作业2

学号	姓名	专业(方向)
18308045	谷正阳	大数据

1. 绘制一个沿固定线路运动的机器人

- 线路可以是圆或任意其它闭合路径
  - 选用'∞'字形
  - 考虑是两个圆形拼接而成
  - 设置在右边的圆形俯视做逆时针运动，左边则为顺时针
  - 需要通过$\Delta\theta_i$的正负来表示顺逆时针
  - 同时可以通过$\Delta\theta_i$的正负判断在左还是在右
  - 状态转移如下：

$$
\begin{cases}
x_i = \frac{\Delta\theta_i}{|\Delta\theta_i|} \cdot r \cdot (1 - \cos(\theta_i)) + x_0 \\
y_i = y_0 \\
z_i = \frac{\Delta\theta_i}{|\Delta\theta_i|} \cdot r \cdot \sin(\theta_i) + z_0 \\
\theta_{i+1} = \theta_i + \Delta\theta_i \\
\Delta\theta_{i+1} = \begin{cases}
-\Delta\theta_i, & \text{if } \theta_{i+1} + \Delta\theta_i > \max\theta \\
-\Delta\theta_i, & \text{if } \theta_{i+1} + \Delta\theta_i < \min\theta \\
\Delta\theta_i, & \text{else}
\end{cases}
\end{cases}
$$

  - 初始状态如下：

$$
\begin{cases}
x_0 = 0 \\
y_0 = 0 \\
z_0 = -50 \\
\theta_0 = 0^\circ \\
\Delta\theta_0 = 1^\circ \\
r = 10 \\
\max\theta = 360^\circ \\
\min\theta = 0^\circ
\end{cases}
$$

- 机器人在任意时刻应面向运动曲线的切线方向
  - 根据切线和半径垂直，可以得知面向方向与z轴负方向夹角为θ_i
- 机器人应该有头、躯干、四肢等基本部分
  - 都用单位箱子变换来表示，单元箱子实现参考课件代码`drawUnitBox`
- 机器人在运动过程中应具有 摆臂 及 抬脚 两个基本动作
  - 实现类似机器人的面向，使用旋转变换
  - 增加了摇头动作

2. 代码

```cpp
#include "myglwidget.h"
#include <math.h>

/*######################################################
##  函数: MyGLWidget
##  函数描述:  MyGLWidget类的构造函数，实例化定时器timer
##  参数描述:
##  parent: MyGLWidget的父对象
######################################################*/

MyGLWidget::MyGLWidget(QWidget *parent)
    :QOpenGLWidget(parent)
{
    timer = new QTimer(this); // 实例化一个定时器
    timer->start(16); // 时间间隔设置为16ms，可以根据需要调整
    connect(timer, SIGNAL(timeout()), this, SLOT(update())); // 连接update()函数，每16ms触发一次update()函数进行重新绘图
}


/*######################################################
##  函数: ~MyGLWidget
##  函数描述:  ~MyGLWidget类的析构函数，删除timer
##  参数描述:  无
######################################################*/
MyGLWidget::~MyGLWidget()
{
    delete this->timer;
}


/*######################################################
##  函数: initializeGL
##  函数描述:  初始化绘图参数，如视窗大小、背景色等
##  参数描述:  无
######################################################*/
void MyGLWidget::initializeGL()
{
    glViewport(0, 0, width(), height());
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glDisable(GL_DEPTH_TEST);
}


/*######################################################
##  函数: drawUnitBox
##  函数描述: 绘制1*1*1的单元箱子
##  参数描述: 无
######################################################*/
void drawUnitBox()
{
    glBegin(GL_TRIANGLE_STRIP);
    glVertex3f(-0.5f, 0.5f, -0.5f);
    glVertex3f(0.5f, 0.5f, -0.5f);
    glVertex3f(-0.5f, -0.5f, -0.5f);
    glVertex3f(0.5f, -0.5f, -0.5f);
    glVertex3f(0.5f, -0.5f, 0.5f);
    glVertex3f(0.5f, 0.5f, -0.5f);
    glVertex3f(0.5f, 0.5f, 0.5f);
    glVertex3f(-0.5f, 0.5f, -0.5f);
    glVertex3f(-0.5f, 0.5f, 0.5f);
    glVertex3f(-0.5f, -0.5f, -0.5f);
    glVertex3f(-0.5f, -0.5f, 0.5f);
    glVertex3f(0.5f, -0.5f, 0.5f);
    glVertex3f(-0.5f, 0.5f, 0.5f);
    glVertex3f(0.5f, 0.5f, 0.5f);
    glEnd();
}
```

```
/*#####################################################
##  函数：drawHead
##  函数描述：绘制头部，有摇头动作
##  参数描述：无
#####################################################*/
void drawHead()
{
    static float theta = 0.0f;
    static float delta_theta = 4.0f;

    glPushMatrix();
    glTranslatef(0.0f, 10.0f, 0.0f);
    glRotatef(theta, 0.0f, 1.0f, 0.0f);
    glScalef(8.0f, 8.0f, 8.0f);
    glColor3f(0.91015625f, 0.5859375f, 0.4765625f);
    drawUnitBox();
    glPopMatrix();

    theta += delta_theta;
    if (theta == 28.0f)
    {
        delta_theta = -4.0f;
    }
    else if (theta == -28.0f)
    {
        delta_theta = 4.0f;
    }
}


/*#####################################################
##  函数：drawBody
##  函数描述：绘制身体
##  参数描述：无
#####################################################*/
void drawBody()
{
    glPushMatrix();
    glScalef(8.0f, 12.0f, 4.0f);
    glColor3f(0.0f, 0.54296875f, 0.54296875f);
    drawUnitBox();
    glPopMatrix();
}


/*#####################################################
##  函数：drawAnArm
##  函数描述：绘制一条胳膊，位置未设置
##  参数描述：无
#####################################################*/
void drawAnArm()
{
    glPushMatrix();
    glScalef(4.0f, 12.0f, 4.0f);
    glColor3f(0.91015625f, 0.5859375f, 0.4765625f);
    drawUnitBox();
    glPopMatrix();
}


/*#####################################################
##  函数：drawArms
##  函数描述：绘制胳膊，有抬臂动作
##  参数描述：无
#####################################################*/
void drawArms()
{
    static float theta = 0.0f;
    static float delta_theta = 1.0f;
```

```
    glPushMatrix();
    glTranslatef(0.0f, 6.0f, 0.0f);
    glRotatef(theta, 1.0f, 0.0f, 0.0f);
    glTranslatef(-6.0f, -6.0f, 0.0f);
    drawAnArm();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(0.0f, 6.0f, 0.0f);
    glRotatef(theta, -1.0f, 0.0f, 0.0f);
    glTranslatef(6.0f, -6.0f, 0.0f);
    drawAnArm();
    glPopMatrix();

    theta += delta_theta;
    if (theta == 30.0f)
    {
        delta_theta = -1.0f;
    }
    else if (theta == -30.0f)
    {
        delta_theta = 1.0f;
    }
}


/*######################################################
##  函数：drawALeg
##  函数描述：绘制一条腿，位置未设置
##  参数描述：无
######################################################*/
void drawALeg()
{
    glPushMatrix();
    glScalef(4.0f, 12.0f, 4.0f);
    glColor3f(0.0f, 0.0f, 0.80078125f);
    drawUnitBox();
    glPopMatrix();
}


/*######################################################
##  函数：drawLegs
##  函数描述：绘制腿，有抬腿动作
##  参数描述：无
######################################################*/
void drawLegs()
{
    static float theta = 0.0f;
    static float delta_theta = 1.0f;

    glPushMatrix();
    glTranslatef(0.0f, -6.0f, 0.0f);
    glRotatef(theta, -1.0f, 0.0f, 0.0f);
    glTranslatef(-2.0f, -6.0f, 0.0f);
    drawALeg();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(0.0f, -6.0f, 0.0f);
    glRotatef(theta, 1.0f, 0.0f, 0.0f);
    glTranslatef(2.0f, -6.0f, 0.0f);
    drawALeg();
    glPopMatrix();

    theta += delta_theta;
    if (theta == 30.0f)
    {
```

```
        delta_theta = -1.0f;
    }
    else if (theta == -30.0f)
    {
        delta_theta = 1.0f;
    }
}


/*#######################################################
##  函数: drawRobot
##  函数描述: 绘制机器人
##  参数描述: 无
#######################################################*/
void drawRobot()
{
    drawHead();
    drawBody();
    drawArms();
    drawLegs();
}


/*#######################################################
##  函数: paintGL
##  函数描述:  绘图函数，实现图形绘制，会被update()函数调用，有围绕∞字形做闭环运动
##  参数描述:  无
#######################################################*/
void MyGLWidget::paintGL()
{
    // Your Implementation
    static float theta = 0.0f;
    static float delta_theta = 1.0f;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0f, width() / height(), 30.0f, 70.0f);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef(delta_theta * 10.0f * (1 - cos(theta * M_PI / 180.0f)), 0.0f, delta_theta * 10.0f * sin(theta * M_PI / 180.0f) - 50.0f);
    glRotatef(theta, 0.0f, 1.0f, 0.0f);
    drawRobot();
    theta += delta_theta;

    if (theta == 360.0f)
    {
        delta_theta = -1.0f;
    }
    else if (theta == 0.0f)
    {
        delta_theta = 1.0f;
    }
}


/*#######################################################
##  函数: resizeGL
##  函数描述:  当窗口大小改变时调整视窗尺寸
##  参数描述:  无
#######################################################*/
void MyGLWidget::resizeGL(int width, int height)
{
    glViewport(0, 0, width, height);
    update();
}
```

3. 结果

见result.mp4