

Дисциплина: Объектно-ориентированное программирование
Лабораторная работа № 3 «Визуализация 3D поверхности»

Основные архитектурные требования:

Программа должна содержать два домена:

- Домен интерфейса
- Домен бизнес-логики

Домен интерфейса

Отвечает только за отображение интерфейса и передачу команд домену бизнес-логики. Загруженные из файла данные не должны храниться в домене интерфейса. Интерфейс должен быть реализован при помощи фреймворка Qt.

Домен бизнес-логики

Отвечает за основную функциональность программы. Именно в нем хранятся загруженные данные, выполняются необходимые операции. Также в этом домене производится подготовка к отрисовке. Диаграмма классов представлена на рисунке 2.

Все ошибки должны быть обработаны, пользователь должен получить о них уведомление. Некритичные ошибки не должны приводить к закрытию программы. При разработке следует руководствоваться принципами ООП.

Для разделения доменов бизнес-логики и интерфейса необходимо использовать паттерн фасад (класс Facade на рисунке 2), который является точкой входа в домен бизнес-логики. Только его методы должны вызываться из интерфейса программы

Задание

Необходимо разработать программу для визуализации функции двух переменных в трехмерном пространстве.

Программа должна предоставлять возможность:

- Загружать значения функции из выбранного пользователем csv-файла.
- Визуализировать нормированные (см. примечание) значения функции в виде поверхности в трехмерном пространстве с учетом заданного шага и диапазона нормировки.
- Смещать камеру (точку обзора) по осям координат (x, y, z).
- Вращать камеру (точку обзора) относительно осей координат (x, y, z).

Программа должна обладать графическим интерфейсом, содержащим:

- кнопку для выбора файла (fileDialog) и поле для вывода его названия.
- зону визуализации.
- поля для ввода шага между соседними точками и диапазона нормировки.
- управляющие элементы для перемещения и поворотов объекта по трем осям.

Входные данные

- csv-файл с матрицей, содержащий N строк и M столбцов в которых записаны числовые значения функции двух переменных.
- шаг (расстояния между соседними точками) по осям X, Y
- диапазон нормировки.
- управляющие команды для перемещений, поворотов и вращений сцены.

Выходные данные (результат)

- Трехмерная каркасная поверхность, отображающая данные из файла

Пример визуализации

На рисунке 1 изображен пример трёхмерной визуализации. Использованный датасет отличается от приведенного в разделе “Пример входных данных” и датасетов из заданий по вариантам.

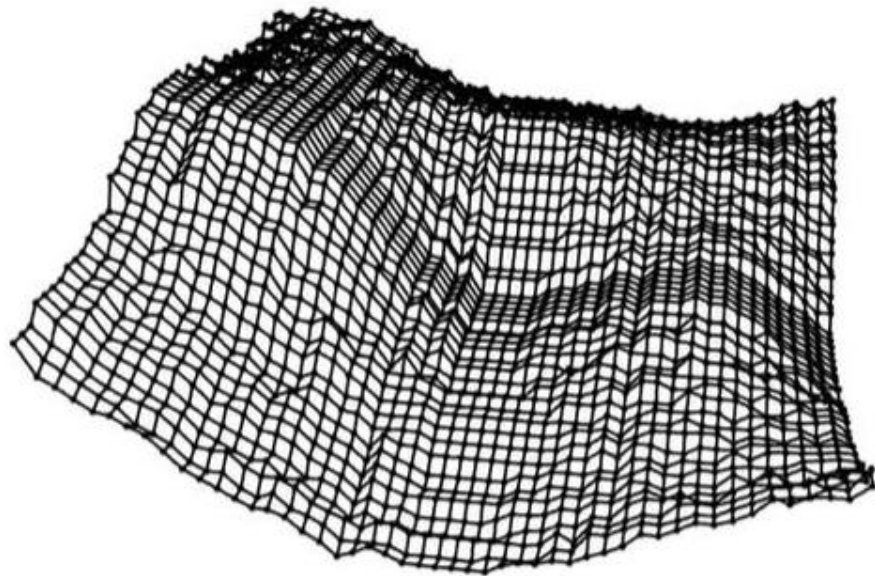


Рисунок 1 Каркасная модель

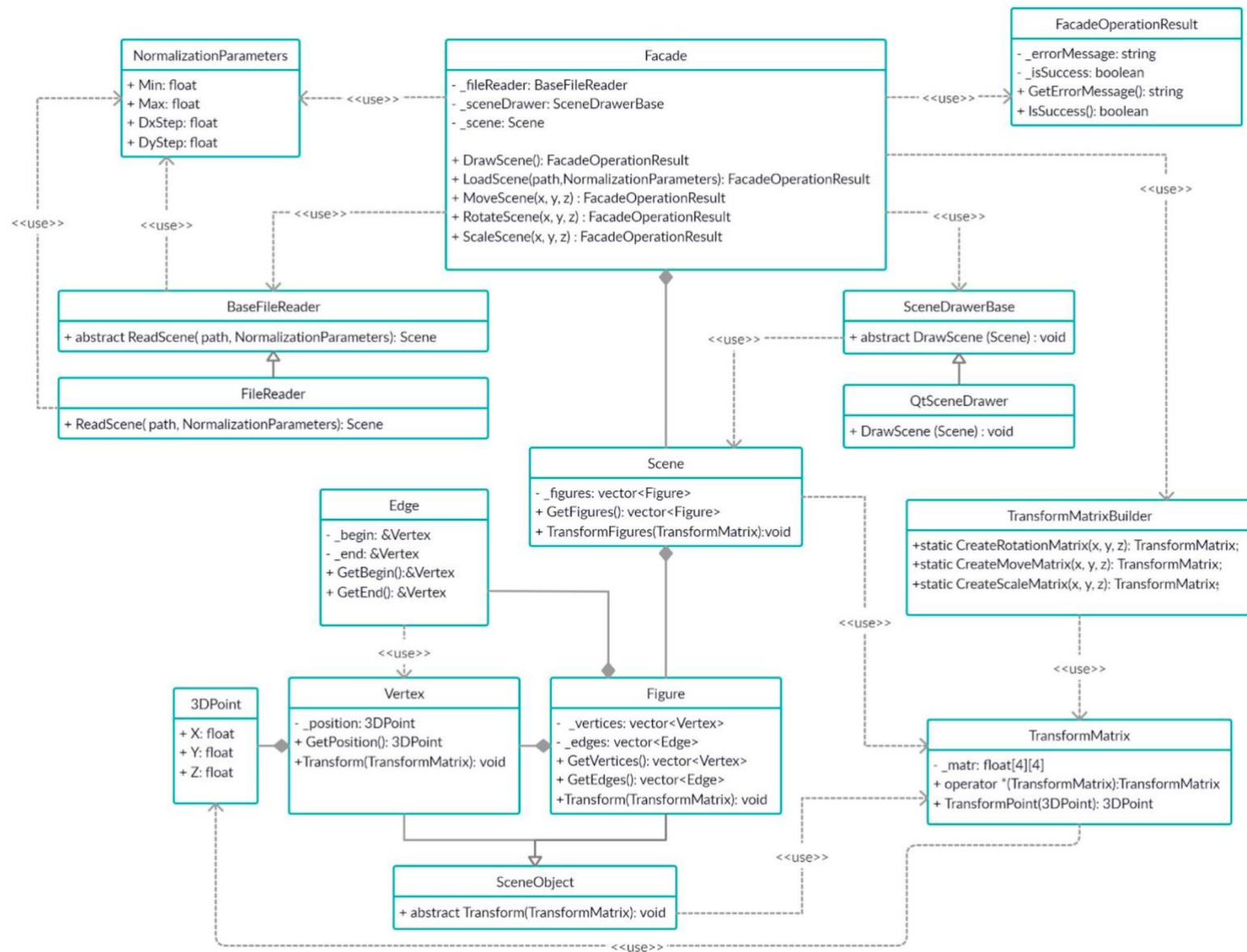


Рисунок 2 Диаграмма классов домена бизнес-логики

Полезная информация

Нормировка данных

Входные данные могут иметь разные значения. Если при выводе расстояние между соседними точками будет равно 1, а значения соседних точек будет отличаться, например на 1000 – работать с визуализированными данными будет затруднительно. Для устранения этой проблемы значения необходимо нормировать - т.е. масштабировать таким образом, чтобы данные укладывались в некоторый фиксированный диапазон.

Пример:

Значения функции: 10, -30, 20, 30

После нормирования к диапазону [-3, 3]: 1, -3, 2, 3.

Формулы для нормировки значений:

x_j – исходное значение

$[a, b]$ – диапазон нормировки

$X' = a + \frac{x - x_{min}}{x_{max} - x_{min}} (b - a)$ – нормированное значение.

Трёхмерные преобразования

Перенос

Перенос точки в трёхмерном пространстве осуществляется простым добавлением константы к соответствующей координате точки:

- по оси X $x' = x + c1$;
- по оси Y $y' = y + c2$;
- по оси Z $z' = z + c3$;

Смещение по каждой координате производится независимо, из чего следует, что для осуществления комплексного смещения сразу по нескольким координатам, необходимо выполнить перенос по каждой из осей по очереди. К примеру, для смещения сразу по осям Oх и Oу сначала необходимо выполнить смещение по оси Oх, а потом для получившегося результата выполнить смещение по Oу.

Масштабирование

Для увеличения или уменьшения объекта по заданной оси, необходимо домножить соответствующую координату на константу.

По оси X $x'=x * c1$;

По оси Y $y'=y * c2$;

По оси Z $z'=z * c3$;

Наиболее частым случаем, является однородное масштабирование объекта, при котором выбирается единый коэффициент масштабирования по всем трём осям ($c1 = c2 = c3 = c$). Коэффициент однородного масштабирования принимает следующие значения: $c = [0, +\infty)$. Причём, для выполнения уменьшения объекта необходимо выбрать $c = [0, 1)$, для выполнения увеличения $c = (1, +\infty)$. Если при масштабировании задать отрицательное число, будет осуществлено масштабирование с отражением относительно плоскости. В рамках задания допускается использование только положительного масштаба.

Вращение

Вращение осуществляется с помощью *матриц поворота*.

Матрица поворота (или матрица направляющих косинусов) – это ортогональная матрица, которая используется для выполнения собственного ортогонального преобразования в евклидовом пространстве. При умножении любого вектора на матрицу поворота длина вектора сохраняется. Определитель матрицы поворота равен единице. Обычно считают, что, в отличие от матрицы перехода при повороте системы координат (базиса), при умножении на матрицу поворота вектора-столбца координаты вектора преобразуются в соответствии с поворотом самого вектора (а не поворотом координатных осей; то есть при этом координаты повернутого вектора получаются в той же, неподвижной системе координат). Однако отличие той и другой матрицы лишь в знаке угла поворота, и одна может быть получена из другой заменой угла поворота на противоположный; та и другая взаимно обратны и могут быть получены друг из друга транспонированием.

Любое вращение в трехмерном пространстве может быть представлено как композиция поворотов вокруг трех ортогональных осей (например, вокруг осей декартовых координат). Этой композиции соответствует матрица, равная произведению соответствующих трех матриц поворота.

Матрицами вращения вокруг оси декартовой системы координат на угол α в трёхмерном пространстве являются:

Вращение вокруг оси x:

$$M_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}$$

Вращение вокруг оси y:

$$M_y(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

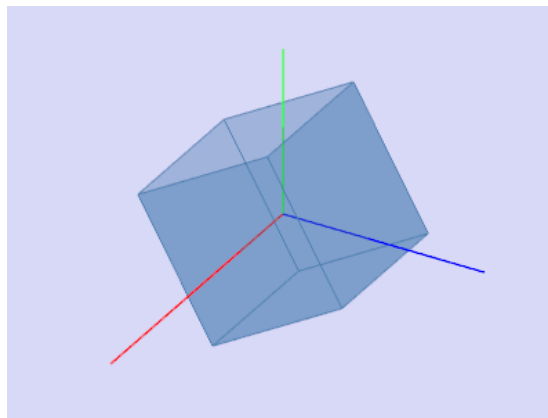
Вращение вокруг оси z:

$$M_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

После преобразований мы получаем формулы:

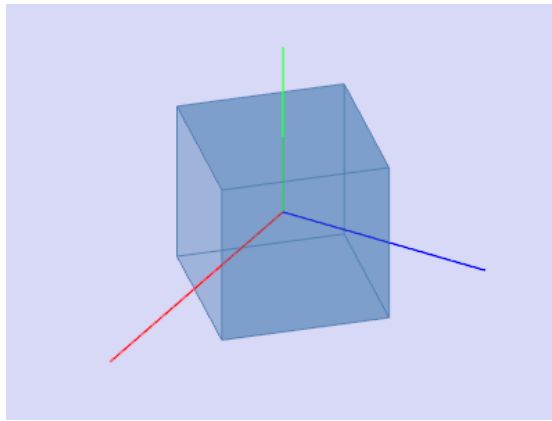
По оси X

- $x'=x$
- $y':=y*\cos(\alpha)-z*\sin(\alpha)$
- $z':=y*\sin(\alpha)+z*\cos(\alpha)$



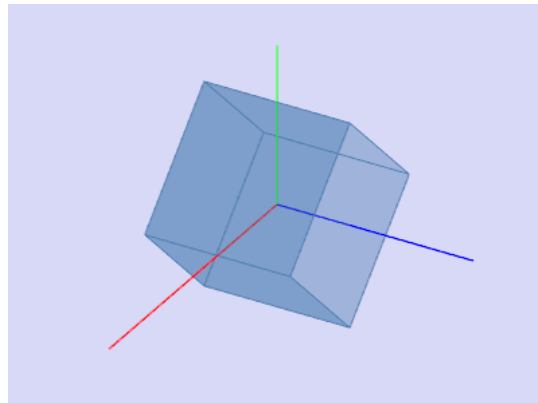
По оси Y

- $x'=x*\cos(\alpha)+z*\sin(\alpha)$
- $y'=y$
- $z'=-x*\sin(\alpha)+z*\cos(\alpha)$



По оси Z

- $x' = x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$
- $y' = x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$
- $z' = z$



Примечание: на графиках красная ось соответствует оси OX, зелёная – OY, синяя – OZ.

Все три поворота делаются независимо друг от друга, т.е. если надо повернуть вокруг осей Oх и Oу, вначале делается поворот вокруг оси Oх, потом применительно к полученной точки делается поворот вокруг оси Oу.

Положительным углом при этом соответствует вращение вектора против часовой стрелки в правой системе координат, и по часовой стрелке в левой системе координат, если смотреть против направления соответствующей оси. Правая система координат связана с выбором правого базиса.