

PROGRAMACION II
PRACTICO EVALUABLE
2019

Se desea obtener un sistema que permita simular un sistema solar. Para tal fin se debe diseñar un conjunto de clases que permitan representar las entidades en juego.

Punto 1:

Definir una clase `CuerpoCeleste` para representar los cuerpos que serán parte del Sistema Solar. La clase debe permitir observar el movimiento de los astros en el sistema. Por simplicidad se supone que el movimiento de los astros se realiza en un plano.

Los atributos que se deben considerar para la representación de un cuerpo celeste son:

- Nombre
- Masa
- Posición en la coordenada X
- Posición en la coordenada Y
- Velocidad según X
- Velocidad según Y

Estos atributos permiten caracterizar un cuerpo celeste y deberán ser considerados como datos miembros privados.

Deben también definirse los métodos públicos para proveer las siguientes funcionalidades:

- Un constructor por defecto y otro con parámetros que permita definir la masa del cuerpo celeste y su posición y velocidad inicial en la simulación.
- Los métodos de carga y visualización de los datos miembro, según sea necesario.
- Un método que calcule la fuerza sobre el cuerpo celeste según la ley de gravitación universal.

$$F = G \frac{m_1 m_2}{r^2}$$

donde

F es el módulo de la fuerza ejercida entre ambos cuerpos, y su dirección se encuentra en el eje que une ambos cuerpos.

G es la constante de gravitación universal.

$$G = 6.67384(80) \times 10^{-11} \text{ N m}^2 \text{ kg}^{-2}$$

Tener en cuenta que, habiendo varios cuerpos celestes en juego, la fuerza total es la suma de los aportes de cada interacción.

- Métodos que permitan calcular la evolución del cuerpo celeste con el paso del tiempo mediante el cálculo de la velocidad (en x e y) y la posición (en x e y) según las ecuaciones siguientes:

- $a_i = F/m$
- $v_i = v_{i-1} + a_i * dt$
- $pos_i = pos_{i-1} + v_i * dt$

donde dt es el paso de evolución temporal que en este caso será de 86400 segundos (es decir un día terrestre)

Punto 2:

Definir la clase `SistemaSolar` que contendrá los cuerpos celestes que componen el sistema.

Los atributos que se deben considerar para esta clase son:

- Cantidad máxima de cuerpos celestes del sistema
- Cantidad actual de cuerpos presentes

- Arreglo dinámico de instancias de la clase CuerpoCeleste
- Deben definirse los métodos públicos para proveer las siguientes funcionalidades:
- Un constructor que debe recibir como parámetro el número máximo de astros a considerar para el sistema y debe alojar la memoria necesaria para ese conjunto.
 - Un método que permita agregar cuerpos celestes al sistema.
 - Un método que realice la evolución del sistema mediante el cálculo de las nuevas coordenadas de cada cuerpo celeste invocando convenientemente los métodos generados en la clase anterior.
 - Un método que permita registrar en un archivo de texto los valores de las sucesivas posiciones de cada astro
 - Un método que grafique la evolución del sistema.

Punto 3:

Generar un programa principal que permita observar la evolución del sistema simulado un periodo de 20 años terrestres.

Se deben obtener dos salidas diferentes como resultado:

- Un archivo de texto con el elenco de las diferentes posiciones en x e y de los astros de la simulación
- Un gráfico que represente la evolución del sistema mostrando el movimiento de los diferentes astros en sus orbitas.

Considerar los siguientes valores de masas, posiciones y velocidades iniciales (reales) de los distintos astros para la simulación:

Cuerpo Celeste	Masa	Posición Inicial en X	Posición Inicial en Y	Velocidad Inicial en X	Velocidad Inicial en Y
Sol	1.98e30	0.	0.	0.	0.
Mercurio	3.28e24	6.99e10	0.	0.	43479.17
Venus	4.83e24	1.09e11	0.	0.	34840.23
Tierra	5.98e24	1.52e11	0.	0.	29476.35
Marte	6.37e23	2.49e11	0.	0.	23025.48
Jupiter	1.9e27	0.	8.16e11	-12723.41	0
Saturno	5.67e26	1.50e12	0.	0.	9370.69

Como ejemplo se propone el siguiente programa principal:

```
#include "../SistemaSolar.h"
#include <winbgim.h>

int main() {

    SistemaSolar ss(9); //sistema solar con un máximo de 9 cuerpos

    //Se crean los cuerpos celestes con nombre, masa, posición x, posición y,
    //velocidad en x, velocidad en y, color de representacion
    CuerpoCeleste sol("Sol",1.98e30, 0., 0.,0., 0., 15);
    CuerpoCeleste mercurio("Mercurio", 3.28e24, 6.99e10, 0.,0., 43479.17,2);
    CuerpoCeleste venus("Venus", 4.83e24, 1.09e11, 0.,0., 34840.23, 3);
    CuerpoCeleste tierra("Tierra", 5.98e24, 1.52e11, 0.,0., 29476.35, 4);
    CuerpoCeleste marte("Marte",6.37e23, 2.49e11, 0.,0., 23025.48, 5);
    CuerpoCeleste jupiter("Jupiter",1.9e27, 0., 8.16e11,-12723.41, 0., 6);
    CuerpoCeleste saturno("Saturno",5.67e26, 1.50e12, 0.,0., 9370.69, 7);

    //Se agregan los cuerpos creados al sistema solar
    ss.AgregarCC(sol);
    ss.AgregarCC(mercurio);
    ss.AgregarCC(venus);
    ss.AgregarCC(tierra);
    ss.AgregarCC(marte);
    ss.AgregarCC(jupiter);
    ss.AgregarCC(saturno);
    initwindow (1200,1200); //inicializa modo grafico
    setcolor(sol.getColor());
    outtextxy (0,0,sol.getNombre());
    setcolor(mercurio.getColor());
    outtextxy (0,30,mercurio.getNombre());
    setcolor(venus.getColor());
    outtextxy (0,60,venus.getNombre());
    setcolor(tierra.getColor());
    outtextxy (0,90,tierra.getNombre());
    setcolor(marte.getColor());
    outtextxy (0,120,marte.getNombre());
    setcolor(jupiter.getColor());
    outtextxy (0,150,jupiter.getNombre());
    setcolor(saturno.getColor());
    outtextxy (0,180,saturno.getNombre());

    //evolucion del sistema durante 20 años terrestres.
    for (int dia=1;dia<=20*365;dia++){
        ss.Simular();
        ss.Archivar();
        ss.Graficar();
    }
}
```

La salida del programa en cuanto a valores calculados debería ser como sigue:

```
sol;796.61;1421.63;
Mercurio;6.96981e+010;3.7566e+009;
Venus;1.08917e+011;3.0102e+009;
Tierra;1.51957e+011;2.54676e+009;
Marte;2.48984e+011;1.9894e+009;
Jupiter;-1.0993e+009;8.15999e+011;
Saturno;1.5e+012;8.09628e+008;

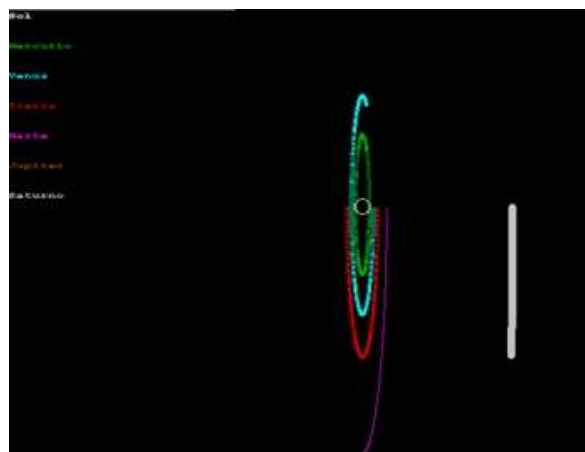
Sol;2388.49;4290.82;
Mercurio;6.9294e+010;7.50231e+009;
Venus;1.08751e+011;6.0181e+009;
Tierra;1.51872e+011;5.0928e+009;
Marte;2.48952e+011;3.97868e+009;
Jupiter;-2.1986e+009;8.15996e+011;
Saturno;1.5e+012;1.61926e+009;

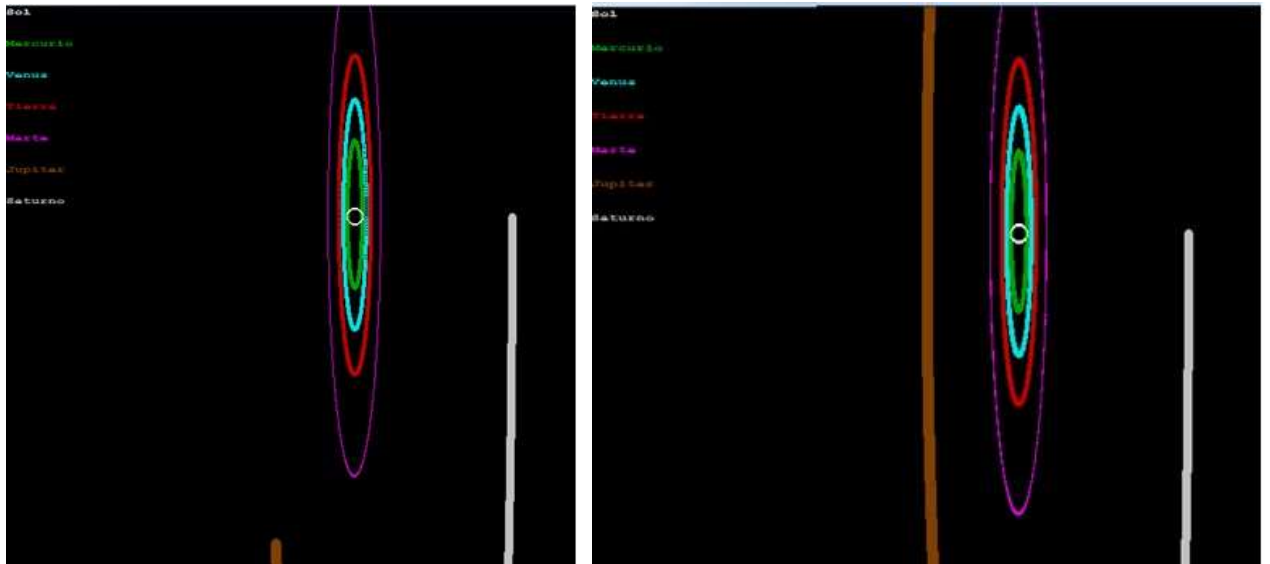
Sol;4773.12;8633.59;
Mercurio;6.86881e+010;1.12262e+010;
Venus;1.08502e+011;9.02141e+009;
Tierra;1.51744e+011;7.63742e+009;
Marte;2.48905e+011;5.9677e+009;
Jupiter;-3.2979e+009;8.15991e+011;
Saturno;1.5e+012;2.42888e+009;

Sol;7946.81;14476;
Mercurio;6.78812e+010;1.49172e+010;
Venus;1.0817e+011;1.20178e+010;
Tierra;1.51573e+011;1.01799e+010;
Marte;2.48841e+011;7.95635e+009;
Jupiter;-4.39719e+009;8.15985e+011;
Saturno;1.5e+012;3.23851e+009;

....
....
```

La salida del programa en cuanto a gráfico de la evolución debería ser como sigue (diferentes momentos de la evolución):





Observe que el trazo (grosor) de la órbita está en relación a la masa del planeta