Для расчета радиотрассы выбираются 2 точки на карте. Эти точки – координаты объектов, между которыми планируется организовать канал связи. Далее вычисляются координаты точек, через которые проходит маршрут радиотрассы. Вычисление координат точек происходит по уравнению прямой, где шаг задается самостоятельно из соображений точности и уровня детализации карты. Чем точнее должен быть расчет радиотрассы, тем больше точек нужно взять, уменьшив расстояние между ними. При детализации другая проблема: если один пиксель – 92 м, то запрашивать каждые 3” бессмыслено, так как будем попадать в одну и ту же точку.

После того, как значения всех координат маршрута радиотрассы вычислены, строится профиль радиотрассы. То есть для каждой координат маршрута определяется значение высоты, это значение высоты наносится на график. Тут важно отметить, что высоты откладываются относительно нулевой линии. Нулевая линия в свою очередь рассчитывается по формуле:

…тыц тут формула+ пояснения что это значит

Далее строится линия связи. Для построения линии связи используются 2 точки: первая точка – высота рельефа первого объекта + высота антенны, а вторая точка - это высота рельефа второго объекта + высота антенны.

Пересечение линии связи профиля радиотрассы говорит о невозможности прохождения сигнала. Если же линия не касается профиля радиотрассы, значит сигнал может пройти.

Сервер карт состоит из 2 частей: сервера и клиентской части. Клиентская часть описана в библиотеке с открытым исходным кодом. На основании нее и были сделаны все выводы о структуре сервера карт, о взаимодействии с приложением, о формировании запроса и о получении ответов.

Еще тут надо сказать, что у клиентской части есть интерфейс, позволяющий посмотреть карту и узнать координаты позиции тайлов

Приложение может взаимодействовать только с клиентом. В качестве запроса оно отправляет размер виджета, который обязательно должен быть заполнен картами. Далее клиент получает размеры виджета, на их основании рассчитывает позицию тайла карты и отправляет запрос на сервер. Сервер выдает изображение в формате qpixmap, но с разными размерами. Если тайл с такими координатами есть, то изображение вернется размерами 256х256, если же нет, то размерами 0х0 (пустой тайл)

Существующая библиотека была разработана для работы с приложениями, реализующими работу с областью карты посредством отображения этой области в некотором виджете. Это не совсем подходит для решения нашей задачи, но мы можем посмотреть, что можно изменить или дописать к этой библиотеке, что позволит решить нашу задачи. Для того, чтобы это посмотреть необходимо сделать детализацию структуру клиентской части, рассмотрев последовательность формирования запроса и получения ответа.

\*Диаграмма последовательностей

Инициатором запроса является приложение. Так как ему для дальнейшей работы нужна область карты, то он делает запрос к серверу карт, а конкретно к mapControl, передавая размер виджета, в котором будет отображаться область карты.

mapControl занимается

При получении управления mapControl инициализирует обращение к LayerManager, передавая в качестве параметров.

При первом запросе от приложения мы настраиваем взаимодействие с клиентом. Делается это при помощи передачи ссылки на экземпляр класса QPaintEvent, который содержит в себе параметры для событий рисования и принадлежит приложению.

В свою очередь события рисования отправляются виджетам, которым необходимо сделать перерисовку содержимого. Например, при передвижении карты.

mapControl связывает класс QWidget с экземпляром класса QPaintEvent. Это нужно для того, чтобы любые изменения на виджете, сразу обрабатывалась QPaintEvent. Но так как основная задача – отрисовка карты, то mapControl передает управление LayerManager, передав в качестве параметра ссылку на объект класса QPainter, занимающийся отрисовкой объектов. И который необходим для отрисовки области карты

layerManager необходим, потому что карта может представлять собой несколько слоев. Например, гибридная карта создает при помощи наложения osm-карты, где обозначены границы государств и подписаны названия городов, на спутниковую карту. И для правильной отрисовки такой карты просто необходим layerManager.

При принятии управления layerManager передает управление Layer, делая это для каждого слоя. Передавая в качестве параметров ссылку на еще один экземпляр класса QPainter, который необходим для отрисовки одного изображения, и координаты центра карты.

Кроме того, layerManager соединяет все тайлы, создавая тем самым область карты и отрисовывает ее поверх виджета при помощи Qpainter, который был передан от mapControl.

При получении управления layer вычисляет значения позиции тайла через смещения.

\*схема алгоритма + рисунок, что получается в итоге

Далее для всех вычисленных позиций идет проверка наличия тайлов в кэше. Если есть то, тайл извлекается из кэша, отрисовывается и отсылается. Если же нет, то происходит передача управления ImageManager.

В свою очередь imageManager делает запрос на сервер, указывая в качестве параметров позицию тайла и увеличение.

… Описать последовательность ответа

Но суть в том, что мы получаем изображение, сохраняем его, а потом перекидываем сигнал, что изображение загружено. Его ловит layermanager, который занимается тем, что запускает метод перерисовки, тем самым запрашивая следующий тайл.

Когда очередь тайлов закончена посылается сигнал об окончании. Его снова ловит layermanager. И снова запускает перерисовку. ХЗ зачем

Пройтись по последовательности еще раз, чтобы описать ее логически

Для решения нашей задачи нам нужно только получать картинки. Отрисовывать их на виджетах не надо. Следовательно, необходимо переписать методы классов, исключив методы отрисовки из них.

\*тут я хочу подробно рассказывать про каждый метод, что вот видите у него есть это, это нам не подходит, поэтому переписываем, для этого делаем наследование.

А вот этот нам подходит, но он ссылается на метод другого клааса, который мы переписывали, значит нам нужно переписать и этот метод, изменив ссылку на другой класс

Кроме того, нам необходимо изменить алгоритм определения позиции тайла так, чтобы он мог работать с географическими координатами.

Для того, чтобы изменять алгоритм опредления позиции тайла, необходимо понимать, в какой проекции мы работает.

Проекции отличаются тем, как они считают Землю (сфера, эллипсоид) и как растягивают карту. В зависимости от того, какая проекция используются разные формулы для перевода географических координат в меркаторские координаты(?).

У нас веб-меркаторская проекция, она очень часто используется для работы с картами в интернете. Для нее формула следующая:

\*тут формула

Если вы не угадали с проекцией, то это скажется в виде погрешности, а при больших увеличениях и в ошибке, при определении позиции тайла.

Мы типо все реализовали. Теперь необходимо проверить, а работает ли это вообще. В качестве проверки будем вводить координаты и получать тайл.

У сервера карт есть графический интерфейс, позволяющий определять позицию тайла. Будем использовать его в качестве сверки.

\*тут будут картинки: что хотели, что получили

Теперь, когда мы поняли, как работает сервер, необходимо создать карту высот, так на сервере ее нет.

Тут есть несколько вариантов:

1. Формат (источник)

\*тут вставить текст из файлика Михаила

В результате выбираем srtm

1. Варианты srtm

- общая карта

- sas planet

Тут проблема в том, что половины карты просто нет

\*вставить рисунок

- qgis встренная карта

Проблема с увеличением (только до 5 слоя, дальше видна только подложка)

- файлы высот

- загрузка в qgis

- полностью

- частями

- рисовка самостоятельно

В итоге выбираем загрузку в qgis часятми

Тут говорим о скрипте, который работает с qgis вместо нас

И также говорим о проблеме, что файлы сохраняются не совсем так, как нам нужно. Имена и каталоги не совпадают. Соу пишем скрипт, который нам все переименует

Загружаем на сервер и смотрим, все ли получилось.

Теперь, когда все закончено необходимо добавить графический интерфейс для взаимодействия с пользователем.

Тут происходит обзор аналогов. Говорим, что чаще всего используются отдельные поля для ввода градусов, секунд, арксекунд.

Ну вывод трассы в отдельное окошечко, которое можно скрыть.

Тестирование интерфейса (?)

Оценка погрешности сравниваем с радио мобиле и открытым приложение на браузере.