



计算机图形学

作业三：星球旋转

数据科学与计算机学院 17大数据与人工智能

17341015 陈鸿峥

双击打开planet.exe即可运行，其他的glut32.dll及glut32.lib为运行所需的动态/静态库。四种基本操作如下：

- 按d键：小星球正方向自转
- 按SHIFT+d键：小星球反方向自转
- 按y键：小星球绕大星球正方向公转
- 按SHIFT+y键：小星球绕大星球反方向公转

设小星球与大星球的距离为 d ，则公转 θ 弧度后小星球的位置为

$$(x, z) = (d \sin \theta, d \cos \theta)$$

在实际做变换时应注意先进行旋转操作，再进行平移，旋转是绕 y 轴旋转。

同时为更好展示 z 轴上的距离远近，我采用了一线性函数，对小星球的半径进行调整。当小星球离我们更近时，即在 z 轴正向，则半径最大；反之，离我们越远，其显示半径越小。

实验结果如下图所示。

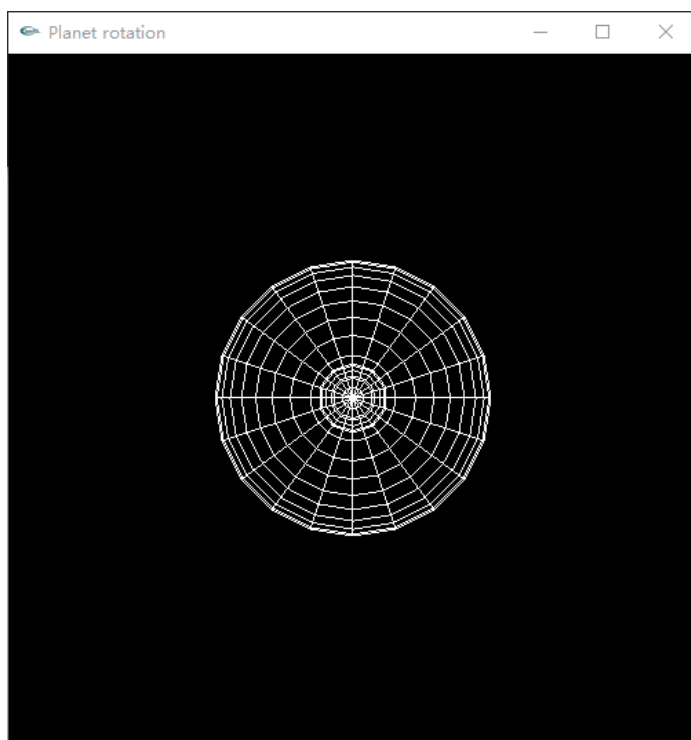


图 1: 初始状态

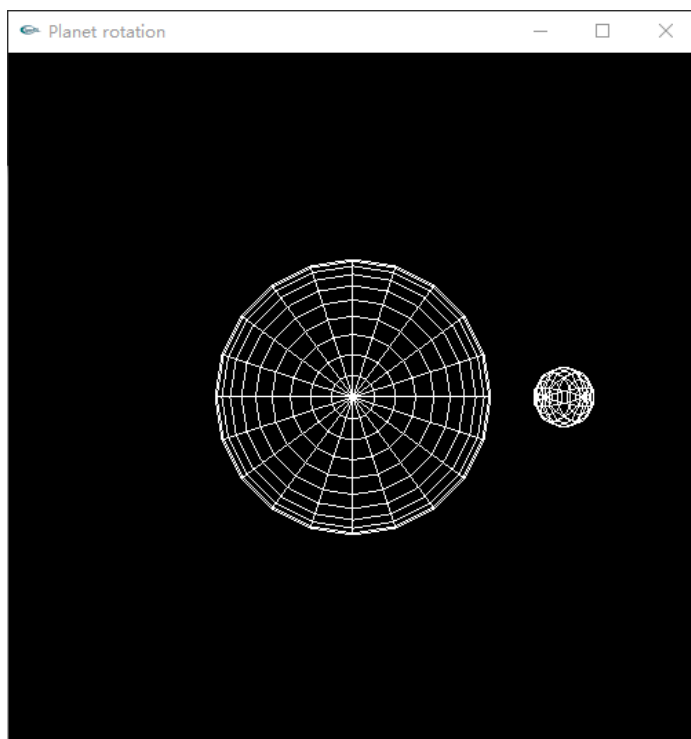


图 2: 旋转后状态

实验的代码如下。

```
#include <windows.h>
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>

const double PI = 2*acos(0.0);

int angRot = 0;
int angRevo = 0;
float distance = 0.8;

void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);

    // Big sphere
    glutWireSphere(0.4f, 20, 20);

    float posX = (float) sin((float)angRevo/180*PI) * distance;
    float posz = (float) cos((float)angRevo/180*PI) * distance;

    glPushMatrix(); // only transform smaller one
    // planet revolution
    glTranslatef(posX,0,posz);
    // planet rotation (firstly self rotate)
    glRotatef(angRot,0,1,0);
    // for visualization, the size of the sphere is changed linearly
    glutWireSphere(0.1f*(posz+2*distance)/(3*distance), 8, 8);
    glPopMatrix();

    glFlush();
}

void keyPressed(unsigned char key, int x, int y)
{
    // int mod = glutGetModifiers(); // GLUT_ACTIVE_SHIFT
    printf("Pressed %c!\n", key);
    switch (key){
        case 'd':angRot = (angRot + 10) % 360;break;
        case 'D':angRot = (angRot - 10) % 360;break;
        case 'y':angRevo = (angRevo + 10) % 360;break;
        case 'Y':angRevo = (angRevo - 10) % 360;break;
    }
    myDisplay();
}
```

```
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);

    glutInitWindowPosition(100, 100);
    glutInitWindowSize(500, 500);

    glutCreateWindow("Planet rotation");

    glutDisplayFunc(myDisplay);
    glutKeyboardFunc(keyPressed);

    // get into display
    glutMainLoop();

    return 0;
}
```