Dibujando Líneas

Funciones del programa:

- **void init(void)**
- **void pixel(int x,int y)**
- **void LineaBres(int xa, int ya, int xb, int yb)**
- **void lineSegment(void)**
- **int main(int argc, char** argv)**
- **void onMouse(int button, int state, int x, int y)**
- **void onMotion(int x, int y)**

Main

El programa comienza la ejecución con la función main(), la cual maneja todo lo referente a la correcta ejecución del programa.

- glutInit()

  Inicia lo necesario para poder trabajar con la librería de GLUT

- glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB)

  Se trabajara con un único buffer en la ventana, además en el formato de colores RGB

- glutInitWindowPosition(50, 100)

  Indica que la ventana que se utilizara para trabajar aparecerá en a partir de los pixeles 50,100 en distancia horizontal y vertical con respecto de la esquina superior izquierda del monitor.

- glutInitWindowSize(500, 500)

  La ventana que se abrirá será de 500 pixeles de ancho y 500 de largo.

- glutCreateWindow("Líneas con el Mouse")

  Se crea la ventana con lo ya anteriormente especificado teniendo como título “Líneas con el Mouse”.

- init()

  Especifica algunos otros atributos para trabajar con la ventana.

- glutDisplayFunc(lineSegment)

  Muestra todo aquello que sea dibujado con la función lineSegment.

- glutMainLoop()

Mantiene la ventana en un ciclo infinito para visualizar lo dibujado.

- glutMouseFunc(onMouse)

Llamará a la función onMouse cada vez que se detecte algún evento con el ratón, por ejemplo un click en el área de dibujado.

- glutMotionFunc(onMotion)

Llamará a la función onMotion cada vez que se este moviendo el mouse por la ventana.

Init (void)

Se encarga de Iniciar la ventana, pero a su vez dando otros atributos a la misma con las siguientes funciones

- glClearColor(0.0, 0.0, 0.0, 0.0)

Establece el fondo del área de trabajo de la ventana con el color (0,0,0), siendo estos en el formato RGB (Red, Green, Blue) y estando en su forma normalizada (0 Mínimo, 1 Máximo)

- glMatrixMode(GL_PROJECTION)

Proyección ortogonal en una zona rectangular bidimensional.

- gluOrtho2D(0.0, 500.0, 0.0, 500.0)

Sistema de referencia para las coordenadas que van de 0 a 500 (en horizontal -> x) y de 0 a 500 (en vertical -> y)

Pixel (int x, int y)

Función que pinta un pixel en las coordenadas x, y que le llegan por parámetro.

- glBegin(GL_POINTS);

Comienza el trazado en el modo GL_POINTS.

- glVertex2i(x, y);

Pinta un punto en la coordenada x, y del plano.

- glEnd();

Termina el trazado de puntos, aquí finaliza la función.

LineaBres (int xa, int ya, int xb, int yb)

Función que nos sirve para trazar una línea entre dos puntos, tomando como primer punto las coordenadas xa, xb y para segundo punto las coordenadas ya, yb.

El trazado de la línea mediante este algoritmo se hace calculando primero las distancias de los puntos en vertical (ya-yb) así como en horizontal (xa-xb). Después de calcular las distancias se pasa a verificar cómo será el incremento, si es positivo, negativo o nulo, tanto en x como en y.

Cuando ya se calcularon los incrementos/decrementos se verifica en que eje (x o y) se hacen más iteraciones y sobre ese se comienza a trazar puntos tomando en cuenta en qué momento se aumenta x y en qué momento y (determinado mediante el parámetro de decisión).

lineSegment (void)

Función encargada del dibujado de las líneas con el mouse, para ello se presenta el siguiente código:

```
glClear(GL_COLOR_BUFFER_BIT);

glColor3f(0.0, 1.0, 0.0); // color de la línea

glClear(GL_COLOR_BUFFER_BIT| GL_DEPTH_BUFFER_BIT);

glDrawPixels(500,500,GL_RGB,GL_UNSIGNED_BYTE,colorArray);

LineaBres(xini,yini,xfin,yfin);//dibuja una línea

glFlush();
```

Para comenzar esta función asigna el color de la línea y dibuja con **DrawPixels** lo que se tenga almacenado, esto se verá reflejado cuando se dibuje más de 1 línea. Así mismo nos aseguramos que una vez que se dibuje una línea, esta pase a ser parte del fondo para que cuando se dibujen más líneas no se sea afectando a las que ya están dibujadas.

Void onMouse(int button, int state, int x, int y)

Con esta función se logra ejecutar cuando se produce algún evento en la ventana. Tiene parámetros los cuales son los datos del mouse, **button** es el botón con el que se creó el evento, **state** es el estado que tiene el botón, si está presionado o fue liberado, **x** y **y** son las coordenadas del mouse.

El código utilizado es el siguiente:

```
if ( (button == GLUT_LEFT_BUTTON) & (state == GLUT_DOWN) ) {
      xini=x;
      yini=abs(500-y);
}
if ( (button == GLUT_LEFT_BUTTON) & (state == GLUT_UP) ) {
      LineaBres(xini,yini,xfin,yfin);
      glReadPixels(0,0,500,500,GL_RGB,GL_UNSIGNED_BYTE,colorArray);
}
```

Donde si el botón fue el izquierdo y está presionado cambiamos las coordenadas iniciales para después dibujar la línea. Las finales se actualizan en con **void onMotion.**

Cuando el botón es el izquierdo, pero este ya se ha soltado entonces dibujamos la línea “definitiva” con las posiciones iniciales que se obtuvieron al hacer el click y las finales que están siendo calculadas con **void onMotion.**

Void onMotion(int x, int y)

Con esta función estamos calculando las coordenadas del mouse, ya que se está ejecutando cada vez que se detecta algún movimiento del mouse sobre la ventana.

El código utilizado es:

```
      xfin=x;
      yfin=abs(500-y);
      glutPostRedisplay();
```

Donde estamos asignado las coordenadas del mouse como finales y actualizando la pantalla para estar borrando las “líneas temporales”. Con glutPostRedisplay().