

计算机图形学

最终报告

3D 足球游戏

小 组：H

成 员：李沅达

毛铭泽

秦文航

李真希

宋一凡

许华捷

岐东睿

一、简述项目

①动机

小组成员比较喜欢运动，平常也比较喜欢看足球比赛和踢足球，也玩过一些足球游戏，所以决定做一个足球小游戏。

②目标

实现一个射门的简单足球游戏。

游戏概述：球员在球场固定的几个点进行射门，根据射中情况判断是否进入下一关

每一关提供三个不同位置的射门机会，全部命中即可进入下一关卡

提供对球的选择

提供对不同射门球员的选择

游戏内容&操作方式：射门时提供不同射门弧线的选择

射门时提供不同射门方向的选择

提供游戏视角的切换

提供重新开始游戏的选择

过关了自动进入下一关，未过关可以继续挑战同一关

射门时，球门前站有守门员(根据数量多少以及移动与否来确定关卡)

鼠标移动选择射门弧线以及方向，左键确定射出

二、项目实施

A.实现参数： `int moveX1=1;`

`int moveY1=1;`

`int moveX2=2;`

`int moveY2=2; //足球运动轨迹`

`int spinX = 0; //人眼位置`

`int spinY = 0;`

`int des = 10;`

`double BX,BY; //球的位置`

`double dx1=0; //守门员移动 x 坐标`

`int sign=1;`

`int sign_ball;`

`int sign_click=0;`

B.守门员移动：

`void OnTimer(int value)`

`{`

`/*double k;`

`k=(moveY1-moveY2)/(moveX1-moveX2);`

```

    if(k<0){

        BX-=0.1;

        BY-=k*0.1;}

    else{

        BX+=0.1;

        BY-=k*0.1;

    }*/

    if(BY>-1&&sign_click)

        BY-=0.5;

    if(BY== -1)

        sign_click=0;

    if(dx1>4)

        sign=0;

    if(dx1<-3)

        sign=1;

    switch(sign){

    case 0:

        dx1-=0.1;

        break;

    case 1:

        dx1+=0.1;

        break;

    }

    glutPostRedisplay();

    glutTimerFunc(30, OnTimer, 1);

}

```

C.地面绘制:

```
void DrawGround()
{
    glPushMatrix();

    GLfloat fExtent = 40.0f;

    GLfloat fStep = 2.5;

    GLfloat y = -10;

    GLfloat fColor;

    GLfloat iStrip, iRun;

    GLint iBounce = 0;

    glShadeModel(GL_FLAT);

    for(iStrip = -fExtent; iStrip <= fExtent; iStrip += fStep)
    {
        glBegin(GL_TRIANGLE_STRIP);

        for(iRun = fExtent; iRun >= -fExtent+30; iRun -= fStep)
        {
            if((iBounce %2) == 0)

                fColor = 139;

            else

                fColor = 238;

            glColor3ub(0, fColor, 0);

            glVertex3f(iStrip, y, iRun);

            glVertex3f(iStrip + fStep, y, iRun);

            iBounce++;
        }
    }
}
```

```

    }

    glEnd();

}

glPopMatrix();

glShadeModel(GL_SMOOTH);

}

```

D.球门绘制：

```

void drawDoor(double x, double y, double z) {

    glPushMatrix();

    glColor3ub(211,211,211);

    //画左门柱

    glTranslated(x,y,z);

    glRotated(90,1,0,0);

    GLUQuadricObj *objCylinder = gluNewQuadric();

    gluCylinder(objCylinder,0.5,0.5,10,30,10);

    //画左衔接处

    glTranslated(1,0,0);

    glRotated(90,1,0,0);

    GLdouble eqn[4]={0.0, -1.0, 0.0, 0.0};

    GLdouble eqn1[4]={-1.0, 0.0, 0.0, 0.0};

    glClipPlane(GL_CLIP_PLANE0,eqn);

    glClipPlane(GL_CLIP_PLANE1,eqn1);

    glEnable(GL_CLIP_PLANE0);

    glEnable(GL_CLIP_PLANE1);
}

```

```

    glutSolidTorus(0.5,1,30,30);

    glDisable(GL_CLIP_PLANE0);

    glDisable(GL_CLIP_PLANE1);

    //画上门柱

    glTranslated(0,-1,0);

    glRotated(90,0,1,0);

    gluCylinder(objCylinder,0.5,0.5,20,30,10);

    //画右衔接处

    glTranslated(0,1,20);

    glRotated(90,0,1,0);

    GLdouble eqn2[4]={0.0, -1.0, 0.0, 0.0};

    GLdouble eqn3[4]={-1.0, 0.0, 0.0, 0.0};

    glClipPlane(GL_CLIP_PLANE2,eqn2);

    glClipPlane(GL_CLIP_PLANE3,eqn3);

    glEnable(GL_CLIP_PLANE2);

    glEnable(GL_CLIP_PLANE3);

    glutSolidTorus(0.5,1,30,30);

    glDisable(GL_CLIP_PLANE2);

    glDisable(GL_CLIP_PLANE3);

    //画右门柱

    glTranslated(-1,0,0);

    glRotated(90,-1,0,0);

    GLUQuadricObj *objCylinder1 = gluNewQuadric();

    gluCylinder(objCylinder1,0.5,0.5,10,30,10);

    glPopMatrix();

}

```

E.球的绘制:

```
void drawBall(double R, double x, double y, double z) {  
  
    glPushMatrix();  
  
    glTranslated(x,y,z);  
  
    glutSolidSphere(R,20,20);  
  
    glPopMatrix();  
  
}
```

F.整体画面绘制:

```
void display(void) {  
  
    //清除缓冲区颜色  
  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    //定义白色  
  
    glColor3f(1.0, 1.0, 1.0);  
  
    //圆点放坐标中心  
  
    glLoadIdentity();  
  
    //从哪个地方看  
  
    gluLookAt(spinX, 0, des, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);  
  
    glPushMatrix();  
  
  
    DrawGround();  
  
    //球门  
  
    drawDoor(-10, 0, 0);  
  
  
    drawSkewed(2, 5, 1, dx1, -1.5, 0);  
  
}
```

```

        if(sign_ball==1){

            drawBall(1 ,BX, -9, BY);

        }

        glPopMatrix();

        glutSwapBuffers();

    }

```

G. 鼠标事件：

```

void mouseClick(int btn, int state, int x, int y) {

    if(btn==GLUT_LEFT_BUTTON){

        sign_click=1;

        switch(state){

            case GLUT_DOWN:

                moveX1 = x;

                moveY1 = y;

                break;

            case GLUT_UP:

                moveX2 = x;

                moveY2 = y;

                break;

        }

    }

}

```

H. 键盘事件（控制视角）：

```
void keyPressed(unsigned char key, int x, int y) {
```

```
    switch (key) {
```

```
        case 'a':
```

```
            spinX -= 2;
```

```
            BX=spinX;
```

```
            BY=des-7;
```

```
            sign_ball=0;
```

```
            break;
```

```
        case 'd':
```

```
            spinX += 2;
```

```
            BX=spinX;
```

```
            BY=des-7;
```

```
            sign_ball=0;
```

```
            break;
```

```
        case 's':
```

```
            des += 2;
```

```
            BX=spinX;
```

```
            BY=des-7;
```

```
            sign_ball=0;
```

```
            break;
```

```
        case 'w':
```

```
            des -= 2;
```

```
            BX=spinX;
```

```
            BY=des-7;
```

```
            sign_ball=0;
```


Professional. 《华章程序员书库:OpenGL 编程指南(原书第 8 版)》，机械工业出版社; 第 1 版 (2014 年 10 月 1 日)。

<http://www.scratchapixel.com/>

<http://ogldev.atspace.co.uk/>