

Índice

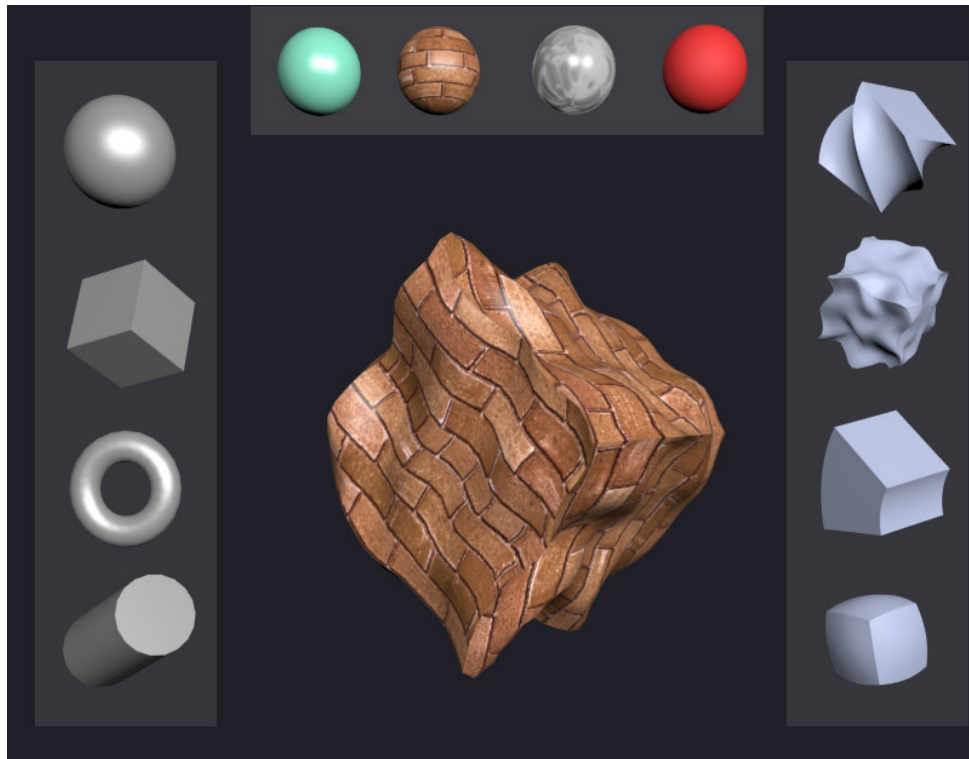
| | |
|--|----------|
| 1. Enunciado | 2 |
| 2. Consideraciones de Diseño | 5 |
| 2.1. Estructura de la aplicación | 5 |
| 3. Compilación | 6 |
| 4. Ejecución | 6 |
| 5. Controles de teclado | 7 |

1. Enunciado

Trabajo Práctico Nº2 – Programación en GPU

Objetivo

Desarrollar una aplicación OpenGL con las siguientes características, utilizando programas de vértices y de fragmentos programados en GLSL para correr en la GPU.



La pantalla se divide en 4 áreas:

Panel Izquierdo: permite seleccionar entre 4 tipos de primitivas (cubo, esfera, toroide y cilindro). La seleccionada se debe mostrar en el centro.

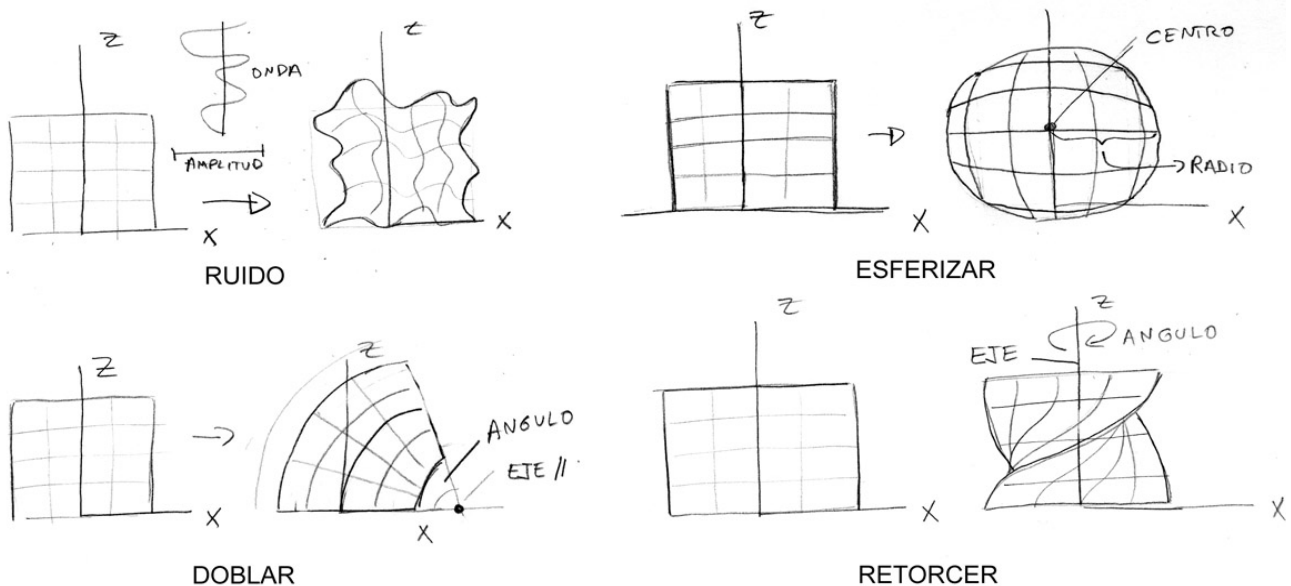
Panel Superior: permite seleccionar entre 4 tipos de materiales o shaders de vértices, que se aplican a la primitiva que esta activa en el centro

- 1) **Material sombreado brillante:** Modelo de Phong con colores fijos
- 2) **Material sombreado texturado:** Idem 1 pero con color difuso y ambiente definido por textura.
- 3) **Material reflectivo:** Idem 1 con la suma de una componente que refleja un entorno definido por un mapa de reflexión cúbico o esférico.

- 4) **Material sombreado semimate:** Idem 1 pero con coeficientes tales que simulen un material semimate.

Panel Derecho: permite seleccionar modificadores (aplican una deformación mediante un programa de vértice) que se aplicarán a la primitiva seleccionada en el panel izquierdo. Habrá 4 tipos:

- 1) **Retorcer:** aplica un retorcimiento alrededor del eje Z en el espacio de modelado definido por un ángulo.
- 2) **Ruido:** aplica una función de ruido definida en el espacio de modelado. Puede implementarse como la suma de varias componentes de ondas senoidales que varían según X,Y,Z. Los parámetros del modificador deberán ser longitud de onda, amplitud de onda y fase.
- 3) **Doblar:** aplicará una deformación de la primitiva alrededor de un eje paralelo al eje Y del espacio de modelado. Los parámetros deberán ser ángulo y distancia eje de deformación al origen de coordenadas.
- 4) **Esferizar:** dado un radio, un centro (x,y,z) y un factor que va entre 0 y 1, el modificador mapea las coordenadas de cada vértice a una esfera definida por los parámetros. El factor control el grado de interpolación entre la posición original y su posición en la esfera. Si $f=1$, todos los vértices están sobre la esfera, para $f=0$ la malla no se modifica.



Animación

Los parámetros de los modificadores deberán estar animados de modo que varíe la deformación en el tiempo. Por ejemplo en el caso del deformador "retorcer" el ángulo puede oscilar entre 0 y un ángulo alfa.

Controles

Arrastrando el mouse sobre la pantalla se deberá poder rotar la primitiva seleccionada alrededor de los ejes X,Y

Iluminación

Definir al menos 2 fuentes de luz puntuales que estén por delante de la escena (detrás de la cámara) para poder ver el efecto de reflexión especular. Las mismas se podrán apagar o prender con comandos de teclado.

Controles de teclado

| <u>Tecla</u> | <u>Acción</u> |
|--------------|----------------------------------|
| 1 | Apagar o prender fuente de luz 1 |
| 2 | Apagar o prender fuente de luz 2 |

Entrega y corrección

La fecha de entrega es el viernes 28 de Octubre

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.

Informe

Explicar la arquitectura de la aplicación, detallando las estructuras de datos, funciones y decisiones de diseño, que se hayan tomado. No es necesario incluir código fuente.

Carátula del Informe: en la primera hoja se deberán incluir los siguientes datos

Trabajo practico nro: X Cuatrimestre: N / YYYY Grupo nro.: Z

| Integrantes | Calificación | Fecha | Corrigió |
|--------------------------|--------------|-------|----------|
| Nombre, Apellido, Padrón | | | |
| Nombre, Apellido, Padrón | | | |

Datos al pie: colocar en **cada** hoja

Cuat: N/YYYY Trabajo Practico nro.: X, Grupo nro: Z, Integrantes: Padrón, Padrón

2. Consideraciones de Diseño

Para realizar el presente trabajo práctico se utilizó el lenguaje de programación GLSL dado que el mismo permite programar la GPU de la tarjeta de video para realizar cálculos complejos con una mejor performance y precisión. Para realizarlo, se tomaron en cuenta las siguientes consideraciones de diseño:

- Se debió utilizar la versión 1.2 de GLSL dado que es la soportada por las tarjetas gráficas de las notebooks de la mayoría de los integrantes del grupo.
- Se decidió agregar la funcionalidad de girar la primitiva seleccionada con respecto a los tres ejes (x,y,z) en ambos sentidos, utilizando teclas específicas del teclado (Ver **Controles del teclado** para mayor información).
- Se agregó el cono como uno de los tipos de primitivas disponibles para seleccionar.
- Para el mapeo de texturas en la superficie reflexiva se utiliza un mapa cúbico.
- Se supone que la reflexión obtenida de la superficie brillante es la máxima.
- La componente ambiental de las luces puntuales es una luz de color negro.
- Se utilizó la librería Il (devil) para cargar las texturas en formato .png, que luego son mapeadas en el mapa cúbico utilizado para el material reflectivo.

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 menús. 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 combinan la iluminación con la textura y el material de la primitiva seleccionada.

3. Compilación

Para compilar la aplicación bajo entorno Linux se provee un archivo 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 tp2.

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:
./tp2

5. Controles de teclado

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

| Tecla | Acción |
|-------|--|
| x | Rotar la primitiva seleccionada con respecto al eje x en el sentido positivo. |
| X | Rotar la primitiva seleccionada con respecto al eje x en el sentido negativo. |
| y | Rotar la primitiva seleccionada con respecto al eje y en el sentido positivo. |
| Y | Rotar la primitiva seleccionada con respecto al eje y en el sentido negativo. |
| z | Rotar la primitiva seleccionada con respecto al eje z en el sentido positivo. |
| Z | Rotar la primitiva seleccionada con respecto al eje z en el sentido negativo. |
| q | Salir de la aplicación. |
| r | Resetear las transformaciones y/o deformaciones aplicadas a la primitiva seleccionada. |
| c | Captura/libera el movimiento del mouse para rotar la primitiva seleccionada. |
| m | Muestra/esconde los menues. |
| 1 | Encender/apagar la fuente de luz 1. |
| 2 | Encender/apagar la fuente de luz 2. |