Московский авиационный институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Компьютерная графика»
Построение полиномиальных кривых

Студент: Семенов И.М.
Преподаватель: Морозов А.В.
Группа: М8О-306Б-18
Вариант: 20
Дата:
Оценка:
Подпись:

Москва, 2020

Условие:

Написать программу, стоящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения

Вариант 20:
NURB-кривая, n = 5, k = 4, узловой вектор равномерный. Веса точек различны и модифицируются.

Материалы:

Лекции по компьютерной графике
Методические указания к лабораторным работам по компьютерной графике
Документация Qt — *doc.qt.io*

Описание программы:

Программа написана на фреймворке Qt, основной функционал реализован с помощью пары из обьектов классов QGraphicsScene и QGraphicsView. Первый из них позволяет представляет с собой сцену, на которую можно добавлять обьекты классов, унаследованных от QGraphicsItem и взаимодействовать с ними. QGraphicsView представляет из себя виджет, на котором сцена отрисовывается. В программе используется класс-наследник QGraphicsView, который ответственен за изменении кривой, при передвижении вершин управляющего многоугольника NURB-кривой.

NURB-кривые являются развитием концепции B-сплайнов и представляют собой более мощный инструмент моделирования. NURB — сплайн расшифровывается как **Non-Uniformed Rational B-Spline**, т.е. **Неравномерный Рациональные B-Сплайн.** Такой сплайн задается следующим выражением.

$$P(t)=\frac{\sum_{i=1}^{n+1} P_i h_i B_{i,k}(t)}{\sum_{i=1}^{n+1} h_i B_{i,k}(t)},\ t_{\min} \le t \le t_{\max},\ 2 \le k \le n+1,$$

В этом выражении P представляют собой вершины управляющего n+1-угольника, h_i — неотрицательные числа, представляющие собой веса точек, а $B_{i,k}(t)$ - базисные функции B-сплайна, определяемые рекурсивными формулами Кокса де Бура:

$$B_{i,1}(t)=\begin{cases} 1 \text{ если } t_i \le t < t_{i+1} \\ 0 \text{ в противном случае} \end{cases}$$

$$B_{i,k}(t) = \frac{(t-t_i)B_{i,k-1}(t)}{t_{i+k-1}-t_i}+\frac{(t_{i+k}-t)B_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}},$$

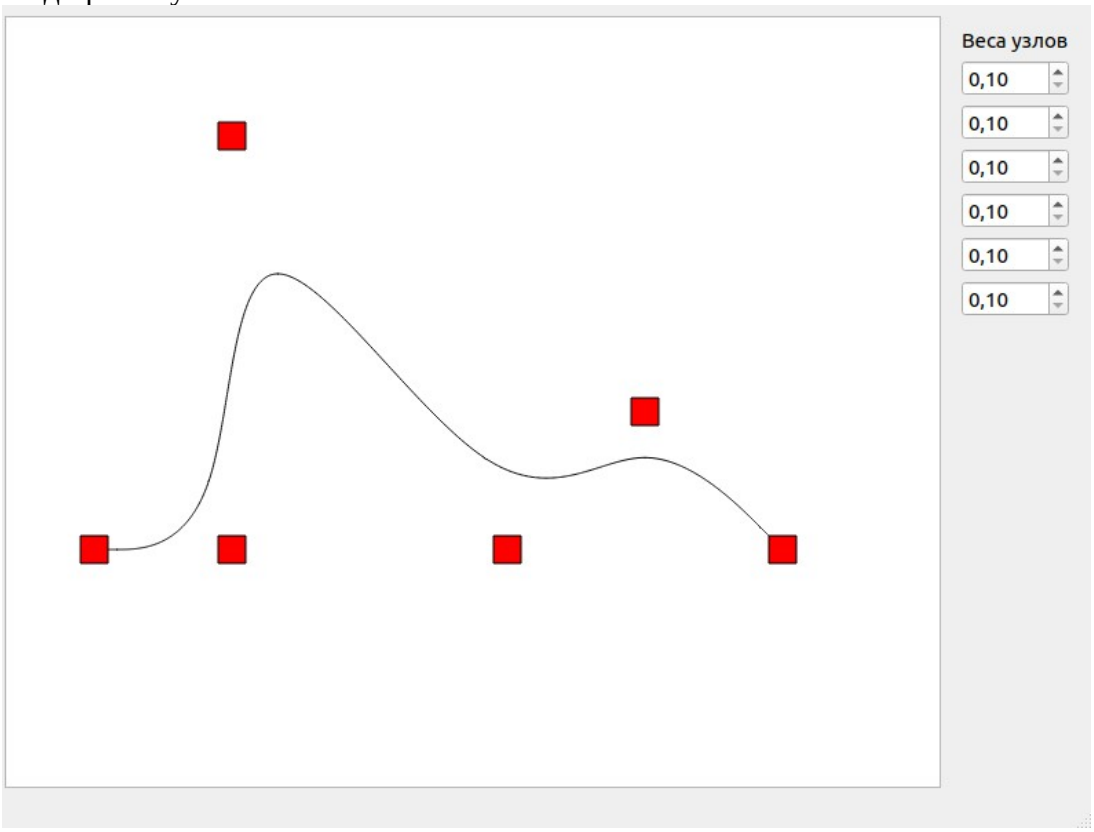
Структура программы

Программа состоит из следующих частей:

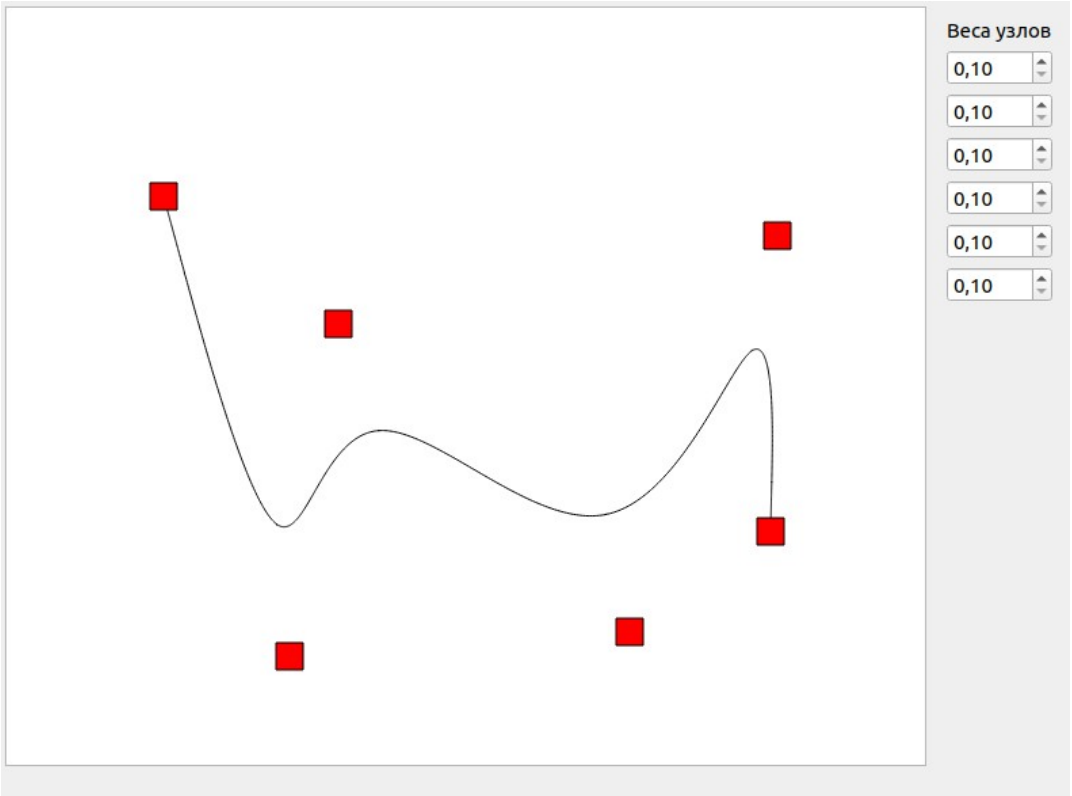
- Класс NURBEstimator — занимается вычислением точек кривой. Содержит вектора весов, узловых коэффициентов и точек управляющего многоугольника. Наибольший интерес представляют две приватные функции.Одна из них — double DeBoor(double,int,int) — рекурсивная функция для вычисления коэффициентов Кокса де Бура. Вторая функция — QPointF CalculatePoint(double) вычисляет непосредственно точку кривой для переданного значения параметра, используя функцию DeBoor, а также вектора весов и точек многоугольника. Учитывая, что коэффициенты Кокса де Бура остаются неизменными до тех пор, пока не поменяется узловой вектор или пока не изменится точность аппроксимации кривой, можно сохранить значения этих коэффициентов, не вычисляя их заново при изменении положения управляющих вершин. Класс дает возможность локальной модификации кривой, возвращая аппроксимацию для тех ее участков, которые были изменены.
- Класс Node — наследник QGraphicsRectItem. Представляет вершину управляющего многоугольника. Содержит номер представляемой вершины и указатель на GraphicsWidget. В этом классе перегружен метод itemChange — при изменении положения узла, вызызвает функцию Update у GraphicsWidget.
- Класс GraphicsWidget — наследник QGraphicsView. Содержит указатель на NURBEstimator, вектор указателей на узлы многоугольника и вектор указателей на участки кривой, представленные как QGraphicsPathItem. Эти вектора инициализируются при создании объекта. При инициализации на графическую сцену добавляются В этом классе реализована логика перерисовки изменившихся участков кривой — при вызове метода Update, принимающего индекс вершины многоугольника и его новую позицию, происходит обращение к хранимому обьекту класса NURBEstimator. После получения обновленных участков кривой, они обновляются и на сцене(вызывается метод QGraphicsPathItem::setPath). При изменении весов, пересчитывается вся кривая.
- Класс ParamHandler — предназначен для передачи вектора весов в GraphicWidget.
- main.cpp и класс MainWindow — нужны для инициализации пользовательского интерфейса и запуска программы.

Демонстрация работы программы

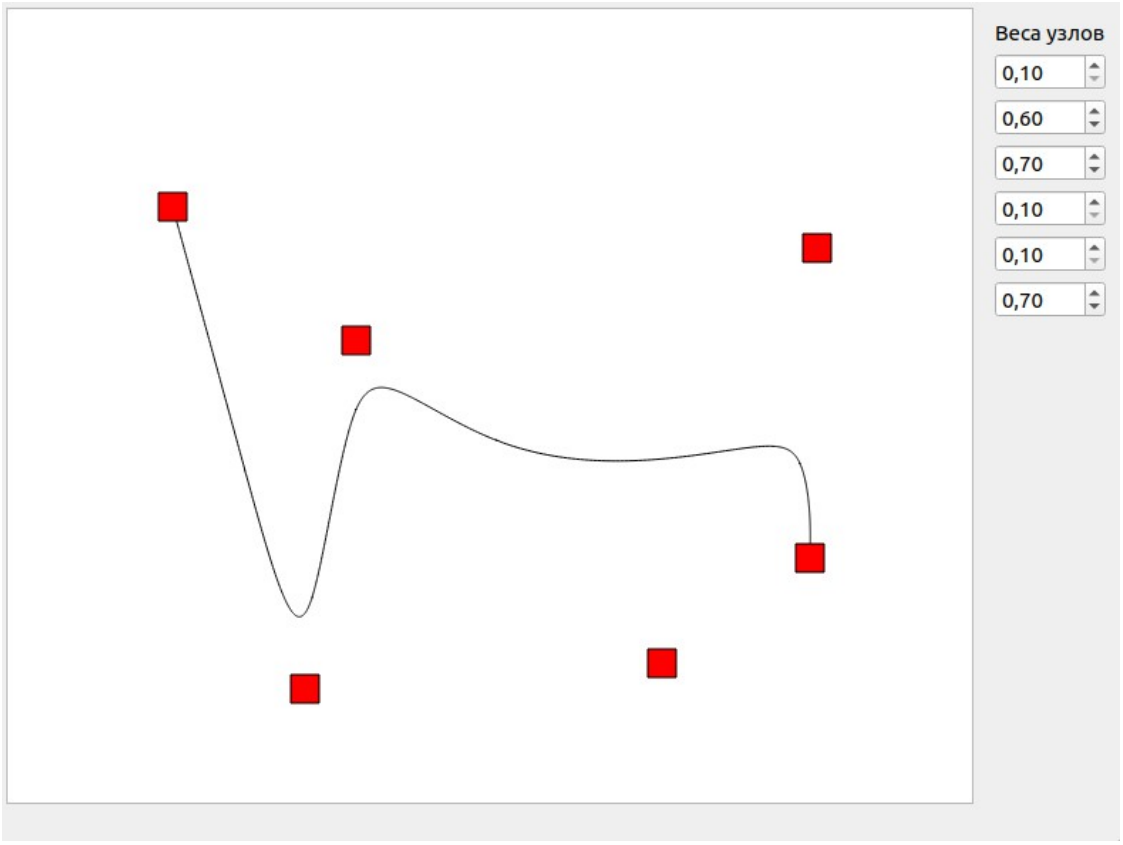
Вид при запуске:



После изменения положения вершин



После изменения весов



Вывод

В кривые обладают отличным свойством — возможностью локальной модификации кривой при изменении положения всего лишь нескольких вершин управляющего многоугольника. Эта особенность выгодно отличает их от кривых Безье и делает их использование более удобным. NURB-сплайны являются даже более гибкими, позволяя задать вектор весов. При необходимости, программу можно доработать, добавив возможность изменения узлового вектора, количества точек многоугольника и порядка кривой.