



UNIVERSIDAD NACIONAL AUTÓNOMA MÉXICO

FACULTAD DE INGENIERÍA

LABORATORIO DE COMPUTACION GRAFICA E INTERACCION
HUMANO – COMPUTADORA

CARLOS ALDAIR ROMAN BALBUENA

PROYECTO FINAL

MANUAL DE USUARIO

NAVARRO OSORIO ARMANDO

PERIODO: 2022-1

CIUDAD DE MEXICO, NOVIEMBRE DE 2021



OBJETIVO

El objetivo principal del desarrollo de este proyecto es que el alumno comprenda, analice, aplique y demuestre los conocimientos adquiridos durante este curso en donde se aplicaron diferentes herramientas de desarrollo de modelado.

DESCRIPCIÓN

Se selecciono un ambiente o un espacio ya sea real o ficticio el cual se va a recrear mediante los métodos de modelado y texturizados vistos durante el desarrollo del curso de esta materia, en este caso se usará el software de Maya, de Gimp y de OpenGL para el desarrollo de este.

El espacio seleccionado a desarrollar en este proyecto es de la serie animada de “Bob Esponja” el cual es el restaurante “Krusty Krab” el cual se va a construir y basarse en las imágenes para ser recreación en maya y posteriormente pasarlo a OpenGL



Los elementos que se recrearon en el espacio fueron los siguientes:

Muñeco de nieve



Bob Esponja



Patricio



Medusa



Almeja



Cabe mencionar que las imágenes de los elementos a representar y del mismo ambiente solo es una representación ya que en algunos no se pudo encontrar o bien crear el modelo bien como se observa en la imagen.

PLANEACIÓN

Para el desarrollo del proyecto se planteo una serie de pasos a seguir para poder obtener un buen resultado al finalizarlo, por lo cual se planteó un diagrama de Gantt con un desarrollo de 7 semana para poder distribuir de la mejor manera el proyecto.

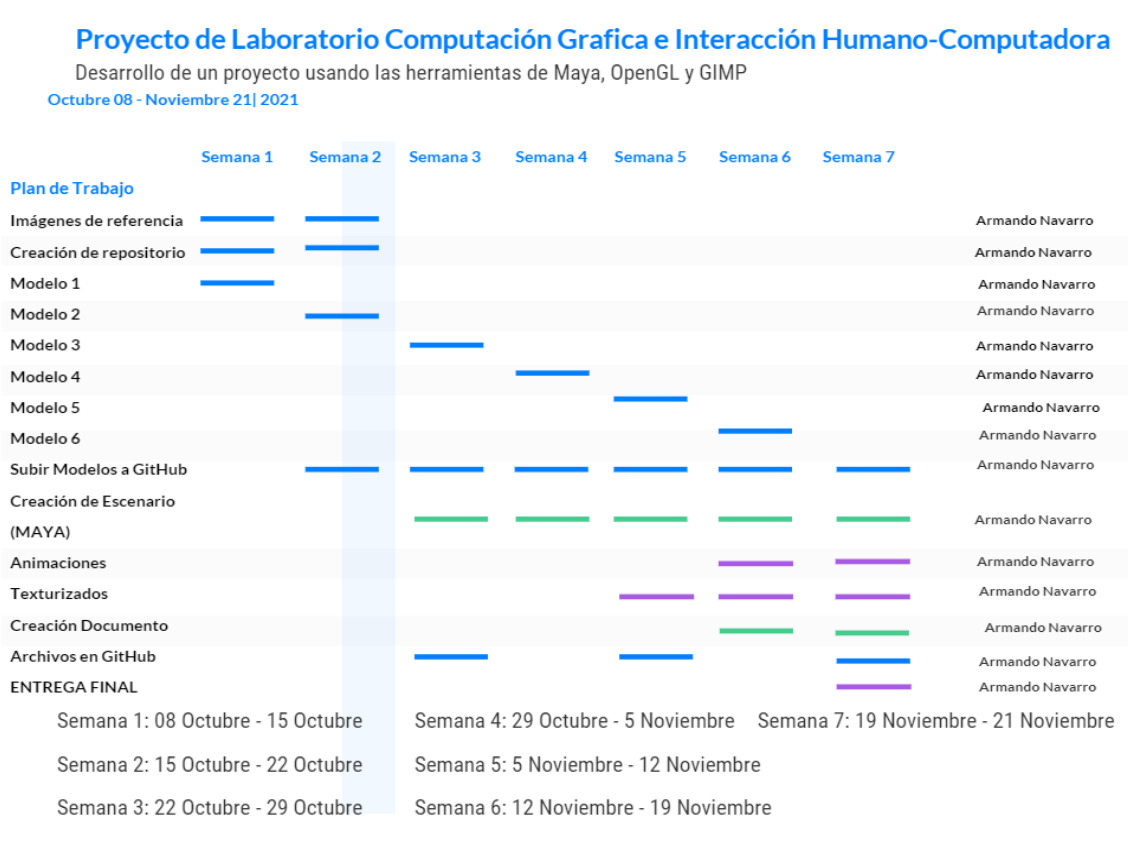
Durante las siete semanas se desarrollaron actividades diversas como es el caso de la texturización, modelado, importación a OpenGL, creación del algunos objetos y la creación del repositorio en donde se aloja nuestro proyecto, a continuación, agregamos la liga del repositorio.

https://github.com/ArmandoNavarroOsorio/Proyecto_Lab_Grafico

En él se podrá todos los documentos relacionados con el proyecto y de una mejor manera el diagrama en caso de que se requiera descargar individualmente.

Para el desarrollo del diagrama de Gantt se empleo la herramienta de www.venngage.com en donde nos proporciona plantillas o bien es su defecto la creación de un diagrama desde cero, se agrega la liga en donde se encuentra el plan de trabajo usado esta herramienta:

<https://infograph.venngage.com/infographics>



Como se observa en el diagrama cada una de las tareas tenia una fecha establecida para ser desarrolladas e implementadas

DESARROLLO

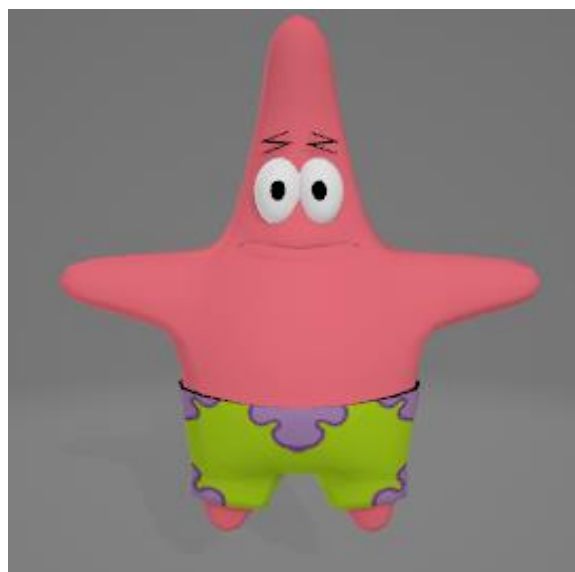
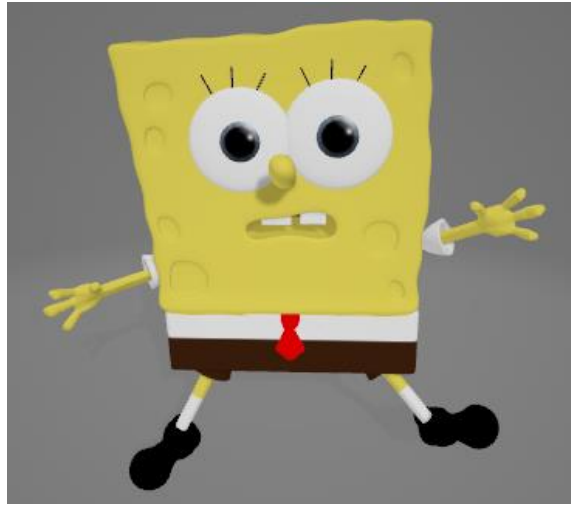
Para el desarrollo del proyecto se opto por el uso de la herramienta de GitGub para poder publicar y almacenar el mismo.



- Como se observa en la imagen se tiene una carpeta “Modelos” donde se fueron almacenando cada uno de los modelos que se fueron entregando en cada semana que el profesor lo solicitaba.
- En la carpeta de “PROYECTO” se encuentra como tal el proyecto llevado a OpenGL y al mismo tiempo se encuentra el ejecutable y todas las herramientas para poder llevar a cabo su ejecución si así se requiere.
- En el archivo de “PROPUESTA” se puede visualizar cual fue la propuesta para poder desarrollar el proyecto.
- En el archivo de “Diagrama” se visualiza como tal la planeación del proyecto en formato de pdf

Se muestran los modelos que se implantaron ya en formato .obj







Explicación de partes del código en donde se observa parte del proyecto.

Carga de los modelos.

```
// Load models
Model restaurante((char*)"Models/RESTAURANTE/RESTAURANTE.obj");
Model almeja((char*)"Models/RESTAURANTE/ALMEJA.obj");
Model nieve((char*)"Models/RESTAURANTE/NIEVE.obj");
Model medusa((char*)"Models/RESTAURANTE/MEDUSA.obj");
Model medusa2((char*)"Models/RESTAURANTE/MEDUSA2.obj");
```

Implementación en ambiente de OpenGL

```
glm::mat4 model(1);
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
restaurante.Draw(shader);
```

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-3.954f, 2.312f, 2.34f));
model = glm::rotate(model, glm::radians(rot), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
almeja.Draw(shader);
```

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-2.212f, 0.655f, 3.271f));
model = glm::rotate(model, glm::radians(rot), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
nieve.Draw(shader);
```



```

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-10.0f + movx, 5.0f, -3.0f + movz));
model = glm::rotate(model, glm::radians(rot2), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
medusa.Draw(shader);

```

```

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.0f + movx1, 3.0f, 3.0f + movz));
model = glm::rotate(model, glm::radians(rot3), glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
medusa2.Draw(shader);

```

-La animación consiste en que al iniciar la ejecución del proyecto automáticamente las medusas se mueven y se puede desactivar con la letra X.

-Las otras animaciones consiste en que se cierra y se abre la almeja al igual que se quita y se come el sobre el muñeco de nieve.

```

if (reinicio)
{
    if (cambio == false) {
        rot -= 0.05f;
        if (rot <= -80.0f) {
            cambio = true;
        }
    }
    if (cambio == true) {
        rot += 0.05f;
        if (rot >= 0.0f) {
            cambio = false;
        }
    }
}

```

```

if (anim)
{
    if (recorrido1)
    {
        if (movx > 25.0f)
        {
            recorrido1 = false;
            recorrido2 = true;
            rot2 = 30.0f;
        }
        else
        {
            movx += 0.03f;
        }
    }
    if (recorrido2)
    {
        if (movx < 5.0f)
        {
            recorrido2 = false;
            recorrido1 = true;
            rot2 = 0.0f;
        }
        else
        {
            movx -= 0.03f;
        }
    }
}

```

```

if (anim)
{
    if (recorrido3)
    {
        if (movx1 < -20.0f)
        {
            recorrido3 = false;
            recorrido4 = true;
            rot3 = -50.0f;
        }
        else
        {
            movx1 -= 0.03f;
        }
    }
    if (recorrido4)
    {
        if (movx1 > 15.0f)
        {
            recorrido4 = false;
            recorrido3 = true;
            rot3 = 30.0f;
        }
        else
        {
            movx1 += 0.03f;
        }
    }
}

```

```

if (keys[GLFW_KEY_Z]) {
    movimiento += 1;
    if (movimiento % 2 == 0)
    {
        reinicio = false;
    }
    else
    {
        reinicio = true;
    }
}

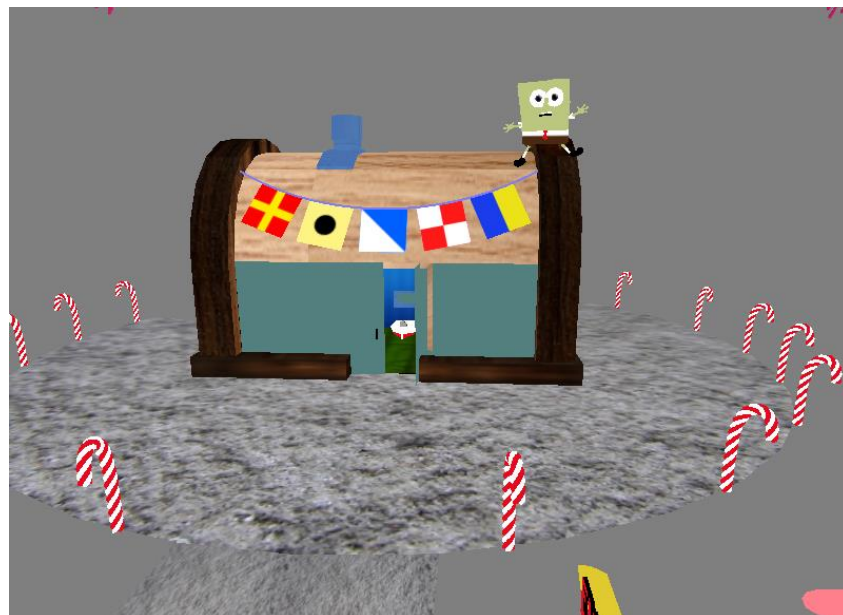
```

```

if (keys[GLFW_KEY_X])
{
    active = !active;
    anim = !anim;
}

```

Ejecución del código:





Algunos problemas presentados durante el desarrollo del proyecto fue la capacidad de la computadora a no tener un buen procesador una tarjeta de video los procesos del ambiente eran muy pesada al momento de cargar y modelar cada uno de ellos, por lo cual se recomendaría por lo menos tener un buen procesador una tarjeta de video aunque sea de 1 vram para que no consuma de más de recursos.