

DESARROLLO DE SISTEMAS DISTRIBUIDOS

Proyecto 2

Elaborado por: Ukranio Coronilla

PROGRAMADOR "NORMAL"

Se sugiere elaborar los ejercicios para proyecto al final de cada práctica con la intención de prepararse para programar el presente proyecto.

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

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

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

Si no compila 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. Por esta ocasión debe dejar las implementaciones, la interfaz y el código principal dentro de un mismo archivo. Debe utilizar las librerías GFX y no necesita modificarlas de modo alguno para que yo pueda compilar su programa.

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 su 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)
- El asteroide no tiene distintos tamaños y/o no viaja a una velocidad inversamente proporcional a su tamaño.
- No se mantienen en la pantalla aproximadamente n asteroides en todo momento.

En caso de que el código no se encuentre orientado a objetos se penalizará además con 5 puntos.

Solo debe subir un archivo con extensión `cpp` a la plataforma MOODLE conteniendo interfaz, implementación y principal.

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 plagiarlo y plagiado con cero.

PROGRAMADOR “PRINCIPIANTE”

Para los programadores que tienen deficiencias para programar, y con intención de que se regularicen, el proyecto 2 consiste en elaborar los siguientes ejercicios de programación en lenguaje C++.

Elaborar los siguientes 26 ejercicios de programación en C++ incluidos en el libro C++ for Everyone (todos ellos se encuentran al final de cada capítulo en el apartado marcado como Programming Exercises):

P1.2, P1.8, P2.19, P2.21, P3.3, P3.7, P3.23, P4.16, P4.20, P4.21, P4.23, P5.3, P5.6, P5.7, P5.14, P5.15, P6.8, P6.16, P6.20, P6.21, P7.8, P7.9, P7.10, P7.13, P9.25, P9.18.

También elaborar los siguientes 10 ejercicios de programación en C++ incluidos en el libro A First Book of C++:

Ejercicio 8 de la página 451, Ejercicio 3 de la página 462, Ejercicio 6 de la página 472, Ejercicio 2 de la página 481, Ejercicio 11 de la página 507, Ejercicio 4 de la página 518, Ejercicio 3 de la página 528, Ejercicio 4 de la página 533, Ejercicio 9 de la página 561, Ejercicio 10 de la página 562.

Para subir los proyectos a MOODLE: Los códigos de los programas elaborados deberán concatenarse en un solo archivo de texto plano, dejando una línea de asteriscos “*” entre cada ejercicio. En caso de que tenga que incluir el programa principal, la interfaz, la implementación, y archivo Makefile separar cada uno de estos archivos con una línea de guiones bajos “_”.

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

Ejemplo:

Antonio_Orlando_Rodriguez_DSD_4CM2_2.txt

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