#include <stdlib.h>

#include <GL/glut.h>

#include<stdio.h>

static GLfloat angle = 0.0;

static GLfloat move=0.0;

int flag=1;

float x=60.0,y=0.0;

void draw();

void car();

int m=-55;

int n=-12;

void init()

{

glClearColor(1.0, 1.0, 1.0, 1.0);

glShadeModel(GL\_SMOOTH);

}

void spinDisplay(void)

{

angle=angle+1.0;

if(angle>360.0)

angle=angle-360.0;

move++;

glutPostRedisplay();

}

void draw(void)

{

if(flag==1)

{

glPushMatrix();

glTranslatef(2,-15,0);

glTranslatef(move,0,0);

glColor3f(0.5,0.3,0.3);

//glColor3f(0.0,0.5,0.0);

glPushMatrix();

glTranslatef(-40,60,0);

glutSolidSphere(5,20,6); //body

glPopMatrix();

glPushMatrix();

glTranslatef(-35,60,0);

glutSolidSphere(5,20,6); //body

glPopMatrix();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-39,55);

glVertex2f(-38,53);

glVertex2f(-38.5,53); //stand

glVertex2f(-39.5,55);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-35,55);

glVertex2f(-34,53);

glVertex2f(-34.5,53); //stand

glVertex2f(-35.5,55);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-41,53);

glVertex2f(-32,53);

glVertex2f(-32,52.5);

glVertex2f(-41,52.5);

glEnd();

glColor3f(0.5,0.3,0.3);

glBegin(GL\_QUADS);

glVertex2i(-40,66);

glVertex2i(-35,66);

glVertex2i(-35,55);

glVertex2i(-40,55);

glEnd();

glBegin(GL\_QUADS);

glVertex2i(-37,68);

glVertex2i(-38,68);

glVertex2i(-38,65);

glVertex2i(-37,65);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-55,67.6);

glVertex2f(-18,67.6);

glVertex2f(-18,67.4); //leaf

glVertex2f(-55,67.4);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-70,60.6);

glVertex2f(-40,61.6);

glVertex2f(-40,58.6); //tail

glVertex2f(-70,59.6);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-67.5,62.5);

glVertex2f(-67.4,61.6);

glVertex2f(-72.4,58.5); //leaf

glVertex2f(-72.5,57.6);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-72.5,62.5);

glVertex2f(-72.5,61.5);

glVertex2f(-67.5,58.5); //leaf

glVertex2f(-67.5,57.5);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-72.5,62.5);

glVertex2f(-72.5,61.5);

glVertex2f(-67.5,58.5); //leaf

glVertex2f(-67.5,57.5);

glEnd();

glFlush();

glPopMatrix();

}

else

{

glPushMatrix();

glTranslatef(2,-15,0);

glTranslatef(move,y,0);

glColor3f(0.5,0.3,0.3);

glPushMatrix();

glTranslatef(-40,x-y,0);

glutSolidSphere(5,20,6); //body

glPopMatrix();

glPushMatrix();

glTranslatef(-35,x-y,0);

glutSolidSphere(5,20,6); //body

glPopMatrix();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-39,55);

glVertex2f(-38,53);

glVertex2f(-38.5,53); //stand

glVertex2f(-39.5,55);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-35,55);

glVertex2f(-34,53);

glVertex2f(-34.5,53); //stand

glVertex2f(-35.5,55);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_QUADS);

glVertex2f(-41,53);

glVertex2f(-32,53);

glVertex2f(-32,52.5);

glVertex2f(-41,52.5);

glEnd();

glColor3f(0.5,0.3,0.3);

glBegin(GL\_QUADS);

glVertex2i(-40,66);

glVertex2i(-35,66);

glVertex2i(-35,55);

glVertex2i(-40,55);

glEnd();

glBegin(GL\_QUADS);

glVertex2i(-37,68);

glVertex2i(-38,68);

glVertex2i(-38,65);

glVertex2i(-37,65);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-55,67.6);

glVertex2f(-18,67.6);

glVertex2f(-18,67.4); //leaf

glVertex2f(-55,67.4);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-70,60.6);

glVertex2f(-40,61.6);

glVertex2f(-40,58.6); //tail

glVertex2f(-70,59.6);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-67.5,62.5);

glVertex2f(-67.4,61.6);

glVertex2f(-72.4,58.5); //leaf

glVertex2f(-72.5,57.6);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-72.5,62.5);

glVertex2f(-72.5,61.5);

glVertex2f(-67.5,58.5); //leaf

glVertex2f(-67.5,57.5);

glEnd();

glBegin(GL\_QUADS);

glVertex2f(-72.5,62.5);

glVertex2f(-72.5,61.5);

glVertex2f(-67.5,58.5); //leaf

glVertex2f(-67.5,57.5);

glEnd();

glFlush();

glPopMatrix();

x=x-0.1;

y-=0.1;

if(x<9)

glutIdleFunc(NULL);

}

}

void car()

{

glPushMatrix();

//glTranslatef(-55,-12,0);

glTranslatef(m,n,0);

glBegin(GL\_QUADS); //car

glColor3f(1,0,0);

glVertex2f(10,-2);

glVertex2f(10,-5);

glVertex2f(-5,-5);

glVertex2f(-5,-2);

glEnd();

glTranslatef(-2,-5,0);//tyres

glColor3f(0,0,0);

glutSolidSphere(1.5,10,10);

glTranslatef(9,0,0);

glColor3f(0,0,0);

glutSolidSphere(1.5,10,10);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);//glass

glVertex2f(-7,5);

glVertex2f(-9,3);

glVertex2f(0,3);

glVertex2f(-2,5);

glEnd();

glPopMatrix();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS); //SKY

//glColor3f(0,0,1);

glColor3f(1.0,0.7,0.0);

glVertex2f(70,0);

glVertex2f(70,70);

glColor3f(1,1,1);

glVertex2i(-70,70);

glVertex2i(-70,-70);

glBegin(GL\_QUADS); //ROAD

glColor3f(0.2,0.2,0.2);

glVertex2f(60,0);

glVertex2f(60,-20);

glVertex2f(-60,-20);

glVertex2f(-60,0);

glEnd();

glBegin(GL\_QUADS); //road line

glColor3f(1,1,1);

glVertex2f(60,-9);

glVertex2f(60,-11);

glVertex2f(-60,-11);

glVertex2f(-60,-9);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(1,1,1);

glVertex2f(60,-19);

glVertex2f(60,-21);

glVertex2f(-60,-21);

glVertex2f(-60,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-55,-19);

glVertex2f(-55,-21);

glVertex2f(-50,-21);

glVertex2f(-50,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-45,-19);

glVertex2f(-45,-21);

glVertex2f(-40,-21);

glVertex2f(-40,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-35,-19);

glVertex2f(-35,-21);

glVertex2f(-30,-21);

glVertex2f(-30,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-25,-19);

glVertex2f(-25,-21);

glVertex2f(-20,-21);

glVertex2f(-20,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-15,-19);

glVertex2f(-15,-21);

glVertex2f(-10,-21);

glVertex2f(-10,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-05,-19);

glVertex2f(-05,-21);

glVertex2f(-00,-21);

glVertex2f(-00,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(05,-19);

glVertex2f(05,-21);

glVertex2f(10,-21);

glVertex2f(10,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(15,-19);

glVertex2f(15,-21);

glVertex2f(20,-21);

glVertex2f(20,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(25,-19);

glVertex2f(25,-21);

glVertex2f(30,-21);

glVertex2f(30,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(35,-19);

glVertex2f(35,-21);

glVertex2f(40,-21);

glVertex2f(40,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(45,-19);

glVertex2f(45,-21);

glVertex2f(50,-21);

glVertex2f(50,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(55,-19);

glVertex2f(55,-21);

glVertex2f(60,-21);

glVertex2f(60,-19);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(1,1,1);

glVertex2f(60,0);

glVertex2f(60,1);

glVertex2f(-60,1);

glVertex2f(-60,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-55,0);

glVertex2f(-55,1);

glVertex2f(-50,1);

glVertex2f(-50,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-45,0);

glVertex2f(-45,1);

glVertex2f(-40,1);

glVertex2f(-40,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-35,0);

glVertex2f(-35,1);

glVertex2f(-30,1);

glVertex2f(-30,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-25,0);

glVertex2f(-25,1);

glVertex2f(-20,1);

glVertex2f(-20,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-15,0);

glVertex2f(-15,1);

glVertex2f(-10,1);

glVertex2f(-10,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(-05,0);

glVertex2f(-05,1);

glVertex2f(-00,1);

glVertex2f(-00,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(05,0);

glVertex2f(05,1);

glVertex2f(10,1);

glVertex2f(10,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(15,0);

glVertex2f(15,1);

glVertex2f(20,1);

glVertex2f(20,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(25,0);

glVertex2f(25,1);

glVertex2f(30,1);

glVertex2f(30,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(35,0);

glVertex2f(35,1);

glVertex2f(40,1);

glVertex2f(40,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(45,0);

glVertex2f(45,1);

glVertex2f(50,1);

glVertex2f(50,0);

glEnd();

glBegin(GL\_QUADS); //road side

glColor3f(0,0,0);

glVertex2f(55,0);

glVertex2f(55,1);

glVertex2f(60,1);

glVertex2f(60,0);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.5,1.0,0.5);

glVertex2f(60,-20);

glVertex2f(60,-60);

glVertex2f(-60,-60);

glVertex2f(-60,-20);

glEnd();

glPushMatrix();

glTranslatef(-40,50,0);

glColor3f(1.0,0.7,0.0);

glutSolidSphere(5,30,25); //SUN

glPopMatrix();

glPushMatrix();

glColor3f(1.0,1.0,1.0);

glTranslatef(-6,38,0);

glutSolidSphere(4,30,25);

glPopMatrix();

/\* glPushMatrix();

glColor3f(1.0,1.0,1.0);

glTranslatef(-5,43,0);

glutSolidSphere(4,30,25);

glPopMatrix();\*/

glPushMatrix();

glColor3f(1.0,1.0,1.0);

glTranslatef(0,45,0);

glutSolidSphere(2,30,25);

glPopMatrix();

glPushMatrix();

glColor3f(1.0,1.0,1.0);

glTranslatef(-2,41,0);

glutSolidSphere(4,30,25);

glPopMatrix();

glPushMatrix();

glColor3f(1.0,1.0,1.0);

glTranslatef(3,40,0);

glutSolidSphere(6,30,25);

glPopMatrix();

draw();

//car();

glPopMatrix();

glEnd();

car();

if(move==100)

{

move=-100;

glutIdleFunc(spinDisplay);

flag=0;

}

//Sleep(30);

glutSwapBuffers();

}

void reshape(int w, int h)

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-60.0,60.0,-60.0,60.0,-1.0,20.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void keyboard(unsigned char key, int x, int y)

{

switch (key)

{

case 27:

exit(0);

break;

case 'q':

//glutIdleFunc(car);

car();

m++;

glutPostRedisplay();

break;

case 'w':

//glutIdleFunc(car);

car();

m--;

glutPostRedisplay();

break;

}

}

void mouse(int button, int state, int x, int y)

{

switch(button)

{

case GLUT\_LEFT\_BUTTON:

if (state == GLUT\_DOWN)

{

//glutIdleFunc(spinDisplay);

spinDisplay();

glutPostRedisplay();

}

break;

case GLUT\_RIGHT\_BUTTON:

glutIdleFunc(NULL);

}

}

int main(int argc, char\* argv[])

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB);

glutInitWindowSize(900,900);

glutInitWindowPosition(0,0);

glutCreateWindow(argv[0]);

init();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutMouseFunc(mouse);

glutKeyboardFunc(keyboard);

glutMainLoop();

return 0;

}