

I Proyecto (Valor 20%) “Sistemas Planetarios”

Objetivo

Desarrollar un programa en donde se apliquen conceptos y fundamentos de programación con respecto al uso de primitivas de código, estructuras de control, diseño de funciones y Arreglos en el lenguaje de programación C++, aplicando criterios de lógica y formas óptimas de codificación para dar soluciones a problemas de origen informático.

Instrucciones

1. Lea y comprenda cuidadosamente lo que se le solicite.
2. El trabajo deberá entregarse en forma digital (tanto los fuentes, ejecutables y documentación) el día y hora indicados.
3. El proyecto será desarrollado de forma individual.
4. El proyecto debe ser desarrollado utilizando el Paradigma de Orientación a Objetos, aplicando el lenguaje C++, mediante la herramienta Dev C++.
5. El proyecto debe ser desarrollado aplicando el modo gráfico.
6. El proyecto será probado en dos ocasiones, la primera será con el profesor y los estudiantes, y la segunda será por parte del profesor a cargo del grupo.
7. Fecha de Entrega
 1. Entrega del enunciado: Martes 13 de agosto del 2019
 2. Revisión y defensa: Martes 10 de setiembre del 2019

Antecedentes

¿Qué es un Sistema Planetario?

Según la definición dada por la Real Academia Española, establece que un Sistema Planetario es “un conjunto formado por una estrella central y sus planetas, satélites y cometas”. Un ejemplo de sistema planetario es el Sistema Solar, el cual cuenta con una estrella central y planetas organizados en torno al Sol. Cuando un sistema planetario está conformado por múltiples estrellas (Binario o múltiple) se denomina Sistema Estelar. En la figura 1, se observa un diagrama de un sistema planetario denominado Sistema Kepler-186, el cual se compara con el Sistema Solar

Antes de 1543 se tenía la creencia de que la Tierra era el centro del Universo, a esta teoría se le denominó Geocentrismo establecido por Ptolomeo, pero esta fue rivalizada por la Teoría Heliocéntrica establecida por Nicolás Copérnico, esta teoría ponía al Sol como centro del sistema y además ubicaba a cada planeta en una órbita circular alrededor del Sol. (Taylor, 2000)

Años después, Johannes Kepler logró acabar con la creencia de que las órbitas planetarias eran circulares y descubrió que las órbitas realmente son elípticas, esto se convirtió en ley, conocida como la Primera Ley de Kepler. (Taylor, 2000)

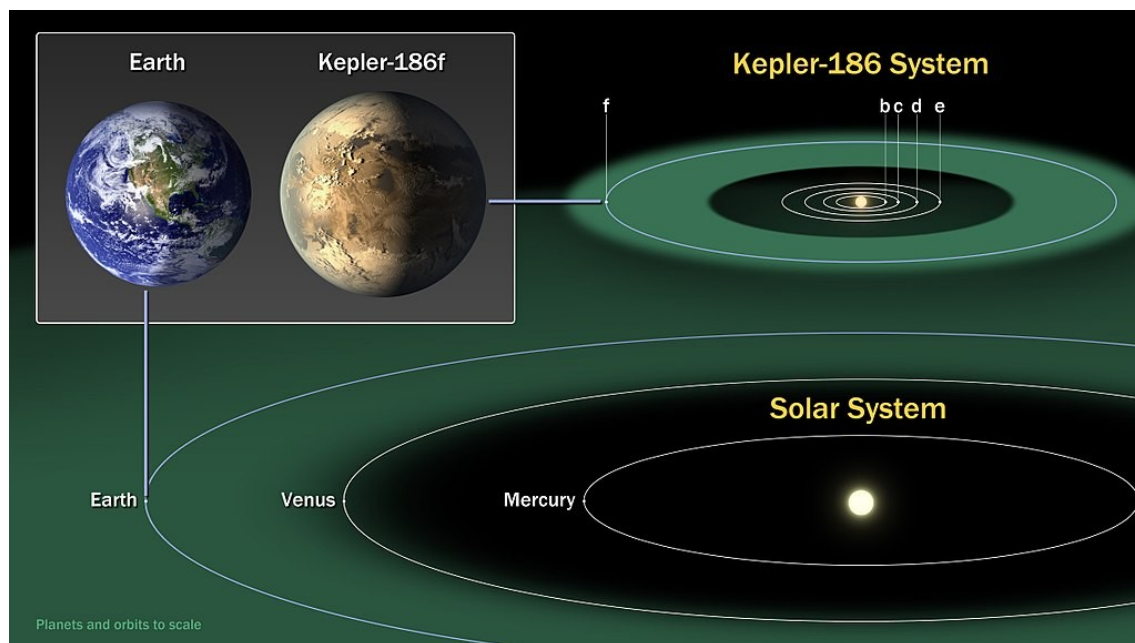


Figura 1. Sistemas planetarios. (Solís, 2018)

Descripción general

Debido a que un sistema planetario, tal como el Sistema Solar, está conformado por una estrella, planetas, satélites, asteroides y cometas, el programa a desarrollar deberá permitir graficar estos sistemas, según la información de cada uno de los cuerpos celestes que se incluyan a este.

A continuación, se indican las características que poseen cada uno de los cuerpos celestes necesarios para la construcción de los sistemas planetarios:

1. **Una estrella:** cuenta con características como el nombre, el diámetro, la temperatura, la clasificación y el color. De acuerdo a la tabla 1, se puede determinar que dependiendo de la clasificación, temperatura o color, se podrán determinar las otras características.

Clasificación	Temperatura (K)	Color	Ejemplo	Código de color HEX
O	40.000 – 25.000	Azul – violeta	I Cephei	965cfa
B	25.000 – 11.000	Blanco – azul	Spica	dddbff
A	11.000 – 7.500	Blanco	Vega	ffffff
F	7.500 – 6.000	Blanco – amarillo	Proción	faf7d2
G	6.000 – 5.000	Amarillo	Sol	fceb00
K	5.000 – 3.500	Naranja	Arturo	ffd257
M	3.500 – 3.000	Rojo	Betelgeuse	ff8870

Tabla 1. Características según clasificación.

2. **Planetas:** son cuerpos celestes que orbitan alrededor de una estrella o remanente de esta. Cuentan con las características básicas como el nombre, diámetro ecuatorial, masa, radio orbital, periodo orbital (años), el periodo de rotación (días) y el color.
3. **Satélites naturales:** son cuerpos celestes que orbitan alrededor de un planeta. Cuentan con características que deberán ser abstraídas.
4. **Asteroides:** son cuerpos rocosos más pequeños que un planeta y mayor que un meteoroides. Pueden agruparse y formar un cinturón alrededor de un cuerpo celeste denominado Cinturón de Asteroides.
5. **Cometas:** son cuerpos celestes de pequeñas dimensiones que cuando se acercan a una estrella deja tras de sí una cola luminosa de miles de kilómetros.

Una vez definidas las características de los cuerpos celestes en un sistema planetario, el programa deberá cumplir con los siguientes requerimientos:

1. Un usuario al abrir el programa, este deberá permitirle crear nuevos sistemas planetarios.
2. Cuando el usuario añade un nuevo sistema planetario, se deberán crear los cuerpos celestes que pertenecerán a ese sistema planetario.
3. El programa deberá almacenar cada sistema planetario creado, mediante la utilización de las estructuras de datos denominadas Arreglos (Arrays).
4. El programa deberá permitir el mantenimiento de los datos ya creados (editar y eliminar).
5. Una vez que un sistema planetario ha sido completado, el usuario podrá graficarlo, por lo tanto el programa lo generará a partir de los datos establecidos por el usuario. La estrella será colocada en el centro del sistema planetario y los demás cuerpos celestes se ubicarán según el radio orbital que posean.
6. Una vez que el sistema planetario se ha graficado, el usuario podrá iniciar la animación, es decir todos los cuerpos empezarán moverse alrededor de su correspondiente cuerpo celeste. La animación de estos sistemas planetarios se comportarán según el Sistema Heliocéntrico de Copérnico, en el que los planetas describen una órbita circular alrededor del Sol.

Bonus: Si el programa en vez de realizar órbitas circulares, realiza órbitas elípticas tal como lo establece la Primera Ley de Kepler, se otorgarán puntos extra. Asimismo, permitirle al usuario crear cinturones de asteroides alrededor de un cuerpo celeste.

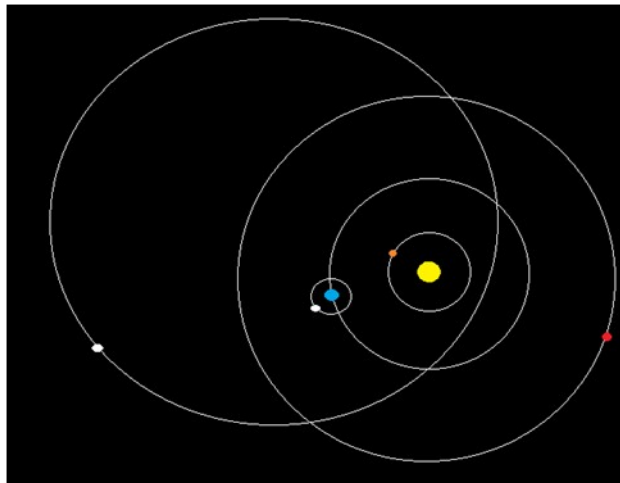


Figura 2. Ejemplo de un Sistema Planetario generado.

En el diagrama de la figura 2 se observa el ejemplo de un sistema planetario en donde incluye una estrella con tres planetas con órbitas circulares, un cometa y uno de los planetas tiene un satélite (luna).

Referencias bibliográficas

- Taylor, S. (2000). *Nuestro sistema solar y su lugar en el cosmos*. New York: Cambridge University Press.
- Solís, J. (2018). *Planetas Extrasolares*. Madrid: Anuario Astronómico del Observatorio de Madrid. Disponible en <http://astronomia.ign.es/rknowsys-theme/images/webAstro/paginas/documentos/Anuario/articuloSolis2018-apares.pdf>

Criterios de evaluación

1. Eficacia (Cumple el funcionamiento solicitado) (70%).

Rubro	Deficiente		Regular		Satisfactorio	
	0	1	2	3	4	5
1. Información (20%)						
1.1. El programa permite agregar, modificar y eliminar cuerpos celestes (13%)						
1.2. El programa permite visualizar la información de cada sistema planetario (5%)						
1.3. Se utilizan Arreglos para almacenar la información (2%)						
2. Graficación (20%)						
2.1. El programa genera automáticamente el sistema planetario seleccionado (15%)						
2.2. Cada cuerpo celeste del sistema planetario es identificable (5%)						
3. Animación (30%)						
3.1. Al ejecutar la animación todos los cuerpos celestes describen órbitas circulares alrededor de su correspondiente cuerpo celeste (25%)						
3.2. Todos los cuerpos celestes describen periodos orbitales diferentes (5%)						
4. Bonus (10%)						
4.1. Los cuerpos celestes pueden describir órbitas elípticas (7%)						
4.2. Se pueden crear cinturones de asteroides alrededor de un cuerpo celeste (3%)						

2. Eficiencia y estructura solicitada (25%)

Rubro	Deficiente 0 – 1	Regular 2 – 3	Satisfactorio 4 – 5
Separación lógica de capas	Se desarrolló la app en una sola clase o no se utilizó el paradigma de Orientación a Objetos	Se utilizan clases con funciones y atributos como contenedores de información, pero no hay división clara de lógicas	Los nombres de clases, atributos y funciones correspondientes, realizan lo que su nombre sugiere.
Código fuente	No presentó código o no es de su autoría	El código no es claro o no está bien estructurado	El código está correctamente indentado, tiene documentación interna de todos los procesos
Tratamiento de errores y validación	El programa se detiene continuamente porque no tiene control de errores o este es mínimo	El programa tiene control de muchos errores a nivel de código pero se ve interrumpido debido a la manipulación del usuario. Presenta errores de lógica que desembocan en comportamientos inesperados	El programa tiene el control interno de todos los errores predecibles, e informa al usuario de los errores que surjan por la manipulación del programa

3. Entrega de Fuentes y Creatividad/Originalidad (5%)

Rubro	Deficiente 0 – 1	Regular 2 – 3	Satisfactorio 4 – 5
Facilidad de uso	El sistema es difícil o imposible de utilizar porque el usuario necesita tener conocimiento previo para usarlo, o bien la app devuelve errores difíciles de resolver	El usuario tiene que investigar cómo utilizarlo, las opciones no están a mano y el sistema responde de manera diferente a las acciones que el usuario espera	El usuario puede realizar todas las funciones que desea sin ser inducido a error y puede encontrar fácilmente lo que necesita
Creatividad y Originalidad	En el proyecto no se demuestra innovación		Capacidad de generar nuevas ideas o conceptos que produjeron soluciones originales
Fuentes	El proyecto no permite ser ejecutado		El Proyecto es completamente funcional