DESARROLLO DE SISTEMAS DISTRIBUIDOS

Proyecto 2

Elaborado por: Ukranio Coronilla

El proyecto 2 se elabora en equipos, tal y como se encuentran organizados en el salón de clases. Este debe ser programado por todos los miembros del equipo sin excepción, lo cual implica que exista un alumno coordinador que va a repartir clases y o métodos para que sean programados por sus demás compañeros y elaborará los stubs necesarios (https://es.wikipedia.org/wiki/Stub). La revisión del proyecto será con un integrante del equipo el cual será escogido al azar y que no será el coordinador, quien en la evaluación, deberá tener la capacidad de programar las líneas de código que el profesor le modifique al proyecto. Dado que la calificación obtenida por el alumno es la que le corresponde a todo el equipo, es imprescindible un buen trabajo de equipo.

https://www.youtube.com/watch?v=IjB0j997euA

El proyecto consiste en programar en C++ para la plataforma LINUX y orientada a objetos, una simulación de n asteroides de distintos tamaños, desplazándose en distintas direcciones y a distintas velocidades en el plano 2D, tal y como se muestra en el siguiente video:

https://www.youtube.com/watch?v=yS7wH5_bUF8

En el caso que dos asteroides colisionen, ambos deberán romperse en pedazos más pequeños de corta duración pues todos los pedazos desaparecerán después de unos segundos.

Para los gráficos se debe utilizar la biblioteca simple para gráficos X11 GFX:

https://www3.nd.edu/~dthain/courses/cse20211/fall2013/gfx/

Si no puede compilar el programa `gfx.c` en Linux probablemente requiera instalar las librerías X11 con:

```
sudo apt-get install libx11-dev
```

El programa debe estar completamente en código C++ y orientado a objetos. Esto significa que debe crear las clases y métodos que considere necesarios así como un programa principal que los utilice. El programa solo recibirá como único parámetro en la línea de comandos el entero n que indica el número de asteroides que se van a mantener coexistiendo en la pantalla.

Para utilizar las librerías GFX que se encuentran en lenguaje C dentro de nuestro código en C++ es necesario agregar al inicio del archivo gfx.h las siguientes líneas:

```
#if defined(__cplusplus)
extern "C" {
#endif
```

Y al final del archivo gfx.h las siguientes líneas:

```
#if defined(__cplusplus)
}
#endif
```

Finalmente para ejemplificar la forma de compilar escriba el siguiente programa con nombre de archivo `animación.cpp`

```
#include "gfx.h"
#include <unistd.h>

using namespace std;

int main()
{
   int t;

   gfx_open(800, 600, "Ejemplo Micro Animacion GFX");
   gfx_color(0,200,100);

   for(t = 0; t < 100; t++){
      gfx_clear();
      gfx_line( t*1+80, t*2+40, t*2+40, t*3+80 );
      gfx_line(t*5+80, t*3+40, t*3+40, t*5+80);
      gfx_flush();
      usleep(41666);  //24 por segundo
   }
   return 0;
}
```

Ahora después de haber descargado los archivos `gfx.h` y `gfx.c` solo debe ejecutar en la línea de comandos los siguientes:

```
gcc gfx.c -c
g++ animacion.cpp -c
g++ gfx.o animacion.o -o animacion -lX11
```

Rubrica: Se va a penalizar la calificación con dos puntos por cada uno de los rubros que no se cumpla en el código:

- El asteroide no se ve “decente” y no parece como los que se muestran en el video.
- El asteroide no va girando mientras avanza.
- El asteroide no aparece aleatoriamente desde cualquier lugar posible fuera de la pantalla (por ejemplo que siempre aparezcan del mismo lado o de dos o tres lados en lugar de los cuatro lados y por todas las ubicaciones posibles).
- El asteroide no tiene distintos tamaños y/o no viaja a una velocidad inversamente proporcional a su tamaño.
- No se destruyen los asteroides al colisionar rompiéndose en pedazos más pequeños.

En caso de que el código no se encuentre orientado a objetos y los códigos de interfaz e implementación para cada clase, así como archivo Makefile, se penalizará además con 5 puntos.

Solo un miembro del equipo debe subir todos los archivos de código fuente en un archivo comprimido, a la plataforma MOODLE con todo el código necesario para crear el programa ejecutable.

El nombre del archivo debe ser el nombre del alumno separado con guion bajo, materia (DSD), grupo, numero de proyecto y extensión cpp. El no cumplir con estos requisitos provocará la disminución de la calificación.
Ejemplo de un nombre de archivo:
Juan_Perez_Molinar_DSD_4CM2_2.cpp

Advertencia: Evite copiar programas y que le sean copiados, cualquier acto de plagio se castigará para plagiario y plagiado con cero.