

Московский авиационный институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Компьютерная графика»
Каркасная визуализация выпуклого многогранника
Удаление невидимых линий

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

Москва, 2020

Условие:

Разработать формат представления многоугольника и процедуру его каркасной отрисовки в ортогографической и изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов и масштабирования многогранника. Обеспечить автоматическое центрирование и изменение размеров изображения при изменении размеров окна.

Вариант 20:

8-гранная прямая правильная усеченная пирамида

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. В программе используется виджет для отрисовки и виджет с параметрами из предыдущей работы. Помимо этого добавлен абстрактный класс фигуры, в котором находятся функции, вычисляющие точки и полигоны фигуры, а так же различные преобразователи координат, позволяющие изменить положение фигуры в пространстве, удалить невидимые линии, получить перспективную проекцию и финальные точки для отрисовки на двумерной плоскости.

Структура программы

Программа состоит из файла `main.cpp`, абстрактного класса `Figure` (а так же класса `Pyramid` — единственного наследника), а так же из классов `Transformer`, `InvisibleDeleter`, `Rasterizer`, `Projecter` — классов, для преобразования точек или полигонов фигуры (не объединенные, однако, в общую иерархию). Так же в программе присутствуют классы `ParamHandler` и `GraphicWidget` — их функционал схож с функционалом аналогичных классов предыдущей работы, но несколько доработан.

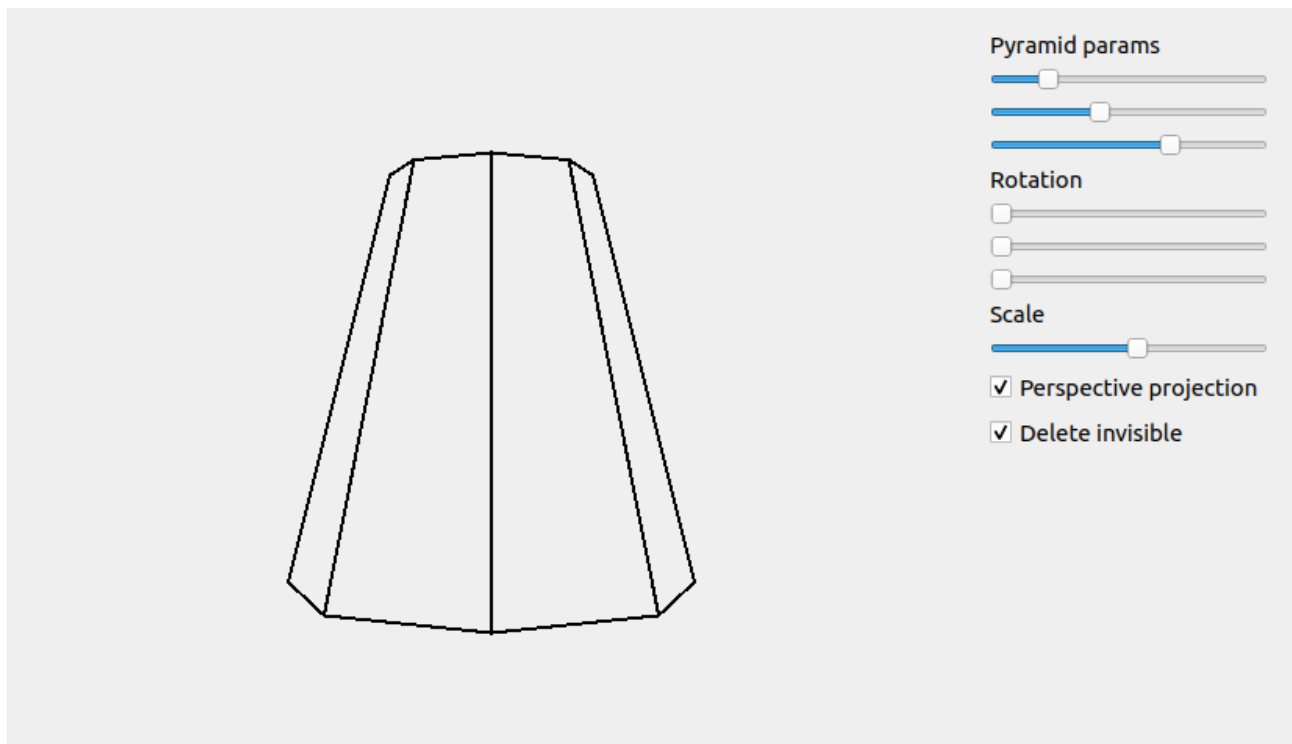
Описание файлов:

- `main.cpp` — содержит точку входа, в которой создается `QApplication` и `MainWindow`, у последнего вызывается метод `show`, после чего приложение входит в `event loop`.
- Класс `MainWindow` — основное окно. В конструкторе создает `ui` на основе формы и соединяет сигнал `ParamHandler::SendParameters` со слотом `GraphicWidget::SetParameters`, для соответствующих объектов, созданных во время создания `ui` и являющихся его детьми. После чего вызывает функцию `Setup` для обработчика параметров.
- Класс `ParamHandler` — предназначен для пересылки параметров в `GraphicWidget`. Так как виджеты теперь представляют собой классы `QSlider` и `QCheckBox`, пришлось изменить поведение класса. Теперь внутри него содержится контейнер, имеющий указатели на интересные виджеты. При изменении любого из них, вызывается событие, отсылающее параметры адресату (в данном случае это `GraphicWidget`). Необходимо вызвать функцию `Setup`, чтобы соединить внутренние виджеты со слотом `UpdateParameters` и добавить указатели на них в контейнер. Это действие не может быть выполнено в конструкторе, так как на момент вызова конструктора, дочерние виджеты еще не инициализированы.
- Класс `GraphicWidget` — занимается отрисовкой (с помощью `QPainter`) фигуры. Управление осуществляется с помощью виджетов в `ParamHandler`. По сути, класс занимается установкой параметров у нескольких объектов, после чего вызывает функцию отрисовки, определенную в классе фигуры.

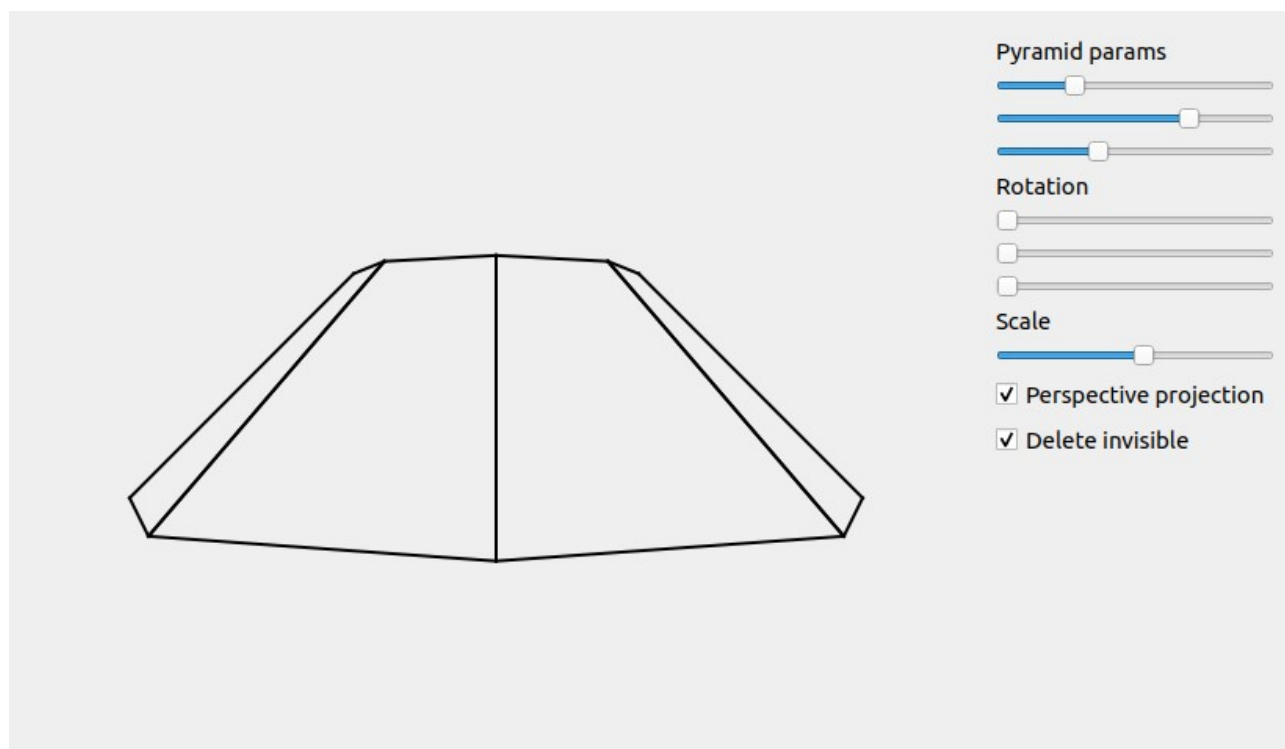
- Класс InvisibleDeleter — реализует алгоритм Джонсона удаления невидимых линий. Алгоритм работает в мировом пространстве и корректно удаляет невидимые полигоны для выпуклых многогранников (а усеченная пирамида является выпуклой).
- Класс Transformer содержит несколько четырехмерных матриц, которые умножаются на точки фигуры, тем самым изменяя масштаб, поворачивая и двигая фигуру в пространстве.
- Класс Projecter искажает фигуру таким образом, чтобы получить перспективную проекцию при окончательном отображении точек фигуры на двумерную плоскость.
- Класс Rasterizer — преобразует фигуру в набор двумерных многоугольников, которые представляют собой полигоны фигуры, спроецированные на плоскость xOz . Эти точки далее отрисовываются на экране с помощью QPainter.

Демонстрация работы программы

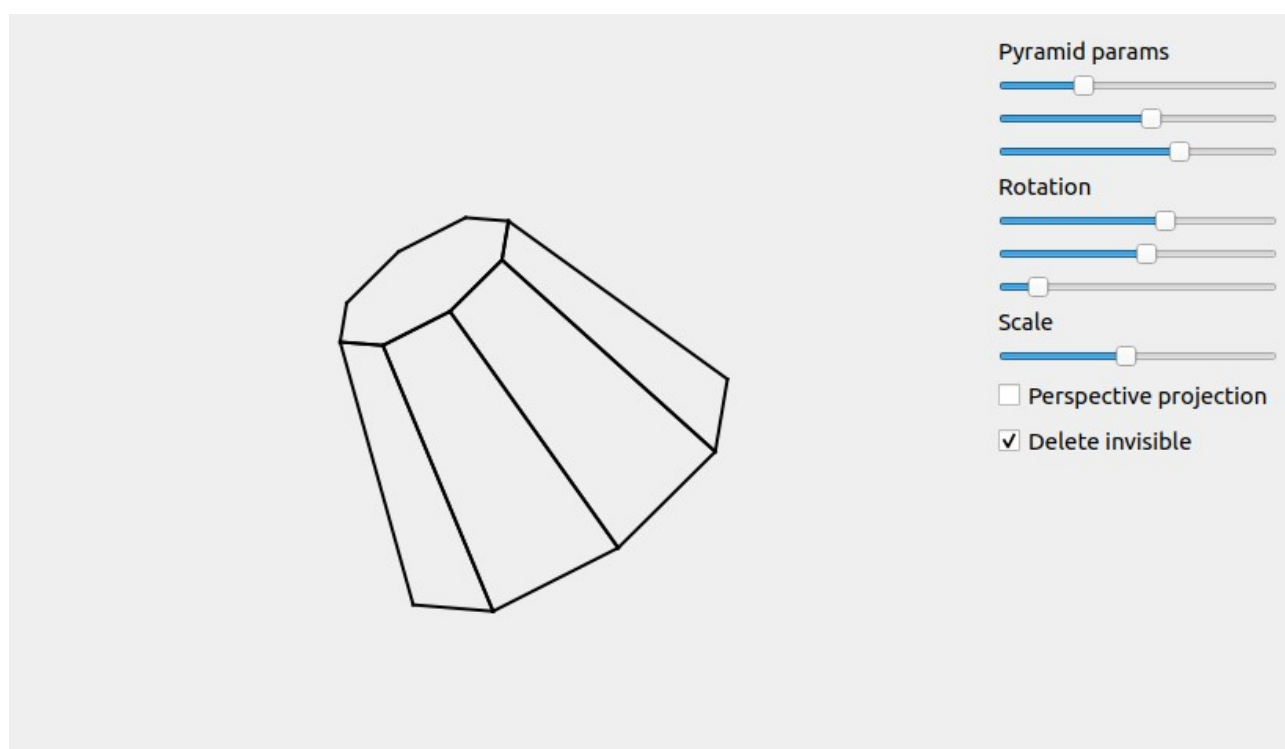
Вид при запуске:



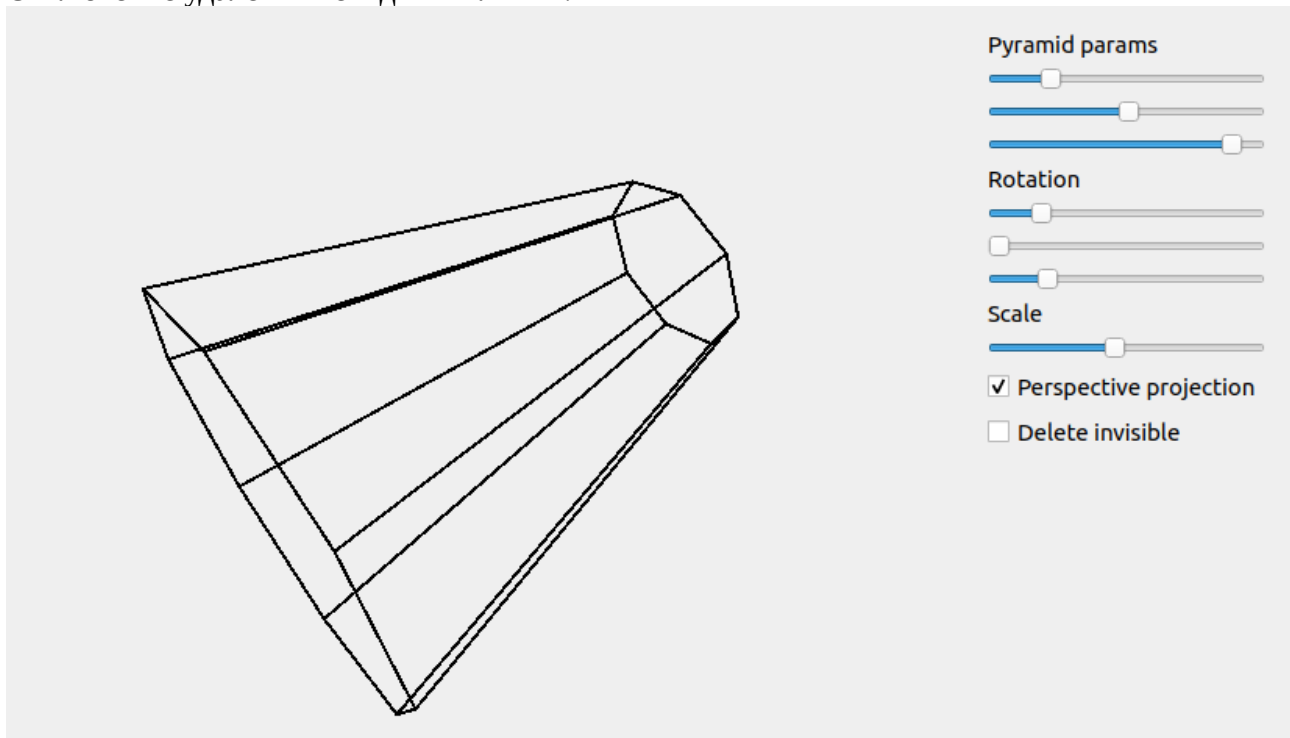
После изменения параметров самой фигуры:



Вращение и отключение перспективной проекции(теперь проекция ортогографическая):



Отключение удаления невидимых линий:



Вывод

В результате работы задача отрисовки трехмерной фигуры на экране была сведена к построению набора отрезков с использованием QPainter. Трансформации фигуры, проецирование и удаление невидимых линий написаны вручную. Такой подход более затратен по времени, уходящем на написание программы, чем использование разнообразных графических библиотек (например, некоторые библиотеки языка Python, которые позволяют буквально в нескольких строках кода отрисовывать сложные фигуры), но позволяет лучше понять процесс.