Московский авиационный институт
(государственный технический университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторные работы

по курсу
«Компьютерная графика»

V семестр

Студент: Баскаков О.А.
Группа: 08-306, номер по списку № 1.
Преподаватель: Измайлов А.А.

Оценка:
Дата:

Москва
2009

Отчет по лабораторной работе №1 по курсу «компьютерная графика»
Студент (или студентка) группы 08-306: Баскаков О.А.
Преподаватель: Измайлов А.А.
Работа выполнена: "9" сентября 2009 г.

Тема: *Построение изображений двумерных кривых.*

Цель работы

Написать программу, строящую изображение заданной замечательной кривой.
Обеспечить центрирование и масштабирование при изменении размеров окна. Дать возможность изменения параметров кривой и числа шагов аппроксимации. Дополнительное задание: обеспечить возможность вращения кривой вокруг центра координат.

Задача

Вариант 1:

$$1.\ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

x, y — декартовы координаты, a, b — константы, значения которых выбираются из соображений наглядности.

Оборудование

Ноутбук с предустановленным QT Creator, видеокарта ATI.

Теоретическая часть

Плоская кривая - кривая, все точки которой лежат в одной плоскости. Существуют следующие аналитические способы задания плоской кривой в декартовых координатах (x,y):

- Fx,y=0 (в неявном виде);
- y=fx; x=g(y) (в явном виде);
- x=x(t), y=y(t) (в параметрическом виде);

Многие кривые на плоскости удобно описывать в полярных координатах r,φ, как функции радиуса-вектора r и полярного угла φ. Декартова и полярная системы связаны друг с другом следующим соотношением:
x=rcosφ, y=rsinφ;

Отображение f: R2->R2 называется аффинным преобразованием, если его можно записать в виде:
fx=Mx+y, где M – невырожденная матрица размеров 2x2.

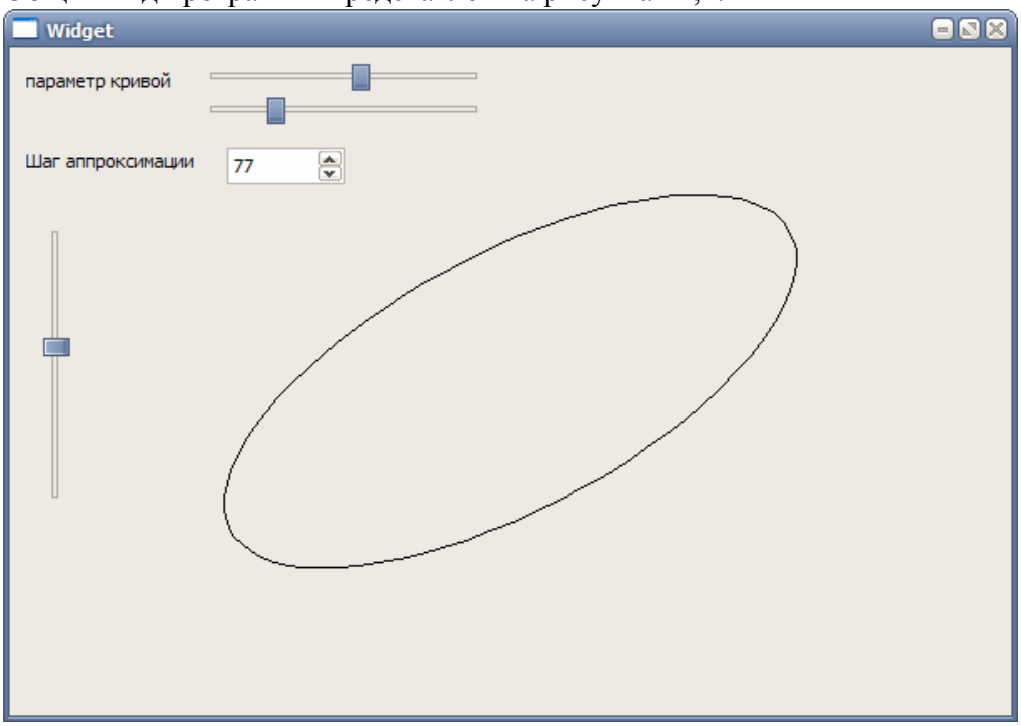
Основные аффинные преобразования:
1.Поворот на угол θ
y1y2=cosθ-sinθsinθcosθx1x2;
2.Масштабирование
y1y2=μ00λx1x2 , где λ,μ>0;
3.Перенос
y1y2=x1x2+a1a2;
Любое невырожденное аффинное преобразование можно выразить как суперпозицию приведенных выше элементарных преобразований.

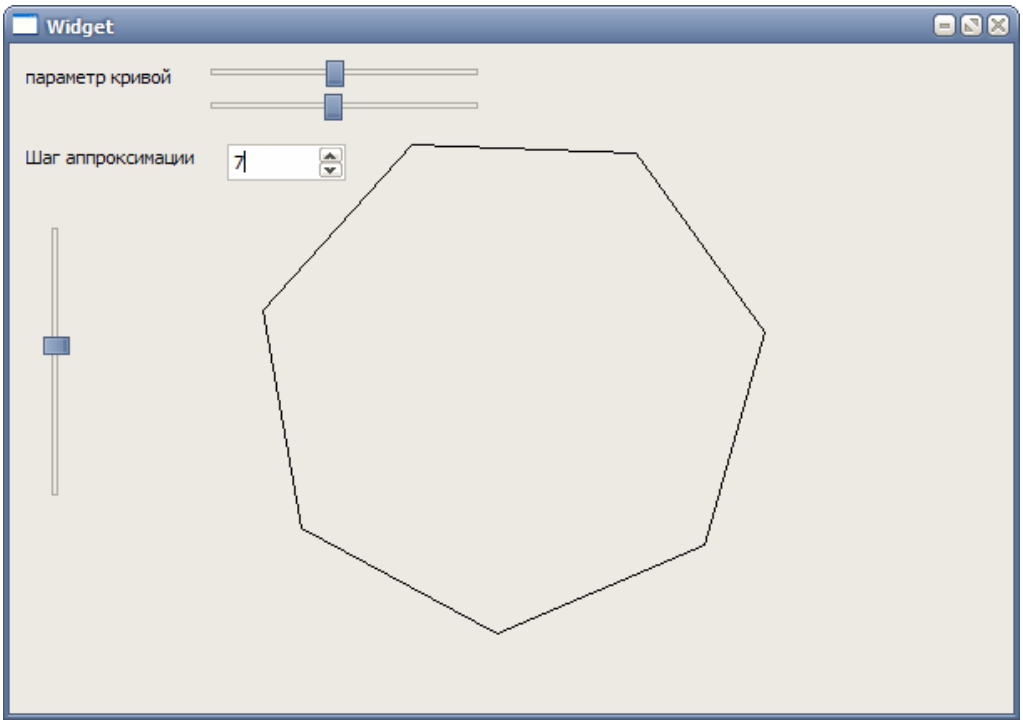
Ход решения

На языке C++ при помощи библиотек Qt была написана программа, которая строит график функции заданной по варианту. Реализовано центрирование, масштабирование, поворот данного графика, а также изменение некоторых его параметров.

График элипса строится поточечно в полярных координатах. Для этого диапазон допустимых значений аргумента функции разбивается на N отрезков в соответствии с числом итераций. Над точкой xi,Y(xi) , где xi - конец отрезка, выполняется аффинное преобразование поворота и результат отображается на экране. Соседние точки отрезков соединяются линиями.

Общий вид программы представлен на рисунках 1,2.





Реализовано изменение параметров A и B, которые задют растяжение элипса по оси, ими же можно масштабировать. Регулятор слева позволяет вращать элипс. Шаг аппроксимации задает переход от многоугольника к почти гладкой кривой.

Замечания

Преобразование координат точек лучше было бы сделать через матрицы.

Выводы

Были написана программа построения изображения заданной плоской кривой. А также получены основные навыки работы с Qt и QPainter.

Исходный код.

```
#include "widget.h"
#include "ui_widget.h"
#include <QPainter>
#include <math.h>


void Widget::paintEvent(QPaintEvent* event)
{
    QPainter p(this);

    float ox, oy;

    ox=this->width()/2;
    oy=this->height()/2;

    float scale;
    scale=(this->width()+this->height())*0.05;

    float rot;
    rot=ui->verticalSlider->value()*atan(1)*8/360;

    float dphi;
    dphi=atan(1)*8/
         ui->spinBox->value();


    float xlast,ylast,xr,yr,x2,y2;
    float phi = 0;

    float a = ui->horizontalSlider_A->value()*0.01;
    float b = ui->horizontalSlider_B->value()*0.01;

        xlast = - b*scale*sin(rot);
        ylast = + b*scale*cos(rot);

    for (phi = 0; phi < atan(1)*8; phi+=dphi)
    {
        xr = a*scale*sin(phi);
        yr = b*scale*cos(phi);

        x2 = xr*cos(rot) - yr*sin(rot);
        y2 = xr*sin(rot) + yr*cos(rot);

        p.drawLine( ox + xlast, oy - ylast, ox + x2, oy - y2);

        xlast = x2;
        ylast = y2;
    }

}
```