

MANUAL DE USUARIO, PROYECTO FINAL

CONTRERAS PERALTA MARCO ISAAC

FACULTAD DE INGENIERÍA

UNAM

GRUPO: 2

FECHA DE ENTREGA:20/11/2019

OBJETIVO:

Demostrar los conocimientos obtenidos en el laboratorio de computación gráfica, aplicándolos en un proyecto donde se modelaron dos cuartos con al menos 5 elementos que dan la característica de realidad. Además de 3 animaciones en total

IMPLEMENTACION:

El desarrollo se obtuvo a través del lenguaje c++ con la librería opengl, todo fue programado pieza por pieza hasta llegar a cada uno de los modelos, que se representan mediante funciones y son mandados a llamar en la función display();

El modelo en general es el club-house (lugar donde se cambian y se distraen los jugadores) del equipo de baseball ST LOUIS CARDINALS, por ende la mayoría de decoraciones es alusivo al logotipo del equipo.

El segundo cuarto representa un gimnasio, donde los jugadores podrían entrenar en sus tiempos libres

En el desarrollo del proyecto se cumplieron los siguientes objetivos:

Modelado de la habitación

Modelado de 7 elementos que a continuación se listan (estos modelos, en su mayoría contienen texturas) y sus respectivas funciones creadas:

Estructura	void clubhouse();	con sus texturas correspondientes
Silla	void silla();	contiene texturas en el asiento, respaldo y tubería
Mesa de ping pong	void pingpong();	contiene textura en la red
Ropero	void guardarropa();	texturizado a madera
Maquina de gym	void m_gym();	la es de textura de colchoneta
pantalla empotrada	void pantalla();	la pantalla tiene la textura de un partido
maquina de dulces	void m_dulces();	la textura consta de de los dulces a vender
Pelota beis	void pelota();	textura para que parezca una pelota de beisbol
bote de basura	void bote();	Texturizado metálico
Maquina de gym 2	void m_gym2();	textura de colchoneta roja
maquina de agua	void m_agua();	la textura consta de la caratula de una maquina expendedora de agua con su garrafon
lugar de pesas	void m_pesas();	Lugar con pesas acomodadas

Código generado de los modelos

```
//función de la estructura
void clubhouse() {
    glPushMatrix();
    glTranslatef(0.0, 0, -5);
    fclubhouse.prisma(10, 15, 0.8, tpuerta.GLindex);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0, 3.0, -5.0);
    fclubhouse.prisma(15, 60, 0.5, tfachada.GLindex);
    glTranslatef(0.0, 0.0, -25.0);
    fclubhouse.prisma(15, 60, 0.5, tpared.GLindex);
    glTranslatef(30.0, 0.0, 12.5);
    fclubhouse.prisma(15, 0.5, 25, tpared.GLindex);
    glTranslatef(-60.0, 0.0, 0.0);
    fclubhouse.prisma(15, 0.5, 25, tpared.GLindex);
    glTranslatef(30.0, 7.3, 0.0);
    fclubhouse.prisma(0.5, 60, 25, ttecho.GLindex);
    glTranslatef(0.0, -14.6, 0.0);
    fclubhouse.prisma(0.5, 60, 25, tpiso.GLindex);
    glTranslatef(0.0, 7.5, 0.0);
    fclubhouse.prisma(14, 5, 3, NULL);
    glTranslatef(30.0, 0.0, -22.0);
    //gym
    fclubhouse.prisma(15, 1, 20, tpared2.GLindex);
    glTranslatef(-60.0, 0.0, 0.0);
    fclubhouse.prisma(15, 1, 20, tpared2.GLindex);
    glTranslatef(30.0, 0.0, -10.0);
    fclubhouse.prisma(15, 60, 0.5, tpared2.GLindex);
    glTranslatef(0.0, 7.3, 10.0);
    fclubhouse.prisma(0.5, 60, 20, ttecho2.GLindex);
    glTranslatef(0.0, -14.6, 0.0);
    fclubhouse.prisma(0.5, 60, 20, tpiso2.GLindex);
    //pasillo
    glTranslatef(-30.0, 7.3, 15.0); //puertaclub
    fclubhouse.prisma(15, 1, 4, tpgym.GLindex);
    glTranslatef(-7.0, 0.0, -5.5);
    fclubhouse.prisma(15, 0.8, 15, tpared2.GLindex);
    glTranslatef(7.0, 0.0, -5); //puertagym
    fclubhouse.prisma(15, 1.2, 4, tpgym.GLindex);
    glTranslatef(-3.7, 0.0, -2); //
    fclubhouse.prisma(15, 6.5, 1, tpared2.GLindex);
    glTranslatef(0, 0.0, 15); //
    fclubhouse.prisma(15, 6.5, 1, tpared2.GLindex);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0, -5.0, -25.0);
    fclubhouse.prisma(0.2, 130, 80, text5.GLindex);

    glPopMatrix();
}

void silla() {
    glPushMatrix();
    glColor3f(1, 1, 1);
    glTranslatef(0, 2, 15);
    fsilla.prisma(0.3, 1, 1, tsilla.GLindex);
    glTranslatef(0, 0.5, 0.36);
    fsilla.prisma(1, 1, 0.3, tsilla.GLindex);
    glTranslatef(0, -1.2, -0.3);
    fsilla.cilindro(0.05, 0.7, 24, ttubo.GLindex);
    glTranslatef(-0.45, 1, -0.3);
    fsilla.prisma(0.5, 0.1, 0.14, ttubo.GLindex);
    glTranslatef(0.9, 0, 0.0);
}
```

```

        fsilla.prisma(0.5, 0.1, 0.1, ttubo.GLindex);
        glTranslatef(0.0, 0.2, 0.3);
        fsilla.prisma(0.1, 0.1, 0.5, ttubo.GLindex);
        glTranslatef(-0.9, 0.0, 0.0);
        fsilla.prisma(0.1, 0.1, 0.5, ttubo.GLindex);
        glPopMatrix();
    }
    void pingpong() {
        fpingpong.prisma(0.3, 10, 3, NULL);
        glColor3f(0, 0, 1);
        glTranslatef(0, 0.0, 0);
        fpingpong.prisma(0.3, 10, 3, NULL);
        glColor3f(1, 1, 1);
        glTranslatef(0, 0.12, 0);
        fpingpong.prisma(0.1, 10, 0.2, NULL);
        glColor3f(1, 0, 0);
        glTranslatef(-3.5, -1.7, 1);
        fpingpong.prisma(3, 0.3, 0.3, NULL);
        glTranslatef(7.5, 0, 0);
        fpingpong.prisma(3, 0.3, 0.3, NULL);
        glTranslatef(0, 0, -2.0);
        fpingpong.prisma(3, 0.3, 0.3, NULL);
        glTranslatef(-7, 0, 0.0);
        fpingpong.prisma(3, 0.3, 0.3, NULL);
        glColor3f(1, 1, 1);
        glTranslatef(3.3, 2.2, 1.0);
        glEnable(GL_ALPHA_TEST);
        fpingpong.prisma(1, 0.2, 3, tred.GLindex);
        glDisable(GL_ALPHA_TEST);
    }
    void guardaropa() {
        glColor3f(1, 1, 1);
        fropero.prisma(13, 10, 0.3, tmadera.GLindex);
        glTranslatef(0, 2.5, 1.3);
        fropero.prisma(0.5, 10, 2.5, tmadera.GLindex);
        glTranslatef(0, 1.8, 0);
        fropero.prisma(0.5, 10, 2.5, tmadera.GLindex);
        glTranslatef(0, 2.3, 0);
        fropero.prisma(0.3, 10, 2.5, tmadera.GLindex);
        glTranslatef(0, -10, 0);
        fropero.prisma(0.3, 10, 2.5, tmadera.GLindex);
        glTranslatef(-1, 8, 0);
        fropero.prisma(4, 0.4, 2.5, tmadera.GLindex);
        glTranslatef(3, 0, 0);
        fropero.prisma(4, 0.4, 2.5, tmadera.GLindex);
        glTranslatef(3, -4.5, 0);
        fropero.prisma(13, 0.4, 2.5, tmadera.GLindex);
        glTranslatef(-10, 0, 0);
        fropero.prisma(13, 0.4, 2.5, tmadera.GLindex);
    }
    void m_gym() {
        glColor3f(1, 1, 1);
        fmaqgym.prisma(0.3, 5, 2, tcolchoneta.GLindex);
        glTranslatef(-2.0, -0.6, 0.0);
        fmaqgym.prisma(1, 0.2, 0.2, NULL);
        glTranslatef(4.0, 0.0, 0.0);
        fmaqgym.prisma(1, 0.2, 0.2, NULL);
        glTranslatef(0.0, 0.7, 0.7);
        fmaqgym.cilindro(0.1, 2, 24, NULL);
        glTranslatef(0.0, 0.0, -1.4);
        fmaqgym.cilindro(0.1, 2, 24, NULL);
        glPushMatrix();
        glRotatef(90, 1, 0, 0);
        glTranslatef(-0.2, -2, -1.5);
        fmaqgym.cilindro(0.1, 5, 24, NULL);
        glPopMatrix();
        glColor3f(0, 0, 0);
        glTranslatef(0, 1.2, 0);
        fmaqgym.cilindro(0.8, 0.3, 24, NULL);
    }

```

```

        glTranslatef(0,2.8, 0);
        fmaqgym.cilindro(0.8, 0.3, 24, NULL);
        glPopMatrix();
    glPopMatrix();
}
void pantalla() {
    glColor3f(1, 1, 1);
    fpantalla.prisma(2, 5, 0.2, tpantalla.GLindex);
    glTranslatef(0.0, 0.0, -0.05);
    glColor3f(0.0, 0.0, 0.0);
    fpantalla.prisma(2.5, 6, 0.2, NULL);
    glTranslatef(0.0, 1, -0.9);
    fpantalla.prisma(0.3, 0.3, 2, NULL);
    glTranslatef(0.0, 0.9, -1);
    fpantalla.prisma(3.0, 0.3, 0.3, NULL);
}
void m_dulces() {
    fmaq_dulces.prisma(8, 3, 3, tdulces.GLindex);
    glTranslatef(0.0, 0.0, -1.5);
    fmaq_dulces.prisma(8, 3, 0.3, tpdulces.GLindex);
    glTranslatef(-1.5, 0.0, 1.5);
    fmaq_dulces.prisma(8, 0.3, 3.0, tpdulces.GLindex);
    glTranslatef(3.0, 0.0, 0.0);
    fmaq_dulces.prisma(8, 0.3, 3.0, tpdulces.GLindex);
    glTranslatef(-1.5, 3.8, 0.0);
    fmaq_dulces.prisma(0.3, 3, 3.0, tpdulces.GLindex);
}
void botebasura() {
    glColor3f(1,1,1);
    fbote.cilindro(0.5, 1.5, 24, tbote2.GLindex);
    glTranslatef(0.0, 1.4, 0);
    fbote.cilindro(0.6, 0.15, 24, tbote2.GLindex);
    glColor3f(0, 0, 0);
    glTranslatef(0.0, -0.05, 0);
    fbote.torus(0.6, 0.2, 24, 24);
}
void m_gym2() {
    glColor3f(1, 1, 1);
    fmaqgym2.prisma(0.3, 5, 2, tcolchonetar.GLindex);
    glTranslatef(-2.0, -0.6, 0.0);
    fmaqgym2.prisma(1, 0.2, 0.2, NULL);
    glTranslatef(4.0, 0.0, 0.0);
    fmaqgym2.prisma(1, 0.2, 0.2, NULL);
    glTranslatef(-6.5, 0.0, 3);
    fmaqgym2.cilindro(0.2, 10, 24, ttubo.GLindex);
    glTranslatef(8, 0.0, 0);
    fmaqgym2.cilindro(0.2, 10, 24, ttubo.GLindex);
    glTranslatef(0, 0.0, -6);
    fmaqgym2.cilindro(0.2, 10, 24, ttubo.GLindex);
    glTranslatef(-8, 0.0, 0);
    fmaqgym2.cilindro(0.2, 10, 24, ttubo.GLindex);
    glColor3f(1, 0, 0);
    glTranslatef(4, 10, 0);
    fmaqgym2.prisma(0.3, 8, 0.3, NULL);
    glTranslatef(0, -5, 0);
    fmaqgym2.prisma(0.3, 8, 0.3, NULL);
    glTranslatef(0, 0, 6);
    fmaqgym2.prisma(0.3, 8, 0.3, NULL);
    glTranslatef(0, 5, 0);
    fmaqgym2.prisma(0.3, 8, 0.3, NULL);
    glTranslatef(4.8, -7, 0);
    fmaqgym2.prisma(0.3, 1.5, 0.3, NULL);
    glTranslatef(0, 0, -6);
    fmaqgym2.prisma(0.3, 1.5, 0.3, NULL);
    glTranslatef(0, 7, 3);
    fmaqgym2.prisma(0.3, 1.5, 0.3, NULL);
    glColor3f(0.5, 0.5, 0.5);
    glTranslatef(-0.8, 0, 0);
    fmaqgym2.prisma(0.3, 0.3, 6, NULL);
}

```

[illegible]

```

glRotatef(-90, 1, 0, 0);
glTranslatef(0, -3, 0);
glRotatef(90, 1, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.2, 1.5, 24, NULL);
glTranslatef(0, 0, 0);
glColor3f(0, 0, 0);
glTranslatef(0, 0.2, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(0, 1.1, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(0, 0, -3);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);

```

```

fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(0, -1.1, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
fmaqpesas.cilindro(0.8, 0.15, 24, NULL);
glTranslatef(-2, 0, 0);
glRotatef(-90, 1, 0, 0);
glPopMatrix();
}

```

Archivo ejecutable:

Dentro de la raíz donde se encuentran los archivos e imágenes, se encuentra un archivo tipo .exe (ejecutable) el cual carga el proyecto sin tener que compilarlo, este archivo se llama: ProyFinalEjecutable.exe, al dar clic sobre el automáticamente se carga el proyecto animado

Animación de algunos elementos:

Pelota: se mueve de un lado a otro: creada por animación compleja, se debe oprimir la tecla "space" para iniciar la animación y una vez más si se quiere reiniciar la misma.

Control de cámara:

Para controlar la cámara se ocupan las teclas **ASWD** para controlar el acercamiento y alejamiento de los objetos, úselo para desplazarse en el modelo

Para controlar el ángulo de vista se usan las **teclas de flechas**, cada una responde para obtener un ángulo de vista mejor, úselo para ver diferentes ángulos de los modelos tridimensionales.

Por último, se puede usar la tecla [RePag](#) para obtener una vista de altura, [AvPag](#) para disminuir esa misma altura.

Estas vistas se pueden combinar para mayor comodidad.

Archivos diversos:

En la raíz del proyecto se pueden encontrar diversos archivos tales como: las cabeceras o archivos .h que contienen código necesario para correr el programa sin errores.

El archivo ProyFinal.cpp es el main del proyecto aquí viene todo el diseño del modelo.

Contiene archivos .tga que son las imágenes de las texturas del proyecto

Codigo para texturas

```
CTexture tpuerta;  
CTexture ttecho;  
CTexture tpared;  
CTexture tfachada;  
CTexture tpiso;  
CTexture tdulces;  
CTexture tpdulces;  
CTexture tpantalla;  
CTexture tred;  
CTexture tmadera;  
CTexture tsilla;  
CTexture ttubo;  
CTexture tpelota1;  
CTexture tbote2;  
CTexture tcolchoneta;  
//segunda parte  
CTexture tpared2;  
CTexture tpgym;  
CTexture ttecho2;  
CTexture tcolchonetar;  
CTexture tagua;  
CTexture tpagua;  
CTexture tpesas;  
CTexture taagua;  
CTexture ttapete;  
CTexture tpiso2;  
  
tpuerta.LoadTGA("puerta.tga");  
tpuerta.BuildGLTexture();  
tpuerta.ReleaseImage();  
  
ttecho.LoadTGA("techo.tga");  
ttecho.BuildGLTexture();  
ttecho.ReleaseImage();  
  
tpared.LoadTGA("metalp.tga");  
tpared.BuildGLTexture();  
tpared.ReleaseImage();  
  
tfachada.LoadTGA("fachada.tga");  
tfachada.BuildGLTexture();  
tfachada.ReleaseImage();  
  
tpiso.LoadTGA("piso1.tga");  
tpiso.BuildGLTexture();
```

```
tpiso.ReleaseImage();

tdulces.LoadTGA("dulces.tga");
tdulces.BuildGLTexture();
tdulces.ReleaseImage();

tpdulces.LoadTGA("pdulces.tga");
tpdulces.BuildGLTexture();
tpdulces.ReleaseImage();

tpantalla.LoadTGA("tv.tga");
tpantalla.BuildGLTexture();
tpantalla.ReleaseImage();

tred.LoadTGA("red.tga");
tred.BuildGLTexture();
tred.ReleaseImage();

tmadera.LoadTGA("madera.tga");
tmadera.BuildGLTexture();
tmadera.ReleaseImage();

tsilla.LoadTGA("silla.tga");
tsilla.BuildGLTexture();
tsilla.ReleaseImage();

ttubo.LoadTGA("tubo.tga");
ttubo.BuildGLTexture();
ttubo.ReleaseImage();

tpelota1.LoadTGA("pelota3.tga");
tpelota1.BuildGLTexture();
tpelota1.ReleaseImage();

tbote2.LoadTGA("bote.tga");
tbote2.BuildGLTexture();
tbote2.ReleaseImage();

tcolchoneta.LoadTGA("colchoneta.tga");
tcolchoneta.BuildGLTexture();
tcolchoneta.ReleaseImage();

tpared2.LoadTGA("pared2.tga");
tpared2.BuildGLTexture();
tpared2.ReleaseImage();

tpgym.LoadTGA("pgym.tga");
tpgym.BuildGLTexture();
tpgym.ReleaseImage();

ttecho2.LoadTGA("tgym.tga");
ttecho2.BuildGLTexture();
ttecho2.ReleaseImage();

tcolchonetar.LoadTGA("colchonetar.tga");
tcolchonetar.BuildGLTexture();
tcolchonetar.ReleaseImage();

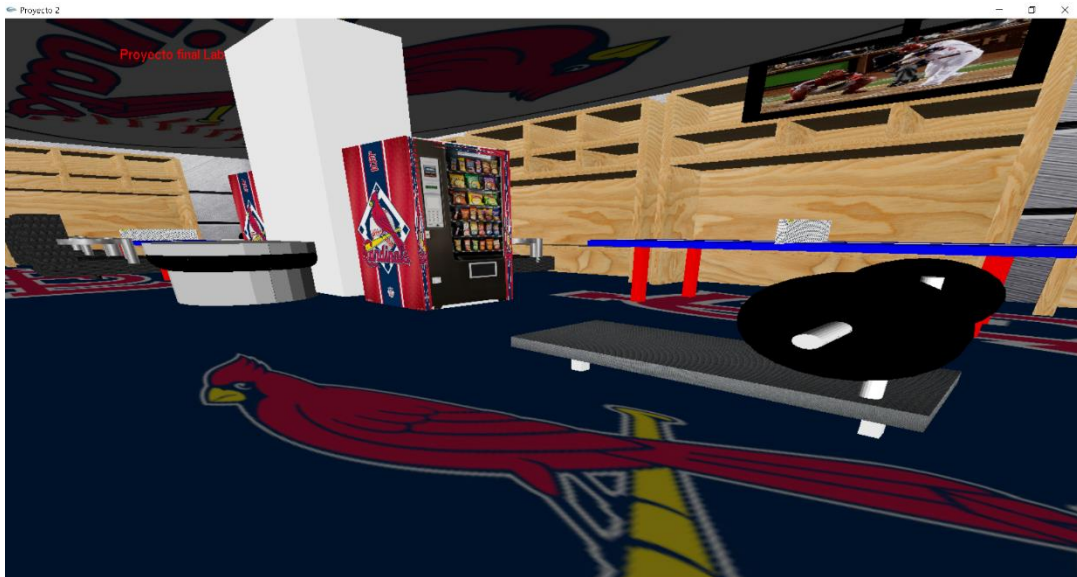
tpesas.LoadTGA("pesas.tga");
tpesas.BuildGLTexture();
tpesas.ReleaseImage();

tagua.LoadTGA("agua.tga");
tagua.BuildGLTexture();
tagua.ReleaseImage();

tpagua.LoadTGA("pagua.tga");
tpagua.BuildGLTexture();
tpagua.ReleaseImage();
```

```
taagua.LoadTGA("agua.tga");  
taagua.BuildGLTexture();  
taagua.ReleaseImage();  
  
ttapete.LoadTGA("tapete.tga");  
ttapete.BuildGLTexture();  
ttapete.ReleaseImage();  
  
tpiso2.LoadTGA("piso2.tga");  
tpiso2.BuildGLTexture();  
tpiso2.ReleaseImage();
```

Algunos resultados:



En contra de la imagen real:



La fachada del clubhouse



Los objetos modelados:

