

Solución

Para este problema he definido los ángulos que iré aumentando, las imágenes a cargar y algunos valores para definir la rotación de la tierra y de la luna.

Defino el tamaño de la pantalla, seguido de la carga de las imágenes del fondo(estrellas) y los elementos (planetas,estrellas, la luna), a estos últimos los redimensiono para que estén acorde al tamaño.

```
// Variables inicializadas
float angulo1,angulo2;
PImage fondo,tierra,luna,sol;
int sunSize,eSize,moonSize;
int x1,x2,y1,y2,r;
//
void setup(){
    size(1280,720);
    angulo1=0.0;
    angulo2=0.0;
    // Redimensiono el fondo
    fondo = loadImage("stars.png");
    fondo.resize(width,height);

    //Cargo las imagenes y las redimensiono para que tenga un tamaño acorde
    tierra = loadImage("earth.png");
    eSize=height/15;
    tierra.resize(eSize,eSize);

    sol = loadImage("sun.png");
    sunSize=height/8;
    sol.resize(sunSize,sunSize);

    luna = loadImage("moon.png");
    moonSize=height/30;
    luna.resize(moonSize,moonSize);
}
```

Después en la función **draw** cargo el fondo de estrellas y lo redimensiono, para definir la rotación elíptica de la tierra alrededor del sol, he usado la funciones paramétricas:

$$x = x_0 + a * \cos(t)$$

$$y = y_0 + b * \text{sen}(t)$$

Mientras que para el movimiento de la luna, he usado la ecuación con respecto a una circunferencia:

$$\begin{aligned}x &= r \cos t \\ y &= r \sin t\end{aligned}, t \in \mathbb{R}$$

Ahora, para que las transformaciones no se apliquen a todos los elementos usamos pushMatrix y popMatrix en la rotación de la tierra y de la luna, incrementando tras cada una de estas los ángulos para que se muevan y roten las imágenes de estas.

```
void draw(){
    //carga el fondo y pongo el sol en el medio de la imagen
    background(fondo);
    image(sol,(width-sunSize)/2,(height-sunSize)/2);

    // ecuaciones paramétricas para el ángulo
    x1 = width/2 + int(300.0 * cos(radians(angulo1)));
    y1 = height/2 + int(300.0 * sin(radians(angulo1)));

    //Antes y después de las transformaciones llamo a push y pop matrix para restablecerlos cambios

    pushMatrix();
    //traslado la tierra y la roto para que de vueltas
    translate(x1, y1);
    rotate(radians(r)); // rotación de 45 grados
    //incremento
    r +=1;
    image(tierra,-eSize/2,-eSize/2);
    popMatrix();

    angulo1 +=1;

    //Aquí hago lo mismo pero para la luna con un movimiento circular
    pushMatrix();
    translate(x1, y1);
    // la función varía ya que ahora es un movimiento circular
    x2 = int(50*cos(angulo2));
    y2 = int(50*sin(angulo2));
    image(luna,x2-moonSize/2,y2-moonSize/2);
    popMatrix();
    angulo2 +=0.1;
}
```