



Московский авиационный институт
(национальный исследовательский университет)

Факультет прикладной математики и информатики
Кафедра вычислительной математики и программирования

Курсовая работа
по курсу «Компьютерная графика»
на тему «Построение линейчатой поверхности»

Студент: Рябыкин А.С.
Преподаватель: Филиппов Г.С.
Дата:
Оценка:
Подпись:

Москва, 2020

Оглавление

1. Постановка задачи	3
2. Описание использованных технологий	3
3. Краткое описание программы	3
4. Работа с приложением.....	4
5. Выводы.....	4
6. Литература	4
7. Приложение	5-6

1. Постановка задачи

Составить и отладить программу, обеспечивающую каркасную визуализацию порции поверхности заданного типа. Исходные данные готовятся самостоятельно и вводятся из файла или в панели ввода данных. Должна быть обеспечена возможность тестирования программы на различных наборах исходных данных. Программа должна обеспечивать возможность управлять количеством изображаемых параметрических линий. Для визуализации параметрических линий поверхности разрешается использовать только функции отрисовки отрезков в экранных координатах.

Вариант

4 – Линейчатая поверхность (направляющие – кубические кривые Безье 3D).

Линейчатая поверхность в дифференциальной геометрии — поверхность, образованная движением прямой линии. Прямые, принадлежащие этой поверхности, называются прямолинейными образующими, а каждая кривая, пересекающая все прямолинейные образующие, направляющей кривой.

2. Описание использованных технологий

Для выполнения графических построений был выбран QPainter из фреймворка Qt, который наравне с QGraphicsScene используется для требуемых целей. Однако, QPainter выполняет более низкоуровневое рисование, в том числе и на виджетах (хотя имеется возможность рисования и на других устройствах рисования), работа с которыми знакома по выполненным перед курсовым проектом лабораторным работам.

QPainter содержит высокооптимизированные функции, позволяющие выполнять большинство рисунков требуемых программами с GUI. Легок в управлении, при этом не лишен функциональности. Основная функциональная возможность - это рисование, и он имеет множество функций для рисования примитивов: `drawPoint()`, `drawPoints()`, `drawLine()`, `drawRect()`, `drawRoundRect()`, `drawEllipse()`, `drawArc()`, `drawPie()`, `drawChord()`, `drawLineSegments()`, `drawPolyline()`, `drawPolygon()`

Для построения интерфейса программы использовались классы `view` (для работы с графическими построениями) и `panel` (для воплощения прочего функционала: аппроксимация, повороты), наследуемые от класса `QWidget`.

3. Краткое описание программы

Для реализации поставленной задачи используются 5 классов:

- `bezie` – задаёт построение кривой Безье.
- `panel` – задаёт функционал управления (кнопки поворотов и слайдер для изменения аппроксимации)

- view – содержит функционал для отрисовки линейчатой поверхности, обработки событий (изменение размера, нажатие и перемещение мыши – для задания точек кривой)
- mainwindow – содержит конструкторы классов view и panel, обработку функционала panel (повороты и аппроксимация)
- main – определяет экземпляр класса mainwindow (задаются первоначальные размеры окна)

Построением поверхности занимается функция PaintEvent(из класса view) которая задаёт две направляющие – кривые Безье, по которым строится поверхность. Затем по массиву точек кривых рисуются соединяющие их линии, необходимые для изменения поверхности.

Функционал управления реализован в классе panel.

4. Работа с приложением

После запуска программы появляется окно, в котором построена поверхность по заданным по умолчанию параметрам местоположения и аппроксимации.

В программе доступны изменения аппроксимации – слайдер на объекте panel, повороты – кнопки на том же объекте того же класса. Функция изменения масштаба реализована при изменении размеров окна.

Для изменения направляющих, формы и размеров поверхности используется работа с экраном с помощью клика мыши и перетаскивания узлов.

5. Выводы

В ходе выполнения данного курсового проекта я закрепил знания по созданию трёхмерных объектов Qt, QWidget, QPainter, которыми я пользовался и при написании лабораторных работ.

6. Литература

<https://doc.qt.io/qt-5/qpainter.html> - Документация по Qt, class QPainter

<https://doc.qt.io/qt-5/qwidget.html> - Документация по Qt, class QWidget

<https://doc.qt.io> – Документация по Qt

7. Приложение



