

## Proyecto Final

### Manual de usuario

#### Objetivo

El alumno conocerá y aplicará las técnicas de modelado, implementación de modelos y animación de manera práctica demostrando su ingenio y creatividad.

#### Alcances

Se creará un entorno en 3D usando OpenGL mediante el modelado en Maya, y usando animaciones hechas mediante transformaciones lineales.

#### Cronograma de actividades realizadas

Actividades			
06/12/2020 - 10/12/2020	11/12/2020 - 16/11/2020	17/12/2020 - 30/12/2020	01/01/2021 - 22/01/2021
Plantear una propuesta	Presentar la propuesta	Empezar a modelar objetos en Maya	Terminar de modelar todos los elementos necesarios para el proyecto
Identificar los objetivos		Empezar a incorporar los modelos a OpenGL	Crear animaciones simples y complejas
Programación del calendario de actividades	Búsqueda de modelos de internet que pudieran ser útiles para el proyecto	Hacer pruebas de compilación del proyecto	Carga del proyecto a GitHub

## Código en Visual Studio

Carga de objetos del proyecto:

```
// Load models

Model ourModel ((char*)"Models/HouseModel/HouseModel.obj");
Model ourModel2 ((char*) "Models/Couch/Couch.obj");
Model ourModel3((char*) "Models/TvStand/tvStand.obj");
Model ourModel4((char*)"Models/Bed/Bed.obj");
Model ourModel5((char*)"Models/Basket/Basket.obj");
Model deskRoom((char*) "Models/Desk/desk.obj");
Model drawerDesk((char*)"Models/Desk/cajon.obj"); // Cajon
Model chairRoom((char*)"Models/Chair/chair.obj");
Model carModel((char*)"Models/Car2/car.obj");
Model bearModel((char*)"Models/Bear/bear.obj");

Model drawerModel((char*)"Models/Drawer/drawer.obj");
Model lampModel((char*)"Models/Lamp/lamp.obj");
Model closetModel((char*)"Models/Closet/closet.obj");
Model tableModel((char*)"Models/Table/table.obj");
Model couch2Model((char*)"Models/Couch/Couch2ndfloor.obj");
Model couch3Model((char*)"Models/Couch/Couch2ndfloor2.obj");
Model plantModel((char*)"Models/Plant/plant.obj");
Model doorModel((char*)"Models/Door/door.obj");
```

Casa:

```
// HOUSE MODEL LOADING AND DRAWING
glUniformMatrix4fv( glGetUniformLocation( shader.Program, "model" ), 1, GL_FALSE, glm::value_ptr( model ) );
ourModel.Draw(shader);

backup = model;

// DOOR MODEL LOADING AND DRAWING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
doorModel.Draw(shader);
```

Cama:

```
// BED LOADING AND DRAWING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
ourModel4.Draw(shader);
```

Silla:

```
// CHAIR LOADING AND DRAWING
model = glm::translate(model, glm::vec3(0.0f+ chair_y, 0.0f , 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
chairRoom.Draw(shader);
model = backup;
```

Escritorio:

```
// DESK LOADING
ourModel5.Draw(shader);
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
deskRoom.Draw(shader);

//DRAWER LOADING
model = glm::translate(model, glm::vec3(0.0f+ drawer_x, 0.0f, 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
drawerDesk.Draw(shader);
model = backup;
```

Lampara:

```
// LAMP LOADING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
lampModel.Draw(shader);
```

Comoda:

```
// DRAWER LOADING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
drawerModel.Draw(shader);
```

Oso:

```
// BEAR LOADING
model = glm::translate(model, glm::vec3(0.0f + bear_x, 0.0f, 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
bearModel.Draw(shader);
model = backup;
```

Closet:

```
// CLOSET LOADING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
closetModel.Draw(shader);
```

Cesta de ropa:

```
// BASKET LOADING
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
ourModel3.Draw(shader);
```

## Animaciones simples

```
if (keys[GLFW_KEY_Y])
{
    openDrawer(); // FUNCION PARA ABRIR EL CAJÓN DEL ESCRITORIO.
}

if (keys[GLFW_KEY_T])
{
    getChairBack(); // FUNCIÓN PARA DESLIZAR LA SILLA HACIA ATRAS
}
```

```
void openDrawer(){
    (drawer_x == 0) ? drawer_x = -0.6 : drawer_x = 0;
}

void getChairBack(){
    (chair_y == 0) ? chair_y = -0.6 : chair_y = 0;
}
```

```
if (keys[GLFW_KEY_K]) { // ANIMACIÓN PARA ABRIR LA PUERTA
    (door == true) ? door = false : door = true;
}
```

## Animaciones complejas

```
if (keys[GLFW_KEY_L]) {
    (carCircuit == false) ? carCircuit = true, cir1=true : carCircuit = false; // INICIAR LA ANIMACION DEL COCHE
}
```

```
if (keys[GLFW_KEY_J]) { // ANIMACION CAIDA DEL OSO
    (bearCir == false)? bearCir = true: bearCir=false, rotBear=0.0f, bear_y=0, cirB1=true; // SE ACTIVA LA ANIMACION DEL OSO
}
```

```

void moveCar() {
    if (carCircuit) {
        if (cir1) {
            (car_z < 10.3) ? car_z = car_z + 0.1 : cir1 = false;
            (cir1 == false) ? cir2 = true, rotCar=45.5 : cir1 = true;
        }

        if (cir2) {
            (car_x < 17.8) ? car_x = car_x + 0.1 : cir2 = false;
            (cir2 == false) ? cir3 = true, rotCar = 47.1 : cir2 = true;
        }

        if (cir3) {
            (car_z > -10.5) ? car_z = car_z - 0.1 : cir3 = false;
            (cir3 == false) ? cir4 = true, rotCar = 48.6 : cir3 = true;
        }

        if (cir4) {
            (car_x > -4.5) ? car_x = car_x - 0.1 : cir4 = false;
            (cir4 == false) ? cir5 = true, rotCar = 0 : cir4 = true;
        }

        if (cir5) {
            (car_z < 7.5) ? car_z = car_z + 0.1 : cir5 = false;
            (cir5 == false) ? cir6 = true, rotCar = 45.5 : cir5 = true;
        }

        if (cir6) {
            (car_x > 0.1) ? car_x = car_x + 0.1 : cir6 = false;
            carCircuit = false;
        }
    }
}

```

```

void moveBear() {
    if (bearCir) {
        if (cirB1) {
            (rotBear < 0.4) ? rotBear = rotBear + 0.05 : cirB1 = false, cirB2=true;
        }

        if (cirB2) {
            (bear_y > -2.1) ? bear_y = bear_y - 0.1 : cirB2 = false;
        }
    }
}

```