#include <iostream>

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

using namespace std;

int winwidth=1600,winheight=900;

int startx=-1200,starty=700,startz=100,deepth=80,widthsmall=200,widthlarge=500,heightlaege=400,heightsmall=200;

int \*\* head=NULL, \*\*clavicle=NULL,\*\*pelvicum=NULL,\*\*leftarm1=NULL,\*\*leftarm2=NULL,\*\*rightarm1=NULL,\*\*rightarm2=NULL,

\*\*leftleg1=NULL,\*\*leftleg2=NULL,\*\*rightleg1=NULL,\*\*rightleg2=NULL;

float pro=0.2;

int time=1, direction=-1;

void createcube(int startx,int starty,int startz,int width,int height ,int deepth,int\*\* cube)

{

cube[0][0]=startx;

cube[0][1]=starty;

cube[0][2]=startz;

cube[1][0]=startx+width;

cube[1][1]=starty;

cube[1][2]=startz;

cube[2][0]=startx+width;

cube[2][1]=starty-height;

cube[2][2]=startz;

cube[3][0]=startx;

cube[3][1]=starty-height;

cube[3][2]=startz;

cube[4][0]=startx;

cube[4][1]=starty;

cube[4][2]=startz-deepth;

cube[5][0]=startx+width;

cube[5][1]=starty;

cube[5][2]=startz-deepth;

cube[6][0]=startx+width;

cube[6][1]=starty-height;

cube[6][2]=startz-deepth;

cube[7][0]=startx;

cube[7][1]=starty-height;

cube[7][2]=startz-deepth;

}

void drawhead()

{

//pro 是偏移的比例

//glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glVertex2f(head[0][0]\*1.0/winwidth+head[0][2]\*pro/winwidth,head[0][1]\*1.0/winheight-head[0][2]\*pro/winheight);

glVertex2f(head[1][0]\*1.0/winwidth+head[1][2]\*pro/winwidth,head[1][1]\*1.0/winheight-head[1][2]\*pro/winheight);

glVertex2f(head[2][0]\*1.0/winwidth+head[2][2]\*pro/winwidth,head[2][1]\*1.0/winheight-head[2][2]\*pro/winheight);

glVertex2f(head[3][0]\*1.0/winwidth+head[3][2]\*pro/winwidth,head[3][1]\*1.0/winheight-head[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(head[4][0]\*1.0/winwidth+head[4][2]\*pro/winwidth,head[4][1]\*1.0/winheight-head[4][2]\*pro/winheight);

glVertex2f(head[5][0]\*1.0/winwidth+head[5][2]\*pro/winwidth,head[5][1]\*1.0/winheight-head[5][2]\*pro/winheight);

glVertex2f(head[6][0]\*1.0/winwidth+head[6][2]\*pro/winwidth,head[6][1]\*1.0/winheight-head[6][2]\*pro/winheight);

glVertex2f(head[7][0]\*1.0/winwidth+head[7][2]\*pro/winwidth,head[7][1]\*1.0/winheight-head[7][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(head[0][0]\*1.0/winwidth+head[0][2]\*pro/winwidth,head[0][1]\*1.0/winheight-head[0][2]\*pro/winheight);

glVertex2f(head[4][0]\*1.0/winwidth+head[4][2]\*pro/winwidth,head[4][1]\*1.0/winheight-head[4][2]\*pro/winheight);

glVertex2f(head[7][0]\*1.0/winwidth+head[7][2]\*pro/winwidth,head[7][1]\*1.0/winheight-head[7][2]\*pro/winheight);

glVertex2f(head[3][0]\*1.0/winwidth+head[3][2]\*pro/winwidth,head[3][1]\*1.0/winheight-head[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(head[1][0]\*1.0/winwidth+head[1][2]\*pro/winwidth,head[1][1]\*1.0/winheight-head[1][2]\*pro/winheight);

glVertex2f(head[5][0]\*1.0/winwidth+head[5][2]\*pro/winwidth,head[5][1]\*1.0/winheight-head[5][2]\*pro/winheight);

glVertex2f(head[6][0]\*1.0/winwidth+head[6][2]\*pro/winwidth,head[6][1]\*1.0/winheight-head[6][2]\*pro/winheight);

glVertex2f(head[2][0]\*1.0/winwidth+head[2][2]\*pro/winwidth,head[2][1]\*1.0/winheight-head[2][2]\*pro/winheight);

glEnd();

glFlush();

}

void drawclavical()

{

//pro 是偏移的比例

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glVertex2f(clavicle[0][0]\*1.0/winwidth+clavicle[0][2]\*pro/winwidth,head[0][1]\*1.0/winheight-head[0][2]\*pro/winheight);

glVertex2f(clavicle[1][0]\*1.0/winwidth+clavicle[1][2]\*pro/winwidth,head[1][1]\*1.0/winheight-head[1][2]\*pro/winheight);

glVertex2f(clavicle[2][0]\*1.0/winwidth+clavicle[2][2]\*pro/winwidth,head[2][1]\*1.0/winheight-head[2][2]\*pro/winheight);

glVertex2f(clavicle[3][0]\*1.0/winwidth+head[3][2]\*pro/winwidth,head[3][1]\*1.0/winheight-head[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(clavicle[4][0]\*1.0/winwidth+head[4][2]\*pro/winwidth,head[4][1]\*1.0/winheight-head[4][2]\*pro/winheight);

glVertex2f(clavicle[5][0]\*1.0/winwidth+head[5][2]\*pro/winwidth,head[5][1]\*1.0/winheight-head[5][2]\*pro/winheight);

glVertex2f(clavicle[6][0]\*1.0/winwidth+head[6][2]\*pro/winwidth,head[6][1]\*1.0/winheight-head[6][2]\*pro/winheight);

glVertex2f(clavicle[7][0]\*1.0/winwidth+head[7][2]\*pro/winwidth,head[7][1]\*1.0/winheight-head[7][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(clavicle[0][0]\*1.0/winwidth+head[0][2]\*pro/winwidth,head[0][1]\*1.0/winheight-head[0][2]\*pro/winheight);

glVertex2f(clavicle[4][0]\*1.0/winwidth+head[4][2]\*pro/winwidth,head[4][1]\*1.0/winheight-head[4][2]\*pro/winheight);

glVertex2f(clavicle[7][0]\*1.0/winwidth+head[7][2]\*pro/winwidth,head[7][1]\*1.0/winheight-head[7][2]\*pro/winheight);

glVertex2f(clavicle[3][0]\*1.0/winwidth+head[3][2]\*pro/winwidth,head[3][1]\*1.0/winheight-head[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(clavicle[1][0]\*1.0/winwidth+head[1][2]\*pro/winwidth,head[1][1]\*1.0/winheight-head[1][2]\*pro/winheight);

glVertex2f(clavicle[5][0]\*1.0/winwidth+head[5][2]\*pro/winwidth,head[5][1]\*1.0/winheight-head[5][2]\*pro/winheight);

glVertex2f(clavicle[6][0]\*1.0/winwidth+head[6][2]\*pro/winwidth,head[6][1]\*1.0/winheight-head[6][2]\*pro/winheight);

glVertex2f(clavicle[2][0]\*1.0/winwidth+head[2][2]\*pro/winwidth,head[2][1]\*1.0/winheight-head[2][2]\*pro/winheight);

glEnd();

glFlush();

}

void drawrect(int \*\* rect )

{

//pro 是偏移的比例

//glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glVertex2f(rect[0][0]\*1.0/winwidth+rect[0][2]\*pro/winwidth,rect[0][1]\*1.0/winheight-rect[0][2]\*pro/winheight);

glVertex2f(rect[1][0]\*1.0/winwidth+rect[1][2]\*pro/winwidth,rect[1][1]\*1.0/winheight-rect[1][2]\*pro/winheight);

glVertex2f(rect[2][0]\*1.0/winwidth+rect[2][2]\*pro/winwidth,rect[2][1]\*1.0/winheight-rect[2][2]\*pro/winheight);

glVertex2f(rect[3][0]\*1.0/winwidth+rect[3][2]\*pro/winwidth,rect[3][1]\*1.0/winheight-rect[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(rect[4][0]\*1.0/winwidth+rect[4][2]\*pro/winwidth,rect[4][1]\*1.0/winheight-rect[4][2]\*pro/winheight);

glVertex2f(rect[5][0]\*1.0/winwidth+rect[5][2]\*pro/winwidth,rect[5][1]\*1.0/winheight-rect[5][2]\*pro/winheight);

glVertex2f(rect[6][0]\*1.0/winwidth+rect[6][2]\*pro/winwidth,rect[6][1]\*1.0/winheight-rect[6][2]\*pro/winheight);

glVertex2f(rect[7][0]\*1.0/winwidth+rect[7][2]\*pro/winwidth,rect[7][1]\*1.0/winheight-rect[7][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(rect[0][0]\*1.0/winwidth+rect[0][2]\*pro/winwidth,rect[0][1]\*1.0/winheight-rect[0][2]\*pro/winheight);

glVertex2f(rect[4][0]\*1.0/winwidth+rect[4][2]\*pro/winwidth,rect[4][1]\*1.0/winheight-rect[4][2]\*pro/winheight);

glVertex2f(rect[7][0]\*1.0/winwidth+rect[7][2]\*pro/winwidth,rect[7][1]\*1.0/winheight-rect[7][2]\*pro/winheight);

glVertex2f(rect[3][0]\*1.0/winwidth+rect[3][2]\*pro/winwidth,rect[3][1]\*1.0/winheight-rect[3][2]\*pro/winheight);

glEnd();

glBegin(GL\_LINE\_LOOP);

glVertex2f(rect[1][0]\*1.0/winwidth+rect[1][2]\*pro/winwidth,rect[1][1]\*1.0/winheight-rect[1][2]\*pro/winheight);

glVertex2f(rect[5][0]\*1.0/winwidth+rect[5][2]\*pro/winwidth,rect[5][1]\*1.0/winheight-rect[5][2]\*pro/winheight);

glVertex2f(rect[6][0]\*1.0/winwidth+rect[6][2]\*pro/winwidth,rect[6][1]\*1.0/winheight-rect[6][2]\*pro/winheight);

glVertex2f(rect[2][0]\*1.0/winwidth+rect[2][2]\*pro/winwidth,rect[2][1]\*1.0/winheight-rect[2][2]\*pro/winheight);

glEnd();

glFlush();

}

void display()

{

//glClearColor(1.0f, 1.0f, 1.0f,0.1f);

glClear(GL\_COLOR\_BUFFER\_BIT);

drawrect(head);

drawrect(clavicle);

drawrect(pelvicum);

drawrect(leftarm1);

drawrect(leftarm2);

drawrect(rightarm1);

drawrect(rightarm2);

drawrect(leftleg1);

drawrect(leftleg2);

drawrect(rightleg1);

drawrect(rightleg2);

}

void moveaction1(int \*\* rect,int\*\* rect1)

{

rect[5][2]=rect[4][2]=rect[0][2]-deepth\*cos(time);

rect1[0][2]=rect1[1][2]=rect[2][2]=rect[3][2]=rect[1][2]+sin(time)\*heightsmall;

rect1[4][2]=rect1[5][2]=rect[7][2]=rect[6][2]=rect[5][2]+sin(time)\*heightsmall;

rect1[3][2]=rect1[2][2]=rect[1][2]-2\*heightsmall\*cos(time);

rect1[7][2]=rect1[6][2]=rect[5][2]-2\*heightsmall\*cos(time);

rect[5][1]=rect[4][1]=rect[0][1]+deepth\*sin(time)\*-1;

rect1[0][1]=rect1[1][1]=rect[3][1]=rect[2][1]=rect[1][1]-1\*heightsmall\*cos(time);

rect1[4][1]=rect1[5][1]=rect[7][1]=rect[6][1]=rect[5][1]-1\*heightsmall\*cos(time);

rect1[3][1]=rect1[2][1]=rect[1][1]-2\*heightsmall\*cos(time);

rect1[7][1]=rect1[6][1]=rect[5][1]-2\*heightsmall\*cos(time);

}

void moveaction2(int \*\* rect,int \*\*rect1,int time)

{

rect[5][2]=rect[4][2]=rect[0][2]-deepth\*cos(time);

rect1[0][2]=rect1[1][2]=rect[2][2]=rect[3][2]=rect[1][2]+sin(time)\*heightsmall;

rect1[4][2]=rect1[5][2]=rect[7][2]=rect[6][2]=rect[5][2]+sin(time)\*heightsmall;

rect1[3][2]=rect1[2][2]=rect[1][2]-2\*heightsmall\*cos(time);

rect1[7][2]=rect1[6][2]=rect[5][2]-2\*heightsmall\*cos(time);

rect[5][1]=rect[4][1]=rect[0][1]+deepth\*sin(time)\*-1;

rect1[0][1]=rect1[1][1]=rect[3][1]=rect[2][1]=rect[1][1]-1\*heightsmall\*cos(time);

rect1[4][1]=rect1[5][1]=rect[7][1]=rect[6][1]=rect[5][1]-1\*heightsmall\*cos(time);

rect1[3][1]=rect1[2][1]=rect[1][1]-2\*heightsmall\*cos(time);

rect1[7][1]=rect1[6][1]=rect[5][1]-2\*heightsmall\*cos(time);

/\* rect[5][2]=rect[4][2]=rect[0][2]-deepth\*cos(time);

rect[2][2]=rect[3][2]=rect[1][2]+sin(time)\*heightsmall;

rect[7][2]=rect[6][2]=rect[5][2]+sin(time)\*heightsmall;

rect[5][1]=rect[4][1]=rect[0][1]+deepth\*sin(time)\*-1;

rect[3][1]=rect[2][1]=rect[1][1]-1\*heightsmall\*cos(time);

rect[7][1]=rect[6][1]=rect[5][1]-1\*heightsmall\*cos(time);\*/

}

void motion()

{

/\*    glViewport(0,0,(GLsizei)w,(GLsizei)h);           //为了选择一个更小的绘图区域，在窗口中定义一个像素矩形，将图像映射到这个矩形中

    glMatrixMode(GL\_PROJECTION);                     //指定哪一个矩阵是当前矩阵（GL\_PROJECTION,对投影矩阵应用随后的矩阵操作）

    glLoadIdentity();                                //将当前的用户坐标系的原点移到了屏幕中心：类似于一个复位操作

    gluOrtho2D(0.0,(GLdouble)w,0.0,(GLdouble)h);     //将当前的可视空间设置为正投影空间,这个函数描述了一个平行修剪空间,意味着离观察者较远的对象看上去不会变小

\*/

for(int i=0;i<8;i++)

{

head[i][0]++;

clavicle[i][0]++;

pelvicum[i][0]++;

leftarm1[i][0]++;

leftarm2[i][0]++;

rightarm1[i][0]++;

rightarm2[i][0]++;

leftleg1[i][0]++;

leftleg2[i][0]++;

rightleg1[i][0]++;

rightleg2[i][0]++;

head[i][1]--;

clavicle[i][1]--;

pelvicum[i][1]--;

leftarm1[i][1]--;

leftarm2[i][1]--;

rightarm1[i][1]--;

rightarm2[i][1]--;

leftleg1[i][1]--;

leftleg2[i][1]--;

rightleg1[i][1]--;

rightleg2[i][1]--;

}

time+=direction;

if(time%20==0)

{

moveaction1(leftarm1,leftarm2);

moveaction1(rightleg1,rightleg2);

moveaction2(leftleg1,leftleg2,-time);

moveaction2(rightarm1,rightarm2,-time);

}

if(time>45||time<-45)

direction=-direction;

glClear(GL\_COLOR\_BUFFER\_BIT);

drawrect(head);

drawrect(clavicle);

drawrect(pelvicum);

drawrect(leftarm1);

drawrect(leftarm2);

drawrect(rightarm1);

drawrect(rightarm2);

drawrect(leftleg1);

drawrect(leftleg2);

drawrect(rightleg1);

drawrect(rightleg2);

}

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

{

head = new int\*[8];

clavicle = new int\*[8];

pelvicum = new int\*[8];

leftarm1 = new int\*[8];

leftarm2 = new int\*[8];

rightarm1 = new int\*[8];

rightarm2 = new int\*[8];

leftleg1 = new int\*[8];

leftleg2 = new int\*[8];

rightleg1 = new int\*[8];

rightleg2 = new int\*[8];

for(int i=0;i<8;i++)

{

head[i] = new int[3];

clavicle[i] = new int[3];

pelvicum[i] = new int[3];

leftarm1[i] = new int[3];

leftarm2[i] = new int[3];

rightarm1[i] = new int[3];

rightarm2[i] = new int[3];

leftleg1[i] = new int[3];

leftleg2[i] = new int[3];

rightleg1[i] = new int[3];

rightleg2[i] = new int[3];

}

createcube(startx-deepth/2,starty,startz,deepth,heightlaege,deepth,head);

createcube(startx-widthlarge/2,starty-100,startz,widthlarge,deepth,deepth,clavicle);

createcube(startx-widthsmall/2,starty-heightlaege,startz,widthsmall,deepth/2,deepth,pelvicum);

createcube(startx-widthlarge/2-deepth,starty-100,startz,deepth,heightsmall,deepth,leftarm1);

createcube(startx-widthlarge/2-deepth,starty-100-heightsmall,startz,deepth,heightsmall,deepth,leftarm2);

createcube(startx+widthlarge/2,starty-100,startz,deepth,heightsmall,deepth,rightarm1);

createcube(startx+widthlarge/2,starty-100-heightsmall,startz,deepth,heightsmall,deepth,rightarm2);

createcube(startx-widthsmall/2-deepth,starty-heightlaege,startz,deepth,heightsmall,deepth,leftleg1);

createcube(startx-widthsmall/2-deepth,starty-heightlaege-heightsmall,startz,deepth,heightsmall,deepth,leftleg2);

createcube(startx+widthsmall/2,starty-heightlaege,startz,deepth,heightsmall,deepth,rightleg1);

createcube(startx+widthsmall/2,starty-heightlaege-heightsmall,startz,deepth,heightsmall,deepth,rightleg2);

glutInit(&argc,argv);

glutInitWindowSize (winwidth,winheight);

glutCreateWindow("Hello,world!");

glutDisplayFunc(display);

//glutReshapeFunc(motion);

glutIdleFunc(&motion);

glutMainLoop();

}