软件说明书

宁宸章 航空航天学院

目录

1 使用说明

1.1 程序介绍

本程序实现了一个卫星轨道的模拟，可以反应轨道参数对轨道的影响，方便航天动力学的同学进行课程学习。轨道参数包括半长轴 a，轨道倾角 i，升交点赤经**Ω**，以及偏心率 c。这四个参数可以唯一确定一个航天器的运动轨道。

1.2 使用说明

程序由 processing 3.5.4 环境开发，使用 java 语言。首先下载该编译环境。随后运行程序，打开 satellite 程序，运行即可看到全屏的模拟界面。如下图所示

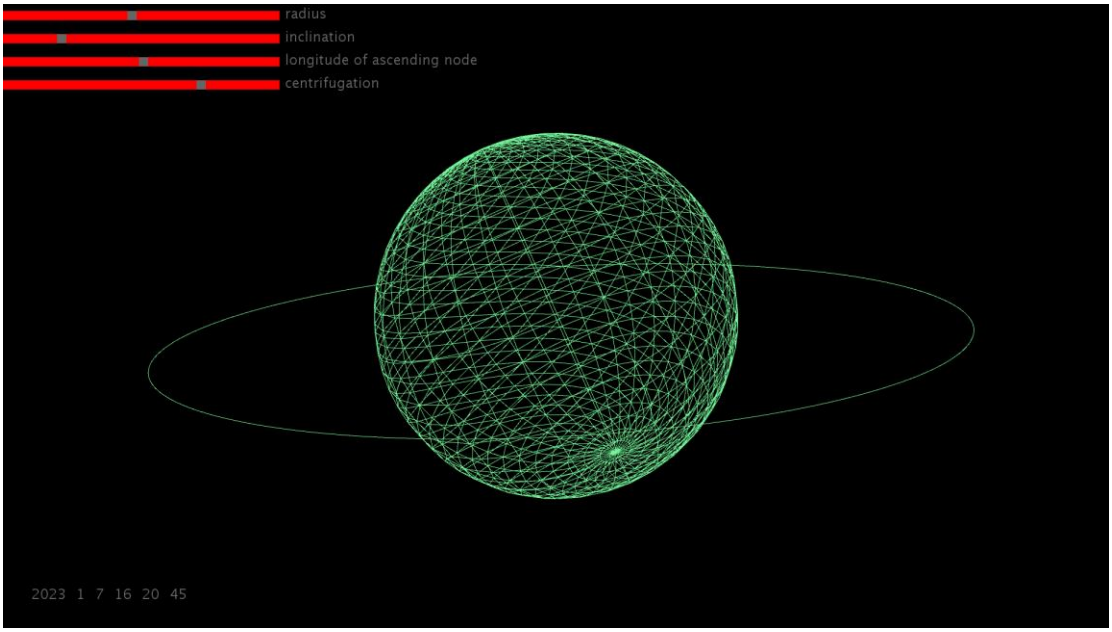

图 1 仿真界面

其中左上角为轨道参数的控制条，用户可以通过拖动灰色滑块进行调节。左下角为系统当前时间。中间即为地球与卫星轨道模型。在参数改变时，轨道会实时变化。鼠标右键单机，可以在开发文件夹下保存当前界面的截图。

2 程序解读

2.1 程序结构

本程序由三个模块构成：satellite，savePicture，scrollbar。其中 satellite 是程序主要部分，负责绘制整个界面。SavePicture 函数可帮助用户记录当前截屏。Scrollbar 中描述了控制条类，用以实现控制条的显示、滑动、参数控制等功能。

2.2 程序原理

2.2.1 Satellite

Satellite 主要有五个部分。首先进行参数声明，定义四个轨道参数和四个对应的控制条对象。

```
float radius;
float inclination; //inclination in degree
float loan; //longitude of ascending node
float c=1; //centrifugation;
boolean firstMousePress = false;
HScrollbar hs1, hs2, hs3, hs4;  // Two scrollbars
float timer = 0;

void setup() {
  fullScreen(P3D);
  textSize(24);
  //create scrollbar
  hs1 = new HScrollbar(0, 20, width/4, 16, 16);
  hs2 = new HScrollbar(0, 60, width/4, 16, 16);
  hs3 = new HScrollbar(0, 100, width/4, 16, 16);
  hs4 = new HScrollbar(0, 140, width/4, 16, 16);
}
```

图 2 satellite 初始化部分

其中 firstMousePress 用于保证同时只有一个控制条被使用，timer 用于模拟地球自转。接下来进行控制条的绘制和参数赋值。

```
  hs1.update();
  hs2.update();
  hs3.update();
  hs4.update();
  hs1.display();
  text("radius", width/4+10, 25, 0);
  hs2.display();
  text("inclination", width/4+10, 65, 0);
  hs3.display();
  text("longitude of ascending node", width/4+10, 105, 0);
  hs4.display();
  text("centrifugation", width/4+10, 145, 0);

  // Get the position of the radius scrollbar
  // and convert to a value to display
  float radius = hs1.getPos()+600;

  // Get the position of the inclination scrollbar
  // and convert to a value to display
  float inclination = (hs2.getPos()-240)/480*2*PI;

  // Get the position of the loan scrollbar
  // and convert to a value to display
  float loan = (hs3.getPos()-240)/480*2*PI;

  // Get the centrifugation of the loan scrollbar
  // and convert to a value to display
  float c = (abs(hs4.getPos()-120))/360;

  //After it has been used in the sketch, set it back to false
  if (firstMousePress) {
    firstMousePress = false;
  }
```

图 3 控制条绘制

随后是自转地球的绘制和系统时间显示。

```
// setup before draw sphere
stroke(#74F599);
noFill();
//lights();
pushMatrix();
translate(width/2, height/2);
//imitate self rotate
rotateY(timer);
rotateX(PI/6);

// rotate with mouse, uncite the following will make the model rotate with mouse
//float thetaX = map(mouseY, 0, width, -PI, PI);
//float thetaY = map(mouseX, 0, width, -PI, PI);
//rotateX(thetaX);
//rotateY(thetaY);

//draw the earth
sphere(300);
popMatrix();

//text time
timer = (timer+0.01)%TWO_PI;
text(year() + "□"  + month()+ "□" + day()+ "□" +hour()+ "□" +
  minute()+ "□" +second()+ "□", 50, height-50, 0);
```

图 4 地球和时间显示

接下来是程序的核心，轨道的绘制。

```
//draw satellite orbit
translate(width/2, height/2);
stroke(#74F599);
rotateX(inclination);
rotateY(loan);
ellipseMode(RADIUS);
//uncite the following will make the model rotate with mouse
//rotateX(thetaX);
//rotateY(thetaY);
ellipse(0, 0, radius/c/2, radius/2);
```

图 5 轨道绘制

最后是当前参数大小的输出和视频保存。

```
//print present value
println("radius");
println(radius);
println("inclination");
println(inclination);
println("loan");
println(loan);
println("centrifugation");
println(c);

//create video
saveFrame();
```

图 6 参数输出与视频保存

2.2.2 SavePicture

该函数用于生成截屏。检测到鼠标右键点击是，则保存当前截屏，命名为 PDE+系统时间。

```
//save picture
void mouseClicked() {
    if (mouseButton == RIGHT) {
    String picName = "PDE_" + year()+ "_" + month()+"_" + day()+"_" +
      hour()+"_" + minute()+"_" + second();
    save(picName + ".png");
    println(" ----> Picture saved.");
  }
}
```

图 7 savePicture 函数

2.2.3 Scrollbar

该部分创建了控制条类。首先是对象变量和声明函数的设置。创建一个新的控制条，输入其大小、位置和按钮尺寸即可。

```
class HScrollbar {
  int swidth, sheight;    // width and height of bar
  float xpos, ypos;       // x and y position of bar
  float spos, newspos;    // x position of slider
  float sposMin, sposMax; // max and min values of slider
  int loose;              // how loose/heavy
  boolean over;           // is the mouse over the slider?
  boolean locked;
  float ratio;

  HScrollbar (float xp, float yp, int sw, int sh, int l) {
    swidth = sw;
    sheight = sh;
    int widthtoheight = sw - sh;
    ratio = (float)sw / (float)widthtoheight;
    xpos = xp;
    ypos = yp-sheight/2;
    spos = xpos + swidth/2 - sheight/2;
    newspos = spos;
    sposMin = xpos;
    sposMax = xpos + swidth - sheight;
    loose = l;
  }
```

图 8 控制条类初始化

接下来是核心成员函数 update，用于实现参数更新和滑块控制。

```
void update() {
  if (overEvent()) {
    over = true;
  } else {
    over = false;
  }
  if (firstMousePress && over) {
    locked = true;
  }
  if (!mousePressed) {
    locked = false;
  }
  if (locked) {
    newspos = constrain(mouseX-sheight/2, sposMin, sposMax);
  }
  if (abs(newspos - spos) > 1) {
    spos = spos + (newspos-spos)/loose;
  }
}

float constrain(float val, float minv, float maxv) {
  return min(max(val, minv), maxv);
}

boolean overEvent() {
  if (mouseX > xpos && mouseX < xpos+swidth &&
    mouseY > ypos && mouseY < ypos+sheight) {
    return true;
  } else {
    return false;
  }
}
```

图 9 update 函数

Display 和 getpose 函数分别用于绘制控制条和返回当前参数值。

```
void display() {
  noStroke();
  fill(255,0,0);
  rect(xpos, ypos, swidth, sheight);
  if (over || locked) {
    fill(0, 255, 0);
  } else {
    fill(102, 102, 102);
  }
  rect(spos, ypos, sheight, sheight);
}

float getPos() {
  // Convert spos to be values between
  // 0 and the total width of the scrollbar
  return spos * ratio;
}
}
```

图 10 display 与 getpose 函数