Departamento de Engenharia Informática

TP3 - Visualização 3D

Camera - lookat

DEI – CG

2017/18

jh, pjmm

Transformações Geométricas (Pipeline)

- 1. Definição **Objectos** e transformações (rotação, translação, escala, …)
- ~~2. Definição de modelos de luz e **cor**~~
- 3. Visualização / Projecção
  - Localização e orientação do **observador**
    - Coordenadas dos objectos em função das coordenadas do observa
  - **Volume de visualização**
    - Tipo de **projecção** (perspectiva/paralela)
- 4. ViewPort

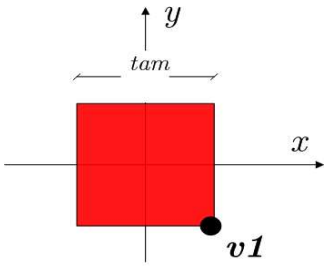
Esquecer a cor

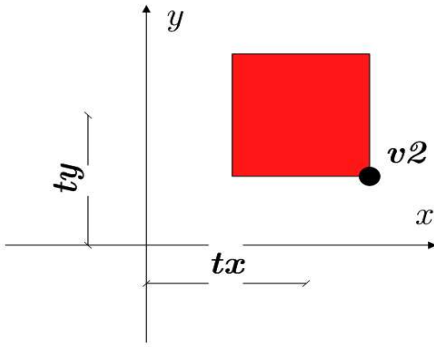
Coordenadas !

Transformações Geométricas

1. Definição **Objectos** e transformações (rotação, translação, escala, …)

```
glTranslatef(tx, ty, 0);                    // transformações
glColor4f(VERMELHO);                        // Cores
glBegin(GL_POLYGON);                        // primitivas (objectos)
            glVertex3i(  0,   0, 0);
            glVertex3i(tam,   0, 0);
            glVertex3i(tam, tam, 0);
            glVertex3i(0, tam, 0);
glEnd();
```



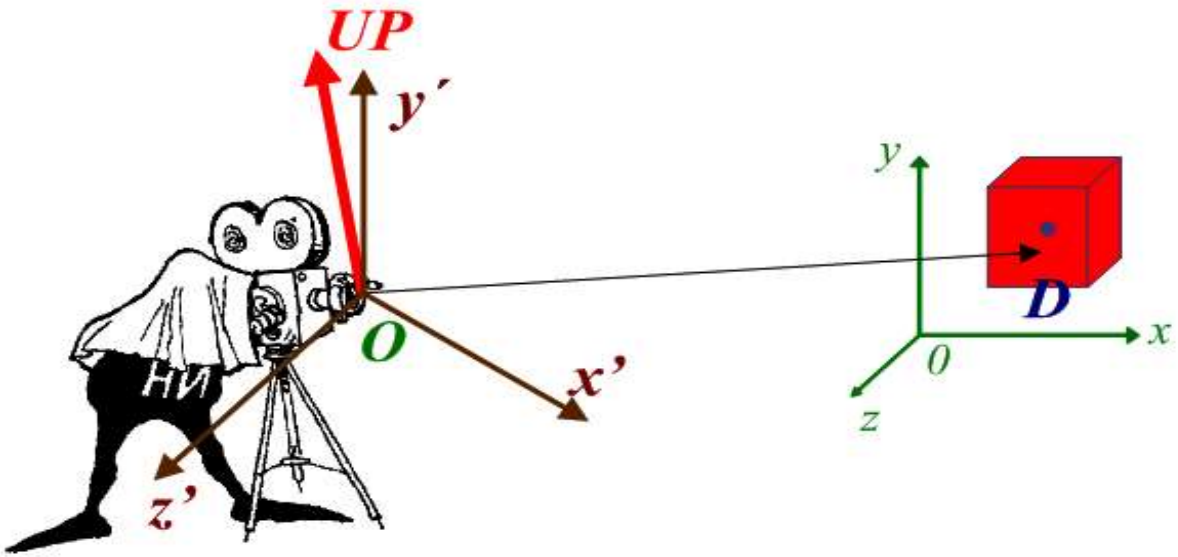


$$\begin{bmatrix} x2 \\ y2 \\ z2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ z1 \\ 1 \end{bmatrix}$$

Transformações Geométricas

- 2. Transformações de Visualização (***Definir a localização do observador e sua orientação***).

gluLookAt(Ox, Oy, Oz, Dx, Dy, dz, UPx, UPy, UPz);

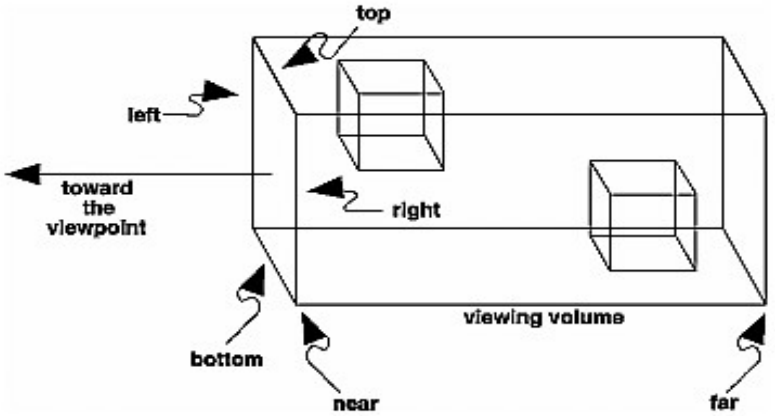


$$\begin{bmatrix} Px' \\ Py' \\ Pz' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{1x} & R_{2x} & R_{3x} & Rx \bullet Q \\ R_{1y} & R_{2y} & R_{3y} & Ry \bullet Q \\ R_{1z} & R_{2z} & R_{3z} & Rz \bullet Q \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix}$$

Transformações Geométricas

- 3. Projecção e Volume de Visualização

glOrtho (left, right, bottom, top, near, far);



Transformações Geométricas

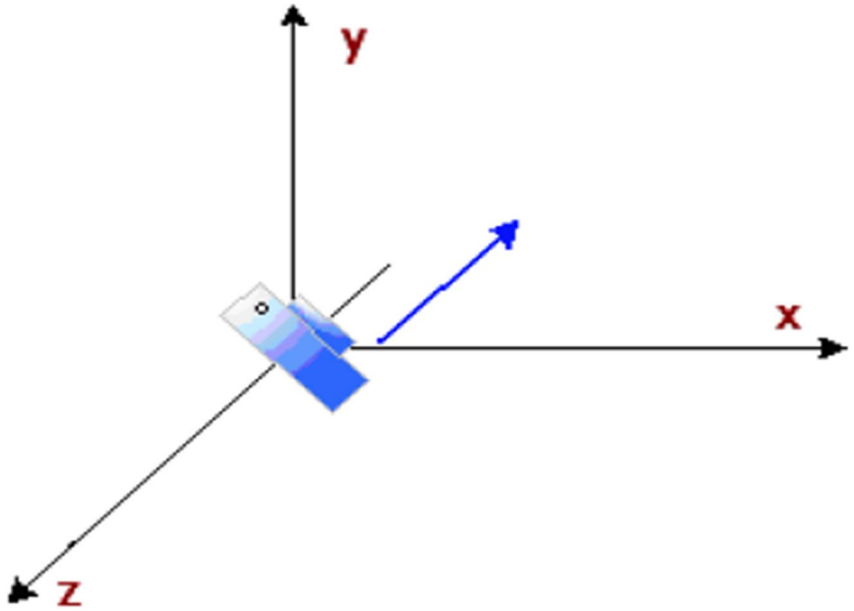
- 3. Projecção e Volume de Visualização

- Por Omissão

glOrtho (-1, 1, -1, 1, -1, 1);

Observador:

- **na origem (0,0,0)**
- **A olhar no sentido contrário ao eixo Z**



Transformações Geométricas

- 3. Projecção e Volume de Visualização

Nota: fez-se, nos trabalhos anteriores

- glOrtho2D(-5, 5, -5, 5)
- [xmin, xmax, ymin, ymax]= (-5, 5, -5, 5)
-
- Observador origem olhar –z
- Volume de visualização (-5,5,-5,5)

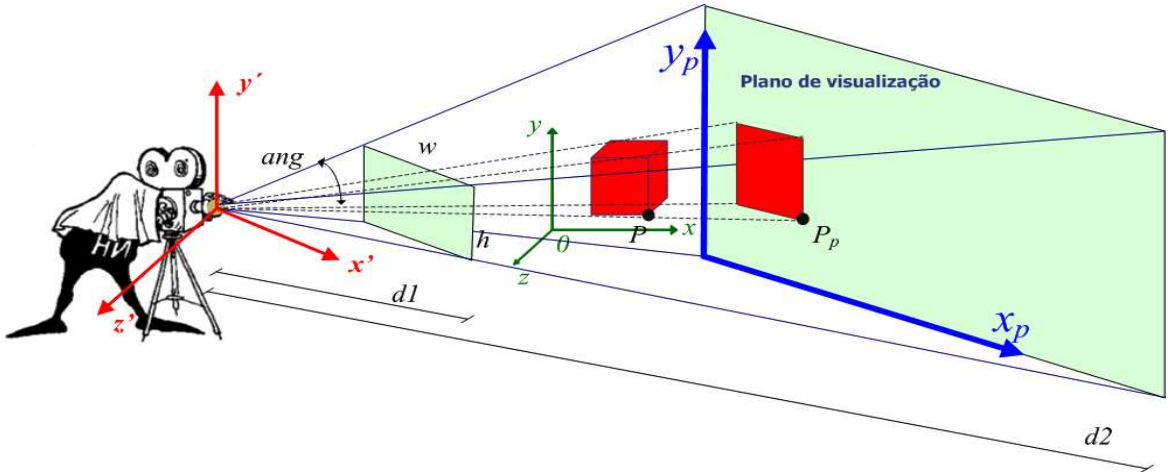
Eixos XY (2D)

Transformações Geométricas

- 3. Projecção e Volume de Visualização

gluPerspective(angulo, wScreen/hScreen, d1, d2);

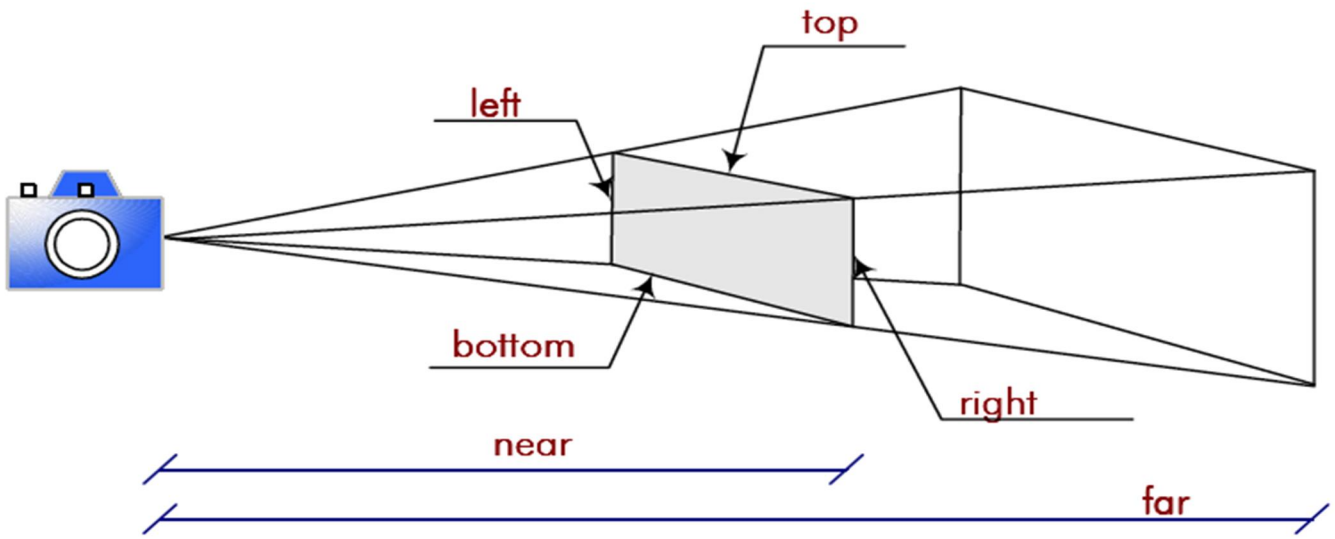
*d1,d2 - distâncias (***sempre positivos***)*



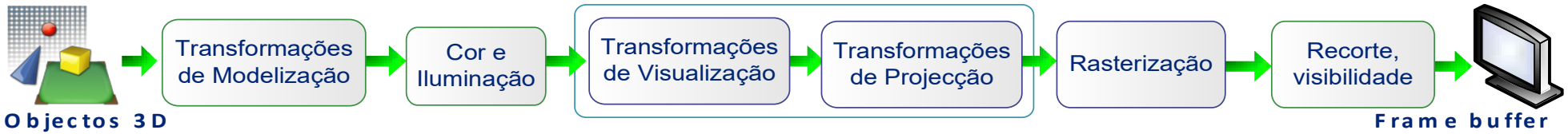
Transformações Geométricas

- 3. Projecção e Volume de Visualização

```
void glFrustum(GLdouble left, GLdouble right, GLdouble bottom,
               GLdouble top, GLdouble near, GLdouble far);
```

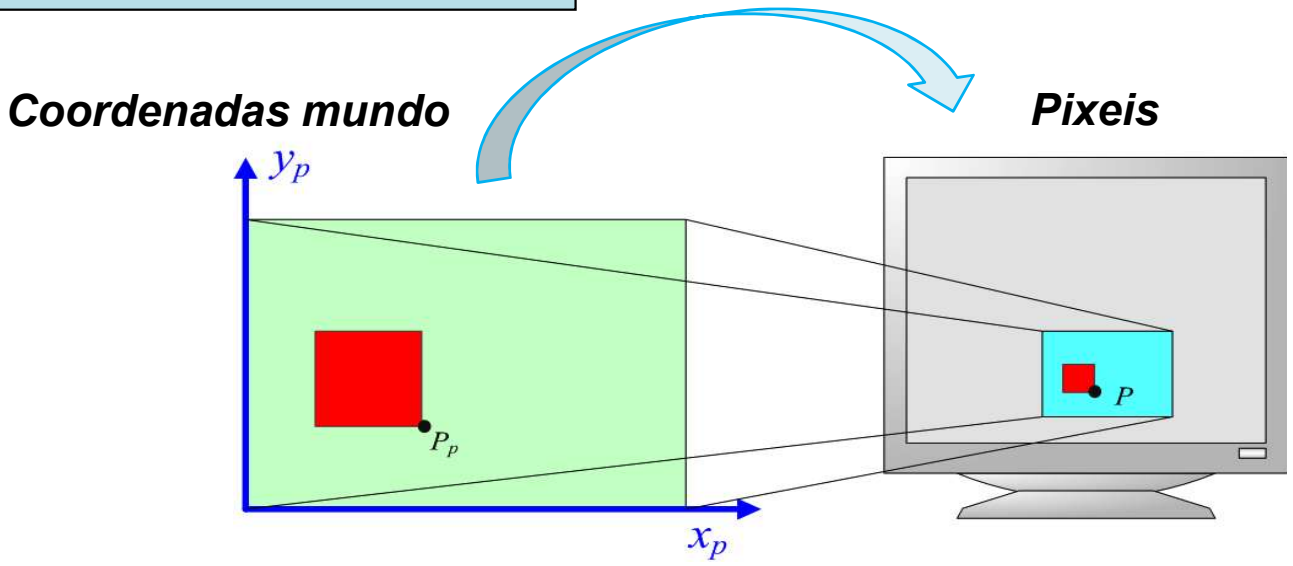


Transformações Geométricas



- 4. ViewPort

glViewport (Xo, Yo, wScreen, hScreen);



Transformações Geométricas

Ordem Conceptual:

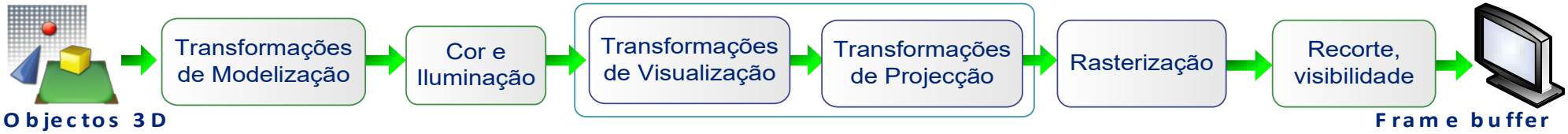
1. **Definir a cena** para ser fotografada, incluindo a especificação dos objectos
   Equivalente à etapa de modelização

2. **Posicionar a câmara** para fotografar a cena
   Equivalente a especificar as transformações de visualização

3. **Seleccionar a lente da câmara** ou ajustar o zoom
   Equivalente a especificar as transformações de projecção geométricas

4. **Determinar o tamanho final da foto** (maior ou menor)
   Equivalente a especificar a janela de visualização (viewport)

Transformações Geométricas



glViewport (0,0,wScreen, hScreen);	Janela Visualização	M1

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-dX, dX, -dY, d\Y, -dZ, dZ);
	Projecção	M2=**Projection**

glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(Ox,Oy,Oz, Dx,Dy,Dz, UPx,UPy,UPz);
	View (observador)	M3a

glTranslatef(tx, ty,0);
glColor4f(VERMELHO);
glBegin(GL_POLYGON);
glVertex3i(0, 0, 0);
glVertex3i(tam, 0, 0);
glVertex3i(tam, tam, 0);
glVertex3i(0, tam, 0);
glEnd();
	Model (objectos)	M3b
		
		M3=**ModelView**

Matrizes ModelView e Projection

- **PROJECTION**
  - Definição do modelo de perspectiva / volume de visualização.

- **MODELVIEW**
  - Posição e orientação os **objectos** na cena de visualização (modelos)
  - Localização e orientação do **observador (visualizador)**

- Para especificar qual a matriz em operação no momento o OpenGL disponibiliza os comandos:

  glMatrixMode (GL_PROJECTION)

  glMatrixMode (GL_MODELVIEW)

- Para permitir que a transformação actual seja re-inicializada existe o comando

  glLoadIdentity().

Visualização 3D

- Trabalho
  - **1. Objectos**
  - **2. Observador**
  - **3**. Projecções
  - **4.** Janela visualização

Visualização 3D

1. Objectos

- **GLUT**

  - **glutSolidCube, glutWireCube**
  - **glutSolidTeapot, glutWireTeapot**

  - glutSolidSphere, glutWireSphere
  - glutSolidCone, glutWireCone
  - glutSolidTorus, glutWireTorus
  - glutSolidDodecahedron, glutWireDodecahedron
  - glutSolidOctahedron, glutWireOctahedron
  - glutSolidTetrahedron, glutWireTetrahedron
  - glutSolidIcosahedron, glutWireIcosahedron

Visualização 3D

1. Objectos

- **GLU**

CYLINDER

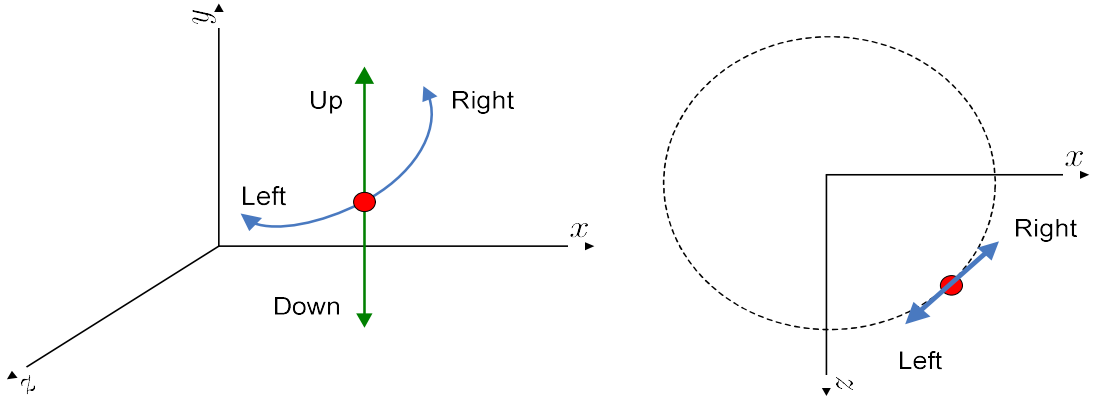
- GLUquadricObj* y = gluNewQuadric ( );
- void gluCylinder(GLUquadricObj * obj,
  GLdouble baseRadius, GLdouble topRadius,
  GLdouble height,
  GLint slices, GLint stacks)

gluCylinder(y, 2, 2, 4, 100, 100);

Tem o mesmo movimento de translação, rotação que o paralelepípedo da aula anterior

Visualização 3D

- **2. Observador**
  - Subir/descer
  - Girar



Visualização 3D

- **3. Projecções**
  - Ortogonal
  - Perspectiva

  - Pode-se esquecer por agora !!

Visualização 3D

- **4. Janela visualização**
  - Viewport
  - Proxima aula !

Observador/camera

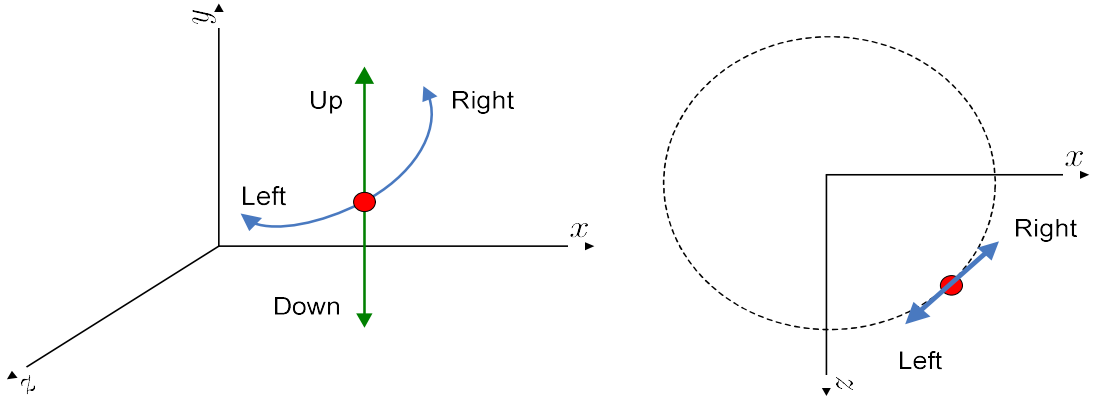
- **Para a aula hoje**

```
gluLookAt( Ox, Oy, Oz, Dx, Dy, dz,  UPx, UPy, UPz );
```

  - **Depende do objectivo !!!**
  - **Três modos diferentes de observar a cena**
    - Observador móvel – foco fixo
    - Observador fixo– foco móvel
    - Observador e foco móveis

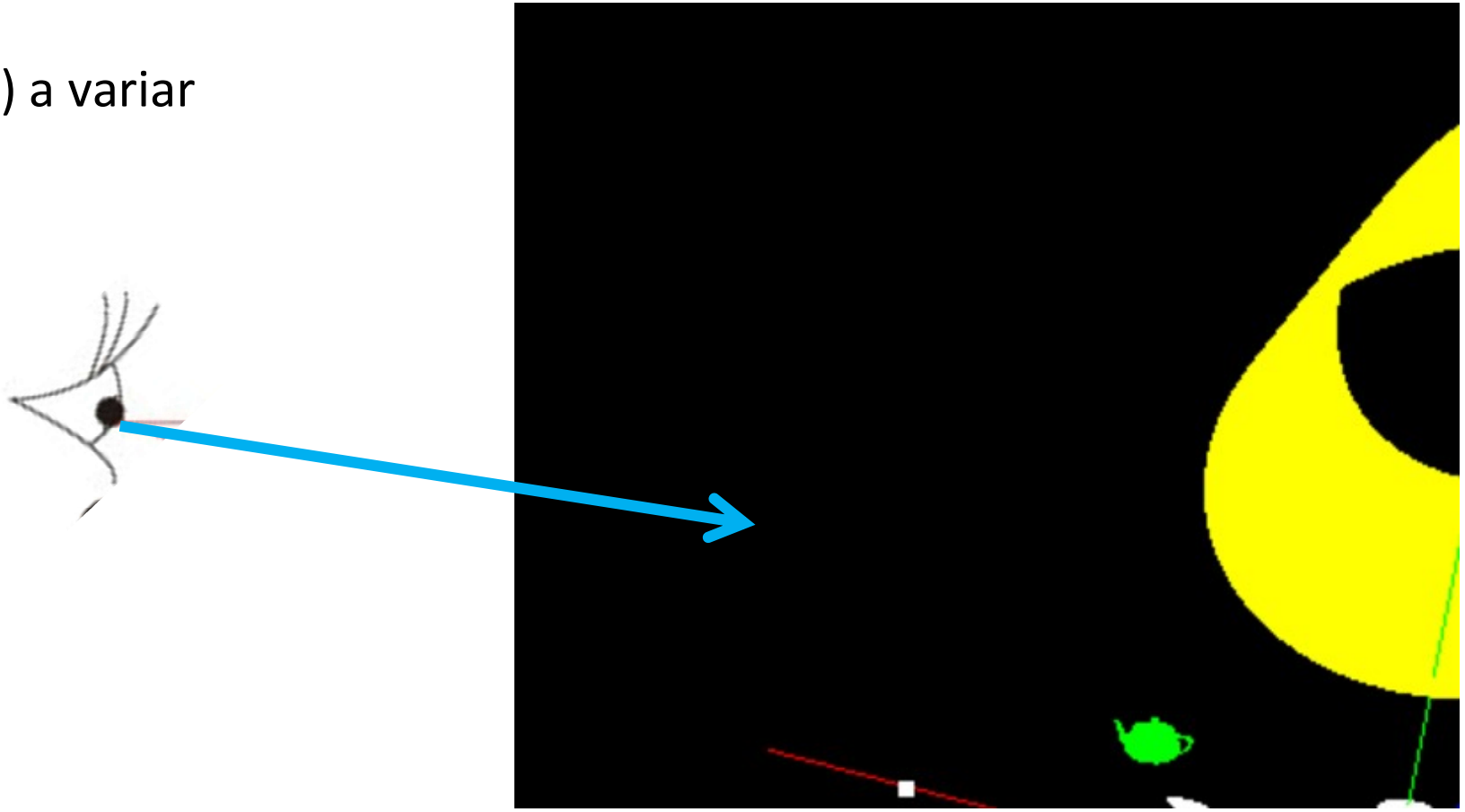
Modo 0

- **Observador móvel – foco fixo**
- Aula passada
  - Subir/descer
  - Girar



Modo 1

- **Observador fixo – foco móvel**
- TY=0
- (Tx,Tz) a variar



Modo 2

- **Observador e foco móveis**
  - (Px,Py) a variar
  - (Tx,Tz) a variar

  - **glookat**(P[0], P[1], P[2], T[0], T[1], T[2], 0,1,0)