



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

Materia: Laboratorio de computación Gráfica

“Documentación del proyecto final”

Profesor: Ing. Carlos Aldair Román Balbuena

Alumno: Carrillo Domínguez Carlos Daniel

Fecha de entrega: 6 de mayo del 2020

Índice

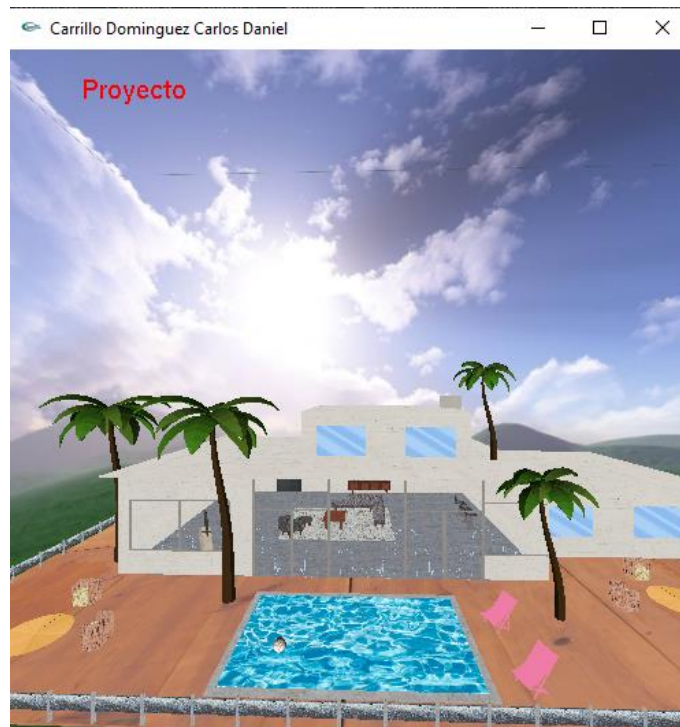
1. Sobre la aplicación	3
2. Manual técnico	5
3. Manual de Usuario y ejemplos gráficos.....	13

Sobre la aplicación

La aplicación fue desarrollada en el lenguaje de programación Visual Basic haciendo uso del entorno de programación **Microsoft Visual Studio**, el código de dicha aplicación se encuentra público en la plataforma Github en el siguiente repositorio: <https://github.com/DannCarrillo/ComputacionGrafica/>.

Esta aplicación fue desarrollada para presentar el Proyecto final del Laboratorio de computación gráfica donde se simula la casa y una habitación de la serie animada “Bojack Horseman”.





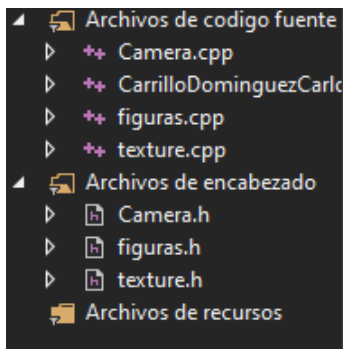
Guía táctica

La aplicación se encuentra programada en Visual Studio.

Descripción de elementos incluidos:

```
#include "texture.h"
#include "figuras.h"
#include "Camera.h"
#include "cmodel/CModel.h"
```

Se incluyo los “.h” donde vienen la realización de figuras, el manejo de cámara, y para cargar los modelos 3d en formato “.3DS”.



main del proyecto.

```
#define MAX_FRAMES 20
int i_max_steps = 90;
int i_curr_steps = 0;

typedef struct _frame
{
    //Variables para GUARDAR Key Frames
    float posX;    //Variable para PosicionX
    float posY;    //Variable para PosicionY
    float posZ;    //Variable para PosicionZ
    float incX;    //Variable para IncrementoX
    float incY;    //Variable para IncrementoY
    float incZ;    //Variable para IncrementoZ

    float giroBancoCen;
    float giroBancoCenInc;
    float giroBancoIzq;
    float giroBancoIzqInc;
    float giroBancoDer;
    float giroBancoDerInc;
}FRAME;
```

Se definio el número de Frames a guardar.

Creamos estructura donde declaramos variables para las posiciones, sus incrementos, y para la rotación.

```
FRAME KeyFrame[MAX_FRAMES];
int FrameIndex=0;
bool play=false;
int playIndex=0;
bool play2 = false;
```

Inicializamos variables que ocuparemos para las animaciones.

```
CCamera objCamera; //Create objet Camera

GLfloat g_lookupdown = 0.0f; // Look Position In The Z-Axis (NEW)

int font=(int)GLUT_BITMAP_HELVETICA_18;

//GLfloat Diffuse[]= { 1.0f, 1.0f, 1.0f, 1.0f }; // Diffuse Light Values
GLfloat Diffuse[]= { 0.5f, 0.5f, 0.5f, 1.0f }; // Diffuse Light Values
GLfloat Specular[] = { 1.0, 1.0, 1.0, 1.0 }; // Specular Light Values
GLfloat Position[]= { 0.0f, 7.0f, -5.0f, 0.0f }; // Light Position
GLfloat Position2[]= { 0.0f, 0.0f, -5.0f, 1.0f }; // Light Position

GLfloat m_diff1[] = { 0.0f, 0.2f, 1.0f, 1.0f }; // Diffuse Light Values
GLfloat m_spec1[] = { 0.0, 0.0, 0.0, 1.0 }; // Specular Light Values
GLfloat m_amb1[] = { 0.0, 0.0, 0.0, 1.0 }; // Ambient Light Values
GLfloat m_s1[] = {18};

GLfloat m_diff2[] = { 0.8f, 0.2f, 0.0f, 1.0f }; // Diffuse Light Values
GLfloat m_spec2[] = { 0.0, 0.0, 0.0, 1.0 }; // Specular Light Values
GLfloat m_amb2[] = { 0.0, 0.0, 0.0, 1.0 }; // Ambient Light Values
GLfloat m_s2[] = {22};
```

Creamos el objeto cámara.

Se creó las luces

Se inicializaron las variables a ocupar para las texturas.

```
CTexture text1; //Madera
CTexture text2; //SillasEXT
CTexture text3; //Alberca
CTexture text4; //Piso
CTexture text5; //barandal
CTexture text6; //Vidrio
CTexture textSM; //Marco de la silla exteriores
CTexture textAlberca;

//*****TEXTURAS CIELO*****
//***Texturas para el cielo***
CTexture textSkyboxUnica;
CTexture textSkyboxFrontal;
CTexture textSkyboxAtras;
CTexture textSkyboxIzquierda;
CTexture textSkyboxDerecha;
CTexture textSkyboxArriba;
CTexture textSkyboxAbajo;
```

```
/**Texturas casa fachada**
CTexture textpared;
CTexture textMarco;
CTexture textPisoCasa;
CTexture textGlassWasser;
CTexture textPuerta;
CTexture textventana;
CTexture textMesa;
CTexture textPata;
CTexture textWall;
CTexture textSilla;
CTexture textTecho;
CTexture textMarble;
CTexture tapete;
CTexture textMesaT;
CTexture textTele;
CTexture textTele2;
CTexture textCuadro1;
CTexture textburo;
CTexture textsillon;
CTexture textsillon2;
CTexture textcojin;
```

```
CTexture textsillon3;
CTexture textbanco;
CTexture textpelota;
CTexture texthumo;
```

```

CFiguras fig1;
CFiguras fig2;
CFiguras fig3;
CFiguras fig4; //Cilindro mesa
CFiguras fig5; //Casa01
CFiguras fig6;
CFiguras fig7; //Para crear Monito
CFiguras figMarco;
CFiguras figtapete;
CFiguras figtele;
CFiguras figtele2;
CFiguras figburo;
CFiguras figcuadro;
CFiguras figsillon;
CFiguras figBanco;
CFiguras figpelota;
CFiguras fighumo;

//Modelos 3D
CModel amaca;
CModel florero;
CModel palmera;

```

Se inicializo las figuras que ocuparemos para modelar la casa y los objetos. También se inicializo los modelos 3d que se ocuparon.

```

float movPX = 0.0;
float movPY = 0.0;
float movPZ = 0.0;
float rotP = 0.0;
bool g_fanimacion = false;
bool g_avanza = false;

void saveFrame ( void )
{
    printf("frameindex %d\n",FrameIndex);

    KeyFrame[FrameIndex].posX=posX;
    KeyFrame[FrameIndex].posY=posY;
    KeyFrame[FrameIndex].posZ=posZ;

    KeyFrame[FrameIndex].giroBancoIzq = giroBancoIzq;
    KeyFrame[FrameIndex].giroBancoCen = giroBancoCen;
    KeyFrame[FrameIndex].giroBancoDer = giroBancoDer;

    FrameIndex++;
}

```

Se inicializaron las variables que se ocuparan para animación.

Se crea la función “saveFrame” en la que se guardan los Frame e imprime en consola el número de frame guardado.


```
void resetElements( void )
{
    posX=KeyFrame[0].posX;
    posY=KeyFrame[0].posY;
    posZ=KeyFrame[0].posZ;

    giroBancoIzq = KeyFrame[0].giroBancoIzq;
    giroBancoCen = KeyFrame[0].giroBancoCen;
    giroBancoDer = KeyFrame[0].giroBancoDer;
}
```

Se crea función para resetear los frame a cero.

```
+void humo() { ... }

+void ciudad () { ... }

+void bara() { ... }

+void amacaa() { ... }

+void salaExt () { ... }

+void alberca () { ... }
```

-Función “humo”, crea un humo utilizando una imagen con transparencia. Utilizando un mapeo de la imagen y rotándola sobre “y”.

-Función “ciudad”, crea la base de la fachada, el piso. Usando: fig3.prisma2(text4.GLindex, text4.GLindex);

-Función “bara”, crea los barandales que rodean la fachada. Usando:

```
fig4.cilindro(0.1,1.5,20.0,text5.GLindex);
fig4.prisma2(text6.GLindex,text6.GLindex);
```

-Función “amacaa”, manda a llamar al modelo 3D de la amaca con: amaca.GLrender(NULL, _SHADED, 1);

-Función “salaExt”, crea la mesa y las sillas que están alrededor de la misma, cambiando su rotación y traslación.

```
fig4.cilindro(2.0, 0.01,180, text1.GLindex);
fig4.cilindro(1, 0.5, 40, text1.GLindex);
fig2.prisma(1.0,1.0,1.0, text2.GLindex);
fig2.prisma(1.5, 1.0, 0.3, textSM.GLindex);
fig2.prisma(1.5, 1.0, 0.3, textSM.GLindex);
```

-Función “alberca”, crea la alberca usando dos prismas diferentes, el primero es el que crea el movimiento.

```
fig3.prisma_anun(text3.GLindex, text3.GLindex);
fig3.prisma(9.5, 0.1,16.,text5.GLindex);
```



```

+void bancooIzq() { ... }

+void bancooCen() { ... }

+void bancooDer() { ... }

+void sillon3() { ... }

+void sillon2() { ... }

+void sillon() { ... }

+void mesaCentro() { ... }

+void buro() { ... }

+void pelota() { ... }

```

-Funciones “bancooIzq”, “bancooCen”, “bancooDer”, crean los bancos con movimiento rotacional sobre su mismo eje. Ocupando “conos”, “cilindros” y “planos”. Se coloca a principio de la función `glRotatef(giroBancoIzq, 0.0, 1.0, 0.0);` Para que el banco gire al incrementar la variable “giroBanco(Izq,Cen,Der)”.

-Función “sillon3”, crea el sillón redondo usando `figsillon.cilindro(2.0,2.5,70.0,textsillon2.GLindex);` `figburo.cono(4.75, 0.5, 30, textburo.GLindex);`

-Función “sillon2”, crea el sillón individual, usando “.prisma” y agregando diferentes texturas y rotaciones.

-Función “sillon”, crea el sillón grande, usando los principios de “sillon2” pero a mayor escala.

-Función “mesaCentro”, crea la mesa de centro usando prismas para la tabla, y cambiando su tamaño para las patas.

-Función “buro”, crea el buro usando prisma para la base y prismas mas pequeños para los detalles, conos para las patas.

-Función, “pelota”, crea un pelota usando `figpelota.esfera(0.4,20,20,textpelota.GLindex);`

```

+void mesaTele(void) { ... }

+void casa3() { ... }

+void casa2() { ... }

+void casa( void ) { ... }

+void cuadro(void) { ... }

+void tele (void) { ... }

```

-Función “mesaTele”, crea el mueble en donde se coloca la televisión, usando prismas para las bases y cambiando las escalas para los detalles.

-Función “Casa3” y “Casa2”, crea la parte de arriba de la casa y la del costado, usando primas planos y unos más pequeños para las ventanas.

-Función “casa”, crea la casa principal donde se colocando los muebles del cuarto. `fig1.skybox2(60.0, 85.0, 85.0, textPata.GLindex);` y prismas con transparencias para las ventanas.

-Función “cuadro”, crea un prisma con textura de una imagen para un cuadro.

-Función “tele”, crea una tele usando, cono como base, y cilindro, también primas para la parte superior.

```

void InitGL ( GLvoid ) { ... }

void pintaTexto(float x, float y, float z, void *font,char *string) { ... }

```

En la función “InitGL”, inicializamos las luces, las texturas a ocupar, ejemplo

```

text1.LoadTGA("casatexturas/mesa1.tga");
text1.BuildGLTexture();
text1.ReleaseImage();

```

y también los modelos 3D: palmera._3dsLoad("Palm_Tree.3ds");

Colocamos los frame a guardar para el movimiento del humo.

En la función “pintaTexto”, crea el texto a ver en la pantalla.

```

void display ( void ) // Creamos la funcion donde se dibuja
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glLoadIdentity();
    glPushMatrix();

    glRotatef(g_lookupdown,1.0f,0,0);
}

```

En la función “display”, nuestra función principal, mandamos a llamar a las funciones creadas, aplicando glPushMatrix(); y glPopMatrix(); para ir dándole forma al ambiente que se simula. Ejemplo de dibujo de cielo:

```

//****CIELO****
glPushMatrix();
glDisable(GL_LIGHTING);
figSkybox.skyboxH(140.0, 140.0, 140.0, textSkyboxFrontal.GLindex,
textSkyboxAtras.GLindex, textSkyboxIzquierda.GLindex, textSkyboxDerecha.GLindex,
textSkyboxArriba.GLindex);
glEnable(GL_LIGHTING);
glPopMatrix();

```

```

void animacion() { ... }

void reshape ( int width , int height ) { ... }

void keyboard ( unsigned char key, int x, int y ) { ... }

void arrow_keys ( int a_keys, int x, int y ) { ... }

```

- Función “animación”, se crean las animaciones como:

```

fig3.text_izq-= 0.001;
fig3.text_der-= 0.001;
if(fig3.text_izq<-1)
fig3.text_izq=0;
if(fig3.text_der<0)
fig3.text_der=1;

```

que crea el movimiento continuo de la alberca.

- Función “keyboard”, crea los casos para poder interactuar con el ambiente, se explica los comandos a usar en el “Manual de usuario”.
- Función “arrow_keys”, crea el movimiento de la cámara.

```

int main ( int argc, char** argv ) // Main Function
{
    int submenu;
    glutInit          (&argc, argv); // Inicializamos OpenGL
    glutInitDisplayMode (GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // Display Mode (Colores RGB y alpha | Buffer Doble )
    glutInitWindowSize (500, 500); // Tamaño de la Ventana
    glutInitWindowPosition (0, 0); // Posicion de la Ventana
    glutCreateWindow    ("Carrillo Dominguez Carlos Daniel"); // Nombre de la Ventana
    //glutFullScreen    ( ); // Full Screen
    InitGL (); // Parametros iniciales de la aplicacion
    glutDisplayFunc     ( display ); //Indicamos a Glut función de dibujo
    glutReshapeFunc     ( reshape ); //Indicamos a Glut función en caso de cambio de tamaño
    glutKeyboardFunc    ( keyboard ); //Indicamos a Glut función de manejo de teclado
    glutSpecialFunc     ( arrow_keys ); //Otras
    glutIdleFunc        ( animacion );
    submenu = glutCreateMenu ( menuKeyFrame );
    glutAddMenuEntry    ("Guardar KeyFrame", 0);
    glutAddMenuEntry    ("Reproducir Animacion", 1);
    glutCreateMenu      ( menu );
    glutAddSubMenu      ("Animacion Monito", submenu);
    glutAttachMenu      (GLUT_RIGHT_BUTTON);
    glutMainLoop        ( ); //
    return 0;
}

```

Controles básicos y ejemplos gráficos

Manual de usuario

1. Movimiento por los planos:

a. Hacia adelante con la tecla:

“W” o “w”

b. Hacia atrás con la tecla:

“S” o “s”

c. Hacia la izquierda con la tecla:

“A” o “a”

d. Hacia la derecha con la tecla:

“D” o “d”

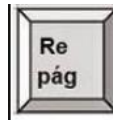
e. Hacia arriba con la tecla:

“RePág”



f. Hacia abajo con la tecla:

“AvPág”



2. Movimiento de la cámara:

a. Vista hacia arriba:

Usando la flecha del teclado hacia arriba



b. Vista hacia abajo:

Usando la flecha del teclado hacia arriba



c. Vista hacia la izquierda:

Usando la flecha del teclado hacia la izquierda



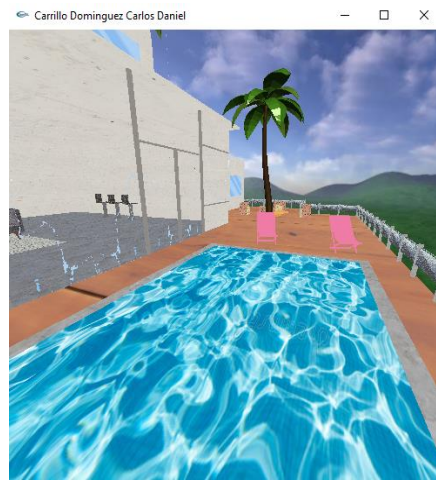
d. Vista hacia la derecha:

Usando la flecha del teclado hacia la derecha



3. Animaciones fijas

a. Alberca

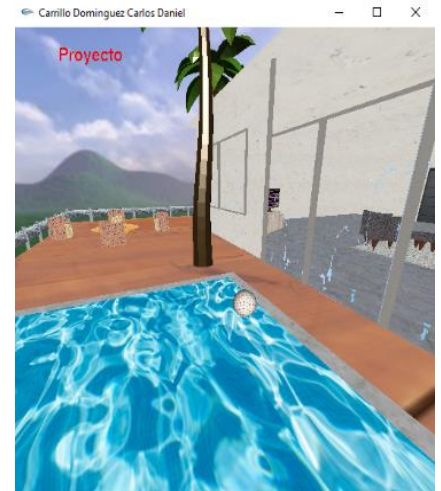
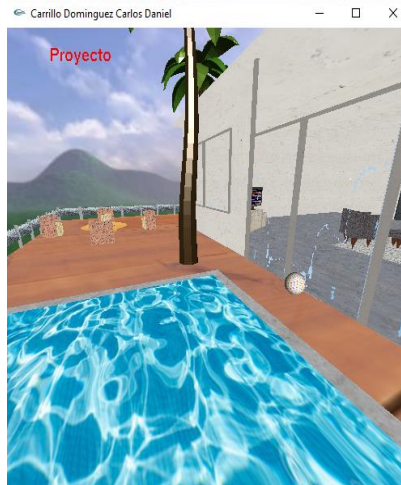
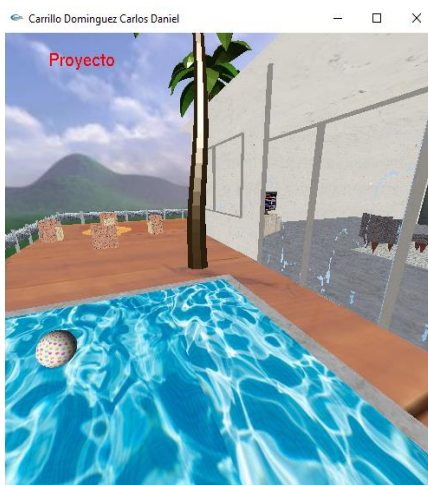


Simula ser una alberca con oleaje natural gracias al viento. La animación se mantiene constante.

4. Interacciones con los objetos:

a. Con pelota

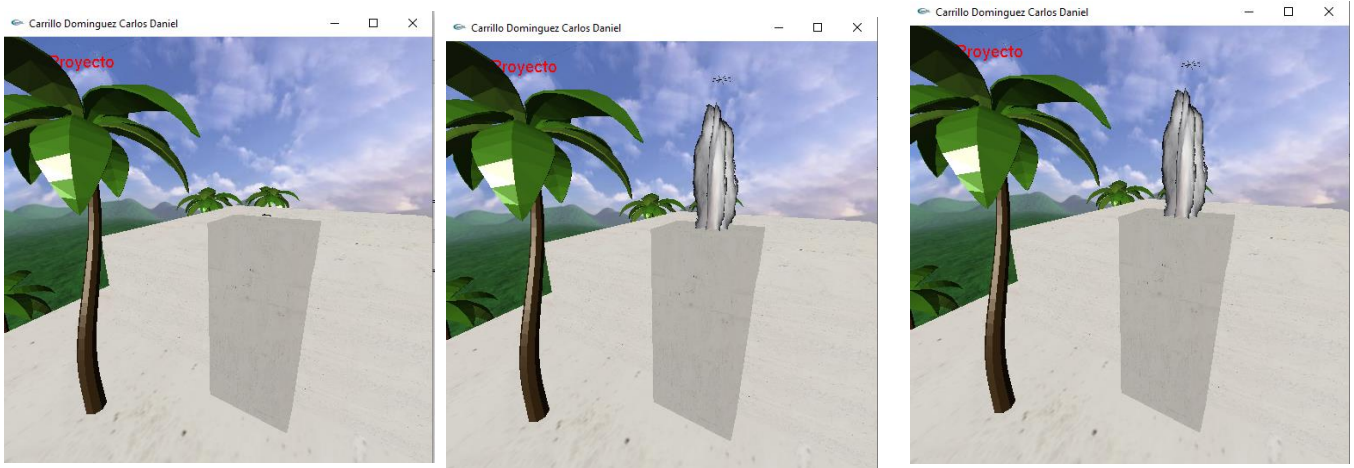
Se activa y desactiva usando la tecla: “I” o “i”



Simula ser una pelota que es llevada por la corriente de la alberca hacia la puerta de cristal y rebota nuevamente a la alberca, manteniéndose en un ciclo.

b. Con chimenea

Se activa y desactiva con la tecla: “L” o “I”



Simula ser una chimenea sacando humo. El humo sale de la chimenea elevándose en el cielo y desaparece para volver a salir.

c. Con los bancos.

1. El banco de la izquierda

Para rotar el banco hacia se izquierda se usa la tecla: “B” (Mayúscula)

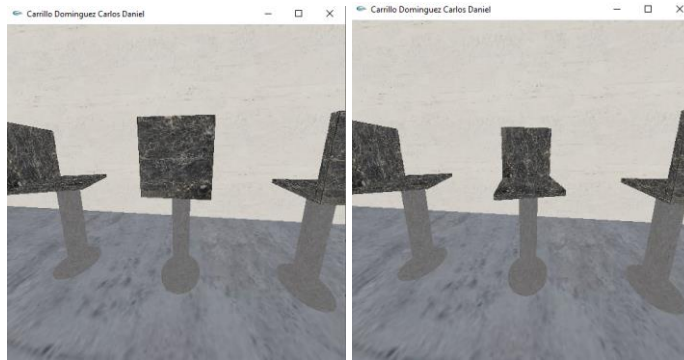
Para rotar el banco hacia se derecha se usa la tecla: “b” (Minúscula)



2. El banco del centro

Para rotar el banco hacia se izquierda se usa la tecla: “N” (Mayúscula)

Para rotar el banco hacia se derecha se usa la tecla: “n” (Minúscula)



3. El banco de la derecha

Para rotar el banco hacia se izquierda se usa la tecla: “M” (Mayúscula)

Para rotar el banco hacia se derecha se usa la tecla: “m” (Minúscula)



Se guardan los movimientos de los banco con la tecla : “K”

d. Para reproducir la animación de los bancos con los movimientos grabados, es con la tecla “O” u “o”.