

Índice

1. Enunciado	2
2. Consideraciones de Diseño	4
2.1. Estructura de la aplicación	4
3. Compilación	6
4. Ejecución	6
5. Controles de teclado	7

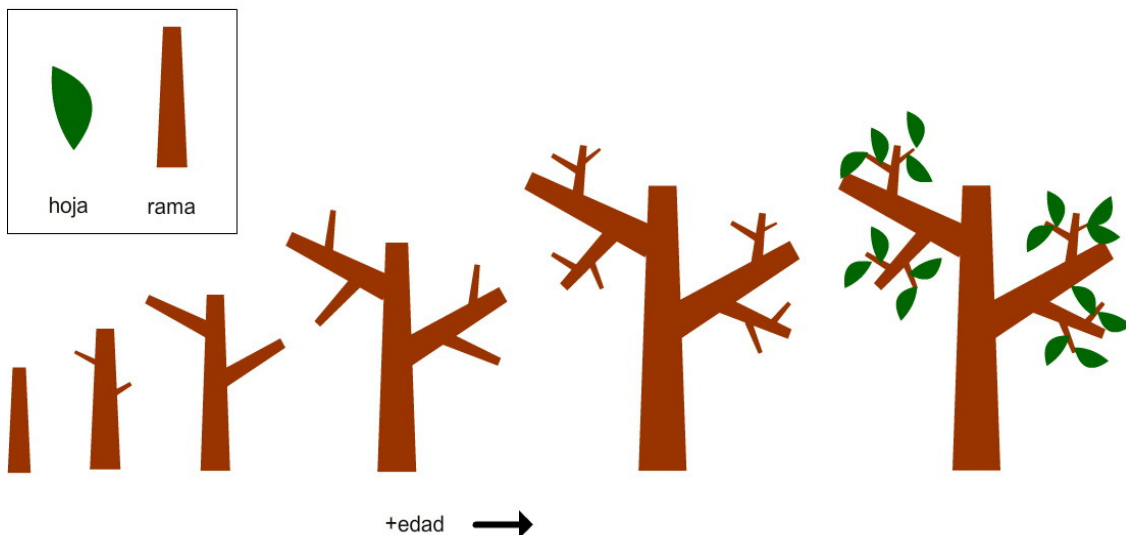
1. Enunciado

Trabajo Práctico N°1 - Transformaciones 3D en OpenGL

Objetivo

Desarrollar una aplicación OpenGL que modele un árbol en 3D con ramas y hojas basadas en las primitivas de ramas y hojas que serán definidas en base a triángulos.

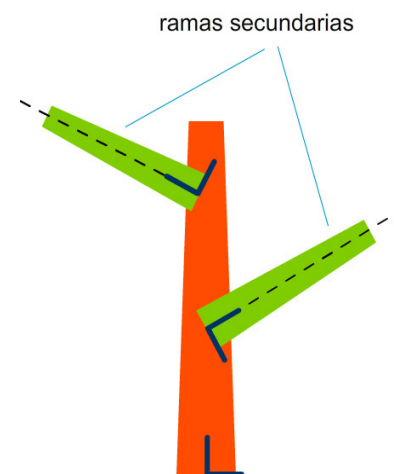
Se deberá construir un modelo jerárquico en donde un parámetro global “edad” controle el grado de desarrollo de la estructura, como se muestra en la figura



Cada rama deberá tener al menos 2 ramas secundarias, cuyo ángulo, posición y escala relativas a la rama padre podrán variar dentro de un rango de máximos y mínimos (ej: ángulo min. y max.), de este modo la estructura final tendrá una apariencia aleatoria.

En el último nivel las ramas se reemplazan por hojas. A medida que aumenta la edad la estructura tendrá más niveles de profundidad (más instancias de ramas). Cada rama deberá variar su escala a medida que crece.

*Si bien las figuras son bidimensionales, la estructura del árbol deberá ser tridimensional.



Animación

La edad deberá variar en función del tiempo para animar el crecimiento del árbol. La velocidad será regulable.

Cámara

Deberá mapearse el movimiento del mouse en X e Y de manera que el punto de vista “orbite” alrededor del árbol.

Controles de teclado

<u>Tecla</u>	<u>Acción</u>
R	Reiniciar la animación de crecimiento
P	Pausar/reanudar animación
Q	incrementar velocidad de crecimiento
A	decrementar velocidad de crecimiento

Entrega y corrección

La fecha de entrega es el 16 de Septiembre.

Además del informe que se detalla a continuación debe entregarse copia del ejecutable y/o código fuente ya sea en DVD o vía email. En este último caso indicar claramente en el nombre del archivo ZIP o RAR: tp nro., grupo y cuatrimestre.

La evaluación del TP será individual, por lo tanto todos los integrantes del grupo deberán estar presentes en la corrección.

2. Consideraciones de Diseño

El presente trabajo práctico se basa en la utilización de un algoritmo recursivo para dibujar y simular el crecimiento del árbol. Este algoritmo, dibuja en cada llamada una rama, o en su defecto una hoja, de un determinado nivel del árbol.

Para realizarlo, se tomaron en cuenta las siguientes consideraciones de diseño:

- Cada rama posee cuatro ramas hijas; o en su defecto cuatro hojas si se tratara del anteúltimo nivel.
- Por defecto, el árbol tiene ocho niveles. Para modificar la cantidad de niveles, se puede pasar por línea de comandos el parámetro con dicha cantidad. Se recomienda no exceder la cantidad de 10 niveles.
- La altura de cada nivel del árbol depende en forma cuadrática de dicho nivel. La misma se actualiza automáticamente con el paso del tiempo, de forma tal de producir la sensación de que el árbol va creciendo hasta alcanzar su edad tope. Se consideraron 10 intervalos de tiempo para lograr que una rama en particular alcance su altura total.
- El diámetro de cada rama depende linealmente de la altura de la misma.
- Para dibujar las hojas se utilizaron dos triángulos, empleando `GL_TRIANGLE_STRIP`, de forma tal de que los mismos conformen un rombo.
- Con el fin de dibujar el tronco y las ramas, se utilizaron triángulos para conformar un cono truncado, junto con sus tapas.
- El ángulo de una rama respecto de su hija varía en función del nivel y en los rangos: $[40; 40 + \text{NivelMáximo}]$ y $[30; 30 + \text{NivelMáximo}]$; correspondiendo el mayor ángulo a las ramas ubicadas a menor altura en la rama padre.
- Después de dibujar cada nivel, se realiza una rotación de 90 grados, de forma tal de lograr repartir las ramas en todo el espacio.

Asimismo, para el manejo de la cámara, se tomaron las siguientes consideraciones de diseño:

- Se puede hacer orbitar la cámara alrededor del árbol utilizando la tecla `z` y `Z` o presionando la tecla `c` para capturar el movimiento del mouse y rotar la cámara con el mismo.
- Se permite acercar la cámara hasta el tronco del árbol utilizando la tecla `+` y alejarse del mismo utilizando la tecla `-`.

2.1. Estructura de la aplicación

Para desarrollar la aplicación se utilizó el lenguaje C++, que permite la programación orientada a objetos con el encapsulamiento de clases. Dentro de las principales, se encuentra la clase `Mundo`; que representa el mundo del trabajo práctico en el cual se suscriben los eventos de OpenGL. Contiene los menús de los distintos paneles y la primitiva seleccionada y responde a las peticiones del usuario.

Se utilizó el patrón Command para el manejo de teclas, movimiento del mouse y el click del mouse en los elementos de los menues. Entre estos últimos se encuentran los comandos cambiar textura, cambiar color, cambiar forma, etc.// Existen shaders de vertices, que realizan principalmente las transformaciones de las deformaciones, shaders de iluminación (basando los cálculos en el modelo de Phong) y shaders de fragmentos que combina la iluminación con la textura y el material.

3. Compilación

Para compilar la aplicación bajo entorno Linux se provee un archivo de makefile. Para utilizar el mismo, debe tenerse instalada la herramienta cmake. Para instalarla, por ejemplo, en una distribución Ubuntu se deben realizar los siguientes pasos:

- 1) sudo apt-get install cmake
- 2) Ingresar la contraseña de root del usuario.
- 3) Aceptar la descarga e instalación de los paquetes necesarios.

Teniendo ya instalada la herramienta, posicionados en el directorio donde se encuentra el trabajo práctico es necesario crear un directorio build de la siguiente forma:

```
mkdir build  
cd build  
cmake ..  
make
```

Una vez realizados estos pasos, se dispondrá del ejecutable de nombre tp1.

4. Ejecución

Para ejecutar la aplicación bajo un entorno Linux se deben realizar los siguientes pasos:

- 1) Posicionarse en el directorio build creado anteriormente.
- 2) Ejecutar:
./tp1

5. Controles de teclado

La aplicación permite realizar las siguientes acciones utilizando las teclas detalladas a continuación:

Tecla	Acción
R	Reiniciar la animación de crecimiento
P	Pausar/reanudar la animación
Q	Incrementar la velocidad de crecimiento
A	Decrementar la velocidad de crecimiento
q	Salir de la aplicación
a	Muestra/esconde los ejes
g	Muestra/esconde la grilla
c	Captura/libera el movimiento del mouse para rotar la cámara
z	Rotar la cámara alrededor del árbol en sentido antihorario
Z	Rotar la cámara alrededor del árbol en sentido horario
+	Acercar la cámara al árbol
-	Alejar la cámara del árbol