4. 3D визуализация

На последнем этапе работ был выполнен выбор инструментов для создания 3D модели. Также был произведен анализ классов, содержащихся в модуле Qt Data Visualization, при помощи которого были разработаны собственные классы, выполняющие 3D визуализацию.

4.1 Выбор инструментов для создания 3D модели

Все реализованные ранее части приложения разрабатывалась под Qt. И было бы намного удобнее реализовать новый метод также с использованием Qt. Удобство использование этого фреймворка обусловлена удобством и простой дальнейшей поддержки этого проекта.

Задача построения 3D модели для какого-то набора данных является довольно распространенной задачей. Следовательно, во фреймворке Qt должны быть уже реализованы модули, позволяющие выполнять 3D визуализацию.

Фреймворк, действительно, содержит 2 модуля, позволяющие выполнить задачу: Qt Data Visualization, Qt OpenGL

Что касается последнего, то с Qt 5 данный модуль считается устаревшим, а со следующей версии он был убран. Так как разрабатываемое приложение планируется поддерживать и дальше, то использование этого модуля для разработки является нецелесообразным.

4.2 Анализ классов модуля Qt Data Visualization

Анализ классов для данного модуля производился при помощи рассмотрения диаграммы классов, представленной на рисунке 4.1

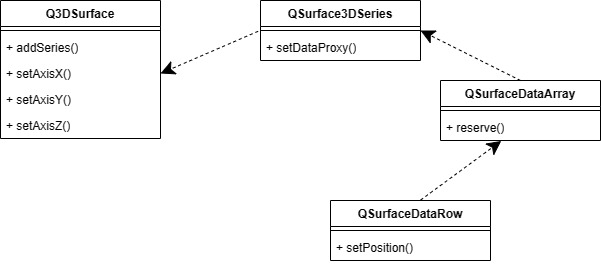


Рисунок 4.1 – Диаграмма классов модуля Qt Data Visualization

В модуле Data Visualization для работы с 3D графикой разработан класс Q3DSurface, его основной задачей является рисование 3х-мерной модели вместе с системой координат. Для задания системы координат используются методы setAxisX, setAxisY, setAxisZ. Для рисования 3D модели используется метод addSeries.

Для отрисовки данных их необходимо сформировать и передать обратно в класс Q3DSurface. Для этой передачи данных был реализован класс QSurface3DSeries, содержащий метод setDataProxy. А для формирования данных используется класс QSurfaceDataArray.

Так как любая 3х-мерная модель состоит из набора 3х-мерных данных, то в модуле был разработан класс QSurfaceDataArray, который сохраняет весь набор данных в 3х-мерный массив для дальнейшего использования. Кроме того, был разработан класс QSurfaceDataRow для создания 3х мерной точки, которая впоследствии записывается в 3х-мерный массив.

4.3 Разработка собственных классов

Диаграмма разработанных классов с использованием классов модуля представлена на рисунке 4.2

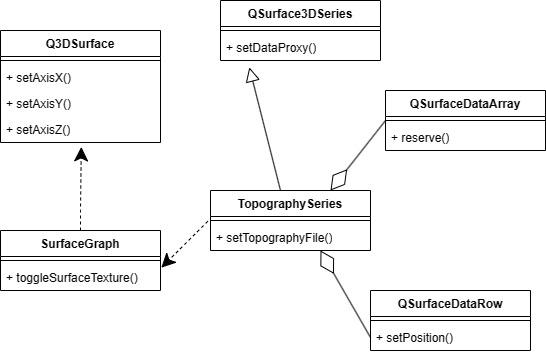


Рисунок 4.2 - Диаграмма разработанных классов с использованием классов модуля

Данные для 3х-мерной модели формируются за счет преобразования каждого тайла, используемого при расчетах, в 3х-мерный массив. То есть координаты пикселя вместе с номером тайла в массиве преобразовываются в координаты X, Z. А значение цвета пикселя, конвертируется в высоту и будет являться координатой Y.

Для формирования данных был разработан класс TopographySeries, который является наследником от класса QSurface3DSeries и в котором добавлен метод setTopographyFile, который и занимается получением данных из тайлов.

Для создания 3х-мерной точки используется класс QSurfaceDataRow, экземпляр которого создается внутри TopographySeries и в который записывается значение полученная в ходе конвертации значений точка. А для сохранения этой точки в массив используется QSurfaceDataArray, экземпляр которого создается внутри TopographySeries.

Далее набор данных передается для отрисовки, но так как QSurface3DSeries был изменен через создание наследника - нового класса, то вместо Q3DSurface был разработан новый класс SurfaceGraph, который вызывает методы TopographySeries для получения и формирования данных для 3D модели. Создание координатных осей остается без изменений, поэтому методы для их создания вызываются из экземпляра класса Q3DSurface.

Листинг кода приведен в приложении Б.

Результат построения 3D модели рельефа для области представлен на рисунке 4.2

Рисунок 4.2 - Результат построения 3D модели рельефа для области