Proyecto final

En este proyecto he diseñado la visión de lo que sería el sistema solar a escala, representando cada uno de los planetas y el Sol, con sus respectivos movimientos alrededor de este.

Descripción del código

Para el desarrollo del sistema solar, en los ficheros modelo.cc y modelo.h, se ha creado la clase esfera,una función para realizar el movimiento de los planetas,llamada giros_planeta y otra para la configuración de la luz para que esta vaya cambiando cada vez que los planetas se muevan,configuracionLuzGlobal().

-Clase esfera

En esta clase tenemos como parámetros el radio,las capas y las divisiones verticales, para poder realizar así la esfera, luego la distancia_centro, que especifica la distancia de esta al centro, y luego para aquellas esferas que se le haga anillo, tenemos como parámetros el radio interno y externo, y el número de segmentos.Además de un puntero a la clase Material implementada en otras prácticas, para así aplicar el material a la esfera.Por último,tenemos como métodos,constructores con parámetros, get de cada uno de los parámetros y el dibuja de la esfera,del anillo y uno para dibujar el conjunto con material y textura.

draw() -> Es el método que dibuja la esfera.

Esta se compone por dos bucles anidados,uno para las capas(dados por los ángulos de latitud,phi) y otra para los cortes o divisiones verticales(dados por los ángulos de longitud theta).
En cada iteración,se calculan cuatro vértices que forman dos triángulos para cada segmento de la esfera.Además se calculan de estos vértices, las coordenadas de textura de estos, es decir se transforman a vértices 2D, y también si las coordenadas 3D las dividimos por el radio obtenemos las normales de cada vértice. Como hemos usado triángulos se usa GL_TRIANGLES para su renderizado.

draw_anillo()->Es el método que dibuja el anillo

Este método es muy parecido al anterior, pero para su renderizado usa GL_TRIANGLES_STRIP ya que en este caso vamos a formar triángulos estirados,debido a que como un anillo su coordenada y=0, solo se calcula los vértices x, z, por ello que en este caso solo usemos theta.Los vértices que se calculan son los del radio interior y los del radio exterior que conforman el anillo,de

éstas se obtienen las coordenadas de textura para que se pueda aplicar la textura correctamente

draw_textura()-> Es el método que añade las dos anteriores pero para que se le pueda añadir textur

-Giros_planeta

Se divide en dos con un if, el movimiento órbital y el lineal para los rebotes.

Movimiento órbital
En esta parte se usan dos bucles independientes, uno para la traslación de los planetas y el otro para la rotación de los mismos.

La traslación se realiza con el calculo solo de las coordenadas x z, debido a que se mueven solo en el plano xz, donde:

x=distancia*cos(anguloRad)
z=distancia*sin(anguloRad)

La rotación se hace teniendo en cuenta la velocidad de órbita de cada planeta, para evitar que el ángulo supere los 360º, este se reinicia cuando lo supera.

Movimiento lineal
En este se usa un bucle para añadir las velocidades lineales a cada uno de las componentes del sistema solar,esta velocidad esta representada por un struct Vertice, cada una de sus componentes representa la velocidad en cada eje,aunque en el eje Y sea siempre 0.

Además aquí se añaden los límites o las paredes invisibles para que una vez se encuentren los planetas con ella reboten hacia el lado contrario.

-detectarColisiones

Esta función se encarga de detectar cuando dos de los planetas estan colisionando. Se conforma de dos bucles anidados, el primero con una componente del sistema solar y se compara con el resto, con la obtención de la distancia entre sus posiciones y se comprueba si esta es menor a la suma de los radios de esos dos planetas, si es así,para hacer el efecto de rebote, se intercambian las componentes de las velocidades de uno a otro y viceversa.

-configurarLuzGlobal

Método para hacer el efecto de que la luz del sol se refleje en todos los planetas por las caras que miren hacia este.

-

En el init se encuentran pues todas las inicializaciones de los planetas y el sol como objeto, ya que la clase principal es esta y luego para realizar la forma de estos se usa esfera que esta dentro de esta.Además encontramos varios vectores con los parámetros de estos planetas, como las diferentes componentes del material de estas, las velocidades de órbita o justo con los objetos(planetas) con los que se trabaja, y con estos vectores pues poder añadirle a las variables de los planetas sus respectivos valores.

Ejecución código

Para la ejecución del proyecto,se debe de hacer el make y luego ./practicasIG,cuando este se este ejecutando puede parar el movimiento con G y puede volver a que se de el movimiento con g,si quiere ver el rebote puede pulsar M, para volver a su órbita m.Si se clica a alguno de los planetas aparecerá su nombre para la distinción de los mismos.