay để mô tả chuyển động của camera
 - Quay dọc theo chiều dài camera gọi là PITCH
 - Quay dọc theo chiều rộng camera gọi là ROLL
 - Quay dọc theo trục dọc gọi là YAW

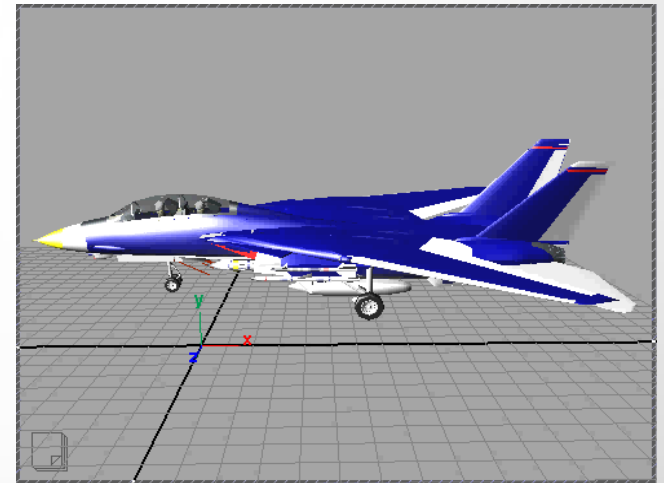
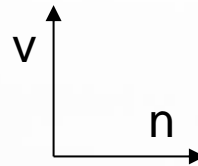


CÁC CHUYỂN ĐỘNG CỦA CAMERA

- Pitch

$$v' = \cos(\theta)v - \sin(\theta)n$$

$$n' = \sin(\theta)n + \cos(\theta)v$$

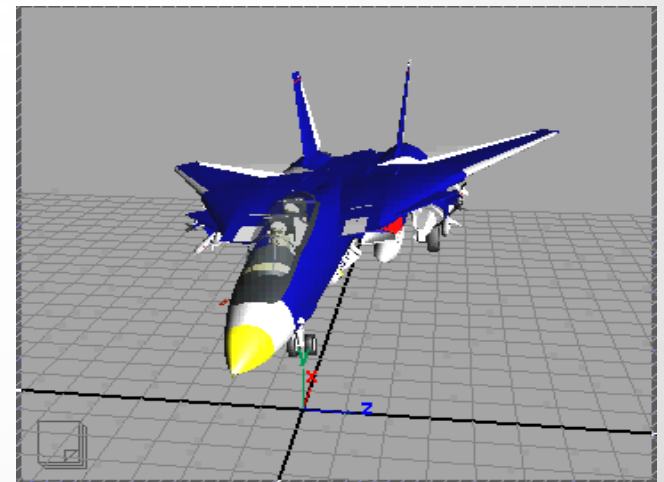
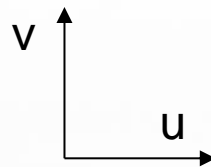


CÁC CHUYỂN ĐỘNG CỦA CAMERA

- Roll

$$u' = \cos(\theta)u + \sin(\theta)v$$

$$v' = \cos(\theta)v - \sin(\theta)u$$



CÁC CHUYỂN ĐỘNG CỦA CAMERA

- Yaw
 - $n' = \cos(\theta)n - \sin(\theta)u$
 - $u' = \sin(\theta)n + \cos(\theta)u$

