//Independence\_Day

#include <windows.h>

#include <GL/glut.h>

#include <GL/glu.h>

#include<math.h>

#include <stdlib.h>

#include<stdio.h>

#include <GL/gl.h>

#include<MMSystem.h>

#define PI 3.1416

GLint i;

GLfloat cx=0,str=500.0,mn=500.0;

GLfloat cx1=0,str1=500.0,mn1=500.0;

GLfloat sr=0.0,sg=0.749,sb=1.0;

float \_rain = 0.0;

bool rainday = false;

void circle(GLdouble rad)

{

GLint points = 50;

GLdouble delTheta = (2.0 \* PI) / (GLdouble)points;

GLdouble theta = 0.0;

glBegin(GL\_POLYGON);

{

for( i = 0; i <=50; i++, theta += delTheta )

{

glVertex2d(rad \* cos(theta),rad \* sin(theta));

}

}

glEnd();

}

void car()

{

glColor3f(1.0, 0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f(-0.52, 0.2, 0);

glVertex3f(-0.9, 0.2, 0);

glVertex3f(-0.87, .5, 0);

glVertex3f(-0.52, .5, 0);

glEnd();

glColor3f(.0, 1.5, .0);

glBegin(GL\_POLYGON);

glVertex3f(-0.43, 0.2, 0);

glVertex3f(-0.52, 0.2, 0);

glVertex3f(-0.52, 0.5, 0);

glVertex3f(-0.46, 0.45, 0);

glEnd();

glColor3f(.0, 1.5, .0);

glBegin(GL\_POLYGON);

glVertex3f(-0.53, 0.5, 0);

glVertex3f(-0.83, 0.5, 0);

glVertex3f(-0.77, 0.8, 0);

glVertex3f(-0.56, 0.75, 0);

glEnd();

glColor3f(.0, .0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.65, 0.2, 0);

glVertex3f(-0.65, 0.5, 0);

glEnd();

glColor3f(.0, 0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.65, 0.5, 0);

glVertex3f(-0.68, 0.79, 0);

glEnd();

glColor3f(.0, 1.0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.75, 0.2, 0);

glVertex3f(-0.75, 0.5, 0);

glEnd();

glColor3f(.0, 1.0, 0);

glBegin(GL\_LINES);

glVertex3f(-0.75, 0.5, 0);

glVertex3f(-0.77, 0.8, 0);

glEnd();

glColor3f(.0, 1.0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.55, 0.2, 0);

glVertex3f(-0.55, 0.5, 0);

glEnd();

glColor3f(.0, 0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.55, 0.5, 0);

glVertex3f(-0.58, 0.77, 0);

glEnd();

glColor3f(1.0, 0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.6, 0.4, 0);

glVertex3f(-0.62, 0.4, 0);

glEnd();

glColor3f(0, 0, 1.0);

glBegin(GL\_LINES);

glVertex3f(-0.7, 0.4, 0);

glVertex3f(-0.72, 0.4, 0);

glEnd();

glColor3f(0.0, .0, 1.0);

glPushMatrix();

glTranslatef(-.78, .2, 0);

circle(0.06);

glPopMatrix();

glColor3f(0.0, .0, 1.0);

glPushMatrix();

glTranslatef(-.5, .2, 0);

circle(0.06);

glPopMatrix();

glColor3f(1.0, .0, 1.0);

glPushMatrix();

glTranslatef(-.5, .2, 0);

circle(0.02);

glPopMatrix();

glColor3f(1.0, 1.0, .0);

glPushMatrix();

glTranslatef(-.78, .2, 0);

circle(0.02);

glPopMatrix();

}

void cloudBig()

{

//left

glPushMatrix();

glTranslatef(4,5.5,0);

circle(4);

glPopMatrix();

//right

glPushMatrix();

glTranslatef(-8,5.5,0);

circle(3.5);

glPopMatrix();

//top left

glPushMatrix();

glTranslatef(-3.5,9,0);

circle(3.5);

glPopMatrix();

//top right

glPushMatrix();

glTranslatef(1,9,0);

circle(3);

glPopMatrix();

//middle

glPushMatrix();

//glColor3f (1, 1 ,1);

glTranslatef(-1.5,5.5,0);

circle(4);

glPopMatrix();

}

//void a() //mini Cloud

void miniCloud()

{

//left

glPushMatrix();

glTranslatef(4,5.5,0);

circle(4);

glPopMatrix();

//right

glPushMatrix();

glTranslatef(-8,5.5,0);

//3.5

circle(3.5);

glPopMatrix();

//top left

glPushMatrix();

glTranslatef(-3.5,9,0);

circle(3.5);

glPopMatrix();

//top right

glPushMatrix();

glTranslatef(1,9,0);

circle(3);

glPopMatrix();

//middle

glPushMatrix();

glTranslatef(-1.5,5.5,0);

circle(4);

glPopMatrix();

}

void singleCloud() //One Single Cloud

{

glPushMatrix();

glColor3f (1,1,1);

glTranslatef(35,10,0);

miniCloud();

glPopMatrix();

glPushMatrix();

glColor3f (1,1,1);

glTranslatef(28,16,0);

miniCloud();

glPopMatrix();

glPushMatrix();

glColor3f (1,1,1);

glTranslatef(20,10,0);

miniCloud();

glPopMatrix();

}

void threeCloud() // Three Cloud

{

glPushMatrix();

glTranslatef(-15.0,30.0,0.0);

glScalef(0.7,0.7,0.0);

singleCloud();

glPopMatrix();

glPushMatrix();

glTranslatef(20.0,25.0,0.0);

glScalef(0.7,0.7,0.0);

singleCloud();

glPopMatrix();

glPushMatrix();

glTranslatef(-60.0,25.0,0.0);

glScalef(0.7,0.7,0.0);

singleCloud();

glPopMatrix();

}

void rect()

{

glRectf(-16.0, -16.0, 16.0, 16.0);

}

GLfloat ss=0.0;

void Day()

{

glBegin(GL\_POLYGON); // blue sky

glColor3f(sr,sg,sb);

glVertex3f(-50,-3,0.0);

glVertex3f(-50,50,0.0);

glVertex3f(80,50,0.0);

glVertex3f(80,-3,0.0);

glEnd();

glPushMatrix(); //sun

glTranslatef(ss,0.0,0.0);

glTranslatef(-20.0,40.0,0.0);

glScalef(1.0,1.5,0.0);

glColor3f(1.0,1.0,0.0);

circle(3);

glPopMatrix();

}

void ground()

{

glColor3f(0.8,0.498039,0.196078);

glPushMatrix();

glTranslatef(-70,-42,0);

glBegin(GL\_POLYGON);

glVertex3f (-10, 0, 0.0);

glVertex3f (-10,10, 0.0);

glVertex3f (600,10, 0.0);

glVertex3f (600,0, 0.0);

glEnd();

}

void night () //black sky

{

glBegin(GL\_POLYGON);

glColor3f(0.0,0.0,0.0);

glVertex3f(-50,-3,0.0);

glVertex3f(-50,50,0.0);

glVertex3f(80,50,0.0);

glVertex3f(80,-3,0.0);

glEnd();

}

void moon() //moon

{

glPushMatrix();

glTranslatef(mn,0.0,0.0);

glTranslatef(20.0,35.0,0.0);

glScalef(1.0,1.5,0.0);

glColor3f(1.0,1.0,1.0);

circle(4.5);

glPopMatrix();

glutPostRedisplay();

}

void triangle(void)

{

glColor3f (0,128,0);

glBegin(GL\_POLYGON);

glVertex3f (0, 0, 0.0);

glVertex3f (9, 13, 0.0);

glVertex3f (18, 0, 0.0);

glEnd();

}

void grass()

{

glPushMatrix();

glColor3f (0.8,0.196078,0.6);

glTranslatef(38,16,0);

glScalef(.1,.1,0);

threeCloud();

glPopMatrix();

}

void Memorial()

{

glColor3f(0.4, 0.3, 0.2) ; //ground 1st

glBegin(GL\_POLYGON);

glVertex2i(2, 0);

glVertex2i (34, 0);

glVertex2i (34, 0);

glVertex2i (34, 1);

glVertex2i (34, 1);

glVertex2i (2, 1);

glVertex2i (2, 1);

glVertex2i (2, 0);

glEnd();

glColor3f(0.5, 0.3, 0.2) ; //ground 2nd

glBegin(GL\_POLYGON);

glVertex2i(3, 1);

glVertex2i (33, 1);

glVertex2i (33, 1);

glVertex2i (33, 2);

glVertex2i (33, 2);

glVertex2i (3, 2);

glVertex2i (3, 2);

glVertex2i (3, 1);

glEnd();

glColor3f(0.7, 0.3, 0.2) ; //ground 3rd

glBegin(GL\_POLYGON);

glVertex2i(4, 2);

glVertex2i (32, 2);

glVertex2i (32, 2);

glVertex2i (32, 3);

glVertex2i (32, 3);

glVertex2i (4, 3);

glVertex2i (4, 3);

glVertex2i (4, 2);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn 1

glBegin(GL\_POLYGON);

glVertex2i(4, 3);

glVertex2i (18, 6);

glVertex2i (18, 6);

glVertex2i (32, 3);

glVertex2i (32, 3);

glVertex2i (4, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trin 1 main

glBegin(GL\_POLYGON);

glVertex2i(4, 3);

glVertex2i (31, 3);

glVertex2i (31, 3);

glVertex2i (18, 6);

glVertex2i (18, 6);

glVertex2i (4, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn2

glBegin(GL\_POLYGON);

glVertex2i(6, 3);

glVertex2i (18, 10);

glVertex2i (18, 10);

glVertex2i (30, 3);

glVertex2i (30, 3);

glVertex2i (6, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn2 main

glBegin(GL\_POLYGON);

glVertex2i(6, 3);

glVertex2i (18, 10);

glVertex2i (18, 10);

glVertex2i (29, 3);

glVertex2i (29, 3);

glVertex2i (6, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn3

glBegin(GL\_POLYGON);

glVertex2i(8, 3);

glVertex2i (18, 15);

glVertex2i (18, 15);

glVertex2i (28, 3);

glVertex2i (28, 3);

glVertex2i (8, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn3 main

glBegin(GL\_POLYGON);

glVertex2i(8, 3);

glVertex2i (18, 15);

glVertex2i (18, 15);

glVertex2i (27, 3);

glVertex2i (27, 3);

glVertex2i (8, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn 4

glBegin(GL\_POLYGON);

glVertex2i(10, 3);

glVertex2i (18, 20);

glVertex2i (18, 20);

glVertex2i (26, 3);

glVertex2i (26, 3);

glVertex2i (10, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn4 main

glBegin(GL\_POLYGON);

glVertex2i(10, 3);

glVertex2i (18, 20);

glVertex2i (18, 20);

glVertex2i (25, 3);

glVertex2i (25, 3);

glVertex2i (10, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn 5

glBegin(GL\_POLYGON);

glVertex2i(12, 3);

glVertex2i (18, 25);

glVertex2i (18, 25);

glVertex2i (24, 3);

glVertex2i (24, 3);

glVertex2i (12, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn5 main

glBegin(GL\_POLYGON);

glVertex2i(12, 3);

glVertex2i (18, 25);

glVertex2i (18, 25);

glVertex2i (23, 3);

glVertex2i (23, 3);

glVertex2i (12, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn 6

glBegin(GL\_POLYGON);

glVertex2i(14, 3);

glVertex2i (18, 30);

glVertex2i (18, 30);

glVertex2i (22, 3);

glVertex2i (22, 3);

glVertex2i (14, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn6 main

glBegin(GL\_POLYGON);

glVertex2i(14, 3);

glVertex2i (18, 30);

glVertex2i (18, 30);

glVertex2i (21, 3);

glVertex2i (21, 3);

glVertex2i (14, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //trn 7

glBegin(GL\_POLYGON);

glVertex2i(16, 3);

glVertex2i (18, 35);

glVertex2i (18, 35);

glVertex2i (20, 3);

glVertex2i (20, 3);

glVertex2i (16, 3);

glEnd();

glColor3f(0.4, 0.4, 0.4) ; //trn7 main

glBegin(GL\_POLYGON);

glVertex2i(16, 3);

glVertex2i (18, 35);

glVertex2i (18, 35);

glVertex2i (19, 3);

glVertex2i (19, 3);

glVertex2i (16, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //middle line.

glBegin(GL\_POLYGON);

glVertex2i(16, 14);

glVertex2i (16, 14);

glVertex2i (16, 14);

glVertex2i (19, 14);

glVertex2i (19, 14);

glVertex2i (19, 14);

glVertex2i (19, 14);

glVertex2i (16, 14);

glEnd();

glColor3f(0.3, 0.3, 0.3) ; //under train main

glBegin(GL\_POLYGON);

glVertex2i(16, 14);

glVertex2i (18, 9);

glVertex2i (18, 9);

glVertex2i (19, 14);

glVertex2i (19, 14);

glVertex2i(16, 14);

glEnd();

glColor3f(0.3, 0.3, 0.3) ; //upper train main

glBegin(GL\_POLYGON);

glVertex2i(16, 14);

glVertex2i (18, 19);

glVertex2i (18, 19);

glVertex2i (19, 14);

glVertex2i (19, 14);

glVertex2i(16, 14);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; //low train

glBegin(GL\_POLYGON);

glVertex2i(16, 3);

glVertex2i (18, 4);

glVertex2i (18, 4);

glVertex2i (20, 3);

glVertex2i (20, 3);

glVertex2i(16, 3);

glEnd();

glColor3f(0.7, 0.3, 0.2) ; //low train main

glBegin(GL\_POLYGON);

glVertex2i(16, 3);

glVertex2i (18, 4);

glVertex2i (18, 4);

glVertex2i (19, 3);

glVertex2i (19, 3);

glVertex2i(16, 3);

glEnd();

glColor3f(1.0, 1.0, 1.0) ; ; //Middle line top to bottom

glBegin(GL\_LINES);

glVertex2i (18, 4);

glVertex2i (18, 9);

glVertex2i (18, 19);

glVertex2i (18, 35);

glEnd();

}

void tree() //green leaves

{

glPushMatrix();

glTranslatef(35,10,0);

miniCloud();

glPopMatrix();

glPushMatrix();

glTranslatef(28,16,0);

miniCloud();

glPopMatrix();

glPushMatrix();

glTranslatef(20,10,0);

miniCloud();

glPopMatrix();

}

void treebody() //tree body

{

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(0,0,0);

glVertex3f(2.5,2,0);

glVertex3f(4.0,-2,0);

glVertex3f(1,-4,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(7,2,0);

glVertex3f(9,2,0);

glVertex3f(8,-2,0);

glVertex3f(4,-2,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(1,-4,0);

glVertex3f(4,-2,0);

glVertex3f(8,-2,0);

glVertex3f(7,-7,0);

glVertex3f(1.5,-7,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(1.5,-7,0);

glVertex3f(7,-7,0);

glVertex3f(6.5,-10,0);

glVertex3f(2,-10,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(2,-10,0);

glVertex3f(6.5,-10,0);

glVertex3f(6.8,-13,0);

glVertex3f(1.5,-13,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(1.5,-13,0);

glVertex3f(6.8,-13,0);

glVertex3f(7,-18,0);

glVertex3f(.5,-18,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(.5,-18,0);

glVertex3f(7,-18,0);

glVertex3f(3,-27,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(.5,-18,0);

glVertex3f(2.5,-23,0);

glVertex3f(-1,-25,0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f (0.502, 0.000, 0.000);

glVertex3f(7,-18,0);

glVertex3f(8,-28,0);

glVertex3f(4,-22,0);

glEnd();

}

//1st Home

void home1()

{

glColor3ub(0,0,10);

glBegin(GL\_POLYGON);

glVertex2d(3,14);

glVertex2d(3,11);

glVertex2d(10,8);

glVertex2d(10,12);

glVertex2d(6,18);

glEnd();

//2

glColor3ub(153,0,0);

glBegin(GL\_POLYGON);

glVertex2d(10,8);

glVertex2d(10,12);

glVertex2d(20,12);

glVertex2d(20,8);

glEnd();

//3

glColor3ub(0,0,200);

glBegin(GL\_POLYGON);

glVertex2d(10,12);

glVertex2d(6,18);

glVertex2d(17,18);

glVertex2d(21,12);

glEnd();

//4

glColor3ub(255,0,255);

glBegin(GL\_POLYGON);

glVertex2d(5,11);

glVertex2d(5,12);

glVertex2d(8,11);

glVertex2d(8,10);

glEnd();

//5

glColor3ub(50,50,50);

glBegin(GL\_POLYGON);

glVertex2d(14,8);

glVertex2d(14,10);

glVertex2d(17,10);

glVertex2d(17,8);

glEnd();

}

void house()

{

glColor3ub(240,0,0);

glBegin(GL\_POLYGON);

glVertex2d(33,23);

glVertex2d(44,23);

glVertex2d(44,30);

glVertex2d(33,30);

glEnd();

glColor3ub(100,105,105);

glBegin(GL\_POLYGON);

glVertex2d(31,30);

glVertex2d(46,30);

glVertex2d(38.5,38);

glEnd();

}

void tree21()

{

glPushMatrix();

glTranslatef(3,8,0);

triangle();

glPopMatrix();

glPushMatrix();

glTranslatef(3.5,14,0);

glScalef(.9,.9,0);

triangle();

glPopMatrix();

glPushMatrix();

glTranslatef(4.5,20,0);

glScalef(.8,.8,0);

triangle();

glPopMatrix();

glPushMatrix();

// 26 hbe

glTranslatef(7,25,0);

glScalef(.5,.5,0);

triangle();

glPopMatrix();

//gora

glPushMatrix();

glBegin(GL\_POLYGON);

glColor3f(0.36,0.25,0.20);

glVertex3f (10, 4, 0.0);

glVertex3f (10, 8, 0.0);

glVertex3f (14, 8, 0.0);

glVertex3f (14, 4, 0.0);

glEnd();

glPopMatrix();

}

void Rain(int value){

if(rainday){

\_rain += 0.01f;

glBegin(GL\_POINTS);

for(int i=1;i<=1000;i++)

{

int x=rand(),y=rand();

x%=1000; y%=1000;

glBegin(GL\_LINES);

glColor3f(1.0, 1.0, 1.0);

glVertex2d(x,y);

glVertex2d(x+50,y+50);

glEnd();

}

glutPostRedisplay();

glutTimerFunc(5, Rain, 0);

glFlush();

}

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glPushMatrix();

Day();

glPopMatrix();

///ground///

glPushMatrix();

glTranslatef(0.0,-35.0,0.0);

glScalef(3.5,3.0,0.0);

glColor3f(0.0, 0.5, 0.0);

rect();

glPopMatrix();

glPushMatrix();

glColor3f (1,1,1);

glTranslatef(cx1,23.0,0.0);

cloudBig();

glPopMatrix();

glPushMatrix();

glColor3f (1,1,1);

glTranslatef(cx1,20.0,0.0);

cloudBig();

glPopMatrix();

///tree typeB 1///

glPushMatrix();

glColor3f(0.133, 0.545, 0.133);

glTranslatef(35.0,-5.0,0.0);

glScalef(.5,1,0);

tree21();

glPopMatrix();

glPushMatrix();

glColor3f(0.133, 0.545, 0.133);

glTranslatef(30.0,-15.0,0.0);

glScalef(.5,1,0);

tree21();

glPopMatrix();

//tree1

glPushMatrix();

glColor3f(0.133, 0.545, 0.133);

glTranslatef(-49.0,9.5,0.0);

glScalef(0.4,0.5,0.0);

tree();

glPopMatrix();

glPushMatrix();

glTranslatef(-40.4,15.4,0.0);

glScalef(0.4,0.5,0.0);

treebody();

glPopMatrix();

glPushMatrix();

moon();

glPopMatrix();

glPushMatrix();

glTranslatef(cx1,0.0,0.0);

glScalef(.8,1,0);

threeCloud();

glPopMatrix();

///home 2///

glPushMatrix();

glTranslatef(-65.0,-20.0,0.0);

house();

glPopMatrix();

///\*\*\*///

///home 1///

glPushMatrix();

glTranslatef(-45.0,-10.0,0.0);

home1();

glPopMatrix();

///Memo///

glPushMatrix();

glTranslatef(-5.0,-5.0,0.0);

glScalef(.5,1,0);

Memorial();

glPopMatrix();

///\*\*\*///

///tree typeB 1///

///tree2///

glPushMatrix();

glColor3f(0.133, 0.545, 0.133);

glTranslatef(-44.0,-1,0.0);

glScalef(0.4,0.5,0.0);

tree();

glPopMatrix();

glPushMatrix();

glTranslatef(-35.0,5.0,0.0);

glScalef(0.4,0.5,0.0);

treebody();

glPopMatrix();

///\*\*\*\*\*\*\*\*home 4\*\*\*\*\*\*\*\*///

///\*\*\*\*\*\*\*\*road\*\*\*\*\*\*\*\*///

glPushMatrix();

glTranslatef(-10.0,15.0,0.0);

ground();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glTranslatef(-10.0,20.0,0.0);

ground();

glPopMatrix();

glPopMatrix();

///\*\*\*\*\*\*\*\*\*\*\*\*///

///car///

glPushMatrix();

glTranslatef(cx, -15, 0);

glScalef(30, 8, 0);

car();

glPopMatrix();

///\*\*\*///

glPushMatrix();

glTranslatef(cx-15, -20, 0);

glScalef(30, 8, 0);

car();

glPopMatrix();

///tree3///

glPushMatrix();

glColor3f(0.133, 0.545, 0.133);

glTranslatef(22.0,-32.5,0.0);

glScalef(0.4,0.5,0.0);

tree();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glTranslatef(30.5,-26.5,0.0);

glScalef(0.4,0.5,0.0);

treebody();

glPopMatrix();

///home 3///

glPushMatrix();

glTranslatef(-55.0,-50.0,0.0);

home1();

glPopMatrix();

glPopMatrix();

///\*\*\*///

///extra//

///Small house 1//

glPushMatrix();

glTranslatef(-45.0,-65.0,0.0);

house();

glPopMatrix();

///tree typeD 1///

glPushMatrix();

glTranslatef(-10.0,-50.0,0.0);

glScalef(.5,1,0);

tree21();

glPopMatrix();

glPushMatrix();

glTranslatef(-50.0,-50.0,0.0);

glScalef(.5,1,0);

tree21();

glPopMatrix();

///Small house 2//

glPushMatrix();

glTranslatef(-5.0,-70.0,0.0);

house();

glPopMatrix();

glFlush();

}

void init(void)

{

glClearColor (0, 0.749, 1, 0);

glOrtho(-50.0,50.0, -50.0, 50.0, -1.0, 1.0);

}

void spinFastDisplay()

{

//Car

cx = cx + .03;

if(cx>70)

cx = -70;

//cloud

cx1 = cx1 + .007;

if(cx1>70)

cx1 = -70;

glutPostRedisplay();

}

void DisplayDay()

{

str=500.0;

sr=0.0;

sg=0.749;

sb=1.0;

ss = 0.0;

mn = 500.0;

glutPostRedisplay();

}

void DisplayNight()

{

str = 0.0;

sr=0.0;

sg=0.0;

sb=0.0;

ss = 500.0;

mn = 0.0;

glutPostRedisplay();

}

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

{

switch (button)

{

case GLUT\_LEFT\_BUTTON:

if (state == GLUT\_DOWN)

{

DisplayDay();

break;

}

case GLUT\_RIGHT\_BUTTON:

if (state == GLUT\_DOWN)

{

DisplayNight();

break;

}

default:

{

break;

}

}

}

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

{

switch(key)

{

case 'r':

{

rainday = true;

Rain(\_rain);

sndPlaySound("1.wav",SND\_ASYNC);

break;

}

case 's':

{

rainday = false;

break;

}

default:

break;

}

}

int main()

{

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (1350, 690);

glutInitWindowPosition (0, 0);

glutCreateWindow ("Independence Day");

init();

glutDisplayFunc(display);

glutIdleFunc(spinFastDisplay);

glutMouseFunc(mouse);

glutKeyboardFunc(keyboard);

glutMainLoop();

return 0;

}