

## Apéndice C. Manual de Programador

Funciones/Iconos de Control

Funciones/Iconos de Selección

Barra de coordenadas

---

### Módulo Programación

Keyframe

Métodos

```
virtual void AnimateCamera(C3dCamera* pCamera, double dTime);
```

```
virtual void AnimateObject(C3dObject* pObject, double dTime);
```

```
virtual void Serialize(CArchive& ar, int iVersion);
```

```
void SetObjectAttributes(C3dObject* pObject);
```

Atributos

```
CKeyFrame* m_pKeyFramePrevious;
```

```
double m_dTime; // Time reference for this keyframe
```

```
GLfloat m_fOrigin[4]; // Objects world origin
```

```
GLfloat m_fScale[3]; // Objects scale
```

```
GLfloat m_fRotation[3]; // Objects rotation
```

```
GLfloat m_fTranslate[3]; // Objects translation about origin
```

```
C3dColor m_Color; // Color component
```

```
// The following colors apply to light objects
```

```
C3dColor m_ColorAmb; // Ambient Color component
```

```
C3dColor m_ColorDif; // Diffuse Color component
```

```
C3dColor m_ColorSpc; // Specular Color component
```

```
GLfloat m_fSpotAngle; // Spotlights' spot angle
```

Program Point

Métodos

Atributos

```
GLfloat m_fOrigin[4]; // Points world origin/*
```

```
GLfloat m_fRotation[4]; // Points world origin
```

```
int trayectoria;
```

```
GLfloat tiempo;
```

```
GLfloat espera;
```

```
GLfloat velocidad;
```

```
C3dColor m_Color;
```

Program Point List

Métodos

Atributos

Funciones/Iconos de Programación

---

## Módulo Animación

World -Animate

Object- Animate

```
void C3dObject::Animate(double dTime)
```

```
double dTimeBetweenFrames = m_dTime-
```

```
m_pKeyFramePrevious->m_dTime;
```

```
double dTimeSinceLastFrame = dTime-m_pKeyFramePrevious->m_dTime;
```

```
double dPercentThisFrame =
```

```
dTimeSinceLastFrame/dTimeBetweenFrames;
```

```
// Calculate the Origin incremental index values
```

```
fKeyIncrement[X] = (GLfloat)((m_fOrigin[X] -  
m_pKeyFramePrevious->m_fOrigin[X]) * dPercentThisFrame);
```

```
fKeyIncrement[Y] = (GLfloat)((m_fOrigin[Y] -  
m_pKeyFramePrevious->m_fOrigin[Y]) * dPercentThisFrame);
```

```
fKeyIncrement[Z] = (GLfloat)((m_fOrigin[Z] -  
m_pKeyFramePrevious->m_fOrigin[Z]) * dPercentThisFrame);
```

```
if(fKeyIncrement[X])
```

```
pObject->m_fOrigin[X] = m_pKeyFramePrevious->m_fOrigin[X] +
```

```
fKeyIncrement[X];
```

```

if(fKeyIncrement[Y])
pObject->m_fOrigin[Y] = m_pKeyFramePrevious->m_fOrigin[Y] +
fKeyIncrement[Y];
if(fKeyIncrement[Z])
pObject->m_fOrigin[Z] = m_pKeyFramePrevious->m_fOrigin[Z] +
fKeyIncrement[Z];

```

Robot-Animante

```
void C3dObjectRobot::Animate(double dTime)
```

```

if (pnewPoint->trayectoria==LIN)
{
anterior=anterior+espera;
double dTimeBetweenFrames=posterior-anterior;
double dTimeSinceLastFrame = (dTime-currenttime)-anterior;
double dPercentThisFrame =
dTimeSinceLastFrame/dTimeBetweenFrames;

```

```

fKeyIncrement[X] = (GLfloat)
((pnewPoint->m_fOrigin[X]-pcurPoint->m_fOrigin[X]) *
dPercentThisFrame);
fKeyIncrement[Y] = (GLfloat)
((pnewPoint->m_fOrigin[Y]-pcurPoint->m_fOrigin[Y]) *
dPercentThisFrame);
fKeyIncrement[Z] = (GLfloat)
((pnewPoint->m_fOrigin[Z]-pcurPoint->m_fOrigin[Z]) *
dPercentThisFrame);
tx=pcurPoint->m_fOrigin[X]+fKeyIncrement[X];
ty=pcurPoint->m_fOrigin[Y]+fKeyIncrement[Y];
tz=pcurPoint->m_fOrigin[Z]+fKeyIncrement[Z];
SetPosition(tx*100,ty*100,tz*100);
}

```

Funciones/Iconos de Animación

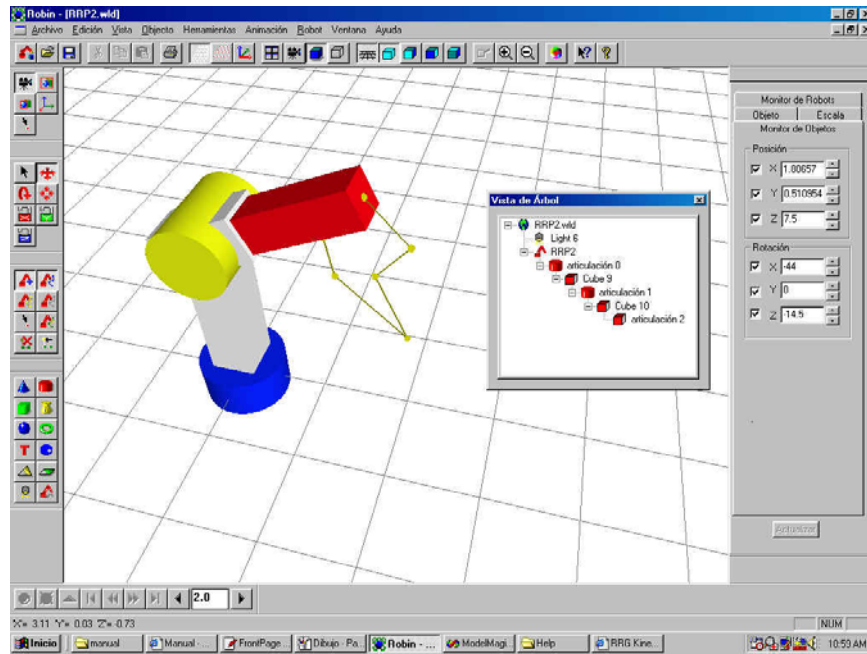


Figura C.1 Vista del mundo

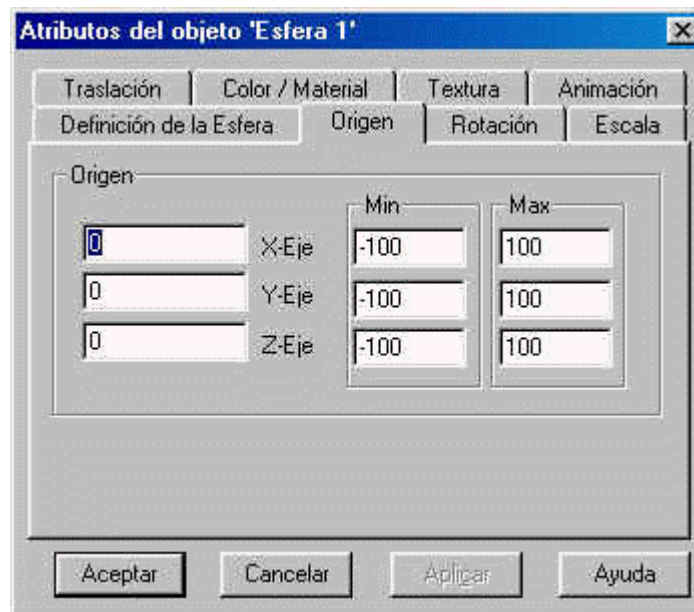


Figura C.2 Origin Dialog

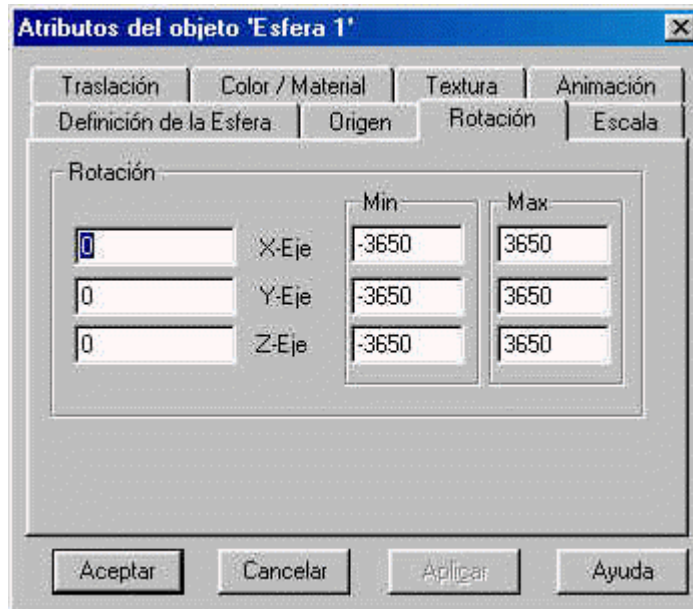


Figura C.3 Rotate Dialog

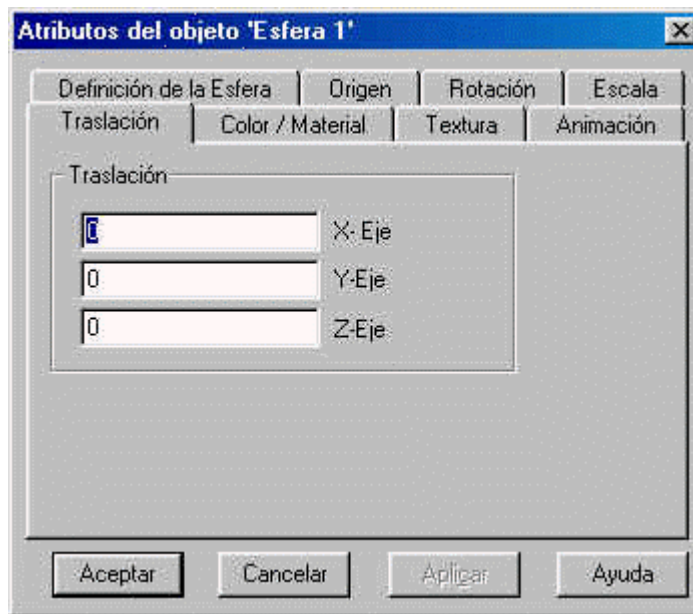


Figura C.4 Translate Dialog

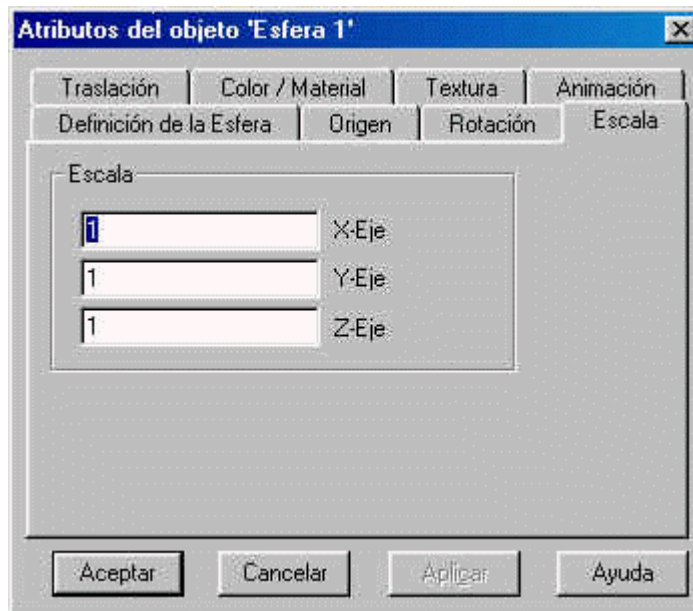


Figura C.5 Scale Dialog

Razo Rodríguez, A. F. 1999. **ROBIN: Herramienta gráfica de simulación de robots industriales en un ambiente virtual**. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas Puebla. Diciembre. Derechos Reservados © 1999.