**FACULTAD DE INGENIERÍA**

INGENIERÍA DE SOFTWARE

**PROYECCIÓN VISUAL EN DEV-C++ CON OPENGL**

**Autor(es):**

Estudiante N1

Estudiante N2

Estudiante N3

Estudiante N4

**Curso:**

Diseño y Desarrollo de Juegos Interactivos I

**Docente:**

Ing MBA Vargas Huaman, Jhonatan Isaac

LIMA – PERÚ

2021-1

ÍNDICE

[**1** **Descripción** 3](#_Toc58123798)

[**2** **Lista de requerimientos** 3](#_Toc58123799)

[**3** **Lista de requerimientos ordenados por prioridad** 3](#_Toc58123800)

[3.1 **Requerimientos (*int main*)** 3](#_Toc58123801)

[**3.1.1** **Implementaciones** 4](#_Toc58123802)

[3.2 **Requerimientos (*void reshape*)** 4](#_Toc58123803)

[**3.2.1** **Implementaciones** 4](#_Toc58123804)

[3.3 **Requerimientos (*void init*)** 4](#_Toc58123805)

[**3.3.1** **Implementaciones** 4](#_Toc58123806)

[3.4 **Requerimientos (*void display*)** 5](#_Toc58123807)

[**3.4.1** **Implementaciones** 5](#_Toc58123808)

[3.5 **Requerimientos (*void idle*)** 5](#_Toc58123809)

[**3.5.1** **Implementaciones** 5](#_Toc58123810)

[3.6 **Requerimientos (*void keyboard*)** 5](#_Toc58123811)

[**3.6.1** **Implementaciones** 5](#_Toc58123812)

[3.7 **Requerimientos (*void keyboards*)** 5](#_Toc58123813)

[**3.7.1** **Implementaciones** 5](#_Toc58123814)

[3.8 **Requerimientos (*int CargarTGA*)** 6](#_Toc58123815)

[**3.8.1** **Implementaciones** 6](#_Toc58123816)

[**4** **Bibliografía** 6](#_Toc58123817)

# **Descripción**

El presente proyecto trata de explicar ….

# **Lista de requerimientos**

**Requerimientos FUNCIONALES:**

La IDE **Dev-C++,** permitirá compilar el programa con el fin de lograr visualizar el funcionamiento de este, a partir de los siguientes detalles:

* **int main ();** para la creación del espacio de trabajo y llamado de funciones.
* **void reshape ();** para la redimensión de la pantalla.
* **void init ();** para la configuración de las luces y validación de carga del archivo tga.
* **void display ();** para la implementación de materiales, transformaciones y animaciones, para la realización del proyecto.
* **void idle ();** para la ejecución de la animación del display.
* **void keyboard ();** Asignación de “keys” para la interacción con el programa.
* **void keyboards ();** Asignación de “keys” usando glutSpecialFunc ().
* **int cargarTGA ();** Para el cargado de archivo tga, para el fondo de pantalla del programa.

**Requerimientos NO FUNCIONALES:**

1. Para la ejecución del programa, se debe contar con la IDE **Dev-C++** y los siguientes requerimientos:

* Microsoft Windows 95, 98, NT 4, 2000, XP
* 32 MB o 64 MB RAM
* 100/400 Mhz Intel compatible CPU
* 200 MB de espacio libre de disco

2. Para la utilización de **OpenGL**

* Cualquier CPU (se recomienda Intel i5/i7/Xeon).
* Cualquier GPU compatible con OpenGL 3.2. (tarjetas gráficas integradas Intel HD 4000 o superior).

3. Para los detalles visuales del **Programa**

* Se asignarán colores, materiales y características de la luz, para luego aplicarlos a los sólidos.
* Se asignará el manejo de teclas para la interacción del usuario con el Programa.

# **Lista de requerimientos ordenados por prioridad**

## **Requerimientos (*int main*)**

Para la realización del “int main”, se requirió de:

1. Inicializar GLUT

2. Establecer el ancho y alto inicial de la ventana

3. Establecer la posición de la pantalla

4. Crear una ventana con el título “Examen Final”

5. Llamar al init ();

6. Registrar lo que devuelve las funciones “display, idle, reshape, keyboard y keyboards”.

7. Ingresar el ciclo de procesamiento de eventos. (glutMainLoop ()).

### **Implementaciones**

## **Requerimientos (*void reshape*)**

Para la realización del “void reshape”, se requirió de:

1. El ViewPort, sirve para proyectar una parte específica de la ventana.

2. Se utilizarán glMatrixMode (GL\_PROJECTION); y glMatrixMode (GL\_MODELVIEW); para la utilización de matrices en el campo proyección y modelo del gráfico.

3. Se usará glLoadIdentity (); para que se borren todas las modificaciones hechas antes de esta misma.

4. Finalmente se implementa el glOrtho, para definir la perspectiva de la cámara hacia el entorno de trabajo.

### **Implementaciones**

## **Requerimientos (*void init*)**

Para la realización del “void init”, se requirió de:

1. Inicializar la posición de la luz para los sólidos.

2. Habilitar y elegir el tipo de iluminación.

3. Definir la vista frontal y mostrar como sólido.

4. Activar el manejo de sombras y profundidad.

5. Habilitar las texturas 2D.

6. Validar el cargado del archivo TGA.

### **Implementaciones**

## **Requerimientos (*void display*)**

Para la realización del “void display”, se requirió de:

1. Inicialización de los materiales.

2. Limpiamos la memoria buffer de color y profundidad.

3. Creamos la matríz.

4. Llamamos al fondo de pantalla, habilitando las texturas 2D.

5. Habilitamos los glPushMatrix (); y glPopMatrix (); para la realización de las animaciones.

6. Definición de materiales.

7. Habilitamos el glutSwapBuffer, para hacer el intercambio de buffers.

### **Implementaciones**

## **Requerimientos (*void idle*)**

Para la realización del “void idle”, se requirió de:

1. Se hará llamado a la función display para realizar el proceso de animación continua cuando no se reciben eventos del sistema de ventanas

### **Implementaciones**

## **Requerimientos (*void keyboard*)**

Para la realización del “void keyboard”, se requirió de:

1. Creación de la función Keyboard, con parámetros de key (unsigned char), debido a que solo aceptará parámetros de código ASCCI los cuales son únicamente positivos, x e y.

2. Establecer un Switch de key, para el registro de ingreso por teclado de instrucciones posteriores.

3. Se establecer los 4 casos principales:

### **Implementaciones**

## **Requerimientos (*void keyboards*)**

Para la realización del “void keyboards”, se requirió de:

1. Creación de la función Keyboards, con parámetros de key(int), debido a que solo aceptará parámetros de código ASCCI los cuales son únicamente positivos, x(int) e y(int).

2. Establecer un switch de key, para el registro de ingreso por teclado de instrucciones posteriores.

3. Se establecer los 4 casos principales:

### **Implementaciones**

## **Requerimientos (*int CargarTGA*)**

Para la realización del “int CargarTGA”, se requirió de:

1. Establecer una estructura “textura”, donde se establece los valores de dibujo, bit por pixel, el largo, ancho y la ID.

2. Definir primero la cabecera TGA, estableciendo los primeros 12 elementos, para verificar si la imagen cargada está en formato TGA.

3. Abrir el archivo en binario y establecer un estándar de verificación de cabecera.

4. Establecemos un puntero imagen que selecciona los parámetros correspondientes.

5. Validar los bits por píxel.

6. Calcular el tamaño de la img.

7. Validar la asignación de memoria.

### **Implementaciones**

# **Bibliografía**

Kilgard, M. (1996). *Index*. Recuperado el 05 de 12 de 2020, de opengl: https://www.opengl.org/resources/libraries/glut/spec3/node113.html