**Computer Graphics Project : “Eid Carnival”**

#include <iostream>

#include <GL/gl.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

#include<windows.h>

#include<mmsystem.h>

#define PI 3.14159265358979323846

using namespace std;

bool night=false;

bool vrain=false;

char carnival[]="EID CARNIVAL";

//

float anglel\_M = 0.0f;

float anglel\_N = 0.0f;

GLfloat speed\_M= 0.0f;

GLfloat speed\_N = 0.0f;

GLfloat trainSpeed=0.02f;

GLfloat trainPos = 0.0f;

GLfloat rainSpeed=0.02f;

GLfloat rainPos = 0.0f;

GLfloat rainSpeedX=0.01f;

GLfloat rainPosX = 0.0f;

GLfloat positionOfCloud1 = 0.0f;

GLfloat speedOfCloud1 = 0.001f;

GLfloat positionOfCloud2 = 0.0f;

GLfloat speedOfCloud2 = 0.002f;

GLfloat positionOfBird = 0.0f;

GLfloat speedOfBird = 0.003f;

void line(double a1,double b1,double a2,double b2) // to draw line

{

glBegin(GL\_LINES);

glVertex2f(a1,b1);

glVertex2f(a2,b2);

glEnd();

}

void triangle(double a1,double b1,double a2,double b2,double a3,double b3) // to draw triangle

{

glBegin(GL\_TRIANGLES);

glVertex2f(a1,b1);

glVertex2f(a2,b2);

glVertex2f(a3,b3);

glEnd();

}

void quad(double a1,double b1,double a2,double b2,double a3,double b3,double a4,double b4) // to draw quad

{

glBegin(GL\_QUADS);

glVertex2f(a1,b1);

glVertex2f(a2,b2);

glVertex2f(a3,b3);

glVertex2f(a4,b4);

glEnd();

}

void printText( float x, float y, char \*st) // to print string

{

int l,i;

l=strlen( st );

glColor3ub(230,196,38);

glRasterPos2f( x, y);

for( i=0; i < l; i++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, st[i]);

}

}

void circle(GLfloat x, GLfloat y, GLfloat radius) // to draw circle

{

int i;

float angle;

glBegin(GL\_POLYGON);

for(i=0;i<100;i++)

{

angle = i\*2\*(M\_PI/100);

glVertex2f(x+(sin(angle)\*radius),y+(cos(angle)\*radius));

}

glEnd();

}

void update(int value)

{

//FOR MARRY GO ROUND

anglel\_M+= speed\_M;

if (anglel\_M>360)

{

anglel\_M-=360;

}

//FOR BOAT

anglel\_N+=speed\_N;

if (anglel\_N> 10)

{

speed\_N = -0.5;

}

else if(anglel\_N<-10)

{

speed\_N=+0.5;

}

if(trainPos<-2)

{

trainPos=1.8;

}

else if(trainPos>1.8)

{

trainPos=-1.8;

}

if(rainPos<-.001)

{

rainPos=.5;

}

if(rainPosX<-0.1)

{

rainPosX=.05;

}

if(vrain)

{

rainPos-=rainSpeed;

rainPosX-=rainSpeedX;

}

glutPostRedisplay();

glutTimerFunc(100, update, 0);

}

void cloudAnimation1(int value)

{

if(positionOfCloud1 < -1.7f)

positionOfCloud1 = 0.6f;

positionOfCloud1 -= speedOfCloud1;

glutPostRedisplay();

glutTimerFunc(40, cloudAnimation1, 0);

}

void cloudAnimation2(int value)

{

if(positionOfCloud2 < -0.6f)

positionOfCloud2 = 1.6f;

positionOfCloud2 -= speedOfCloud2;

glutPostRedisplay();

glutTimerFunc(40, cloudAnimation2, 0);

}

void daysky()

{

glColor3ub(100, 160, 210);

glBegin(GL\_QUADS);

glColor4f(1.0f, 1.0f, 1.0f, 1.0f);

glVertex2f(-1.0,0);

glVertex2f(1.0,0);

glColor4f(0.6f, 0.8f, 1.0f, 1.0f);

glVertex2f(1.0,1.0);

glVertex2f(-1.0,1.0);

glEnd();

}

void nightsky()

{

glColor3ub(13, 42, 67);

glBegin(GL\_QUADS);

//glColor3f(0.0f, 0.0f, 0.0f);

glVertex2f(-1.0,0);

glVertex2f(1.0,0);

glVertex2f(1.0,1.0);

glVertex2f(-1.0,1.0);

glEnd();

}

void birdAnimation(int value)

{

if(positionOfBird > 1.1f)

positionOfBird = -1.1f;

positionOfBird += speedOfBird;

glutPostRedisplay();

glutTimerFunc(20, birdAnimation, 0);

}

void bird()//blue bird

{

glPushMatrix();

glTranslatef(positionOfBird, 0.0f, 0.0f);

int i;

GLfloat mmm=0.062f; GLfloat nnn=.801f; GLfloat radiusmmm =.01f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(0, 0, 255);

glVertex2f(mmm, nnn); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

mmm + (radiusmmm \* cos(i \* twicePi / triangleAmount)),

nnn + (radiusmmm \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(-0.02f,0.8f);

glVertex2f(-0.01f,0.79f);

glVertex2f(0.0f,0.78f);

glVertex2f(0.04f,0.77f);

glVertex2f(0.07f,0.79f);

glVertex2f(0.081f,0.8f);

glEnd();

glBegin(GL\_TRIANGLES);

glVertex2f(0.055f,0.8f);

glVertex2f(0.03f,0.8f);

glVertex2f(0.02f,0.84f);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(242, 242, 242 );

glVertex2f(0.055f,0.8f);

glVertex2f(0.024f,0.8f);

glVertex2f(0.0f,0.83f);

glEnd();

glPopMatrix();

}

void stars()

{

glPointSize( 2.5 );

glBegin(GL\_POINTS);

glColor3ub(247, 240, 188);

glVertex2f(0.8f,0.95f);

glVertex2f(0.9f,0.9f);

glVertex2f(0.95f,0.7f);

glVertex2f(0.8f,0.8f);

glVertex2f(0.7f,0.9f);

glVertex2f(0.6f,0.8f);

glVertex2f(0.5f,0.75f);

glVertex2f(0.4f,0.9f);

glVertex2f(0.3f,0.7f);

glVertex2f(0.25f,0.9f);

glVertex2f(0.25f,0.7f);

glVertex2f(0.1f,0.9f);

glVertex2f(0.15f,0.75f);

glVertex2f(0.0f,0.8f);

glVertex2f(-0.7f,0.9f);

glVertex2f(-0.8f,0.95f);

glVertex2f(-0.4f,0.8f);

glVertex2f(-0.8f,0.95f);

glVertex2f(-0.9f,0.9f);

glVertex2f(-0.95f,0.7f);

glVertex2f(-0.8f,0.8f);

glVertex2f(-0.7f,0.9f);

glVertex2f(-0.6f,0.8f);

glVertex2f(-0.5f,0.75f);

glVertex2f(-0.4f,0.9f);

glVertex2f(-0.3f,0.7f);

glVertex2f(-0.25f,0.9f);

glVertex2f(-0.25f,0.7f);

glVertex2f(-0.15f,0.75f);

glVertex2f(-0.1f,0.9f);

glEnd();

}

void sun()

{

int i;

GLfloat x= -0.84f; GLfloat y=0.88f; GLfloat radius =0.1f;

int triangleAmount = 100;

glColor3f(1.0f, 1.0f, 0.0f);

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(x, y);

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

}

void moon()

{

int i;

GLfloat x= -0.87f; GLfloat y=0.88f; GLfloat radius =0.07f;

int triangleAmount = 100;

glColor3ub(247, 240, 188);

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(x, y);

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

}

void cloud1()

{

glPushMatrix();

glTranslatef(positionOfCloud1, 0.05f, 0.0f);

int i;

GLfloat x=.5f; GLfloat y=.90f; GLfloat radius =.05f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(255, 240, 255);

glVertex2f(x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a=.55f; GLfloat b=.87f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(a, b); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

a + (radius \* cos(i \* twicePi / triangleAmount)),

b + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c=.45f; GLfloat d=.87f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

c + (radius \* cos(i \* twicePi / triangleAmount)),

d + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e=.52f; GLfloat f=.84f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e + (radius \* cos(i \* twicePi / triangleAmount)),

f+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=.6f; GLfloat h=.86f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

g + (radius \* cos(i \* twicePi / triangleAmount)),

h+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glPopMatrix();

}

void cloud2()

{

glPushMatrix();

glTranslatef(positionOfCloud2, -0.02f, 0.0f);

int i;

GLfloat x=-.5f; GLfloat y=.84f; GLfloat radius =.05f;

int triangleAmount = 20;

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(255, 240, 255);

glVertex2f(x, y); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / triangleAmount)),

y + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat a=-.55f; GLfloat b=.81f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(a, b); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

a + (radius \* cos(i \* twicePi / triangleAmount)),

b + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat c=-.45f; GLfloat d=.81f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(c, d); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

c + (radius \* cos(i \* twicePi / triangleAmount)),

d + (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat e=-.52f; GLfloat f=.78f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(e, f); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

e + (radius \* cos(i \* twicePi / triangleAmount)),

f+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

GLfloat g=-.6f; GLfloat h=.80f;

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(g, h); // center of circle

for(i = 0; i <= triangleAmount;i++) {

glVertex2f(

g + (radius \* cos(i \* twicePi / triangleAmount)),

h+ (radius \* sin(i \* twicePi / triangleAmount))

);

}

glEnd();

glPopMatrix();

}

void myDisplay1(void)

{

glClearColor(0.53f,0.81f,0.92f,0.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

daysky();

sun();

cloud1();

cloud2();

bird();

glFlush();

}

void myDisplay2(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

nightsky();

stars();

moon();

glFlush();

}

void backGround()

{

//---------------------------------------------SKY

/\* if(! night){glColor3ub(100, 160, 210);}

else{ glColor3ub(13, 42, 67);}\*/

//quad( -1,-1,1,-1 ,1,1 ,-1,1 );

//----------------------------------green ->back

glColor3ub(4,88,50);

triangle( .2,-.1,1,-.1 ,.6,.35 );

//----------------------------------green ->front

glColor3ub(83, 112, 46);

glBegin(GL\_POLYGON);

glVertex2f(-.8,0);

glVertex2f(-1,0);

glVertex2f(-1,-1);

glVertex2f(1,-1);

glVertex2f(1,0);

glVertex2f(.4,0);

glVertex2f(-.3,.35);

glEnd();

}

void tent()

{

//----------------------------------------------- Large

glColor3ub (211, 47, 47);

quad(0.193, 0.051,0.193, -0.076,.421, -.076,.421, .051);

glColor3ub (230, 0, 5);

triangle(0.152, 0.051,0.46, 0.051,.309, 0.18 );

glPushMatrix();//large wall strip->right

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031-0.031-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031-0.031-0.031-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031-0.031-0.031-0.031-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glPushMatrix();

glTranslated(-0.031-0.031-0.031-0.031-0.031-0.031,0,0);

glColor3ub (244,143,177);

quad( 0.392, 0.052,0.392, -0.076,.408, -.076,.408, .052 );

glPopMatrix();

glColor3ub (0, 0, 0);

line( 0.152,.051,0.46, 0.051 );

glColor3ub (244,143,177);//roof strip->right

triangle( 0.384, 0.051,0.422, 0.051 , .309, 0.18);

glColor3ub (244,143,177);

triangle(0.322, 0.051 ,0.345, 0.051 ,.309, 0.18 );

glColor3ub (244,143,177);

triangle(0.27, 0.051 ,0.294, 0.051 , .309, 0.18);

glColor3ub (244,143,177);

triangle( 0.197, 0.051, 0.234, 0.051 ,.309, 0.18 );

glColor3ub (120,40,31);//tent gate

triangle( 0.277, -0.076,0.34, -0.076 ,.3085, 0.0228 );

}

void gate()

{

glPushMatrix();

glScaled(.9,.7,1);

glTranslated(-.1,-.22,0);

//--------------------------left

glPushMatrix();

glScaled(1,2,1);

glColor3ub(230,196,38);

quad( -.415, -.37,-.402, -.37 ,-.402, -.293 ,-.415, -.293 );

quad( -.390, -.37,-.373, -.37 ,-.373, -.316 ,-.390, -.316 );

quad( -.360, -.37, -.334, -.37,-.334, -.138 , -.360, -.138);

quad( -.407, -.31, -.355, -.32,-.355, -.31 ,-.407, -.32 );

glPopMatrix();

//-------------------------right

glPushMatrix();

glScaled(1,2,1);

glRotated(180,0,1,0);

glTranslated(.25,0,0);

glColor3ub(230,196,38);

quad( -.415, -.37,-.402, -.37 ,-.402, -.293 ,-.415, -.293 );

quad( -.390, -.37,-.373, -.37 ,-.373, -.316 ,-.390, -.316 );

quad( -.360, -.37,-.334, -.37 ,-.334, -.138 ,-.360, -.138 );

quad( -.407, -.31,-.355, -.32 ,-.355, -.31 ,-.407, -.32 );

glPopMatrix();

//-------------------------frame

quad( -.345, -.37,.1, -.37 ,.1, -.2 ,-.345, -.2 );

glColor3ub(0,140,190);

quad( -.345, -.37, .1, -.37,.1, -.2 ,-.345, -.2 );

glPushMatrix();

printText(-.2, -.3,carnival);

glPopMatrix();

//--------------------------fence

glPushMatrix();

glColor3ub(230,196,38);

quad( -1,-.65,-1,-.67,-.4,-.67,-.4,-.65);

quad( .15,-.65,.15,-.67,.6,-.67,.6,-.65);

glPopMatrix();

glPushMatrix();

glTranslated(0,-.05,0);

glColor3ub(230,196,38);

quad( -1,-.65,-1,-.67,-.4,-.67,-.4,-.65);

quad( .15,-.65,.15,-.67,.6,-.67,.6,-.65);

glPopMatrix();

glPushMatrix();// fence front corner

glTranslated(.95,-.37,0);

quad( -.360, -.37, -.334, -.37,-.334, -.138 , -.360, -.138);

glPopMatrix();

glPushMatrix();// fence back corner

glScaled(1,.51,1);

glTranslated(.935,.6,0);

quad( -.360, -.37, -.334, -.37,-.334, -.138 , -.360, -.138);

glPopMatrix();

glPushMatrix();

glTranslated(.1,0,0);

quad(.5,-.55,.51,-.55,.49,.2,.48,.2);

glPopMatrix();

glPopMatrix();

}

void rain()

{

float x=-1.0;

float temp=0.9;

for (int j=0;j<20;j++)

{

float y=temp;

for (int l=0;l<30;l++)

{

glPushMatrix();

glTranslatef(rainPosX,rainPos,0.0f);

glBegin(GL\_LINES);

glColor3ub(222, 222, 222);

glVertex2f(x,y);

x+=.05;

y+=.1;

glVertex2f(x,y);

glEnd();

y=temp;

x=x+.1;

glPopMatrix();

}

temp-=.2;x=-1;

}

glEnd();

}

void Ttree()

{

glPushMatrix();

glScaled(.6,.6,1);

glBegin(GL\_QUADS);

glColor3ub(79, 38, 56);

glVertex2f(-0.01f,0.4f);

glVertex2f(-0.01f,0.0f);

glVertex2f(0.01f,0.0f);

glVertex2f(0.01f,0.4f);

glEnd();

glColor3ub(82, 194, 41);

glBegin(GL\_TRIANGLES);

glVertex2f(0.0f,0.4f);

glVertex2f(-0.15f,0.3f);

glVertex2f(0.15f,0.3f);

glEnd();

glPushMatrix();

glTranslatef(0.0f,0.05f,0.0f);

glBegin(GL\_TRIANGLES);

glVertex2f(0.0f,0.45f);

glVertex2f(-0.15f,0.3f);

glVertex2f(0.15f,0.3f);

glEnd();

glPopMatrix();

glPopMatrix();

}

void tree()

{

glPushMatrix();

glScaled(.8,.8,1);

int i;

int lineAmount = 21;

GLfloat x=.0f; GLfloat y=.0f; GLfloat radius =.1f;

//GLfloat radius = 0.8f; //radius

GLfloat twicePi = 2.0f \* PI;

glBegin(GL\_QUADS);

glColor3ub(108, 48, 23 );

glVertex2f(0.75f,-0.7f);

glVertex2f(0.78f,-0.7f);

glVertex2f(0.78f,-0.4f);

glVertex2f(0.75f,-0.4f);

glEnd();

x=0.7f; y=-.4f; radius =.1f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(43, 117, 18 );

for(i = 0; i <= lineAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / lineAmount)),

y + (radius\* sin(i \* twicePi / lineAmount))

);

}

glEnd();

x=0.8f; y=-.4f; radius =.1f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(43, 117, 18);

for(i = 0; i <= lineAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / lineAmount)),

y + (radius\* sin(i \* twicePi / lineAmount))

);

}

glEnd();

x=0.75f; y=-.3f; radius =.08f;

glBegin(GL\_TRIANGLE\_FAN);

glColor3ub(43, 117, 18);

for(i = 0; i <= lineAmount;i++) {

glVertex2f(

x + (radius \* cos(i \* twicePi / lineAmount)),

y + (radius\* sin(i \* twicePi / lineAmount))

);

}

glEnd();

glPopMatrix();

}

void character()

{

//---------------------------------------head

glColor3ub(250,220,190);//face

glBegin(GL\_POLYGON);

glVertex2f(.8435,-.0941);

glVertex2f(.8789,-.081);

glVertex2f(.888,-.0);

glVertex2f(.808,-.0);

glVertex2f(.808,-.081);

glEnd();

glColor3ub(240,240,240);// cap

quad( .888,-.01,.808,-.01 ,.82,.05 ,.883,.05 );

glColor3ub(0,0,0);//eye->left

circle(.8577,-.0426,.005);

line(.8477,-.0374,.8707,-.0312);

glPushMatrix();//eye->right

glRotated(180,0,1,0);

glTranslated(-1.68,0,0);

glColor3ub(0,0,0);

circle(.8577,-.0426,.005);

line(.8477,-.0374,.8707,-.0312);

glPopMatrix();

line(.839,-.045,.839,-.0686);//nose

line(.83,-.077,.84,-.08);//mouth

line(.84,-.08,.85,-.077);

glColor3ub(240,240,240);//body

glBegin(GL\_POLYGON);

glVertex2f(.824,-.0941);

glVertex2f(0.816,-.1073);

glVertex2f(.8067,-.3);

glVertex2f(.8947,-.3);

glVertex2f(.878,-.10006);

glEnd();

glColor3ub(25,60,80);// left hand

line(.87,-.145,.81,-.145);

line(.87,-.17,.81,-.17);

glColor3ub(250,220,190);

quad(.79,-.145,.79,-.17,.81,-.17,.81,-.145);

glColor3ub(240,240,240);

quad(.81,-.145,.81,-.17,.87,-.17,.87,-.145);

glPushMatrix();

glTranslated(-.02,.03,0);

glColor3ub(25,60,80);//right hand

line(.87,-.145,.81,-.145);

line(.87,-.17,.81,-.17);

glColor3ub(250,220,190);

quad(.79,-.145,.79,-.17,.81,-.17,.81,-.145);

glColor3ub(240,240,240);

quad(.81,-.145,.81,-.17,.87,-.17,.87,-.145);

glPopMatrix();

glColor3ub(25,60,80);//leg

quad(.83,-.3,.83,-.37,.85,-.37,.85,-.3);

quad(.86,-.3,.86,-.37,.88,-.37,.88,-.3);

glColor3ub(240,240,240);//shoe

quad(.825,-.34,.825,-.37,.85,-.37,.85,-.34);

quad(.855,-.34,.855,-.37,.88,-.37,.88,-.34);

}

void mosque()

{

glScaled(1.3,1.4,1);

glTranslated(-.18,0,0);

glPushMatrix();

glColor3ub(210,200,160);

glTranslated(0,-.02,0);

quad(.63,-.02,.63,-.113,.948,-.113,.948,-.02);

glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

//glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd left window

glPopMatrix();

glPushMatrix();

glTranslated(.2,0,0);

//glColor3ub(60,120,60);//right window

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

//glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd right window

glPopMatrix();

glPopMatrix();

quad( .77,-.03 ,.77 ,-.113 ,.8 ,-.113 ,.8 ,-.03 );//door

glColor3ub(60,120,60);

quad(.62,.01,.63,-.02,.948,-.02,.959,.01);

glPopMatrix();

glPushMatrix();

glColor3ub(210,200,160);

glScaled(.8,1,1);

glTranslated(.2,.104,0);

quad(.63,-.02,.63,-.113,.948,-.113,.948,-.02);

glColor3ub(60,120,60);//left window

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd left window

glPopMatrix();

glPushMatrix();

glTranslated(.2,0,0);

glColor3ub(60,120,60);//right window

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd right window

glPopMatrix();

glPopMatrix();

glColor3ub(60,120,60);

quad(.62,.01,.63,-.02,.948,-.02,.959,.01);

glPopMatrix();

glPushMatrix();//---------------------minaret

glColor3ub(210,200,160);

glScaled(.6,1,1);

glTranslated(.53,.225,0);

quad(.63,-.02,.63,-.113,.948,-.113,.948,-.02);

if(night) //left window

{

glColor3ub(247, 240, 188);

}

else

{

glColor3ub(60,120,60);

}//left window

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

//glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd left window

glPopMatrix();

glPushMatrix();

glTranslated(.2,0,0);

//glColor3ub(60,120,60);//right window

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );

glPushMatrix();

glTranslated(.05,0,0);

//glColor3ub(60,120,60);

quad( .64,-.038 ,.64 ,-.094 ,.68 ,-.094 ,.68 ,-.038 );// 2nd right window

glPopMatrix();

glPopMatrix();

glColor3ub(60,120,60);

quad(.62,.01,.63,-.02,.948,-.02,.959,.01);

glPopMatrix();

if(night){glColor3ub(130,178,65);}else{glColor3ub(150,190,65);}

glBegin(GL\_POLYGON);

glVertex2f( .893,.235 );

glVertex2f( .9,.26 );

glVertex2f( .9 ,.32 );

glVertex2f( .789 ,.4 );//top

glVertex2f( .682 ,.32 );

glVertex2f( .682 ,.26 );

glVertex2f( .689 ,.235);

glEnd();

//---------------------pillar left

glPushMatrix();

glPushMatrix();

glColor3ub(210,200,160);

glScaled(.1,5,1);

glTranslated(5.6,.111,0);

quad(.63,-.02,.63,-.113,.948,-.113,.948,-.02);

glPopMatrix();

glPushMatrix();

glScaled(.1,.4,1);

glTranslated(5.6,1.15,0);

glColor3ub(60,120,60);

quad(.62,.01,.63,-.02,.948,-.02,.959,.01);//under minaret

glPopMatrix();

glPushMatrix();

glScaled(.17,.4,1);

glTranslated(2.97,.925,0);

if(night){glColor3ub(130,178,65);}else{glColor3ub(150,190,65);}//minaret left

glBegin(GL\_POLYGON);

glVertex2f( .893,.235 );

glVertex2f( .9,.26 );

glVertex2f( .9 ,.32 );

glVertex2f( .789 ,.4 );//top

glVertex2f( .682 ,.32 );

glVertex2f( .682 ,.26 );

glVertex2f( .689 ,.235);

glEnd();

glPopMatrix();

glPopMatrix();

//--------------------------------pillar right

glPushMatrix();

glTranslated(.305,0,0);

glPushMatrix();

glColor3ub(210,200,160);

glScaled(.1,5,1);

glTranslated(5.6,.111,0);

quad(.63,-.02,.63,-.113,.948,-.113,.948,-.02);

glPopMatrix();

glPushMatrix();

glScaled(.1,.4,1);

glTranslated(5.6,1.15,0);

glColor3ub(60,120,60);

quad(.62,.01,.63,-.02,.948,-.02,.959,.01);//under minaret

glPopMatrix();

glPushMatrix();

glScaled(.17,.4,1);

glTranslated(2.97,.925,0);

if(night){glColor3ub(130,178,65);}else{glColor3ub(150,190,65);}//minaret left

glBegin(GL\_POLYGON);

glVertex2f( .893,.235 );

glVertex2f( .9,.26 );

glVertex2f( .9 ,.32 );

glVertex2f( .789 ,.4 );//top

glVertex2f( .682 ,.32 );

glVertex2f( .682 ,.26 );

glVertex2f( .689 ,.235);

glEnd();

glPopMatrix();

glPopMatrix();

glPushMatrix();

glColor3ub(200,143,93);

glTranslated(.12,0,0);

quad(.435,-.61,.515,-.61,.5,.0,.45,.0);

glPopMatrix();

}

void trainLine()

{

glPushMatrix();

glTranslated(0,-.455,0);

glPushMatrix();

glScaled(1,10,1);

glTranslated(0,.448,0);

glColor3ub(128,116,120);

quad( -1,-.5,-1,-.488,1,-.488,1,-.5);

glPopMatrix();

glColor3ub(82,55,48);

quad( -1,-.5,-1,-.488,1,-.488,1,-.5);

glPushMatrix();

glTranslated(0,.06,0);

quad( -1,-.488,-1,-.5,1,-.5,1,-.488);

glPopMatrix();

//quad(-.99,-.48,-.99,);

glPopMatrix();

}

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

{

switch (key) {

case 'm':

anglel\_M += speed\_M;

speed\_M+=0.75;

update(0);

break;

case 'M':

speed\_M=0.0;

break;

case 'b':

speed\_N+=0.4;

update(0);

break;

case 'B':

speed\_N = 0.0;

break;

//Train Start

case 'w':

trainPos-=trainSpeed;

update(0);

break;

case 's':

trainPos+=trainSpeed;

update(0);

break;

case 'g':

PlaySound("trainsound.wav",NULL,SND\_ASYNC | SND\_LOOP | SND\_FILENAME);

break;

case 'G':

PlaySound(NULL,NULL,SND\_ASYNC | SND\_LOOP | SND\_FILENAME);

break;

case 'n':

night=true;

break;

case 'N':

night=false;

break;

case 'r':

vrain=true;

break;

case 'R':

vrain=false;

break;

case 'e':

exit(0);

break;

}

}

void iceCart()

{

glColor3ub(100, 30, 22);

quad(-0.8245,-0.3044,-0.8245,-0.3088,-0.8009,-0.3088,-0.8009,-0.3044);

glColor3ub(110, 44, 0);

quad(-0.9227,-0.2979,-0.9149,-0.3052,-0.8247,-0.3052,-0.8175,-0.2979);

glColor3ub(160, 64, 0);

glBegin(GL\_POLYGON);

glVertex2f(-0.9149,-0.3052);

glVertex2f(-0.9149,-0.3181);

glVertex2f(-0.8998,-0.3343);

glVertex2f(-0.8433,-0.3341);

glVertex2f(-0.8247,-0.3181);

glVertex2f(-0.8247,-0.3052);

glEnd();

glColor3ub(250, 215, 160);

glBegin(GL\_POLYGON);

glVertex2f(-0.9035,-0.2536);

glVertex2f(-0.9125,-0.262);

glVertex2f(-0.9125,-0.2979);

glVertex2f(-0.8274,-0.2979);

glVertex2f(-0.8274,-0.262);

glVertex2f(-0.835,-0.2536);

glEnd();

glColor3ub(236, 240, 241);

quad(-0.9083,-0.2645,-0.9083,-0.2951,-0.8346,-0.2951,-0.8346,-0.2645);

glColor3ub(245, 176, 65);

triangle(-0.895,-0.273,-0.8573,-0.2603,-0.868,-0.2443);

glPushMatrix();

glColor3ub(125, 60, 152);

glTranslatef(-0.8539,-0.2455, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.0095;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glColor3ub(93, 173, 226);

glTranslatef(-0.8581,-0.2589, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.0095;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glColor3ub(236, 64, 122);

glTranslatef(-0.8678,-0.2448, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.0095;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//WHEELS

glPushMatrix();

glColor3ub(39, 55, 70);

glTranslatef(-0.843f,-0.3343f, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.015;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glColor3ub(121, 125, 127 );

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.0057;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

glPushMatrix();

glColor3ub(39, 55, 70);

glTranslatef(-0.899f,-0.3343f, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.015;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glColor3ub(121, 125, 127 );

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.0057;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

}

void item()

{

glPushMatrix();

//BOAT

glTranslatef(-0.4363f,0.0185f, 0.0f);

glRotatef(anglel\_N, 1.0, 0.0, 1.0);

//Main Boat

glColor3ub(255, 87, 34);

triangle(-0.0816,0.0214,-0.1293,0.034,-0.0816,-0.0123);

quad(-0.0816,0.0086,-0.0816,-0.0123,0.0772,-0.0123,0.0772,0.0086);

triangle(0.0772,-0.0123,0.1255,0.034,0.0772,0.0233);

quad(0.053,0.0248,0.0271,0.0248,0.0271,0.0086,0.053,0.0086);

quad(-0.0255,0.0248,-0.0508,0.0248,-0.0508,0.0086,-0.0255,0.0086);

triangle(0,0.043,-0.0093,0.0086,0.009,0.0086);

//BARS

glColor3ub(62, 39, 35);

triangle(0,0.1919,-0.0402,0.0248,-0.0304,0.0248);

triangle(0,0.1919,0.0351,0.0248,0.0447,0.0248);

glPopMatrix();

glColor3ub(121, 125, 127);

triangle(-0.4348,0.2,-0.5555,-0.018,-0.54,-0.018);

triangle(-0.4348,0.2,-0.3278,-0.018,-0.3123,-0.018);

glPushMatrix();

glColor3ub(216, 67, 21);

glTranslatef(-0.4348f,0.18, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.03;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//MARRY GO ROUND

glPushMatrix();

glTranslatef(-0.82,0.142f, 0.0f);

glRotatef(anglel\_M, 0.0, 0.0, 1.0);

//MARRY GO ROUND

glColor3ub(60,120,60);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.11;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glColor3ub(60,120,60);

glBegin(GL\_LINE\_LOOP);

glVertex2f(0.0,0.116);

glVertex2f(-0.12,0);

glVertex2f(0,-0.117);

glVertex2f(0.12,0);

glEnd();

//bars

glColor3ub(60,120,60);

quad(-0.0017,0.1289,-0.0017,-0.135,0.0017,-0.135,0.0017,0.1289);

quad(-0.13204,0.0026,-0.13204,0,0.13204,0,0.13204,0.0026);

//BLOCK over ring.

glColor3ub(133, 193, 233);

quad(-0.0135,0.118,-0.0135,0.092,0.0127,0.092,0.0127,0.118);

glColor3ub(231, 76, 60);

quad(-0.1271,0.0142,-0.1271,-0.0123,-0.1005,-0.0123,-0.1005,0.0142);

glColor3ub(244, 208, 63);

quad(0.0985,0.0132,0.0985,-0.0132,0.125,-0.0132,0.125,0.0132);

glColor3ub(142, 68, 173);

quad(-0.0129,-0.0992,-0.0129,-0.1257,0.01337,-0.1257,0.01337,-0.0992);

//AXIS of ROTATION

//glColor3ub(236, 240, 241);

glColor3ub(60,120,60);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.01;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//BASE OF MERRY GO ROUND

glPushMatrix();

glColor3ub(120, 40, 31 );

triangle(-0.82,0.1423,-0.89,-0.0134,-0.85,-0.0134);

triangle(-0.82,0.1423,-0.78,-0.0174,-0.74,-0.0174);

quad(-0.825,0.1423,-0.825,0.1344,-0.81527,0.1344,-0.81527,0.1423);

glPopMatrix();

//TICKET COUNTER

glColor3ub(244, 208, 63);

glPushMatrix();

glTranslatef(-0.656f,-0.02f, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.04;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//ROOF TOP

glColor3ub(244, 208, 63);

glBegin(GL\_POLYGON);

glVertex2f(-0.6046,-0.014);

glVertex2f(-0.6173,-0.005);

glVertex2f(-0.6963,-0.005);

glVertex2f(-0.709,-0.014);

glVertex2f(-0.7246,-0.014);

glVertex2f(-0.7246,-0.03);

glVertex2f(-0.5896,-0.03);

glVertex2f(-0.5896,-0.014);

glEnd();

//ROOF

glColor3ub(211, 47, 47 );

quad(-0.7246,-0.03,-0.7606,-0.0508,-0.5519,-0.0508,-0.5896,-0.03);

glColor3ub(244, 143, 177);

quad(-0.7031,-0.03,-0.7175,-0.0508,-0.59435,-0.0508,-0.61134,-0.03);

glColor3ub(211, 47, 47 );

quad(-0.668,-0.03,-0.6752,-0.0508,-0.6365,-0.0508,-0.6443,-0.03);

glColor3ub(211, 47, 47 );

quad(-0.7508,-0.05,-0.7508,-0.1747,-0.5628,-0.1747,-0.5628,-0.05);

//GLASS

glColor3ub(174, 214, 241);

quad(-0.7408,-0.0527,-0.7408,-0.1153,-0.5723,-0.1153,-0.5723,-0.0527);

glPushMatrix();

glColor3ub(255, 255, 255);

glTranslatef(-0.6578f,-0.1071f, 0.0f);

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/50 ;

float r=0.02;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glPopMatrix();

//BARS IN BOTTOM

glColor3ub(244, 143, 177);

quad(-0.7408,-0.1153,-0.7408,-0.1733,-0.5723,-0.1733,-0.5723,-0.1153);

glColor3ub(211, 47, 47 );

quad(-0.722,-0.1153,-0.722,-0.1733,-0.7034,-0.1733,-0.7034,-0.1153);

quad(-0.684,-0.1153,-0.684,-0.1733,-0.6669,-0.1733,-0.6669,-0.1153);

quad(-0.6471,-0.1153,-0.6471,-0.1733,-0.629,-0.1733,-0.629,-0.1153);

quad(-0.6096,-0.1153,-0.6096,-0.1733,-0.5917,-0.1733,-0.5917,-0.1153);

glColor3ub(211, 47, 47 );

triangle(-0.7606,-0.0508,-0.74,-0.0678,-0.7175,-0.0508);

glColor3ub(244, 143, 177);

triangle(-0.7175,-0.0508,-0.6964,-0.0678,-0.6752,-0.0508);

glColor3ub(211, 47, 47 );

triangle(-0.6752,-0.0508,-0.6554,-0.0678,-0.6365,-0.0508);

glColor3ub(244, 143, 177);

triangle(-0.6365,-0.0508,-0.6121,-0.0678,-0.59435,-0.0508);

glColor3ub(211, 47, 47 );

triangle(-0.59435,-0.0508,-0.5705,-0.0678,-0.5519,-0.0508);

//ICE CART

//ice cream shape

iceCart();

}

void bogy()

{

glPushMatrix();

//body=====================

glBegin(GL\_POLYGON);

glColor3ub(128,15,46);

glVertex2f(.0f,.1f);

glVertex2f(.02f,.12f);

glVertex2f(.02f,.32f);

glVertex2f(-.38f,.32f);

glVertex2f(-.4f,.3f);

glVertex2f(0.0f,0.3f);

glEnd();

glBegin(GL\_QUADS);

glColor3ub(153,170,177);

glVertex2f(-.4f,.3f);

glVertex2f(-.4f,.1f);

glVertex2f(.0f,.1f);

glVertex2f(.0f,.3f);

glEnd();

//window==================

glBegin(GL\_QUADS);

if(night)

{

glColor3ub(247, 240, 188);

}

else

{

glColor3ub(37,47,53);

}

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

//

glPushMatrix();

glTranslated(.07,0,0);

glBegin(GL\_QUADS);

//glColor3ub(200,255,255);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslated(.14,0,0);

glBegin(GL\_QUADS);

//glColor3ub(200,255,255);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslated(.21,0,0);

glBegin(GL\_QUADS);

//glColor3ub(200,255,255);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPushMatrix();

glColor3ub(128,15,46);

glTranslated(0,-.1,0);

glScalef(.4f,.4f,0.0f);

circle(-.7,.5,.1);

glTranslated(0.4f,0.0f,0.0f);

circle(-.7,.5,.1);

glPopMatrix();

glPopMatrix();

}

void completeTrain()

{

glPushMatrix();

glTranslated(trainPos,-1,0);

bogy();

glPushMatrix();

glTranslated(.45,0,0);

bogy();

glPopMatrix();

glPushMatrix();

glTranslated(.9,0,0);

bogy();

glPopMatrix();

glPushMatrix();

glTranslated(-.45,0,0);

bogy();

glPushMatrix();

glTranslated(0,.14,0);

glBegin(GL\_QUADS);

glColor3ub(37,47,53);

glVertex2f(-.35f,.25f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.25f);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslated(.35,-.05,0);

glBegin(GL\_QUADS);

glColor3ub(0,0,0);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslated(.8,-.05,0);

glBegin(GL\_QUADS);

glColor3ub(0,0,0);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPushMatrix();

glTranslated(1.25,-.05,0);

glBegin(GL\_QUADS);

glColor3ub(0,0,0);

glVertex2f(-.35f,.23f);

glVertex2f(-.35f,.18f);

glVertex2f(-.3f,.18f);

glVertex2f(-0.3f,.23f);

glEnd();

glPopMatrix();

glPopMatrix();

glPopMatrix();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

if(night)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

nightsky();

stars();

moon();

}

else

{

glClearColor(0.53f,0.81f,0.92f,0.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

daysky();

sun();

cloud1();

bird();

cloud2();

}

backGround();

glPushMatrix();

glTranslated(-.1,0,0);

tent();

//----------------------------------------Small Left

glPushMatrix();

glScaled(0.33,1,1);

glTranslated(.149,-.15,0);

tent();

glPopMatrix();

//---------------------------------------Small right

glPushMatrix();

glScaled(0.33,1,1);

glTranslated(1.13,-.15,0);

tent();

glPopMatrix();

glPopMatrix();

//--------------------------Ice cart on TENT

glPushMatrix();

glTranslated(.9,0,0);

iceCart();

glPopMatrix();

//---------------------------------------mosque

glPushMatrix();

glTranslated(-.015,0,0);

mosque();

glPopMatrix();

//---------------------------------------gate

glPushMatrix();

gate();

glPopMatrix();

//----------------------------------------TREE

glPushMatrix();

glTranslated(0,-.15,0);

tree();

glPopMatrix();

glPushMatrix();

glTranslated(.3,-.15,0);

tree();

glPopMatrix();

glPushMatrix();

glTranslated(-0.68,0.48,0);

tree();

glPopMatrix();

//tria tree

glPushMatrix();

glTranslated(.38,-.58,0);

Ttree();

glPopMatrix();

glPushMatrix();

glTranslated(.15,-.58,0);

Ttree();

glPopMatrix();

glPushMatrix();

glTranslated(-.6,-.6,0);

Ttree();

glPopMatrix();

//////////

if(!night && !vrain)

{

//--------------------------------------character right

glPushMatrix();

glTranslated(.15,0,0);

glScaled(.8,.8,1);

character();

//--------------------------------------character left

glPushMatrix();

glRotated(180,0,1,0);

glTranslated(-1.6,0.005,0);

character();

glPopMatrix();

glPopMatrix();

}

//-------------------------------------------LABU

glPushMatrix();

glScaled(1.3,1.7,1);

glTranslated(.2,0,0);

item();

glPopMatrix();

//-------------------------------Train

trainLine();

glPushMatrix();

completeTrain();

glPopMatrix();

glPushMatrix();

if(vrain)

{

rain();

}

glPopMatrix();

glutSwapBuffers();

glFlush();

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

cout<<"\n Press 'm' for rotating the Marry Go Round.\n";

cout<<"\n Press 'Shift m' for stop the rotation of Marry Go Round.\n";

cout<<"\n Press 'b' for swinging the Boat.\n";

cout<<"\n Press 'Shift b' for stop the swinging.\n";

cout<<"\n Press 'n' for Night mood.\n";

cout<<"\n Press 'Shift n' for Day mood.\n";

cout<<"\n Press 'w' for Train move forward.\n";

cout<<"\n Press 's' for Train move backward.\n";

cout<<"\n Press 'g' to stop train sound.\n";

cout<<"\n Press 'e' for exit.\n";

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(1800, 900);

glutInitWindowPosition(50, 50);

glutCreateWindow("Carnival");

glutDisplayFunc(display);

glutTimerFunc(40, cloudAnimation1, 0);

glutTimerFunc(40, cloudAnimation2, 0);

glutTimerFunc(20, birdAnimation, 0);

glutKeyboardFunc(keyboardHandle);

glutTimerFunc(100, update, 0);

glutMainLoop();

return 0;

}