#include<GL/glut.h>

#include<stdio.h>

#include<stdlib.h>

void timer(int);

int j;

//void init();

GLfloat x=220.0,y=490.0,b=850;

void myDisplay();

void reshape(int ,int);

void myCrt();

void delay();

void rendersecene();

void title();

void init();

void pts(int);

void colorMenu(int id);

void mykey(unsigned char key, int x, int y);

float color[5][3]={{1.0,1.0,1.0},{1.0,0.2,0.7,},{1.0,1.0,0.0},{0.2,1.0,0.4},{0.0,1.0,1.0}};

GLfloat clo[5][3]={{1.0,0.0,0.0},{1.0,0.0,1.0},{0.0,0.0,1.0},{0.0,1.0,1.0},{1.0,1.0,0.0}};

int color\_Menu,flag=1,i;

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_RGB|GLUT\_DOUBLE);

glutInitWindowSize(5000,5000);

glutInitWindowPosition(0,0);

glutCreateWindow("Cathode Ray Tube");

glutKeyboardFunc(mykey);

glutCreateMenu(colorMenu);

glutAddMenuEntry("START",1);

glutAddMenuEntry("STOP",2);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

// init();

glutDisplayFunc(rendersecene);

glutReshapeFunc(reshape);

//glutTimerFunc(0,timer,0);

glutMainLoop();

return 0;

}

void delay()

{

int i,j;

for(i=0;i<5000;i++)

for(j=0;j<5000;j++)

;

}

void drawstring(float x,float y,char \*string,int col)

{

char \*c;

glColor3fv(color[col]);

glRasterPos2f(x,y);

for(c=string;\*c!='\0';c++)

{

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24,\*c);

}

}

void title()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0,0.0,0.0);

drawstring(300.0,900.0,"SHREI DHARMASTHALA INSTITUTE OF TECHNOLOGY",1);

drawstring(350.0,850.0,"DEPARTMENT OF COMPUTER SCIENCE",1);

drawstring(300.0,800.0,"COMPUTER GRAPHICS AND VISULIZATION LABORATORY",1);

drawstring(415.0,750.0,"A MINI PROJECT ",3);

drawstring(450.0,700.0,"ON",3);

drawstring(400.0,650.0,"CATHODE RAY TUBE",3);

drawstring(50.0,600.0,"SUBMITED BY :",2);

drawstring(50.0,550.0,"NAME : ROHAN K M , ROHAN R G ",4);

drawstring(50.0,500.0,"UNIVERSITY NUMBER : 4SU16CS089 , 4SU16CS090 ",4);

drawstring(700.0,600.0,"GUIDED BY : ",2);

drawstring(700.0,550.0,"Prof.ARJUN K ",4);

drawstring(700.0,500.0,"Prof.CHANDRAPPA S ",4);

drawstring(425.0,350.0,"(Press Y to Start)",0);

drawstring(425.0,300.0,"(Press Q to quit)",0);

glutSwapBuffers();

//glutSwapBuffers();

}

void rendersecene()

{

if(flag)

{

int d=10;

title();

while(d>0)

{

delay();

d--;

}

flag=0;

}

}

void pts(int i)

{

glLoadIdentity();

glPointSize(5);

glBegin(GL\_POINTS);

glColor3f(1.0,1.0,1.0);

glVertex2f(x+i,y);

glEnd();

glutSwapBuffers();

}

void myDisplay()

{

glBegin(GL\_LINES);

for(i=0;i<=15;i++)

{

if(b>=150)

{

glColor3fv(clo[rand()%4]);

glVertex2f(700,490);

glVertex2f(900,b);

glVertex2f(900,850);

glVertex2f(900,130);

b-=50;

}

else

{

glVertex2f(700,490);

glVertex2f(900,130);

}

}

glEnd();

glutSwapBuffers();

}

void reshape(int w ,int h)

{

glViewport(0,0,w,h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,1000.0,0.0,1000.0);

glMatrixMode(GL\_MODELVIEW);

}

void colorMenu(int id)

{

int i, c,d;

switch(id)

{

case 1:

for(i=0;i<480;i++)

{

pts(i);

if(i==220)

{

glBegin(GL\_QUADS);

glColor3f(1.0,1.0,1.0);

glVertex2f(450,460);

glVertex2f(450,560);

glVertex2f(490,570);

glVertex2f(490,450);

glEnd();

}

if(i==280)

{

glBegin(GL\_QUADS);///plates

glColor3f(1.0,1.0,0.0);

glVertex2f(500,530);

glVertex2f(530,550);

glVertex2f(620,550);

glVertex2f(590,530);

glEnd();

glBegin(GL\_QUADS);///plates

glColor3f(1.0,1.0,0.0);

glVertex2f(500,430);

glVertex2f(530,450);

glVertex2f(620,450);

glVertex2f(590,430);

glEnd();

}

if(i==70)

{

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

{

glPointSize(5);

glBegin(GL\_POINTS);

glColor3fv(clo[rand()%5]);

glVertex2f(335,425);

glVertex2f(360,445);

glVertex2f(353,455);

glVertex2f(375,435);

glVertex2f(325,485);

glVertex2f(345,475);

glVertex2f(375,470);

glVertex2f(375,535);

glVertex2f(335,555);

glVertex2f(348,549);

glVertex2f(356,515);

glEnd();

}

}

}

myDisplay();

break;

case 2:exit(0);

break;

}

}

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

{

if ((key == 'Y' || key == 'y'))

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glClearColor(0.0, 0.0, 0.0, 0.0);

myCrt();

glutSwapBuffers();

// glutSwapBuffers();

}

else if ((key == 'Q' || key == 'q') )

{

exit(0);

}

}

void myCrt()

{

int i;

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);///big box

glColor3f(0.5,0.5,0.5);

glVertex2f(150,380);

glVertex2f(150,600);

glVertex2f(700,600);

glVertex2f(700,380);

glEnd();

drawstring(300.0,900.0,"(Click Right Mouse Button For Options)",0);

glBegin(GL\_QUADS);

glColor3f(0.0,1.0,1.0);

glVertex2f(300,520);

glVertex2f(300,580);

glVertex2f(320,580);

glVertex2f(320,520);

glVertex2f(320,580);

glVertex2f(400,580);

glVertex2f(400,560);

glVertex2f(320,560);

glVertex2f(380,580);

glVertex2f(400,580);

glVertex2f(400,520);

glVertex2f(380,520);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.0,1.0,1.0);

glVertex2f(300,400);

glVertex2f(300,460);

glVertex2f(320,460);

glVertex2f(320,400);

glVertex2f(320,400);

glVertex2f(320,420);

glVertex2f(380,420);

glVertex2f(380,400);

glVertex2f(380,400);

glVertex2f(400,400);

glVertex2f(400,460);

glVertex2f(380,460);

glEnd();

glBegin(GL\_QUADS);///PLATES

glColor3f(0.8,0.8,0.8);

glVertex2f(440,450);

glVertex2f(440,550);

glVertex2f(480,560);

glVertex2f(480,440);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.0,0.0,0.0);

glVertex2f(450,460);

glVertex2f(450,560);

glVertex2f(490,570);

glVertex2f(490,450);

glEnd();

glBegin(GL\_QUADS);///plates

glColor3f(1.0,1.0,1.0);

glVertex2f(500,530);

glVertex2f(530,550);

glVertex2f(620,550);

glVertex2f(590,530);

glEnd();

glBegin(GL\_LINES);//////////////////////////////////screen////////////////

glColor3f(1.0,1.0,1.0);

glVertex2f(900,850);

glVertex2f(900,130);

glEnd();

glBegin(GL\_QUADS);///plates

glColor3f(1.0,1.0,1.0);

glVertex2f(500,430);

glVertex2f(530,450);

glVertex2f(620,450);

glVertex2f(590,430);

glEnd();

glBegin(GL\_LINES);//side upper cross line

glColor3f(1.0,1.0,1.0);

glVertex2f(700,600);

glVertex2f(900,850);

glEnd();

glBegin(GL\_LINES);//side lower cross line

glColor3f(1.0,1.0,1.0);

glVertex2f(700,380);

glVertex2f(900,130);

glEnd();

////////this is between big box and connector

glBegin(GL\_QUADS);

glColor3f(1.0,0.0,0.0);

glVertex2f(100,380);

glVertex2f(100,600);

glVertex2f(150,600);

glVertex2f(150,380);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.0,0.0,1.0);

glVertex2f(150,460);

glVertex2f(150,520);

glVertex2f(220,520);

glVertex2f(220,460);

glEnd();

//////////////////////////////////////////////////////////////////////////////////////connectors

glBegin(GL\_QUADS);

glColor3f(0.6,0.6,0.0);

glVertex2f(50,400);

glVertex2f(50,430);

glVertex2f(100,430);

glVertex2f(100,400);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.6,0.6,0.0);

glVertex2f(50,450);

glVertex2f(50,480);

glVertex2f(100,480);

glVertex2f(100,450);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.6,0.6,0.0);

glVertex2f(50,500);

glVertex2f(50,530);

glVertex2f(100,530);

glVertex2f(100,500);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.6,0.6,0.0);

glVertex2f(50,550);

glVertex2f(50,580);

glVertex2f(100,580);

glVertex2f(100,550);

glEnd();

glutSwapBuffers();

//glutSwapBuffers();

}