

Universidad nacional Autónoma de
México

Facultad de Ingeniería

Laboratorio de Computación Gráfica

Profesora: Ing. Elizabeth Fonseca Chávez



Entrega de práctica N°5:

Interactividad

Ortiz Servín Belem Valeria

Fecha de entrega: marzo/2009

OBJETIVOS:

1. Preparar una ventana Windows para su uso con OpenGL.
2. Realizar un programa iterativo mediante el uso del mouse.
3. Lograr interactividad con objetos en la escena.
4. Lograr interactividad con la cámara en la escena.

CUESTIONARIO PREVIO

1. Investigue y reporte que hace la función "glutIdleFunc()";

GLUT provee una función que controla la aplicación mientras esta en estado de "espera" mientras esta sin hacer nada aparentemente. Esta función es glutIdleFunc(void (*)(void)), GLUT llama a esta función en todo momento, logrando una animación.

2. Investigue y reporte que hace función "glutMotionFunc()";

La rutina glutMotionFunc(); es llamada por el sistema cuando el raton se mueve por la ventana de trabajo con uno o mas botones pulsados.

El desplazamiento del ratón se utiliza en la función movimiento() para definir, de acuerdo a la operación que se vaya a realizar, la magnitud del ángulo a rotar, el desplazamiento del objeto o la magnitud del escudo. Esta función la debe definir el programador y es llamada por glutMotionFunc(), otra de las rutinas dedicadas a procesar eventos de entrada. La rutina glutMotionFunc() es llamada por el sistema cuando el ratón se mueve por la ventana de trabajo con uno o mas botones pulsados . De forma análoga existe la función glutPassiveMotionFunc(), que es llamada cuando el ratón se mueve dentro de la ventana sin tener ningún botón pulsado. El modo de empleo de ambas funciones es el siguiente:

```
void glutMotionFunc(void (*func)(int x, int y));  
void glutPassiveMotionFunc(void (*func)(int x, int y));
```

Si bien estas funciones nos ayudan a establecer cuando se ah movido el ratón y la magnitud de su desplazamiento, debemos disponer de un medio para indicar al sistema la operación que deseamos realizar sobre el objeto en cuestión. Esto es posible realizarlo por medio del ratón, gracias al sencillo sistema de menú desplegables que es posible construir con Glut.



3. Investigue y reporte que hace la función `glutDisplayFunc()`

GLUT procesa tres tipos de eventos: los eventos de ventanas, menús y globales. Los primeros indican cuando es necesario redimensionar o dibujar una ventana, y cuando se produce un evento de entrada por ventana. Los eventos de menús se definen mediante la llamada `glutCreateMenu()`, y los globales procesan el transcurso del tiempo y el uso de los menús.

Así por ejemplo, `glutDisplayFunc` establece cuando se debe redibujar la ventana. El modo de utilizar esta rutina es el siguiente void `glutDisplayFunc(void (*func)(void))`; el argumento de `glutDisplayFunc` es un puntero a una función `Display` que debe definir el usuario. En nuestro caso es la función `redibuja()`, que a su vez llama `dibujaBox()`, la encargada de dibujar el cubo. Para cada ventana que se crea se debe establecer la respectiva `glutdisplayFunc()`, pues en caso contrario se produce error.

GLUT determina cuando la pantalla de devolución de llamada debe ser activada sobre la base de la ventana de estado que vuelva. Para visualizar el estado de una ventana puede ser llamado explícitamente o implícitamente (`glutPostRedisplay`) como resultado de la ventana de los daños reportados por el sistema de ventanas. El envío múltiple de ventanas vuelve en una ventana unidos por GLUT para reducir al mínimo el número de llamadas de la ventana de llamada.

GLUT determina cuando la pantalla de devolución de llamada debe ser activada sobre la base de la ventana de estado que vuelva. Cuando se establece una superposición de una ventana, pero no hay una superposición de pantalla de devolución de llamada registrada, la pantalla de devolución de llamada se utiliza para la superposición de ambos `redisplaying normal` y `avión` (es decir, que será llamado cuando el estado o volver vuelva estado de superposición se establece). En este caso, la capa de uso no es implícitamente cambiado a la entrada a la pantalla de llamada.

Cuando se crea una ventana, no existe devolución de llamada para mostrar la ventana. Es responsabilidad del programador para instalar una pantalla de devolución de llamada de la ventana antes de que la ventana se muestra. Una pantalla de devolución de llamada debe ser registrada por cualquier ventana que se muestra. Si una ventana se visualiza una pantalla de devolución de llamada sin estar registradas, se produce un error fatal. Pasando a `NULL` `glutDisplayFunc` es ilegal como de GLUT 3.0; no hay manera de "" de baja una pantalla de devolución de llamada (aunque otro de devolución de llamada de rutina siempre puede ser registrado).

4. Investigue y reporte que hace la función `glutIdleFunc()`.

GLUT provee una función que controla la aplicación mientras esta en estado de “espera” mientras esta sin hacer nada aparentemente. Esta función es `glutIdleFunc(void (*)(void))`, GLUT llama a esta función en todo momento, logrando una animación.

5. Investigue y reporte que hace la función `glutMainLoop()`;

Esta función se encarga de pasar el control del flujo del programa a la GLUT, de manera que cada vez que ocurra un “evento” sean llamadas las funciones definidas como callbacks hasta que la ventana se cierre.

Ejercicio 1

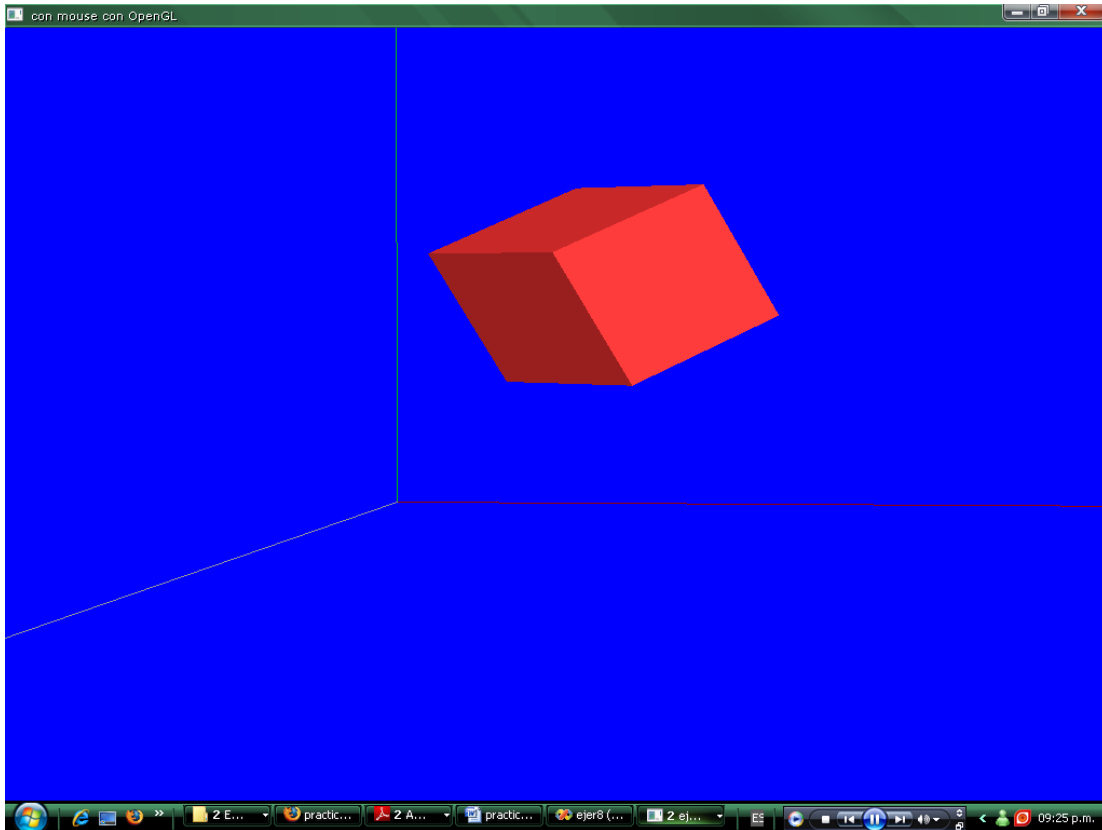
Para este experimento tenemos un programa que luego de iniciada la ventana Windows, pone a la PC a esperar un evento de teclado, ratón o joystick. La función `glutMainLoop()` hace que el computador espere la ocurrencia de un evento para entonces invocar a la función respectiva (callback). Mientras no ocurra un evento, GLUT invocará la función `glutIdleFunc()`, la cual a su vez invoca la generación de la escena en la función “`RenderScene()`”.

1. Vamos a crear un nuevo proyecto dentro de la carpeta “prácticas_g”. Así que presione en secuencia menú “**File->New Project**” o simplemente presione simultáneamente las teclas “**ctrl+shift+N**”. El nuevo proyecto se llamará “**ejer08**”.
2. Localice el archivo el archivo “**ejemplomouse.cpp**” y cópielo de su origen a la carpeta “**...prácticas_g\ejer08**”.
3. Presione la secuencia menú “**Project->Add Existing Item**” o simplemente presione la combinación de teclas “**Shift+Alt+A**”.
4. El programa consta de las siguientes partes:
 - a. Sección de variables globales.
 - b. Subrutina “**luces()**” para habilitar la ecuación de iluminación de OpenGL.
 - c. Subrutina “**EjesReferencia()**” para dibujar los ejes del sistema de referencia de OpenGL.habilitar
 - d. Subrutina “**cubo()**” Para definir una lista de primitivas que luego serán invocadas.
 - e. Subrutina “**idle()**” Es una subrutina que se invoca mientras GLUT no percibe ningún evento de teclado, ratón o joystick y que se encarga de redibujar la escena tan pronto como sea posible.
 - f. Subrutina “**ControlMovimientoRaton()**” Esta subrutina es un callback que se invoca en respuesta a un evento de ratón.
 - g. Subrutina “**RenderScene()**” Para invocar las subrutinas anteriores.

h. Subrutina "main()" Para crear una ventana Windows para uso con OpenGL.

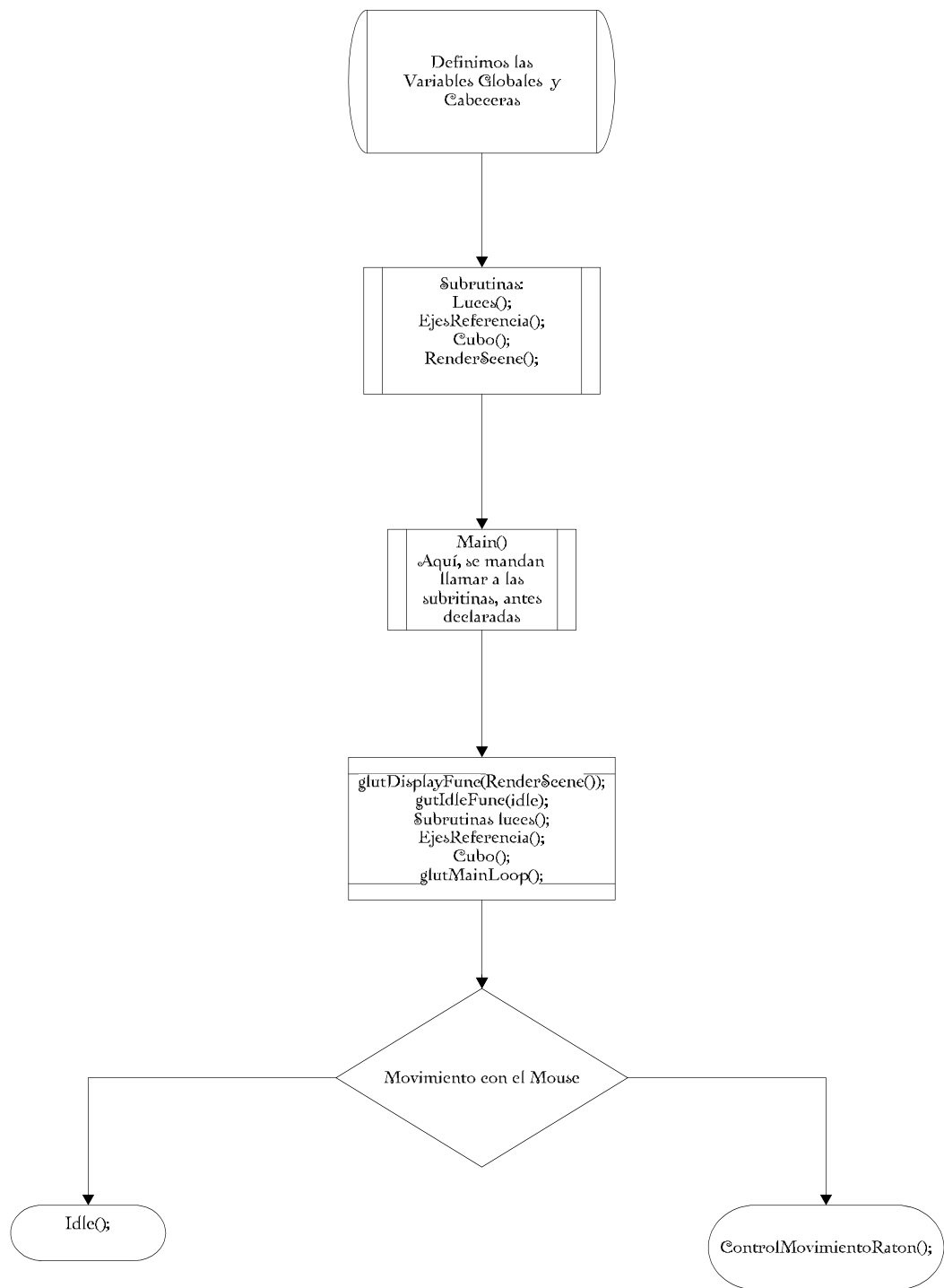
6. Construya y ejecute. 6. Reporte la gráfica obtenida.

Figura Obtenida.



Ejercicio 2

7. Elabore un diagrama de bloques que ilustre el funcionamiento del programa del ejercicio 1.



Ejercicio 3

8. Modifique el programa del ejercicio para que sea la cámara la que se mueva en respuesta a eventos del ratón.

