#include <iostream>

#include <Windows.h>

#include <GL\glew.h>

#include <GL\freeglut.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

#include <cmath>

#include <string>

#define STB\_IMAGE\_IMPLEMENTATION

#include"stb\_image.h"

unsigned int bg2;

unsigned int bg3;

unsigned int bg4;

unsigned int bg5;

#define GEAR 1

#define PI 3.14159265

int submenu;

int then = 0;

int moveup = 0;

int moveright = 0;

int movedown = 0;

int moveleft = 0;

int flashcolor = 1;

int flashcolor2 = 2;

int flashcolor3 = 3;

int spread2 = 0;

int spread3 = 0;

int spreadall = 0;

int flashtext1 = 1;

int flashtext2 = 1;

int flashtext3 = 1;

int flashtext4 = 1;

int flashtext99 = 1;

GLfloat rotx = 70.0;

GLfloat roty = 180.0;

GLfloat global\_ambient\_on[] = { 0.5f, 0.5f, 0.5f, 1.0f };

GLfloat global\_ambient\_off[] = { 0.5f, 0.5f, 0.5f, 0.0f };

GLfloat dir\_ambient[] = { 0.3, 0.3, 0.3, 1.0 };

GLfloat dir\_diffuse[] = { 0.4, 0.4, 0.4, 1.0 };

GLfloat dir\_specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat dir\_position[] = { 1.0,1.0, 1.0, 0.0 };

GLfloat mat\_specular[] = { 0,0.6,0.6,1.0 };

GLint n\_teeth = 10;

GLfloat t\_radius;

GLfloat g\_rot;

GLfloat g\_speed = .9;

bool inv\_rot = false;

bool light = false;

float toRad(float angle) {

return ((angle)\*PI) / 180;

}

void init(void)

{

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, global\_ambient\_on);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, dir\_ambient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, dir\_diffuse);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, dir\_specular);

glLightfv(GL\_LIGHT0, GL\_POSITION, dir\_position);

glEnable(GL\_LIGHT0);

glEnable(GL\_COLOR\_MATERIAL);

glColorMaterial(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMateriali(GL\_FRONT, GL\_SHININESS, 30);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

glShadeModel(GL\_SMOOTH);

glEnable(GL\_NORMALIZE);

glNewList(GEAR, GL\_COMPILE);

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(0.5, 0.5, 0.5);

glVertex3f(0, 0, -10);

for (float angle = 0; angle <= 360; angle += 0.5)

glVertex3f(60 \* cos(toRad(angle)), 60 \* sin(toRad(angle)), -5);

glEnd();

glBegin(GL\_QUAD\_STRIP);

glColor3f(0.5, 0, 0);

for (float angle = 0; angle <= 360; angle += 0.5) {

glVertex3f(60 \* cos(toRad(angle)), 60 \* sin(toRad(angle)), 10);

glVertex3f(60 \* cos(toRad(angle)), 60 \* sin(toRad(angle)), -10);

}

glEnd();

for (float i = 0; i < n\_teeth;i++) {

t\_radius = (360 / n\_teeth) \* i;

glPushMatrix();

glRotatef(t\_radius, 0, 0, 1);

glTranslatef(0, ((60 \* PI) / 16) + 50, 0);

glScalef((60 \* PI \* 2) / 16, 20, 20);

glutSolidCube(0.9);

glPopMatrix();

}

glEndList();

}

void drawScene() {

glInitNames();

glPushName(0);

glPushMatrix();

//glColor3f(0,0,1);

glRotatef(-180, 0, 0, 0);

glRotatef(-g\_rot, 0, 0, 1);

glLoadName(GEAR);

glCallList(GEAR);

glPopMatrix();

}

void men()

{

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 110, 495 + 90, -180);//change 0 to -180

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

glColor3ub(255, 191, 128);

glPushMatrix();

glTranslatef(540 - 110, 495 + 90, -180);//rotate -180 hade

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540 - 110, 495 + 90);

glVertex2i(540 - 110, 490 + 90); //nose

glVertex2i(531 - 110, 500 + 90);

glVertex2i(537 - 110, 500 + 90);//eyebrow

glVertex2i(543 - 110, 500 + 90);

glVertex2i(549 - 110, 500 + 90);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14 - 110, 494 + 1 + 90);

glVertex2i(540 - 14 - 110, 490 + 1 + 90);

glVertex2i(538 - 14 - 110, 489 + 1 + 90);

glVertex2i(538 - 14 - 110, 495 + 1 + 90);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554 - 110, 495 + 90);

glVertex2i(556 - 110, 496 + 90);

glVertex2i(556 - 110, 491 + 90);

glVertex2i(554 - 110, 490 + 90);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);//change hair color

glVertex2i(527 - 110, 503 + 90);

glVertex2i(553 - 110, 503 + 90);

glVertex2i(547 - 110, 509 + 90);

glVertex2i(533 - 110, 509 + 90);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533 - 110, 498 + 90);

glVertex2i(535 - 110, 498 + 90);

glVertex2i(535 - 110, 496 + 90);

glVertex2i(533 - 110, 496 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545 - 110, 498 + 90);

glVertex2i(547 - 110, 498 + 90);

glVertex2i(547 - 110, 496 + 90);

glVertex2i(545 - 110, 496 + 90);

glEnd();

// mouth

glBegin(GL\_POLYGON);

glVertex2i(535 - 110, 487 + 90);

glVertex2i(540 - 110, 485 + 90);

glVertex2i(545 - 110, 487 + 90);

glVertex2i(540 - 110, 487 + 90);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(55, 150, 70);

glVertex2i(529 - 110, 480 + 90);

glVertex2i(551 - 110, 480 + 90);

glVertex2i(566 - 110, 469 + 90);

glVertex2i(561 - 110, 461 + 90);

glVertex2i(556 - 110, 465 + 90);

glVertex2i(556 - 110, 445 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(524 - 110, 465 + 90);

glVertex2i(519 - 110, 460 + 90);

glVertex2i(514 - 110, 469 + 90);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565 - 110, 468 + 90);

glVertex2i(575 - 110, 453 + 90);

glVertex2i(567 - 110, 454 + 90);

glVertex2i(562 - 110, 462 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 - 110, 453 + 90);

glVertex2i(556 - 110, 438 + 90);

glVertex2i(556 - 110, 445 + 90);

glVertex2i(567 - 110, 454 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515 - 110, 468 + 90);

glVertex2i(505 - 110, 453 + 90);

glVertex2i(513 - 110, 454 + 90);

glVertex2i(518 - 110, 462 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505 - 110, 453 + 90);

glVertex2i(524 - 110, 438 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(513 - 110, 454 + 90);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(150, 112, 30);

glVertex2i(556 - 110, 445 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(524 - 110, 440 + 90);

glVertex2i(524 - 110, 440 + 90);

glVertex2i(556 - 110, 440 + 90);

glEnd();

// collar

glBegin(GL\_POLYGON);

glColor3ub(200, 140, 110 + 90);

glVertex2i(529 - 110, 480 + 90);

glVertex2i(551 - 110, 480 + 90);

glVertex2i(546 - 110, 470 + 90);

glVertex2i(534 - 110, 470 + 90);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(20, 140, 110);

glVertex2i(540 - 110, 477 + 90);

glVertex2i(545 - 110, 470 + 90);

glVertex2i(535 - 110, 470 + 90);

glEnd();

// buttons

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 110, 465 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 110, 458 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 110, 451 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

/// pant

glBegin(GL\_POLYGON);

glColor3ub(180, 80, 230);

glVertex2i(555 - 110, 440 + 90);

glVertex2i(525 - 110, 440 + 90);

glVertex2i(520 - 110, 405 + 90);

glVertex2i(530 - 110, 405 + 90);

glVertex2i(533 - 110, 438 + 90);

glVertex2i(550 - 110, 405 + 90);

glVertex2i(560 - 110, 405 + 90);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 110);

glVertex2i(530 - 110, 405 + 90);

glVertex2i(530 - 110, 396 + 90);

glVertex2i(512 - 110, 396 + 90);

glVertex2i(520 - 110, 405 + 90);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 110);

glVertex2i(550 - 110, 405 + 90);

glVertex2i(550 - 110, 396 + 90);

glVertex2i(568 - 110, 396 + 90);

glVertex2i(560 - 110, 405 + 90);

glEnd();

}

void menred()

{

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 110, 495 + 90, -180);//change 0 to -180

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

if (flashcolor2 % 2 == 0)

{

glColor3ub(242, 29, 29);

}

else

{

glColor3ub(255, 191, 128);

}

flashcolor2++;

glPushMatrix();

glTranslatef(540 - 110, 495 + 90, -180);//rotate -180 hade

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540 - 110, 495 + 90);

glVertex2i(540 - 110, 490 + 90); //nose

glVertex2i(531 - 110, 500 + 90);

glVertex2i(537 - 110, 500 + 90);//eyebrow

glVertex2i(543 - 110, 500 + 90);

glVertex2i(549 - 110, 500 + 90);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14 - 110, 494 + 1 + 90);

glVertex2i(540 - 14 - 110, 490 + 1 + 90);

glVertex2i(538 - 14 - 110, 489 + 1 + 90);

glVertex2i(538 - 14 - 110, 495 + 1 + 90);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554 - 110, 495 + 90);

glVertex2i(556 - 110, 496 + 90);

glVertex2i(556 - 110, 491 + 90);

glVertex2i(554 - 110, 490 + 90);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);//change hair color

glVertex2i(527 - 110, 503 + 90);

glVertex2i(553 - 110, 503 + 90);

glVertex2i(547 - 110, 509 + 90);

glVertex2i(533 - 110, 509 + 90);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533 - 110, 498 + 90);

glVertex2i(535 - 110, 498 + 90);

glVertex2i(535 - 110, 496 + 90);

glVertex2i(533 - 110, 496 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545 - 110, 498 + 90);

glVertex2i(547 - 110, 498 + 90);

glVertex2i(547 - 110, 496 + 90);

glVertex2i(545 - 110, 496 + 90);

glEnd();

// mouth

glBegin(GL\_POLYGON);

glVertex2i(535 - 110, 487 + 90);

glVertex2i(540 - 110, 485 + 90);

glVertex2i(545 - 110, 487 + 90);

glVertex2i(540 - 110, 487 + 90);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(55, 150, 70);

glVertex2i(529 - 110, 480 + 90);

glVertex2i(551 - 110, 480 + 90);

glVertex2i(566 - 110, 469 + 90);

glVertex2i(561 - 110, 461 + 90);

glVertex2i(556 - 110, 465 + 90);

glVertex2i(556 - 110, 445 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(524 - 110, 465 + 90);

glVertex2i(519 - 110, 460 + 90);

glVertex2i(514 - 110, 469 + 90);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565 - 110, 468 + 90);

glVertex3i(575 - 115, 453 + 90, 2);

glVertex3i(567 - 115, 454 + 90, 2);

glVertex2i(562 - 110, 462 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 - 115, 453 + 90);

glVertex3i(556 - 128, 438 + 130, 2);

glVertex3i(556 - 128, 445 + 130, 2);

glVertex2i(567 - 115, 454 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515 - 110, 468 + 90);

glVertex2i(505 - 110, 453 + 90);

glVertex2i(513 - 110, 454 + 90);

glVertex2i(518 - 110, 462 + 90);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505 - 110, 453 + 90);

glVertex2i(524 - 110, 438 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(513 - 110, 454 + 90);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(150, 112, 30);

glVertex2i(556 - 110, 445 + 90);

glVertex2i(524 - 110, 445 + 90);

glVertex2i(524 - 110, 440 + 90);

glVertex2i(524 - 110, 440 + 90);

glVertex2i(556 - 110, 440 + 90);

glEnd();

// collar

glBegin(GL\_POLYGON);

glColor3ub(200, 140, 110 + 90);

glVertex2i(529 - 110, 480 + 90);

glVertex2i(551 - 110, 480 + 90);

glVertex2i(546 - 110, 470 + 90);

glVertex2i(534 - 110, 470 + 90);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(20, 140, 110);

glVertex2i(540 - 110, 477 + 90);

glVertex2i(545 - 110, 470 + 90);

glVertex2i(535 - 110, 470 + 90);

glEnd();

// buttons

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 110, 465 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 110, 458 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 110, 451 + 90, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

/// pant

glBegin(GL\_POLYGON);

glColor3ub(180, 80, 230);

glVertex2i(555 - 110, 440 + 90);

glVertex2i(525 - 110, 440 + 90);

glVertex2i(520 - 110, 405 + 90);

glVertex2i(530 - 110, 405 + 90);

glVertex2i(533 - 110, 438 + 90);

glVertex2i(550 - 110, 405 + 90);

glVertex2i(560 - 110, 405 + 90);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 110);

glVertex2i(530 - 110, 405 + 90);

glVertex2i(530 - 110, 396 + 90);

glVertex2i(512 - 110, 396 + 90);

glVertex2i(520 - 110, 405 + 90);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 110);

glVertex2i(550 - 110, 405 + 90);

glVertex2i(550 - 110, 396 + 90);

glVertex2i(568 - 110, 396 + 90);

glVertex2i(560 - 110, 405 + 90);

glEnd();

}

void woman()

{

//face

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540, 495, -180);

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

glColor3ub(255, 191, 128);

glPushMatrix();

glTranslatef(540, 494, -180);

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540, 494);

glVertex2i(540, 490); //nose

glVertex2i(531, 498);

glVertex2i(532, 499);

glVertex2i(532, 499);

glVertex2i(537, 498);//eyebrow

glVertex2i(549, 498);

glVertex2i(548, 499);

glVertex2i(548, 499);

glVertex2i(543, 498);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14, 494 + 1);

glVertex2i(540 - 14, 490 + 1);

glVertex2i(538 - 14, 489 + 1);

glVertex2i(538 - 14, 495 + 1);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554, 495);

glVertex2i(556, 496);

glVertex2i(556, 491);

glVertex2i(554, 490);

glEnd();

//ear ring right

glBegin(GL\_POLYGON);

glColor3ub(255, 85, 90);

glVertex2i(539 - 14, 492);

glVertex2i(542 - 14, 485);

glVertex2i(536 - 14, 485);

glEnd();

//ear ring left

glBegin(GL\_POLYGON);

glColor3ub(255, 85, 90);

glVertex2i(551, 485);

glVertex2i(555, 492);

glVertex2i(558, 485);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(525, 499);

glVertex2i(549, 509);

glVertex2i(540, 512);

glVertex2i(528, 507);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(540, 507);

glVertex2i(549, 509);

glVertex2i(552, 507);

glVertex2i(555, 499);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533, 496);

glVertex2i(535, 496);

glVertex2i(535, 494);

glVertex2i(533, 494);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545, 496);

glVertex2i(547, 496);

glVertex2i(547, 494);

glVertex2i(545, 494);

glEnd();

//mouth

glBegin(GL\_POLYGON);

glColor3ub(150, 50, 50);

glVertex2i(534, 487);

glVertex2i(540, 484);

glVertex2i(546, 487);

glVertex2i(540, 485);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(160, 150, 250);

glVertex2i(529, 480);

glVertex2i(551, 480);

glVertex2i(566, 469);

glVertex2i(561, 460);

glVertex2i(556, 465);

glVertex2i(556, 445);

glVertex2i(524, 445);

glVertex2i(524, 465);

glVertex2i(519, 460);

glVertex2i(514, 469);

glEnd();

//neck

glBegin(GL\_POLYGON);

glColor3ub(255, 190, 128);

glVertex2i(533, 480);

glVertex2i(547, 480);

glVertex2i(545, 471);

glVertex2i(535, 471);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565, 468);

glVertex2i(575, 453);

glVertex2i(567, 454);

glVertex2i(562, 462);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575, 453);

glVertex2i(556, 438);

glVertex2i(556, 445);

glVertex2i(567, 454);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515, 468);

glVertex2i(505, 453);

glVertex2i(513, 454);

glVertex2i(518, 462);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505, 453);

glVertex2i(524, 438);

glVertex2i(524, 445);

glVertex2i(513, 454);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(10, 120, 130);

glVertex2i(556, 445);

glVertex2i(524, 445);

glVertex2i(524, 440);

glVertex2i(556, 440);

glEnd();

/// leg

glBegin(GL\_POLYGON);

glColor3ub(155, 190, 128);

glVertex2i(555, 440);

glVertex2i(525, 440);

glVertex2i(520, 405);

glVertex2i(530, 405);

glVertex2i(533, 438);

glVertex2i(550, 405);

glVertex2i(560, 405);

glEnd();

//skirt

glBegin(GL\_POLYGON);

//glColor3ub(180,80,90);

glVertex2i(524, 440);

glVertex2i(556, 440);

glVertex2i(566, 410);

glVertex2i(514, 410);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(180, 0, 0);

glVertex2i(530, 405);

glVertex2i(530, 396);

glVertex2i(528, 396);

glVertex2i(528, 404);

glVertex2i(522, 396);

glVertex2i(512, 396);

glVertex2i(520, 405);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(180, 0, 0);

glVertex2i(550, 405);

glVertex2i(550, 396);

glVertex2i(552, 396);

glVertex2i(552, 404);

glVertex2i(558, 396);

glVertex2i(568, 396);

glVertex2i(560, 405);

glEnd();

}

void womanred()

{

//face

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540, 495, -180);

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

if (flashcolor3 % 2 == 0)

{

glColor3ub(242, 29, 29);

}

else

{

glColor3ub(255, 191, 128);

}

flashcolor3++;

glPushMatrix();

glTranslatef(540, 494, -180);

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540, 494);

glVertex2i(540, 490); //nose

glVertex2i(531, 498);

glVertex2i(532, 499);

glVertex2i(532, 499);

glVertex2i(537, 498);//eyebrow

glVertex2i(549, 498);

glVertex2i(548, 499);

glVertex2i(548, 499);

glVertex2i(543, 498);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14, 494 + 1);

glVertex2i(540 - 14, 490 + 1);

glVertex2i(538 - 14, 489 + 1);

glVertex2i(538 - 14, 495 + 1);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554, 495);

glVertex2i(556, 496);

glVertex2i(556, 491);

glVertex2i(554, 490);

glEnd();

//ear ring right

glBegin(GL\_POLYGON);

glColor3ub(255, 85, 90);

glVertex2i(539 - 14, 492);

glVertex2i(542 - 14, 485);

glVertex2i(536 - 14, 485);

glEnd();

//ear ring left

glBegin(GL\_POLYGON);

glColor3ub(255, 85, 90);

glVertex2i(551, 485);

glVertex2i(555, 492);

glVertex2i(558, 485);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(525, 499);

glVertex2i(549, 509);

glVertex2i(540, 512);

glVertex2i(528, 507);

glEnd();

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(540, 507);

glVertex2i(549, 509);

glVertex2i(552, 507);

glVertex2i(555, 499);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533, 496);

glVertex2i(535, 496);

glVertex2i(535, 494);

glVertex2i(533, 494);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545, 496);

glVertex2i(547, 496);

glVertex2i(547, 494);

glVertex2i(545, 494);

glEnd();

//mouth

glBegin(GL\_POLYGON);

glColor3ub(150, 50, 50);

glVertex2i(534, 487);

glVertex2i(540, 484);

glVertex2i(546, 487);

glVertex2i(540, 485);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(160, 150, 250);

glVertex2i(529, 480);

glVertex2i(551, 480);

glVertex2i(566, 469);

glVertex2i(561, 460);

glVertex2i(556, 465);

glVertex2i(556, 445);

glVertex2i(524, 445);

glVertex2i(524, 465);

glVertex2i(519, 460);

glVertex2i(514, 469);

glEnd();

//neck

glBegin(GL\_POLYGON);

glColor3ub(255, 190, 128);

glVertex2i(533, 480);

glVertex2i(547, 480);

glVertex2i(545, 471);

glVertex2i(535, 471);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565, 468);

glVertex3i(575 + 5, 453, 2);

glVertex3i(567 + 5, 454, 2);

glVertex2i(562, 462);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 + 5, 453);

glVertex3i(556 - 18, 438 + 40, 2);

glVertex3i(556 - 18, 445 + 40, 2);

glVertex2i(567 + 5, 454);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515, 468);

glVertex2i(505, 453);

glVertex2i(513, 454);

glVertex2i(518, 462);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505, 453);

glVertex2i(524, 438);

glVertex2i(524, 445);

glVertex2i(513, 454);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(10, 120, 130);

glVertex2i(556, 445);

glVertex2i(524, 445);

glVertex2i(524, 440);

glVertex2i(556, 440);

glEnd();

/// leg

glBegin(GL\_POLYGON);

glColor3ub(155, 190, 128);

glVertex2i(555, 440);

glVertex2i(525, 440);

glVertex2i(520, 405);

glVertex2i(530, 405);

glVertex2i(533, 438);

glVertex2i(550, 405);

glVertex2i(560, 405);

glEnd();

//skirt

glBegin(GL\_POLYGON);

//glColor3ub(180,80,90);

glVertex2i(524, 440);

glVertex2i(556, 440);

glVertex2i(566, 410);

glVertex2i(514, 410);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(180, 0, 0);

glVertex2i(530, 405);

glVertex2i(530, 396);

glVertex2i(528, 396);

glVertex2i(528, 404);

glVertex2i(522, 396);

glVertex2i(512, 396);

glVertex2i(520, 405);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(180, 0, 0);

glVertex2i(550, 405);

glVertex2i(550, 396);

glVertex2i(552, 396);

glVertex2i(552, 404);

glVertex2i(558, 396);

glVertex2i(568, 396);

glVertex2i(560, 405);

glEnd();

}

void lamppost99()

{

//post

glColor3ub(170, 170, 220);

double len = 180;

double thick = 10;

glPushMatrix();

glTranslatef(550 + 55, 520, 70.0);

glScalef(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//sphere

glColor3ub(160, 160, 210);

glPushMatrix();

glLoadIdentity();

glTranslatef(550 + 55, 600, 70);

glutSolidSphere(10, 20, 20);

glPopMatrix();

//bar right

glColor3ub(155, 155, 205);

double len0 = 80;

double thick0 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(585 + 55, 590, 0);

glScalef(len0, thick0, len0);

glutSolidCube(1.0);

glPopMatrix();

//bar left

//bulb right

glColor3f(1, 1, 1.0);

glPushMatrix();

glLoadIdentity();

glTranslatef(612 + 60, 565, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

//lantern right

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(613 + 55, 569, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//bulb left

}

void road()

{

int x, y;

glColor3f(1.0, 1.0, 1.0);

for (x = 0;x < 1000;x = x + 60)

{

glBegin(GL\_POLYGON);

glVertex2f(x, 352.5 + 19);

glVertex2f(x, 357.5 + 19);

glVertex2f(x + 30, 357.5 + 19);

glVertex2f(x + 30, 352.5 + 19);

glEnd();

}

for (y = 650;y > 0;y = y - 60)

{

glBegin(GL\_POLYGON);

glVertex2f(822, y);

glVertex2f(826, y);

glVertex2f(826, y - 30);

glVertex2f(822, y - 30);

glEnd();

}

}

void lamppost1()

{//post

glColor3ub(170, 170, 220);

double len = 180;

double thick = 10;

glPushMatrix();

glTranslatef(650, 520 - 300, 70.0);

glScalef(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//lantern right

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(713, 569 - 300, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//sphere

glColor3ub(160, 160, 210);

glPushMatrix();

glLoadIdentity();

glTranslatef(650, 600 - 300, 70);

glutSolidSphere(10, 20, 20);

glPopMatrix();

//bar right

glColor3ub(155, 155, 205);

double len0 = 80;

double thick0 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(685, 590 - 300, 0);

glScalef(len0, thick0, len0);

glutSolidCube(1.0);

glPopMatrix();

//bar left

glColor3ub(155, 155, 205);

double len1 = 85;

double thick1 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(615, 590 - 300, 70);

glScalef(len1, thick1, len1);

glutSolidCube(1.0);

glPopMatrix();

//lantern left

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(587, 569 - 300, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//bulb right

glColor3f(1, 1, 1);

glPushMatrix();

glLoadIdentity();

glTranslatef(717, 565 - 300, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

//bulb left

glColor3f(1, 1, 1);

glPushMatrix();

glLoadIdentity();

glTranslatef(590, 565 - 300, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

}

void lamppost7()

{//post

glColor3ub(170, 170, 220);

double len = 180;

double thick = 10;

glPushMatrix();

glTranslatef(150, 520 - 300, 70.0);

glScalef(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//lantern right

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(213, 569 - 300, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//sphere

glColor3ub(160, 160, 210);

glPushMatrix();

glLoadIdentity();

glTranslatef(150, 600 - 300, 70);

glutSolidSphere(10, 20, 20);

glPopMatrix();

//bar right

glColor3ub(155, 155, 205);

double len0 = 80;

double thick0 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(185, 590 - 300, 0);

glScalef(len0, thick0, len0);

glutSolidCube(1.0);

glPopMatrix();

//bar left

glColor3ub(155, 155, 205);

double len1 = 85;

double thick1 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(115, 590 - 300, 70);

glScalef(len1, thick1, len1);

glutSolidCube(1.0);

glPopMatrix();

//lantern left

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(87, 569 - 300, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//bulb right

glColor3f(1, 1, 1);

glPushMatrix();

glLoadIdentity();

glTranslatef(217, 565 - 300, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

//bulb left

glColor3f(1, 1, 1);

glPushMatrix();

glLoadIdentity();

glTranslatef(90, 565 - 300, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

}

void lamppost2()

{

//post

glColor3ub(170, 170, 220);

double len = 180;

double thick = 10;

glPushMatrix();

glTranslatef(650 + 293, 520, 70.0);

glScalef(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//lantern right

//sphere

glColor3ub(160, 160, 210);

glPushMatrix();

glLoadIdentity();

glTranslatef(650 + 293, 600, 70);

glutSolidSphere(10, 20, 20);

glPopMatrix();

//bar right

//bar left

glColor3ub(155, 155, 205);

double len1 = 80;

double thick1 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(615 + 293, 590, 70);

glScalef(len1, thick1, len1);

glutSolidCube(1.0);

glPopMatrix();

//lantern left

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(587 + 293, 569, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//bulb left

glColor3f(1, 1, 1.0);

glPushMatrix();

glLoadIdentity();

glTranslatef(591 + 293, 565, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

}

void lamppost4()

{

//post

glColor3ub(170, 170, 220);

double len = 180;

double thick = 10;

glPushMatrix();

glTranslatef(650 + 293, 520 - 300, 70.0);

glScalef(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//lantern right

//sphere

glColor3ub(160, 160, 210);

glPushMatrix();

glLoadIdentity();

glTranslatef(650 + 293, 600 - 300, 70);

glutSolidSphere(10, 20, 20);

glPopMatrix();

//bar right

glColor3ub(155, 155, 205);

double len1 = 80;

double thick1 = 5;

glPushMatrix();

glLoadIdentity();

glTranslatef(615 + 293, 590 - 300, 70);

glScalef(len1, thick1, len1);

glutSolidCube(1.0);

glPopMatrix();

//lantern left

glColor3ub(170, 170, 220);

glPushMatrix();

glLoadIdentity();

glTranslatef(587 + 293, 569 - 300, 0);

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

glutSolidCone(22, 22, 3, 2);

glPopMatrix();

//bulb left

glColor3f(1, 1, 1);

glPushMatrix();

glLoadIdentity();

glTranslatef(591 + 293, 565 - 300, 0);

glutSolidSphere(7, 20, 20);

glPopMatrix();

}

void text1()

{

glBegin(GL\_POLYGON);

if (flashtext1 == 1)

{

glColor3f(0.0, 1.0, 0.0);

flashtext1 = 0;

}

else

{

if (flashtext1 == 0)

{

Sleep(200);

flashtext1 = 2;

}

glColor3f(0.0, 0.0, 0.0);

}

glVertex2i(830 - 665, 220);

glVertex2i(1260 - 665 + 20, 220);

glVertex2i(1260 - 665 + 20, 70);

glVertex2i(830 - 665, 70);

glEnd();

char string2[] = "None Infected By COVID-19 Yet.";

void\* font2 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

int x = -130, y = 120, z = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 80 + y, 70 + z);

for (k = 0;k < strlen(string2);k++)

glutBitmapCharacter(font2, string2[k]);

}

void text2()

{

glBegin(GL\_POLYGON);

if (flashtext2 == 1)

{

glColor3f(255.0, 0.0, 0.0);

flashtext2 = 0;

}

else

{

if (flashtext2 == 0)

{

Sleep(200);

flashtext2 = 2;

}

glColor3f(0.0, 0.0, 0.0);

}

glVertex2i(830 - 665, 220);

glVertex2i(1260 - 665 + 20, 220);

glVertex2i(1260 - 665 + 20, 70);

glVertex2i(830 - 665, 70);

glEnd();

char string2[] = "One person is Affected and becomes COVID-19 Positive";

void\* font2 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

int x = -130, y = 120, z = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 80 + y, 70 + z);

for (k = 0;k < strlen(string2);k++)

glutBitmapCharacter(font2, string2[k]);

}

void text3()

{

glBegin(GL\_POLYGON);

if (flashtext3 == 1)

{

glColor3f(255.0, 0.0, 0.0);

flashtext3 = 0;

}

else

{

if (flashtext3 == 0)

{

Sleep(200);

flashtext3 = 2;

}

glColor3f(0.0, 0.0, 0.0);

}

glVertex2i(830 - 665, 220);

glVertex2i(1260 - 665 + 20, 220);

glVertex2i(1260 - 665 + 20, 70);

glVertex2i(830 - 665, 70);

glEnd();

char string3[] = "Now the count reaches to 2 positives.";

void\* font3 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

int x = -130, y = 120, z = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 80 + y, 70 + z);

for (k = 0;k < strlen(string3);k++)

glutBitmapCharacter(font3, string3[k]);

}

void text4()

{

glBegin(GL\_POLYGON);

if (flashtext4 == 1)

{

glColor3f(255.0, 0.0, 0.0);

flashtext4 = 0;

}

else

{

if (flashtext4 == 0)

{

Sleep(200);

flashtext4 = 2;

}

glColor3f(0.0, 0.0, 0.0);

}

glVertex2i(830 - 665, 220);

glVertex2i(1260 - 665 + 20, 220);

glVertex2i(1260 - 665 + 20, 70);

glVertex2i(830 - 665, 70);

glEnd();

char string3[] = "Further positive cases rises to 3.";

void\* font3 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

int x = -130, y = 120, z = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 80 + y, 70 + z);

for (k = 0;k < strlen(string3);k++)

glutBitmapCharacter(font3, string3[k]);

}

void text99()

{

glBegin(GL\_POLYGON);

if (flashtext99 == 1)

{

glColor3f(0.0, 1.0, 0.0);

flashtext99 = 0;

}

else

{

if (flashtext99 == 0)

{

Sleep(200);

flashtext99 = 2;

}

glColor3f(0.0, 0.0, 0.0);

}

glVertex2i(830 - 665, 220);

glVertex2i(1260 - 665 + 20, 220);

glVertex2i(1260 - 665 + 20, 70);

glVertex2i(830 - 665, 70);

glEnd();

char string2[] = "If you wear a MASK ";

void\* font2 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

int x = -130, y = 120, z = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 80 + y, 70 + z);

for (k = 0;k < strlen(string2);k++)

glutBitmapCharacter(font2, string2[k]);

char string3[] = "and Mantain a Safe Distance,";

void\* font3 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 60 + y, 70 + z);

for (k = 0;k < strlen(string3);k++)

glutBitmapCharacter(font3, string3[k]);

char string4[] = "you can save many Lives\n along with yours.";

void\* font4 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 40 + y, 70 + z);

for (k = 0;k < strlen(string4);k++)

glutBitmapCharacter(font4, string4[k]);

char string5[] = "Social Distancing is the key";

void\* font5 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 20 + y, 70 + z);

for (k = 0;k < strlen(string5);k++)

glutBitmapCharacter(font5, string5[k]);

char string6[] = "So Keep a Safe Distance";

void\* font6 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, 0 + y, 70 + z);

for (k = 0;k < strlen(string6);k++)

glutBitmapCharacter(font6, string6[k]);

char string7[] = "and Wear a MASK always.";

void\* font7 = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3f(815 - 500 + x, -20 + y, 70 + z);

for (k = 0;k < strlen(string7);k++)

glutBitmapCharacter(font7, string7[k]);

}

void drawText(int x, int y, int z, const char\* s)

{

int i = 0;

glColor3f(1.0, 1.0, 1.0);

glRasterPos3i(x, y, z);

for (i = 0; s[i] != '\0'; i++)

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

}

void bus\_stop()

{

/\* ground \*/

glColor3ub(100, 100, 100);

glBegin(GL\_POLYGON);

glVertex3i(340 - 200, 470, -110);

glVertex3i(680 - 200, 470, -110);

glVertex3i(710 - 200, 500, -240);

glVertex3i(370 - 200, 500, -240);

glEnd();

glColor3ub(100, 100, 100);

glBegin(GL\_POLYGON);

glVertex3i(340 - 200, 470, -110);

glVertex3i(679 - 200, 470, -110);

glVertex3i(679 - 200, 450, -110);

glVertex3i(340 - 200, 450, -110);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(680 - 200, 470, -110);

glVertex3i(710 - 200, 500, -240);

glVertex3i(710 - 200, 480, -240);

glVertex3i(680 - 200, 450, -110);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(710 - 200, 500, -240);

glVertex3i(710 - 200, 480, -240);

glVertex3i(370 - 200, 480, -240);

glVertex3i(370 - 200, 500, -240);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(370 - 200, 480, -240);

glVertex3i(370 - 200, 500, -240);

glVertex3i(340 - 200, 470, -110);

glVertex3i(340 - 200, 450, -110);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_LINE\_STRIP);

glVertex3i(340 - 200, 470, -110);

glVertex3i(680 - 200, 470, -110);

glVertex3i(710 - 200, 500, -240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_LINES);

glVertex3i(680 - 200, 470, -110);

glVertex3i(680 - 200, 450, -110);

glEnd();

/\* left \*\*/

glColor3ub(10, 50, 80);

glBegin(GL\_POLYGON);

glVertex3i(370 - 200, 610, -140);

glVertex3i(400 - 200, 625, -200);

glVertex3i(400 - 200, 490, -200);

glVertex3i(370 - 200, 480, -140);

glEnd();

/\* mid \*\*/

glColor3ub(10, 170, 80);

glBegin(GL\_POLYGON);

glVertex3i(395 - 200, 580, -200);

glVertex3i(690 - 200, 580, -200);

glVertex3i(690 - 200, 520, -200);

glVertex3i(395 - 200, 520, -200);

glEnd();

glColor3f(0, 0, 0);

glBegin(GL\_LINES);

glVertex3i(395 - 200, 580, -200);

glVertex3i(690 - 200, 580, -200);

glVertex3i(690 - 200, 520, -200);

glVertex3i(395 - 200, 520, -200);

glEnd();

/\* right \*\*/

glColor3ub(10, 50, 80);

glBegin(GL\_POLYGON);

glVertex3i(690 - 200, 625, -200);

glVertex3i(670 - 200, 610, -140);

glVertex3i(670 - 200, 475, -140);

glVertex3i(690 - 200, 490, -200);

glEnd();

/\* chair \*\*/

glColor3ub(0, 10, 20);

glBegin(GL\_POLYGON);

glVertex3i(425 - 200, 530, -180);

glVertex3i(520 - 200, 530, -180);

glVertex3i(500 - 200, 515, -150);

glVertex3i(405 - 200, 515, -150);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex3i(485 - 200, 515, -163);

glVertex3i(485 - 200, 480, -163);

glVertex3i(495 - 200, 515, -170);

glVertex3i(495 - 200, 487, -170);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex3i(420 - 200, 515, -160);

glVertex3i(420 - 200, 480, -160);

glVertex3i(430 - 200, 515, -170);

glVertex3i(430 - 200, 487, -170);

glEnd();

/\* \* 2nd chair \* \*/

glColor3ub(0, 10, 20);

glBegin(GL\_POLYGON);

glVertex3i(560 - 200, 530, -180);

glVertex3i(655 - 200, 530, -180);

glVertex3i(635 - 200, 515, -150);

glVertex3i(540 - 200, 515, -150);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex3i(560 - 200, 515, -160);

glVertex3i(560 - 200, 480, -160);

glVertex3i(572 - 200, 515, -170);

glVertex3i(572 - 200, 487, -170);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex3i(620 - 200, 515, -163);

glVertex3i(620 - 200, 480, -163);

glVertex3i(630 - 200, 515, -170);

glVertex3i(630 - 200, 487, -170);

glEnd();

/\* upper \*/

glColor3ub(250, 0, 0);

glBegin(GL\_POLYGON);

glVertex3i(350 - 200, 619, -120);

glVertex3i(699 - 200, 619, -120);

glVertex3i(699 - 200, 600, -120);

glVertex3i(350 - 200, 600, -120);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(350 - 200, 620, -120);

glVertex3i(700 - 200, 620, -120);

glVertex3i(720 - 200, 640, -240);

glVertex3i(380 - 200, 640, -240);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(700 - 200, 620, -120);

glVertex3i(720 - 200, 640, -240);

glVertex3i(720 - 200, 620, -240);

glVertex3i(700 - 200, 600, -120);

glEnd();

glBegin(GL\_POLYGON);

glVertex3i(350 - 200, 600, -120);

glVertex3i(350 - 200, 620, -120);

glVertex3i(380 - 200, 640, -240);

glVertex3i(380 - 200, 620, -240);

glEnd();

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_LINES);

glVertex3i(350 - 200, 620, -120);

glVertex3i(700 - 200, 620, -120);

glVertex3i(700 - 200, 620, -120);

glVertex3i(720 - 200, 640, -240);

glVertex3i(700 - 200, 620, -120);

glVertex3i(700 - 200, 600, -120);

glEnd();

drawText(300,605,0,"BUS-STOP");

glFlush();

}

void man1()

{

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 290, 495 + 2, -180);

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

glColor3ub(255, 191, 128);

glPushMatrix();

glTranslatef(540 - 290, 495 + 2, -180);

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

// glVertex2i(540-290,495+2);

// glVertex2i(540-290,490+2); //nose

glVertex2i(531 - 290, 500 + 2);

glVertex2i(537 - 290, 500 + 2);//eyebrow

glVertex2i(543 - 290, 500 + 2);

glVertex2i(549 - 290, 500 + 2);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14 - 290, 494 + 1 + 2);

glVertex2i(540 - 14 - 290, 490 + 1 + 2);

glVertex2i(538 - 14 - 290, 489 + 1 + 2);

glVertex2i(538 - 14 - 290, 495 + 1 + 2);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554 - 290, 495 + 2);

glVertex2i(556 - 290, 496 + 2);

glVertex2i(556 - 290, 491 + 2);

glVertex2i(554 - 290, 490 + 2);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(527 - 290, 503 + 2);

glVertex2i(553 - 290, 503 + 2);

glVertex2i(547 - 290, 509 + 2);

glVertex2i(533 - 290, 509 + 2);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533 - 290, 498 + 2);

glVertex2i(535 - 290, 498 + 2);

glVertex2i(535 - 290, 496 + 2);

glVertex2i(533 - 290, 496 + 2);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545 - 290, 498 + 2);

glVertex2i(547 - 290, 498 + 2);

glVertex2i(547 - 290, 496 + 2);

glVertex2i(545 - 290, 496 + 2);

glEnd();

// mouth

glBegin(GL\_POLYGON);

glColor3ub(0, 102, 130);

glVertex2i(548 - 290, 448 + 48);

glVertex2i(531 - 290, 448 + 48);

glVertex2i(531 - 290, 440 + 48);

//glVertex2i(524-290,440+48);

glVertex2i(548 - 290, 440 + 48);

glEnd();

//beard

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(538 - 290, 480 + 2);

glVertex2i(542 - 290, 480 + 2);

glVertex2i(542 - 290, 484 + 2);

glVertex2i(538 - 290, 484 + 2);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(355, 250, 170);

glVertex2i(529 - 290, 480 + 2);

glVertex2i(551 - 290, 480 + 2);

glVertex2i(566 - 290, 469 + 2);

glVertex2i(561 - 290, 461 + 2);

glVertex2i(556 - 290, 465 + 2);

glVertex2i(556 - 290, 445 + 2);

glVertex2i(524 - 290, 445 + 2);

glVertex2i(524 - 290, 465 + 2);

glVertex2i(519 - 290, 460 + 2);

glVertex2i(514 - 290, 469 + 2);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565 - 290, 468 + 2);

glVertex2i(575 - 290, 453 + 2);

glVertex2i(567 - 290, 454 + 2);

glVertex2i(562 - 290, 462 + 2);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 - 290, 453 + 2);

glVertex2i(556 - 290, 438 + 2);

glVertex2i(556 - 290, 445 + 2);

glVertex2i(567 - 290, 454 + 2);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515 - 290, 468 + 2);

glVertex2i(505 - 290, 453 + 2);

glVertex2i(513 - 290, 454 + 2);

glVertex2i(518 - 290, 462 + 2);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505 - 290, 453 + 2);

glVertex2i(524 - 290, 438 + 2);

glVertex2i(524 - 290, 445 + 2);

glVertex2i(513 - 290, 454 + 2);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(110, 12, 30);

glVertex2i(556 - 290, 445 + 2);

glVertex2i(524 - 290, 445 + 2);

glVertex2i(524 - 290, 440 + 2);

glVertex2i(524 - 290, 440 + 2);

glVertex2i(556 - 290, 440 + 2);

glEnd();

// collar

glBegin(GL\_POLYGON);

glColor3ub(200, 140, 110 + 2);

glVertex2i(529 - 290, 480 + 2);

glVertex2i(551 - 290, 480 + 2);

glVertex2i(546 - 290, 470 + 2);

glVertex2i(534 - 290, 470 + 2);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(20, 140, 110);

glVertex2i(540 - 290, 477 + 2);

glVertex2i(545 - 290, 470 + 2);

glVertex2i(535 - 290, 470 + 2);

glEnd();

// buttons

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 290, 465 + 2, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 290, 458 + 2, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 290, 451 + 2, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

/// pant

glBegin(GL\_POLYGON);

glColor3ub(180, 380, 220);

glVertex2i(555 - 290, 440 + 2);

glVertex2i(525 - 290, 440 + 2);

glVertex2i(520 - 290, 405 + 2);

glVertex2i(530 - 290, 405 + 2);

glVertex2i(533 - 290, 438 + 2);

glVertex2i(550 - 290, 405 + 2);

glVertex2i(560 - 290, 405 + 2);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(400, 10, 10);

glVertex2i(530 - 290, 405 + 2);

glVertex2i(530 - 290, 396 + 2);

glVertex2i(512 - 290, 396 + 2);

glVertex2i(520 - 290, 405 + 2);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(400, 10, 10);

glVertex2i(550 - 290, 405 + 2);

glVertex2i(550 - 290, 396 + 2);

glVertex2i(568 - 290, 396 + 2);

glVertex2i(560 - 290, 405 + 2);

glEnd();

}

void man()

{

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 220, 495 + 76, -180);

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

glColor3ub(255, 191, 128);

glPushMatrix();

glTranslatef(540 - 220, 495 + 76, -180);

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540 - 220, 495 + 76);

glVertex2i(540 - 220, 490 + 76); //nose

glVertex2i(531 - 220, 500 + 76);

glVertex2i(537 - 220, 500 + 76);//eyebrow

glVertex2i(543 - 220, 500 + 76);

glVertex2i(549 - 220, 500 + 76);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14 - 220, 494 + 1 + 76);

glVertex2i(540 - 14 - 220, 490 + 1 + 76);

glVertex2i(538 - 14 - 220, 489 + 1 + 76);

glVertex2i(538 - 14 - 220, 495 + 1 + 76);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554 - 220, 495 + 76);

glVertex2i(556 - 220, 496 + 76);

glVertex2i(556 - 220, 491 + 76);

glVertex2i(554 - 220, 490 + 76);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(527 - 220, 503 + 76);

glVertex2i(553 - 220, 503 + 76);

glVertex2i(547 - 220, 509 + 76);

glVertex2i(533 - 220, 509 + 76);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533 - 220, 498 + 76);

glVertex2i(535 - 220, 498 + 76);

glVertex2i(535 - 220, 496 + 76);

glVertex2i(533 - 220, 496 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545 - 220, 498 + 76);

glVertex2i(547 - 220, 498 + 76);

glVertex2i(547 - 220, 496 + 76);

glVertex2i(545 - 220, 496 + 76);

glEnd();

// mouth

glBegin(GL\_POLYGON);

glVertex2i(535 - 220, 487 + 76);

glVertex2i(540 - 220, 485 + 76);

glVertex2i(545 - 220, 487 + 76);

glVertex2i(540 - 220, 487 + 76);

glEnd();

//beard

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(538 - 220, 480 + 76);

glVertex2i(542 - 220, 480 + 76);

glVertex2i(542 - 220, 484 + 76);

glVertex2i(538 - 220, 484 + 76);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(55, 50, 70);

glVertex2i(529 - 220, 480 + 76);

glVertex2i(551 - 220, 480 + 76);

glVertex2i(566 - 220, 469 + 76);

glVertex2i(561 - 220, 461 + 76);

glVertex2i(556 - 220, 465 + 76);

glVertex2i(556 - 220, 445 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(524 - 220, 465 + 76);

glVertex2i(519 - 220, 460 + 76);

glVertex2i(514 - 220, 469 + 76);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565 - 220, 468 + 76);

glVertex2i(575 - 220, 453 + 76);

glVertex2i(567 - 220, 454 + 76);

glVertex2i(562 - 220, 462 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 - 220, 453 + 76);

glVertex2i(556 - 220, 438 + 76);

glVertex2i(556 - 220, 445 + 76);

glVertex2i(567 - 220, 454 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515 - 220, 468 + 76);

glVertex2i(505 - 220, 453 + 76);

glVertex2i(513 - 220, 454 + 76);

glVertex2i(518 - 220, 462 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505 - 220, 453 + 76);

glVertex2i(524 - 220, 438 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(513 - 220, 454 + 76);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(150, 12, 30);

glVertex2i(556 - 220, 445 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(524 - 220, 440 + 76);

glVertex2i(524 - 220, 440 + 76);

glVertex2i(556 - 220, 440 + 76);

glEnd();

// collar

glBegin(GL\_POLYGON);

glColor3ub(200, 140, 110 + 76);

glVertex2i(529 - 220, 480 + 76);

glVertex2i(551 - 220, 480 + 76);

glVertex2i(546 - 220, 470 + 76);

glVertex2i(534 - 220, 470 + 76);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(20, 140, 110);

glVertex2i(540 - 220, 477 + 76);

glVertex2i(545 - 220, 470 + 76);

glVertex2i(535 - 220, 470 + 76);

glEnd();

// buttons

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 220, 465 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 220, 458 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 220, 451 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

/// pant

glBegin(GL\_POLYGON);

glColor3ub(80, 80, 230);

glVertex2i(555 - 220, 440 + 76);

glVertex2i(525 - 220, 440 + 76);

glVertex2i(520 - 220, 405 + 76);

glVertex2i(530 - 220, 405 + 76);

glVertex2i(533 - 220, 438 + 76);

glVertex2i(550 - 220, 405 + 76);

glVertex2i(560 - 220, 405 + 76);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 10);

glVertex2i(530 - 220, 405 + 76);

glVertex2i(530 - 220, 396 + 76);

glVertex2i(512 - 220, 396 + 76);

glVertex2i(520 - 220, 405 + 76);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 10);

glVertex2i(550 - 220, 405 + 76);

glVertex2i(550 - 220, 396 + 76);

glVertex2i(568 - 220, 396 + 76);

glVertex2i(560 - 220, 405 + 76);

glEnd();

}

void manred()

{

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 220, 495 + 76, -180);

glutSolidTorus(1, 10, 100, 90);

glPopMatrix();

if (flashcolor % 2 == 0)

{

glColor3ub(242, 29, 29);

}

else

{

glColor3ub(255, 191, 128);

}

flashcolor++;

glPushMatrix();

glTranslatef(540 - 220, 495 + 76, -180);

glutSolidTorus(7, 7, 100, 90);

glPopMatrix();

glColor3ub(0, 0, 0);

glBegin(GL\_LINES);

glVertex2i(540 - 220, 495 + 76);

glVertex2i(540 - 220, 490 + 76); //nose

glVertex2i(531 - 220, 500 + 76);

glVertex2i(537 - 220, 500 + 76);//eyebrow

glVertex2i(543 - 220, 500 + 76);

glVertex2i(549 - 220, 500 + 76);//eyebrow

glEnd();

//ear right

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(540 - 14 - 220, 494 + 1 + 76);

glVertex2i(540 - 14 - 220, 490 + 1 + 76);

glVertex2i(538 - 14 - 220, 489 + 1 + 76);

glVertex2i(538 - 14 - 220, 495 + 1 + 76);

glEnd();

//ear left

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(554 - 220, 495 + 76);

glVertex2i(556 - 220, 496 + 76);

glVertex2i(556 - 220, 491 + 76);

glVertex2i(554 - 220, 490 + 76);

glEnd();

//hair

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(527 - 220, 503 + 76);

glVertex2i(553 - 220, 503 + 76);

glVertex2i(547 - 220, 509 + 76);

glVertex2i(533 - 220, 509 + 76);

glEnd();

// eyes

glBegin(GL\_POLYGON);

glVertex2i(533 - 220, 498 + 76);

glVertex2i(535 - 220, 498 + 76);

glVertex2i(535 - 220, 496 + 76);

glVertex2i(533 - 220, 496 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(545 - 220, 498 + 76);

glVertex2i(547 - 220, 498 + 76);

glVertex2i(547 - 220, 496 + 76);

glVertex2i(545 - 220, 496 + 76);

glEnd();

// mouth

glBegin(GL\_POLYGON);

glVertex2i(535 - 220, 487 + 76);

glVertex2i(540 - 220, 485 + 76);

glVertex2i(545 - 220, 487 + 76);

glVertex2i(540 - 220, 487 + 76);

glEnd();

//beard

glBegin(GL\_POLYGON);

glColor3ub(0, 0, 0);

glVertex2i(538 - 220, 480 + 76);

glVertex2i(542 - 220, 480 + 76);

glVertex2i(542 - 220, 484 + 76);

glVertex2i(538 - 220, 484 + 76);

glEnd();

//shirt

glBegin(GL\_POLYGON);

glColor3ub(55, 50, 70);

glVertex2i(529 - 220, 480 + 76);

glVertex2i(551 - 220, 480 + 76);

glVertex2i(566 - 220, 469 + 76);

glVertex2i(561 - 220, 461 + 76);

glVertex2i(556 - 220, 465 + 76);

glVertex2i(556 - 220, 445 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(524 - 220, 465 + 76);

glVertex2i(519 - 220, 460 + 76);

glVertex2i(514 - 220, 469 + 76);

glEnd();

//hands

glBegin(GL\_POLYGON);

glColor3ub(255, 191, 128);

glVertex2i(565 - 220, 468 + 76);

glVertex3i(575 - 225, 453 + 76, 2);

glVertex3i(567 - 225, 454 + 76, 2);

glVertex2i(562 - 220, 462 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(575 - 225, 453 + 76);

glVertex3i(556 - 238, 438 + 116, 2);

glVertex3i(556 - 238, 445 + 116, 2);

glVertex2i(567 - 225, 454 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(515 - 220, 468 + 76);

glVertex2i(505 - 220, 453 + 76);

glVertex2i(513 - 220, 454 + 76);

glVertex2i(518 - 220, 462 + 76);

glEnd();

glBegin(GL\_POLYGON);

glVertex2i(505 - 220, 453 + 76);

glVertex2i(524 - 220, 438 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(513 - 220, 454 + 76);

glEnd();

// belt

glBegin(GL\_POLYGON);

glColor3ub(150, 12, 30);

glVertex2i(556 - 220, 445 + 76);

glVertex2i(524 - 220, 445 + 76);

glVertex2i(524 - 220, 440 + 76);

glVertex2i(524 - 220, 440 + 76);

glVertex2i(556 - 220, 440 + 76);

glEnd();

// collar

glBegin(GL\_POLYGON);

glColor3ub(200, 140, 110 + 76);

glVertex2i(529 - 220, 480 + 76);

glVertex2i(551 - 220, 480 + 76);

glVertex2i(546 - 220, 470 + 76);

glVertex2i(534 - 220, 470 + 76);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3ub(20, 140, 110);

glVertex2i(540 - 220, 477 + 76);

glVertex2i(545 - 220, 470 + 76);

glVertex2i(535 - 220, 470 + 76);

glEnd();

// buttons

glColor3ub(0, 0, 0);

glPushMatrix();

glTranslatef(540 - 220, 465 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 220, 458 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

glPushMatrix();

glTranslatef(540 - 220, 451 + 76, 0);

glutSolidTorus(1, 1, 100, 90);

glPopMatrix();

/// pant

glBegin(GL\_POLYGON);

glColor3ub(80, 80, 230);

glVertex2i(555 - 220, 440 + 76);

glVertex2i(525 - 220, 440 + 76);

glVertex2i(520 - 220, 405 + 76);

glVertex2i(530 - 220, 405 + 76);

glVertex2i(533 - 220, 438 + 76);

glVertex2i(550 - 220, 405 + 76);

glVertex2i(560 - 220, 405 + 76);

glEnd();

//shoe left

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 10);

glVertex2i(530 - 220, 405 + 76);

glVertex2i(530 - 220, 396 + 76);

glVertex2i(512 - 220, 396 + 76);

glVertex2i(520 - 220, 405 + 76);

glEnd();

//shoe right

glBegin(GL\_POLYGON);

glColor3ub(100, 10, 10);

glVertex2i(550 - 220, 405 + 76);

glVertex2i(550 - 220, 396 + 76);

glVertex2i(568 - 220, 396 + 76);

glVertex2i(560 - 220, 405 + 76);

glEnd();

}

void tree1d()

{

//trunk1

glColor3ub(95, 6, 5);

double len = 70;

double thick = 20;

glPushMatrix();

glTranslated(110, 150 + 330, 0.0);

glScaled(thick, len, thick);

glutSolidCube(1.0);

glPopMatrix();

//leaves1

glColor3f(0.0, 0.9, 0.0);

glPushMatrix();

glLoadIdentity();

glTranslated(100, 200 + 280, 0.0);

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

glutSolidCone(60, 100, 3, 2);

glPopMatrix();

//leaves2

glColor3f(0.0, 0.9, 0.0);

glPushMatrix();

glLoadIdentity();

glTranslated(97, 250 + 280, 0.0);

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

glutSolidCone(60, 80, 3, 2);

glPopMatrix();

// leaves3

glColor3f(0.0, 0.9, 0.0);

glPushMatrix();

glLoadIdentity();

glTranslated(95, 290 + 280, 0);

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

glutSolidCone(60, 70, 3, 2);

glPopMatrix();

}

/\*

/-------------------------------------------------------------------/

// FUNCTION text

/-------------------------------------------------------------------/

\*/

void text()

{

char string[] = "STAY HOME! STAY SAFE!";

void\* font = GLUT\_BITMAP\_TIMES\_ROMAN\_24;

int k;

glColor3f(1, 0.85, 0);

glRasterPos3f(680, 602, -120);

for (k = 0;k < strlen(string);k++)

glutBitmapCharacter(font, string[k]);

}

/\*void intro()

{

glColor3ub(0, 0, 0);

glBegin(GL\_POLYGON);

glVertex2i(0, 650);

glVertex2i(600, 650);

glVertex2i(800, 250);

glVertex2i(0, 250);

glEnd();

glColor3ub(0,0, 0);

glBegin(GL\_POLYGON);

glVertex2i(600, 650);

glVertex2i(1000, 650);

glVertex2i(1000, 250);

glVertex2i(600, 250);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_POLYGON);

glVertex2i(600, 450);

glVertex2i(1000, 450);

glVertex2i(1000, 0);

glVertex2i(600, 0);

glEnd();

glColor3ub(0, 0, 0);

glBegin(GL\_POLYGON);

glVertex2i(0, 450);

glVertex2i(800, 450);

glVertex2i(800, 0);

glVertex2i(0, 0);

glEnd();

}\*/

void loadBackground4(void)

{

glGenTextures(1, &bg5);

glBindTexture(GL\_TEXTURE\_2D, bg5);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

int width, height, nrChannels;

unsigned char\* data = stbi\_load("Scene\_1\_Final.png", &width, &height, &nrChannels, STBI\_rgb\_alpha);

if (data)

{

glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, width, height, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, data);

//glGenerateMipmap(GL\_TEXTURE\_2D);

}

else

{

std::cout << "Failed to load bg1" << std::endl;

}

stbi\_image\_free(data);

}

void display13()

{

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

glDisable(GL\_TEXTURE\_2D);

glEnable(GL\_TEXTURE\_2D);

glColor3f(1, 1, 1);

glBindTexture(GL\_TEXTURE\_2D, bg5);

glBegin(GL\_QUADS);

glVertex3f(0, 0, 10);

glTexCoord2f(0, 0);

glVertex3f(0, 650, 10);

glTexCoord2f(0, 1);

glVertex3f(1000, 650, 10);

glTexCoord2f(1, 1);

glVertex3f(1000, 0, 10);

glTexCoord2f(1, 0);

glMatrixMode(GL\_PROJECTION);

glEnd();

glFlush();

glDisable(GL\_TEXTURE\_2D);

glutSwapBuffers();

}

void texti()

{

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

//display13();

glClearColor(0.33f, 0.33f, 0.33f, 0.33f);

glBegin(GL\_POLYGON); // left strip1

glColor3f(1.0, 1.0, 1.0);

glVertex3f(105.0, 490.0, 0.0);

glVertex3f(75.0, 490.0, 0.0);

glVertex3f(190.0, 610.0, 0.0);

glVertex3f(220.0, 610.0, 0.0);

glEnd();

glBegin(GL\_POLYGON); // right strip

glColor3f(1.0, 1.0, 1.0);

glVertex3f(790.0, 55.0, 0.0);

glVertex3f(820.0, 55.0, 0.0);

glVertex3f(935.0, 155.0, 0.0);

glVertex3f(905.0, 155.0, 0.0);

glEnd();

glBegin(GL\_LINES); //left line

glColor3f(1.0, 1.0, 1.0);

glVertex2f(120.0, 75.0);

glVertex2f(120.0, 575.0);

glEnd();

glBegin(GL\_LINES); //top line

glColor3f(1.0, 1.0, 1.0);

glVertex2f(120.0, 575.0);

glVertex2f(880.0, 575.0);

glEnd();

glBegin(GL\_LINES); //right line

glColor3f(1.0, 1.0, 1.0);

glVertex2f(880.0, 575.0);

glVertex2f(880.0, 75.0);

glEnd();

glBegin(GL\_LINES); // down line

glColor3f(1.0, 1.0, 1.0);

glVertex2f(120.0, 75.0);

glVertex2f(880.0, 75.0);

glEnd();

glFlush();

drawText(350, 540, 0, "BANGALORE INSTITUTE OF TECHNOLOGY");

drawText(320, 510, 0, "COMPUTER SCIENCE AND ENGINEERING DEPARTMENT");

drawText(280, 480, 0, "18CSL67-COMPUTER GRAPHICS AND VISUALIZATION LABORATORY");

drawText(450, 450, 0, "MINI PROJECT");

drawText(425, 400, 0, " \" SOP OF COVID-19 \" ");

drawText(405, 350, 0, "LIKHTIH B 1BI18CS073(B1)");

drawText(405, 320, 0, "MAHAVEER 1BI18CS075(B1)");

drawText(660, 260, 0, "UNDER THE GUIDANCE OF");

drawText(670, 230, 0, "Asst Prof Bhanushree.K.J");

drawText(660, 210, 0, "Dept of CSE,BIT-Bangalore");

drawText(400, 100, 0, "Click on the left button to start the COVID-19 Spread");

drawText(150, 280, 0, "LAB INCHARGES-B1");

drawText(150, 250, 0, "1)Asst Prof.N.Thanuja");

drawText(170, 230, 0, "Dept of CSE,BIT-Bangalore");

drawText(150, 190, 0, "2)Dr.B.N.Shankar Gowda");

drawText(170, 170, 0, "Dept of CSE,BIT-Bangalore");

glFlush();

}

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

{

if (btn == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

{

then = 1;

glutPostRedisplay();

}

}

void display(void) {

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, 1000, 10.0, 650, -2000, 1500);

glClearColor(0.0, 0.255, 0.0, 0);

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

glMatrixMode(GL\_MODELVIEW);

if (then == 0)

{

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

glClearColor(0.678, 0.847, 0.82,0);

glClearColor(0.0, 0.0, 0.0, 0);

display13();

glutPostRedisplay();

glutSwapBuffers();

glFlush();

//intro();

//texti();

}

if (then == 1)

{

if ((moveup <= 215) && (spread2 == 0))

{

tree1d();

//tree2d();

road();

lamppost1();

lamppost2();

//treed();

lamppost4();

lamppost99();

lamppost7();

text1();

text();

bus\_stop();

man1();

man();

men();

woman();

glPushMatrix();

glTranslatef(290, 340 + moveup, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(315, 300 + moveup, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(340, 340 + moveup, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

}

else if ((moveup > 215) && (spread2 == 0))

{

tree1d();

//tree2d();

road();

lamppost1();

lamppost2();

//treed();

lamppost4();

lamppost99();

lamppost7();

text();

text2();//one person is affected

bus\_stop();

man1();

manred();

men();

woman();

if ((moveright > 0) && (moveright <= 42) && (spread2 == 0))

{

glPushMatrix();

glTranslatef(290 - (moveright / 0.5), 515 + moveright, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(345 + (moveright / 0.5), 545 + moveright, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - (moveright / 0.5), 545 - moveright, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(340 + (moveright / 0.5), 515 - moveright, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

}

else if ((moveright > 0) && (moveright > 42))

{

spread2 = 1;

}

}

else if ((spread2 == 1) && (spread3 == 0))

{

tree1d();

//tree2d();

road();

lamppost1();

lamppost2();

//treed();

lamppost99();

lamppost4();

lamppost7();

text3();

text();

bus\_stop();

man1();

manred();

menred();

woman();

glPushMatrix();

glTranslatef(290 - 88, 515 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - 88, 545 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

if ((movedown > 0) && (movedown <= 42))

{

glPushMatrix();

glTranslatef(395 - (movedown / 0.5), 540 + movedown, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(450 + (movedown / 0.5), 570 + movedown, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(410 - (movedown / 0.5), 570 - movedown, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(445 + (movedown / 0.5), 540 - movedown, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

}

else if ((movedown > 0) && (movedown > 42))

{

spread3 = 1;

}

}

else if ((spread3 == 1) && (spreadall == 0))

{

tree1d();

//tree2d();

road();

lamppost1();

lamppost2();

//treed();

lamppost99();

lamppost4();

lamppost7();

text4();

text();

bus\_stop();

man1();

manred();

menred();

womanred();

glPushMatrix();

glTranslatef(290 - 88, 515 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - 88, 545 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(445 + 88, 540 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

if ((moveleft > 0) && (moveleft < 44))

{

glPushMatrix();

glTranslatef(290 - (moveleft / 0.5), 515 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(345 + (moveleft / 0.5), 545 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - (moveleft / 0.5), 545 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(340 + (moveleft / 0.5), 515 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(395 - (moveleft / 0.5), 540 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(450 + (moveleft / 0.5), 570 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(410 - (moveleft / 0.5), 570 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(445 + (moveleft / 0.5), 540 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(510 - (moveleft / 0.5), 460 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(565 + (moveleft / 0.5), 490 + moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(525 - (moveleft / 0.5), 490 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(560 + (moveleft / 0.5), 460 - moveleft, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

}

else if ((moveleft > 0) && (moveleft > 42))

{

spreadall = 1;

}

}

else if (spreadall == 1)

{

tree1d();

//tree2d();

road();

lamppost1();

lamppost2();

lamppost99();

//treed();

lamppost4();

lamppost7();

text99();

text();

bus\_stop();

man1();

manred();

menred();

womanred();

glPushMatrix();

glTranslatef(290 - 88, 515 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - 88, 545 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(445 + 88, 540 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(290 - 88, 515 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(345 + 88, 545 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(295 - 88, 545 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(340 + 88, 515 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(395 - 88, 540 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(450 + 88, 570 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(410 - 88, 570 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(445 + 88, 540 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(510 - 88, 460 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(565 + 88, 490 + 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(525 - 88, 490 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

glPushMatrix();

glTranslatef(560 + 88, 460 - 44, 10);

glRotatef(rotx, 0, 0, 0);

glRotatef(roty, 0, 1, 0);

glRotatef(-45, 0, 0, 0);

drawScene();

//man();

glPopMatrix();

}

}

glutSwapBuffers();

}

static void SpecialKeyFunc(int Key, int x, int y)

{

switch (Key)

{

case GLUT\_KEY\_UP:

moveup = moveup + 5;

glutPostRedisplay();

break;

case GLUT\_KEY\_RIGHT:

if (moveup > 215)

{

moveright = moveright + 2;

}

glutPostRedisplay();

break;

case GLUT\_KEY\_DOWN:

if (moveright > 42)

{

movedown = movedown + 2;

}

glutPostRedisplay();

break;

case GLUT\_KEY\_LEFT:

if (movedown > 42)

{

moveleft = moveleft + 2;

}

glutPostRedisplay();

break;

}

}

void processSelection(int xPos, int yPos)

{

GLfloat fAspect; // Screen aspect ratio

// Space for selection buffer

GLuint selectBuff[64];

// Hit counter and viewport storeage

GLint hits, viewport[4];

// Setup selection buffer

glSelectBuffer(64, selectBuff);

// Get the viewport

glGetIntegerv(GL\_VIEWPORT, viewport);

// Switch to projection and save the matrix

glMatrixMode(GL\_PROJECTION);

glPushMatrix();

// Change render mode

glRenderMode(GL\_SELECT);

glLoadIdentity();

gluPickMatrix(xPos, viewport[3] - yPos, 4, 4, viewport);

fAspect = (float)viewport[2] / (float)viewport[3];

gluPerspective(45.0f, fAspect, 1.0, 800.0);

display();

hits = glRenderMode(GL\_RENDER);

if (hits >= 1) {

//ProcessSelection(selectBuff);

inv\_rot = !inv\_rot;

}

else

std::cout << "Hai cliccato spazio vuoto" << std::endl;

glMatrixMode(GL\_PROJECTION);

glPopMatrix();

}

void reshape(int w, int h) {

// Prevent a divide by zero

if (h == 0)

h = 1;

// Set Viewport to window dimensions

glViewport(0, 0, w, h);

// Calculate aspect ratio of the window

float fAspect = (GLfloat)w / (GLfloat)h;

// Set the perspective coordinate system

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(65.0, (GLfloat)w / (GLfloat)h, 1.0, 800.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void mouse\_callback(int button, int state, int x, int y) {

if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN)

processSelection(x, y);

}

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

switch (key) {

case 'w':

rotx += 5.0;

break;

case 's':

rotx -= 5.0;

break;

case 'a':

roty += 5.0;

break;

case 'd':

roty -= 5.0;

break;

case 'l':

light = !light;

if (light) {

glEnable(GL\_LIGHT0);

global\_ambient\_on[4] = 1.0f;

}

else {

glDisable(GL\_LIGHT0);

global\_ambient\_on[4] = 0.0f;

}

break;

case '+':

g\_speed += 0.1;

break;

case '-':

g\_speed -= 0.1;

break;

}

glutPostRedisplay();

}

void loadBackground(void)

{

glGenTextures(1, &bg2);

glBindTexture(GL\_TEXTURE\_2D, bg2);

// set the bg1 wrapping/filtering options (on the currently bound bg1 object)

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

// load and generate the bg1

int width, height, nrChannels;

unsigned char\* data = stbi\_load("Scene\_2\_Final.png", &width, &height, &nrChannels, STBI\_rgb\_alpha);

if (data)

{

glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, width, height, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, data);

//glGenerateMipmap(GL\_TEXTURE\_2D);

}

else

{

std::cout << "Failed to load bg1" << std::endl;

}

stbi\_image\_free(data);

}

void timer(int value)

{

glutPostRedisplay();

}

void display10() {

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

glDisable(GL\_TEXTURE\_2D);

glEnable(GL\_TEXTURE\_2D);

glColor3f(1, 1, 1);

glBindTexture(GL\_TEXTURE\_2D, bg2);

glBegin(GL\_QUADS);

glVertex3f(0, 0, 10);

glTexCoord2f(0, 0);

glVertex3f(0, 650, 10);

glTexCoord2f(0, 1);

glVertex3f(1000, 650, 10);

glTexCoord2f(1, 1);

glVertex3f(1000, 0, 10);

glTexCoord2f(1, 0);

glMatrixMode(GL\_PROJECTION);

glEnd();

glFlush();

glDisable(GL\_TEXTURE\_2D);

glutSwapBuffers();

}

void loadBackground2(void)

{

glGenTextures(1, &bg3);

glBindTexture(GL\_TEXTURE\_2D, bg3);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

int width, height, nrChannels;

unsigned char\* data = stbi\_load("Scene\_3\_Final.png", &width, &height, &nrChannels, STBI\_rgb\_alpha);

if (data)

{

glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, width, height, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, data);

//glGenerateMipmap(GL\_TEXTURE\_2D);

}

else

{

std::cout << "Failed to load bg1" << std::endl;

}

stbi\_image\_free(data);

}

void display11()

{

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

glDisable(GL\_TEXTURE\_2D);

glEnable(GL\_TEXTURE\_2D);

glColor3f(1, 1, 1);

glBindTexture(GL\_TEXTURE\_2D, bg3);

glBegin(GL\_QUADS);

glVertex3f(0, 0, 10);

glTexCoord2f(0, 0);

glVertex3f(0, 650, 10);

glTexCoord2f(0, 1);

glVertex3f(1000, 650, 10);

glTexCoord2f(1, 1);

glVertex3f(1000, 0, 10);

glTexCoord2f(1, 0);

glMatrixMode(GL\_PROJECTION);

glEnd();

glFlush();

glDisable(GL\_TEXTURE\_2D);

glutSwapBuffers();

}

void loadBackground3(void)

{

glGenTextures(1, &bg4);

glBindTexture(GL\_TEXTURE\_2D, bg4);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

int width, height, nrChannels;

unsigned char\* data = stbi\_load("Scene\_4\_Final.png", &width, &height, &nrChannels, STBI\_rgb\_alpha);

if (data)

{

glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGBA, width, height, 0, GL\_RGBA, GL\_UNSIGNED\_BYTE, data);

//glGenerateMipmap(GL\_TEXTURE\_2D);

}

else

{

std::cout << "Failed to load bg1" << std::endl;

}

stbi\_image\_free(data);

}

void display12()

{

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

glDisable(GL\_TEXTURE\_2D);

glEnable(GL\_TEXTURE\_2D);

glColor3f(1, 1, 1);

glBindTexture(GL\_TEXTURE\_2D, bg4);

glBegin(GL\_QUADS);

glVertex3f(0, 0, 10);

glTexCoord2f(0, 0);

glVertex3f(0, 650, 10);

glTexCoord2f(0, 1);

glVertex3f(1000, 650, 10);

glTexCoord2f(1, 1);

glVertex3f(1000, 0, 10);

glTexCoord2f(1, 0);

glMatrixMode(GL\_PROJECTION);

glEnd();

glFlush();

glDisable(GL\_TEXTURE\_2D);

glutSwapBuffers();

}

void menufunc(int n)

{

switch (n)

{

case 1: for (int i = 0;i < 5000;i++) {

display10();

}

break;

case 2: for(int i = 0;i < 5000;i++) {

display11();

}

break;

case 3: for (int i = 0;i < 5000;i++) {

display12();

}

break;

case 4: exit(0);

break;

}

}

void idle() {

if (!inv\_rot)

g\_rot -= g\_speed;

else

g\_rot += g\_speed;

glutPostRedisplay();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(1200, 800);

glutInitWindowPosition(0, 0);

glutCreateWindow("COVID-19");

init();

submenu = glutCreateMenu(menufunc);

glutCreateMenu(menufunc);

glutAddMenuEntry("PRECAUTION", 1);

glutAddMenuEntry("SYMPTOM", 2);

glutAddMenuEntry("VACCINE", 3);

glutAddMenuEntry("EXIT", 4);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

glutReshapeFunc(reshape);

glutDisplayFunc(display);

loadBackground();

loadBackground2();

loadBackground3();

loadBackground4();

glutTimerFunc(500, timer, 0);

glEnable(GL\_DEPTH\_TEST);

glutMouseFunc(mouse\_callback);

glutIdleFunc(idle);

glutKeyboardFunc(keyboard);

glutMouseFunc(mouse);

glutSpecialFunc(SpecialKeyFunc);

glutMainLoop();

return 0;

}