

DISEÑO ASISTIDO POR ORDENADOR

4º Curso Ingeniería Informática

Dpto. Lenguajes y Sistemas Informáticos

Curso 2005/06

J.C. Torres

TEMA 3. Técnicas de interacción

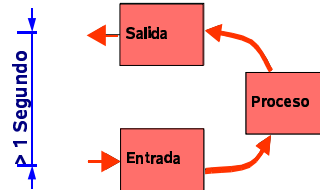
- 3.1 Estructura de un sistema gráfico interactivo
- 3.2 Posicionamiento
- 3.3 Selección
- 3.4 Transformaciones y edición
- 3.5 Control de la cámara

Bibliografía

Foley J.D. et al: "Computer Graphics. Theory and Practice".
 Hearn D.D.; Baker M.P.: "Gráficas por computadora".
 Newman W.M.; Sproull R.F.: "Principles of Interactive Computer Graphics".
 McGraw-Hill 1981
 Sproull R.; Southerland: "Device independent Graphics". McGraw-Hill 1986

J.C. Torres

3.1 Interacción: **Sistemas gráficos no interactivos**



Procesos complejos que no se pueden realizar mientras se modifica el modelo

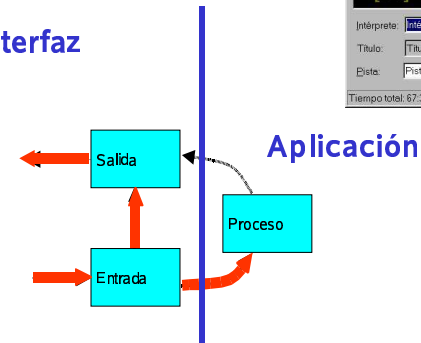
- Conversión de formato
- Generación de imágenes realistas
- Cálculo de momento de inercia de un sólido

J.C. Torres

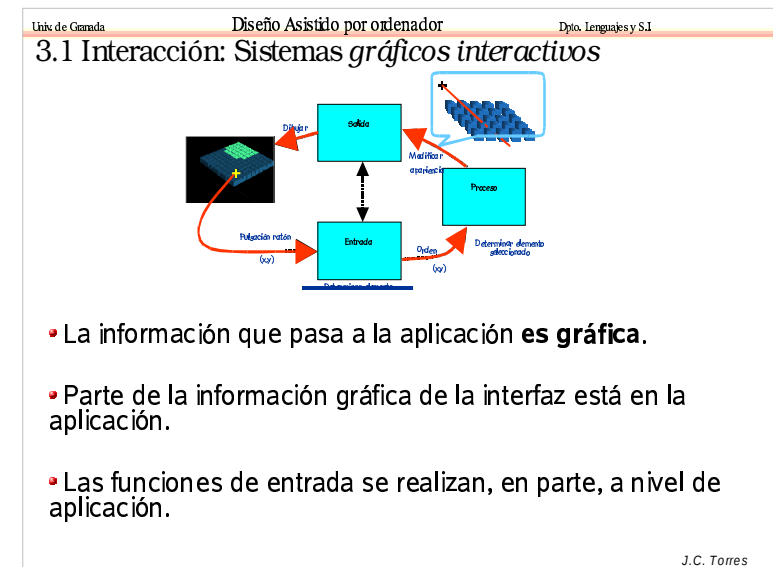
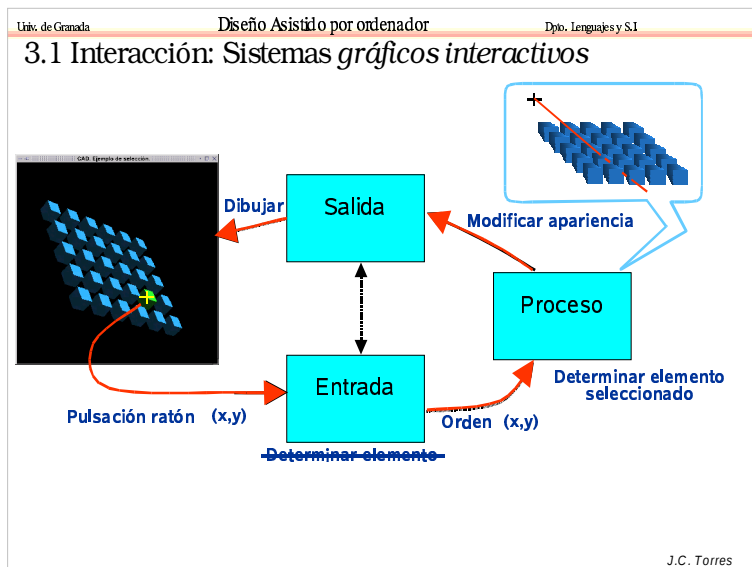
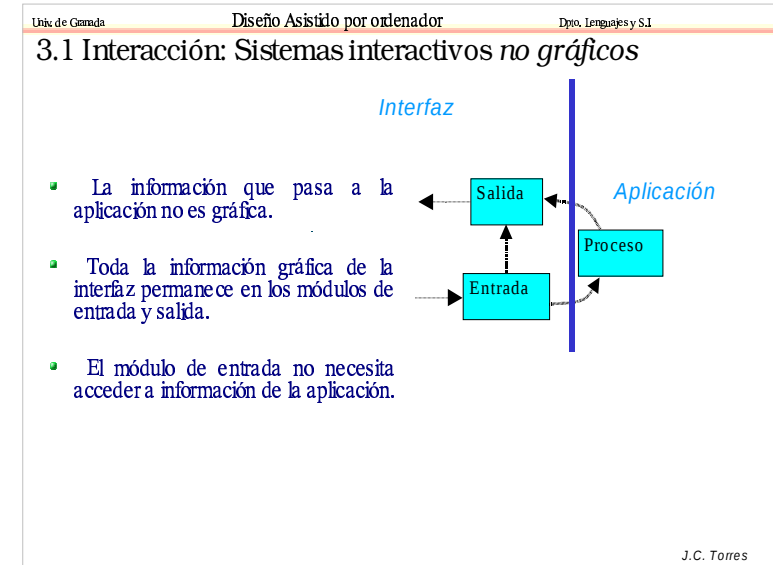
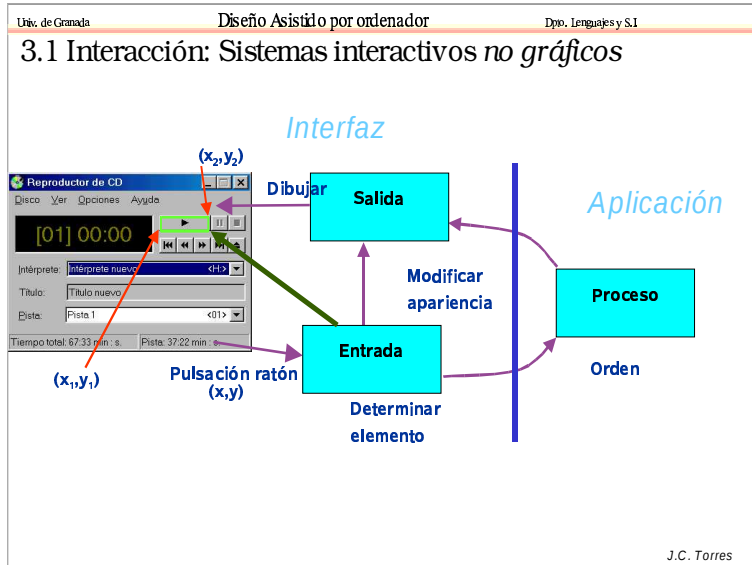
3.1 Interacción: **Sistemas interactivos no gráficos**

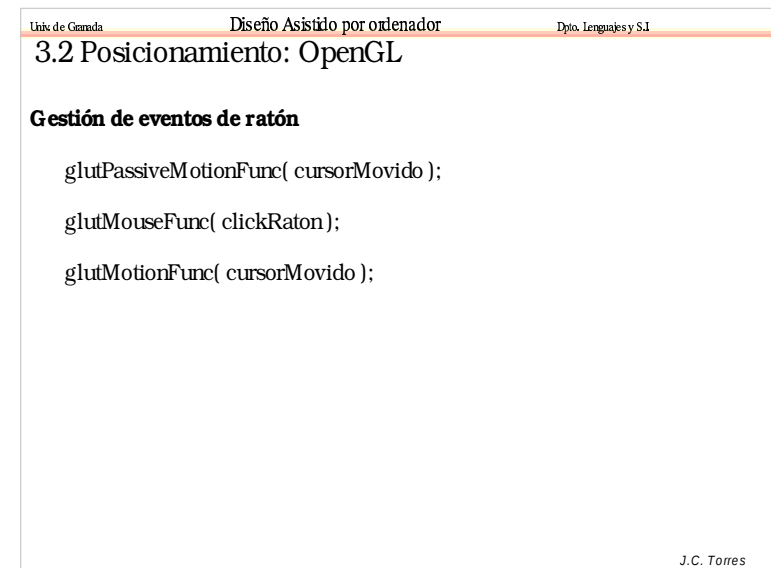
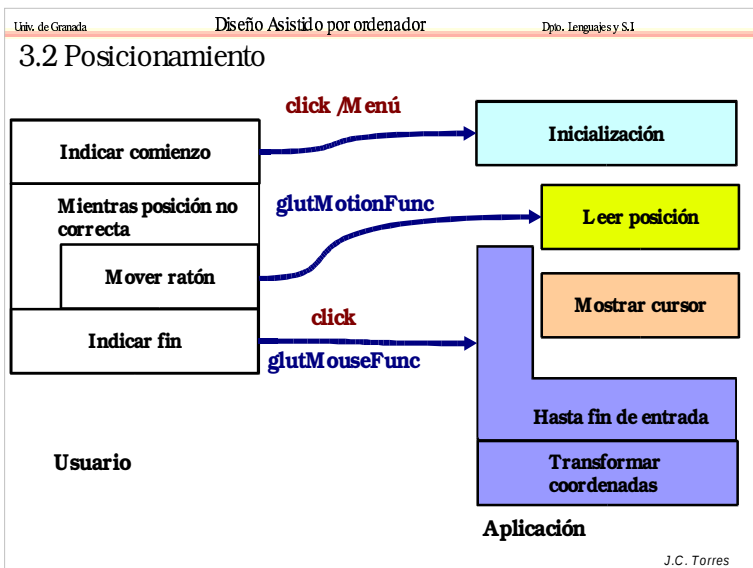
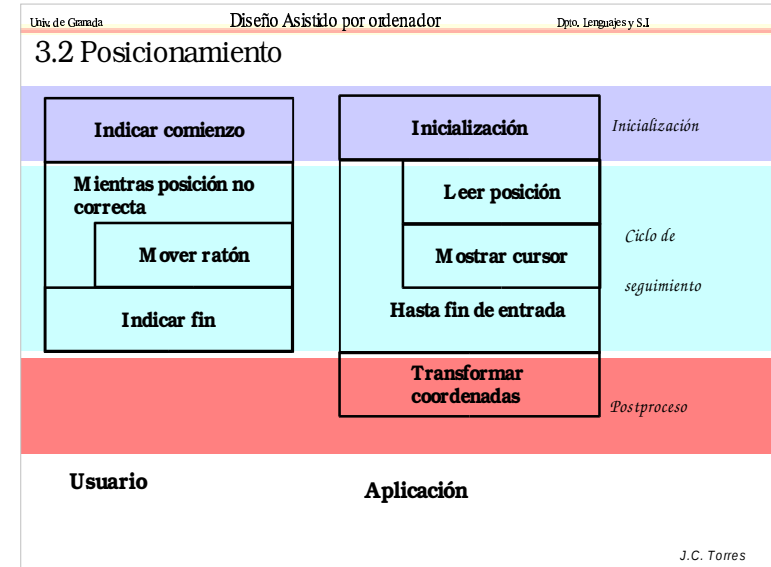
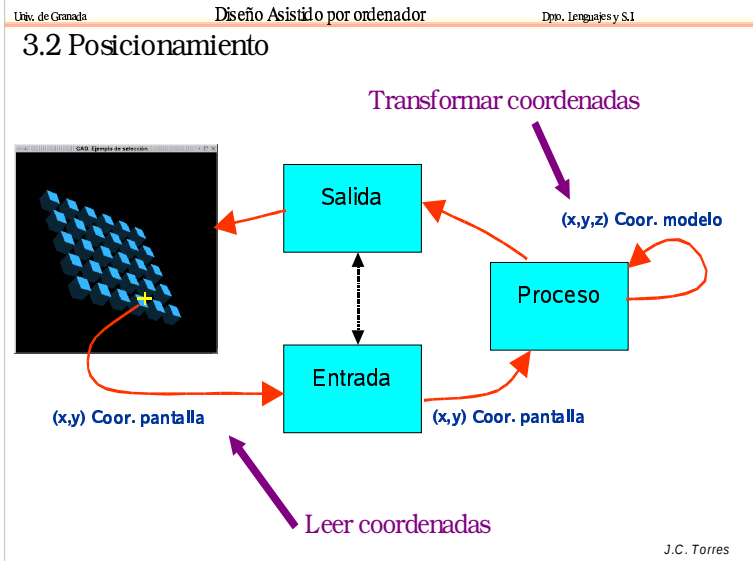
Interfaz

≤ 0.1 Segundo



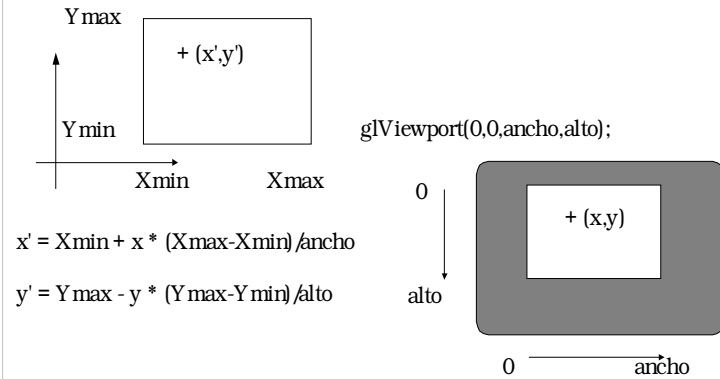
J.C. Torres





3.2 Posicionamiento: *Cálculo de posición*

```
glOrtho(Xmin,Xmax,Ymin,Ymax,Zmin,Zmax);
```



J.C. Torres

3.2 Posicionamiento: OpenGL

Ejemplo: gestión de cursor software

```
static void Dibuja( void )
{
    glClearColor(0,0,0,1);
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glPushMatrix();
```

```
    drawCursor(); //Realimentación del cursor
```

```
//Introduce la geometria
```

```
.....
```

J.C. Torres

3.2 Posicionamiento: OpenGL

```
void drawCursor()
{
    if(drawingcursor) {
        glPushAttrib(GL_CURRENT_BIT);
        if(softCursorEnabled) {
            glBegin(GL_LINES);
            ..... //Dibuja el cursor en (cursorx,cursory)
            glEnd();
        }
        glPopAttrib(); }
}

void cursorMovido( int x, int y ){
    cursorx = 2.0* ((1.0*x) /VAncho -0.5); //en coord.
    cursory = - 2.0* ((1.0*y) /VAlto - 0.5); //normalizadas
    glutPostRedisplay();
}
```

J.C. Torres

3.2 Posicionamiento: OpenGL

```
void clickRaton( int boton, int estado, int x, int y )
{
    float xc,yc;
    switch ( boton ) {
        case GLUT_LEFT_BUTTON:
            if ( estado == GLUT_DOWN ) { //Se ha pulsado
                softCursorEnabled= 1; //Activar realimentación
                cursorMovido(x,y); } //Actualizar pos. cursor
                getCursor(&xc,&yc); //Obtiene posicion en coord. de modelo
                addBox(xc, yc); //Añade de primer punto
            else { //Se ha soltado
                cursorMovido(x,y); //Actualizar pos. cursor
                disableDrawCursor(); //No realimentar
                getCursor(&xc,&yc); //Posición coord. modelo
                addBox(xc, yc); //Acción de la aplicación
                glutPostRedisplay(); } //Actualizar imagen
            break;
        case ...
    }
}
```

J.C. Torres

3.2 Posicionamiento: OpenGL

```

addBox( float X, float Y){
if(firstPoint){
    lastX = X;
    lastY = Y;
    firstPoint=0;
    setFeedbackMode(Feedback,X,Y);}
else { Model[N][0]=lastX;
    Model[N][1]=lastY;
    Model[N][2]=X;
    Model[N][3]=Y;
    firstPoint=1;
    N++;
    setFeedbackMode(0,0,0);
    }
}

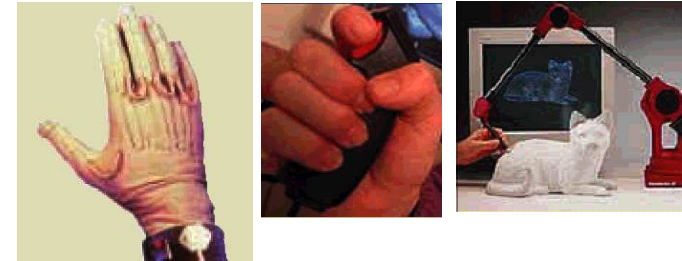
void getCursor(float *xc, float *yc) {
    *xc = cursorx*WAncho;
    *yc = cursory*WAlto;
}

```

J.C. Torres

3.2 Posicionamiento 3D

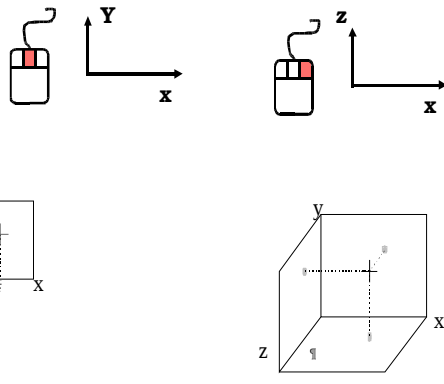
- 3 grados de libertad
- Realimentación
- Restricciones



J.C. Torres

3.2 Posicionamiento 3D

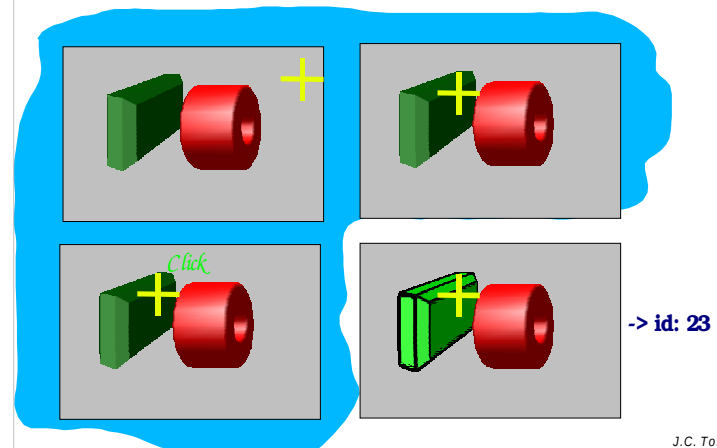
- 3 grados de libertad
- Realimentación
- Restricciones



J.C. Torres

3.3 Selección (pick)

Acción usuario -> Identificador de componente del modelo

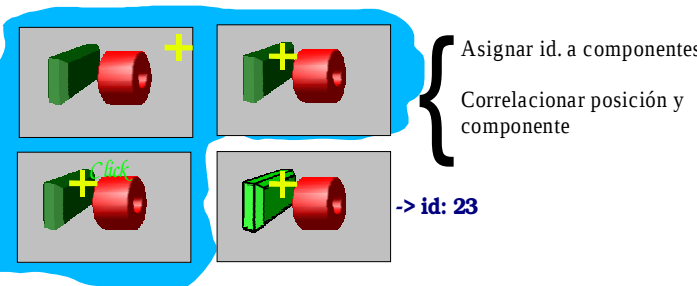


J.C. Torres

Univ. de Granada Diseño Asistido por ordenador Dpto. Lenguajes y S.I.

3.3 Selección (*pick*)

Acción usuario -> Identificador de componente del modelo



Asignar id. a componentes
Correlacionar posición y componente

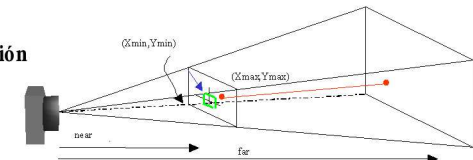
-> id: 23

J.C. Torres

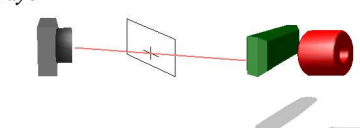
Univ. de Granada Diseño Asistido por ordenador Dpto. Lenguajes y S.I.

3.3 Selección (*pick*)

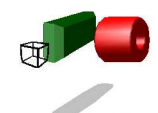
a) Visualización



b) Intersección con rayo



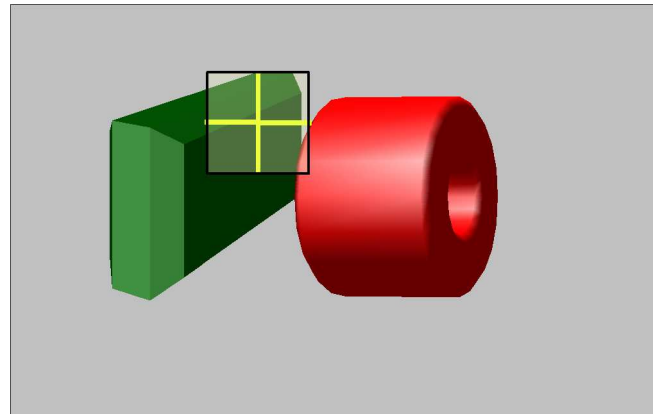
c) en el modelo
(posicionamiento 3D)



J.C. Torres

Univ. de Granada Diseño Asistido por ordenador Dpto. Lenguajes y S.I.

3.3 Selección (*pick*)



J.C. Torres

Univ. de Granada Diseño Asistido por ordenador Dpto. Lenguajes y S.I.

3.3 Selección (*pick*). OpenGL

```

glLoadName(i) Sustituye el nombre activo por i
glPushName(i) Apila el nombre i
glPopName() Desapila un nombre
glInitNames()
....
glPushMatrix();
..  glPushName(i);           //Identificador del pato
    glPushName(PatoId);      //Identificador de parte
    glPushMatrix();          //Dibujar cuerpo
    ...
    glPopMatrix();
    glLoadName(PatasId);     //Cambiamos id de parte
    glPushMatrix();          //Dibujar las patas
    ....
    glPopMatrix();
    glPopName();             //Quitamos id parte
    glPopName();             //Quitamos id pato
glPopMatrix();
  
```

J.C. Torres

3.3 Selección (*pick*): OpenGL

```
void pick(int x, int y) {
    GLuint selectBuf[BUFSIZE];
    GLint hits, viewport[4];
```

```
glGetIntegerv (GL_VIEWPORT, viewport);
glSelectBuffer (BUFSIZE, selectBuf);
```

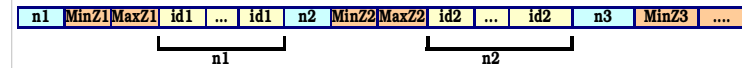
```
glRenderMode (GL_SELECT);
glInitNames();
```

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
gluPickMatrix ( x, viewport[3] - y, 5.0, 5.0, viewport[0]);
viewtrans(); // Sin inicializar matriz de proyección
dibujar ();
```

```
hits = glRenderMode (GL_RENDER);
setviewtrans(); // Inicializando matriz de proyección
```

C. Torres

3.3 Selección (*pick*): OpenGL.



```
hits = glRenderMode (GL_RENDER);
if(hits>0) {
    b=0;
    Zmin = -1;
    for(k=0;k<hits;++k){ // Determinar el más próximo
        if(buff[b]!=0){
            z = buff[b+1];
            if(Zmin== -1 || z<Zmin) {
                *i=buff[b+3];
                *parte=buff[b+4];
                Zmin=z; }
            b+=buff[b]+3; }
    return *i; }
```

J.C. Torres

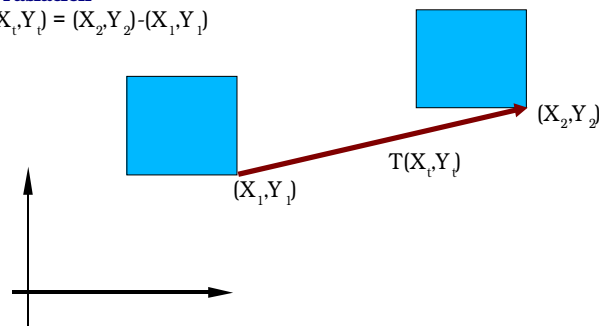
3.4 Transformaciones geométricas

a) Interacción: Arrastre

b) Cálculo de parámetros

Traslación

$$(X_t, Y_t) = (X_2, Y_2) - (X_1, Y_1)$$



J.C. Torres

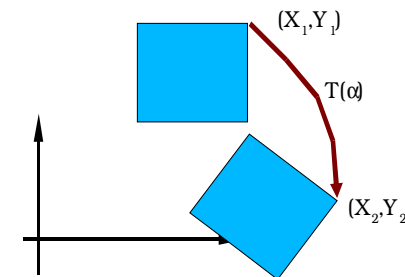
3.4 Transformaciones geométricas

a) Interacción: Arrastre

b) Cálculo de parámetros

Rotación

$$\alpha = \text{angulo}(X_2, Y_2) - \text{angulo}(X_1, Y_1)$$



J.C. Torres

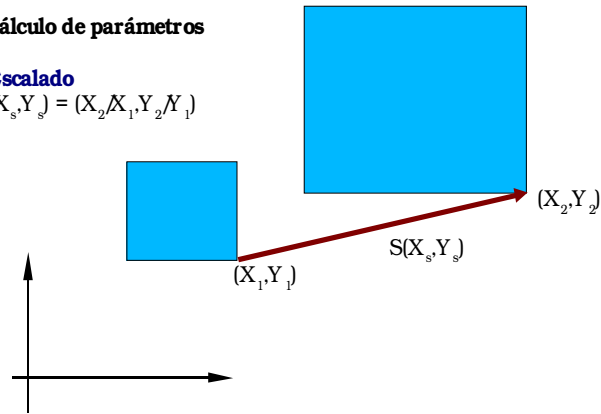
3.4 Transformaciones geométricas

a) Interacción: Arrastre

b) Cálculo de parámetros

Escalado

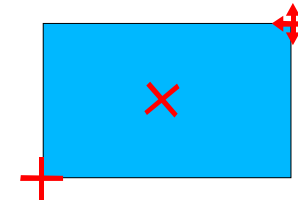
$$(X_s, Y_s) = (X_2/X_1, Y_2/Y_1)$$



J.C. Torres

3.4 Transformaciones geométricas: realimentación y actuadores

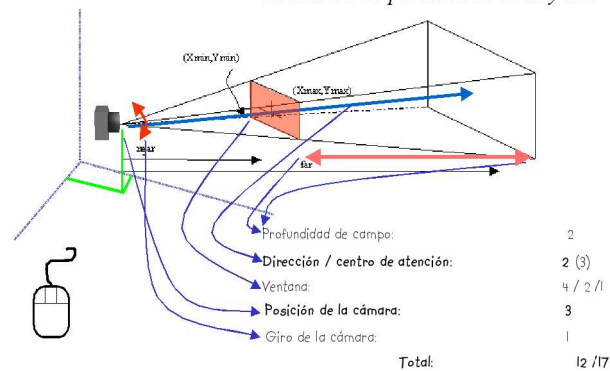
- Realimentación => visualizar el modelo durante el proceso.
- Las acciones sobre el modelo se pueden mostrar como elementos a manipular.



J.C. Torres

3.5 Control de la vista

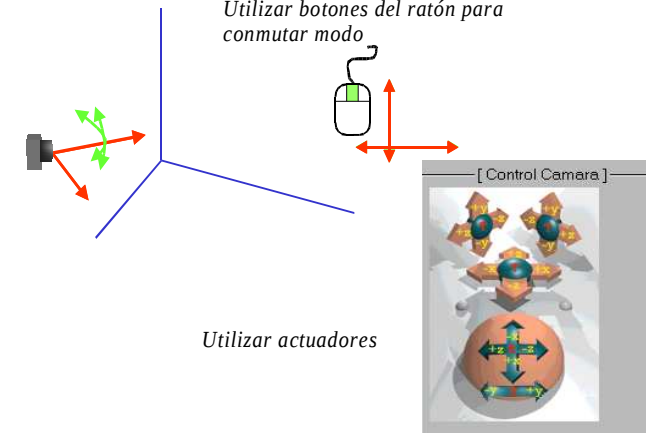
El número de parámetros es muy alto



J.C. Torres

3.5 Control de la vista

Utilizar botones del ratón para conmutar modo



Utilizar actuadores

J.C. Torres

3.5 Control de la vista

Debe haber coherencia entre el movimiento del dispositivo y de la cámara

