**Слайд 1**

Данная магистерская диссертация посвящена разработке прототипа военной online-стратегии на реальных картах с использование данных проекта OpenStreetMap.

**Слайд 2**

Если посмотреть статистику по существующим многопользовательским браузерным играм то можно увидеть что большинство игр это MMORPG (Massively multiplayer online role-playing game) - массовая многопользовательская ролевая онлайн-игра, либо стратегии (пошаговые или реального времени). Суть данных игр сводится к управлению доступными игроку ресурсами для достижения превосходства над своими противниками. Например в военных стратегиях нужно управлять боевыми единицами(юнитами) с целью занять более выгодную по отношению к войскам противника позицию. При столкновении с боевыми единицами противника, как правило, происходит бой, юниты наносят урон друг другу пока один из них не будет уничтожен. Действия происходят на определенной местности, поэтому при разработке игр подобного рода возникает потребность в создании карт игрового пространства, в котором разворачивается действие игры. Карты важны, так как местоположение, рельеф, дороги и наличие ресурсов влияет на действия игрока. Поэтому карты должны быть разнообразны и интересны для игроков. Разнообразие, сложность и непредсказуемость используемых карт очень существенно влияют на сохранение интереса к игре. Еще один аспект игр на реальных картах – образовательный. Игры традиционно не связаны с обучением, но могут вызвать внутреннюю мотивацию учеников. На примере военной стратегии на реальных картах возможно повысить мотивацию учеников к изучению истории или географии. Для создания интересной пользователям игры, целесообразно кроме построения маршрутов по дорогам, включить возможность использовать данные рельефа и автоматически определять факт окружения юнитов игрока. Под окружением юнита понимается такое состояние, что отсутствует путь по дорогам от этого юнита до своей базы обеспечения, такой чтобы на любом его участке он не находился в пределах радиуса действия юнитов противника.

**Слайд 3**

1. Анализ существующих игр использующих реальные географические карты.
2. Анализ возможностей и ограничений картографических сервисов для получения маршрутов движения, данных рельефа и определения окружения.
3. Анализ возможности создания сервисов высотных данных, погодных данных и сервиса маршрутов, выбор платформы для создания этих сервисов и прототипа игры.
4. Создание прототипа online стратегии на реальных географических картах, использующей данные рельефа, погодные данные и граф дорожной сети для построения маршрутов и определения окружения юнитов.

**Слайд 4**

Monopoly City Streets — глобальная онлайн-игра на основе Google Maps.Разработчик: [Tribal DDB](http://en.wikipedia.org/wiki/Tribal_DDB).Жанр: многопользовательская онлайн стратегия реального времени.

Фиолетовым цветом отмечены улицы, доступные для покупки, а синим — уже принадлежащие другим игрокам. На данном скриншоте представлена свободная для покупки улица (так же показано предложение найти ближайшее здание оппонента и взорвать его. В игре есть три типа рейтинга: локальный (городской), региональный(рейтинг по стране), мировой. Цель игры - стать первым в рейтинге.

**World of the Living Dead**

Разработчик: Ballardia.

Жанр: РПГ, стратегия

Действие игры разворачивается постапокалиптическом Лос-Анджелесе, захваченном зомби. Игроку предстоит командовать несколькими отрядами оставшихся в живых людей, бродящими по пустынным улицам и борющимися за выживание. Для отображения игрового мира ранее использовался движок Google Maps, позднее разработчики перешли на Open Street Map.

**War2map**

Разработчик: Маттео Муратори

Жанр: смесь казуальной игры с военной многопользовательской онлайн стратегией реального времени.

В игре, пользователи могут развертывать войска, строить инфраструктуру, расширять и защищать территории, собирать налоги, торговать игровыми объектами с другими пользователями по всему миру. Сражаясь с другими игроками, игрок набирает очки опыта. Внутриигровые деньги производятся налоговыми органами, расположенными на территории подконтрольной игроку. Цель игры развитие инфраструктуры, войск, экономики расширение территории.

**Parallel Kingdom**

Разработчик: PerBlue

Жанр: ролевая игра (РПГ)

Parallel Kingdom (Параллельное Королевство)-игра подобная WoW, основанная на Google Maps для Android и IOS, а так как эта игра не нуждается в получении данных о координатах пользователей, то может быть воспроизведена на ПК или Mac с помощью симулятора, например, BlueStacks. Игра просто использует Google Maps, как фон, однако вызвала интерес нескольких миллионов пользователей. Это доказывает, что даже такая малая связь с реальным миром привлекает несколько миллионов игроков[3]! В Параллельном Королевстве пользователь собирает предметы и ресурсы, охотится монстров и устанавливает расширяет области своего царства.

**Слайд 5**

Картографические сервисы были созданы как ответ на отсутствие свободных географические данных[5]. Google запустила свой сервис онлайн-карт в феврале 2005 года. Google, отличается от конкурентов более удобным интерфейсом, качественной поддержкой и большей интерактивностью. В октябре 2011 года Google решила начать взимать плату за доступ к

Google Maps API один раз в день при превышении пределов использования.

Чем более популярен сайт или приложение, тем больше его риски того, что придется платить за то, чтобы продолжать отображать карту Google. API предоставляет ряд служебных программ для управления картами и добавления содержания на карту с помощью разнообразных служб, что позволяет создавать на веб-сайтах надежные приложения, использующие карты. API также содержит средства для вычисления маршрутов, определения высот отдельных точек, групп точек и путей, вычисления рассояний между точками и др.

JavaScript Maps API V3 – это бесплатная служба, доступная для всех веб-сайтов, бесплатных для потребителей.

**Ограничения картографического сервиса Google**

Картографические сервисы подобные Google имеют серьезные ограничения, препятствующие полноценному использованию их в популярной игре.

На использование службы API маршрутов Google накладывается ограничение, составляющее 2500 запросов маршрутов в день. Поиск автомобильного, велосипедного или пешеходного маршрута учитывается при расчете ежедневной квоты как один запрос. Поиск маршрута общественного транспорта учитывается как 4 запроса. Отдельные запросы автомобильных, велосипедных или пешеходных маршрутов могут содержать до 8 промежуточных путевых точек. Пользователи API Google Карт для организаций в течение дня могут запрашивать до 100 000 маршрутов, для каждого из которых допускается до 23 путевых точек. Для маршрутов общественного транспорта путевые точки не поддерживаются.

API маршрутов можно применять только для отображения результатов на карте Google. Использование данных службы маршрутов без отображения карты, для которой они запрашивались, запрещено. Кроме того, при вычислении маршрутов формируются данные об авторских правах и предупреждения, которые каким-либо способом должны быть показаны пользователю.

API v3 имеют дневной лимит использования 25000 просмотров. Примерно 0,32% пользователей API превышают лимит[5]. Эта политика Google привела к тому что много разработчиков приложений отказались от Google Maps в пользу OpenStreetMap.

На использование API высотных данных Google накладываются ограничения в отношении количества запросов: не более 2500 в день (для пользователей API Карт для организаций – не более 100 000 в день). В каждом запросе разрешается указывать до 512 местоположений, но не более 25 000 в день (для пользователей API Карт для организаций – 1 000 000). Это ограничение предотвращает злоупотребление интерфейсом API высотных данных и его использование в иных целях. Ограничение может быть изменено в будущем без предварительного уведомления. Кроме того, для предотвращения злоупотреблений службой существует ограничение на количество запросов. Если превышено суточное (24 часа) ограничение или имеет место злоупотребление другого рода, API высотных данных может временно прекратить работу. При неоднократных нарушениях доступ к нему может быть заблокирован.

Для запросов по одной точке API высотных данных возвращает данные с максимальной точностью. Для пакетных запросов по нескольким точкам могут быть возвращены менее точные данные, особенно если эти точки удалены друг от друга, так как выполняется сглаживание данных.

Кроме того, прежде чем выполнять кодирование адресов URL нужно убедиться в том, что адреса URL для API высотных данных содержат не более 2048 символов. API высотных данных можно использовать только для отображения результатов на карте Google. Использование данных службы высотных данных без отображения карты, для которой они запрашивались, запрещено

API Карт это программный интерфейс, с помощью которого можно установить Яндекс.Карты и необходимый инструментарий для работы с ними в своём веб-приложении или на сайте[88]

JavaScript API

Static API Карт

Геокодер Маршрутизатор

**Ограничения API Яндекс.Карт[28]**

* API Яндекс.Карт предназначен для отображения карты реальным пользователям. Нельзя применять его только для получения и обработки данных, например геокодирования точек или прокладки маршрутов без вывода их в интерфейсе.
* Недопустимо скрывать или менять логотипы и копирайты на карте, а также загораживать их другими элементами.
* Бесплатный API Яндекс.Карт можно использовать только на общедоступных сайтах и в бесплатных приложениях. У сервисов с авторизацией должна быть открытая и бесплатная регистрация.
* Бесплатный API Карт нельзя использовать в модулях бухгалтерских программ, интранетах и других закрытых системах, десктоп-приложениях и программах, игровых проектах, а также системах мониторинга и диспетчеризации транспорта.
* Также в бесплатном API есть ограничение — в сутки можно делать не более 25 000 запросов к геокодеру.

Геокодирование и Маршрутизатор:

* Функция геокодирования может быть использована только для сайтов или мобильных приложений, использующих Сервис. Пользователь имеет право обрабатывать полученные координаты исключительно для отображения их посредством Сервиса. Количество обращений к функции геокодирования ограничено 25000 запросов для одного сайта/мобильного приложения в сутки.
* Маршрутизатор может быть использован только для сайтов или мобильных приложений, использующих Сервис. Количество обращений к Маршрутизатору ограничено 25 000 запросов для одного сайта/мобильного приложения в сутки.

Используя Сервис, Пользователь не имеет права:

* Создавать на основе Сервиса игровые проекты.

Таким образом, компания Яндекс напрямую запрещает использлвание своего сервиса для создания игр.

OpenStreetMap (дословно «*открытая карта улиц*»), сокращённо OSM — некоммерческий веб-картографический проект по созданию силами сообщества участников-пользователей Интернета подробной свободной и бесплатной географической карты мира. Проект охватывает всю поверхность земного шара. Главной целью проекта является построение не собственно карты, а базы данных, содержащей сведения о точках на земной поверхности. Таким образом, на основе собранных в рамках проекта данных можно создавать карты различного вида и другие сервисы.

Проект OSM был начат, потому что большинство доступных карт, таким Google или Bing, имеют юридические или технические ограничения на их использование: много контента недоступно из-за авторских прав, принадлежащих картографическим компаниям (например, NAVTEQ или TeleAtlas) или национальным органам (например, Ordnance Survey from United Kingdom или IGN - National Geographic Institute). OpenStreetMap в этом отношении выгодно отличается. Все известные данные предоставлены в открытом доступе, так же, как в Википедии. Таким образом, две основные движущие силы, стоящие за созданием и ростом OSM были ограничения на использование или доступность картографической информации большей части мира и появление недорогих портативных устройств спутниковой навигации. OSM является одним из самых активных в мире открытых проектов с более чем 1,5 миллионов зарегистрированных редакторов

**Доступные API**

OpenStreetMap предоставляет несколько API для создания, изменения и чтения географических данных. Кроме того, некоторые API предоставляет расширенные возможности для запросов (например, геокодирования или обратного геокодирования).

Доступны API приведены ниже:

* API: OpenStreetMap API для редактирования, поиска и сохранения сырых геоданных в базе данных OpenStreetMap.
* XAPI: OpenStreetMap API (только для чтения), основанный на модифицированной версия стандартного API, который обеспечивает улучшеный поиск и более широкие возможности запросов.
* Nominatim: OpenStreetMap поисковая система для поиска данных OSM по

названию и адресу и преобразования адресов обратно в OSM точки (т.е. обратного геокодирования).

**Модель данных**

Для того чтобы использовать API, существует необходимость, чтобы объяснить модель данных, которая используется OpenStreetMap. OSM данные хранятся в формате XML. В основном, это список экземпляров трех примитивов данных[5]:

* Node - узел, одна геопространственная точка.
* Way, путь - взаимосвязь, по крайней мере 2 узлов, которые описывают
* линейная фигуру (например, улица, тропинка, железная дорога, река, ...).
* Relation, используется для группировки путей или узлов объектов, которые географически связаны друг с другом (т.е. соединены или рядом друг с другом).
* Карты OpenStreetMap двумерные, без отображения высот над уровнем моря, изолиний. Однако существуют проекты, которые отображают рельефные карты, используя данные о высотах из сторонних свободных источников.

**Слайд 6**

Алгоритм Дейкстры, например, вычисляет расстояние до всех возможных точек в сети, которые находятся ближе, чем цель, в достижении которой мы заинтересованы. Очевидно, не имеет смысла вычислять все эти расстояния, если мы заинтересованы только в пути между двумя точками. Поэтому разработаны и продолжают разрабатываться более быстрые методы - так называемые иерархические подходы(Contraction hierarchies) [7]. Такие методы предполагают разделение работы на две части. Во время первой фазы, называемой фазой предварительной обработки (preprocessing) вычисляются дополнительные данные, которые ускоряют запросы в ходе второй фазы. Используя априорно известные свойства транспортной сети, быстрые алгоритмы позволяют получить оптимальный маршрут за миллисекунды, ценой нескольких часов предобработки. При предобработке узлам графа назначается разный уровень иерархии, при назначении уровня иерархии могут применяться разные подходы, например может учитываться тип дороги и т.д. Алгоритм поиска обычно двунаправленный алгоритм Дейкстры) устроен так, что (при поиске пути принимает во внимание только те дуги, которые ведут в узлы с более высокой иерархией в одном направлении и наоборот в другом направлении. Из-за этого ограничения алгоритм не сможет найти пути на необработанном предварительно графе. Поэтому, по определенным правилам в граф добавляются дополнительные дуги, называемые «кратчайшими», которые представляют существующие кратчайшие пути в графе. Когда предобработка закончится, мы получим граф, в котором узлы имеют иерархию и добавлены дополнительные «кратчайшие» дуги. Для поиска может использоваться двунаправленный алгоритм Дейкстры. Это классическоий алгоритм Дейкстры с некоторыми изменениями. Алгоритм ищет с начального узла в одном направлении и от конечного узла в другом направлении (это алгоритм классической двунаправленного Дейкстры), но он использует дуги, которые направлены в сторону вышележащих узлов в иерархии в одном направлении и дуги, которые направлены к нижележащим узлам в иерархии в другом направлении. Если кратчайший путь существует, эти два поиска встретятся в каком-то узеле *V*. Кратчайший путь из *S* в *T* состоит из путей от *S* до *V* и от *V* до *Т*.

**Слайд 7**

Spatialite[21] это однопользовательская СУБД на базе SQLite с поддержкой пространственных данных и алгоритмов их обработки. Также, Spatialite позволяет подключать источники данных в формате CVS/TXT и shape-файлы ESRI [29]. Для установки достаточно скачать и установить оболочку и инструменты с сайта разработчика.( https://www.gaia-gis.it ) Spatialite включает в себя программу импорта дорожной сети в формате файла OSM, после чего позволяет построить на базе имеющейся информации граф дорожной сети. Предоставляет функциональность для работы с геоданными, как PostGIS или Oracle Spatial. Но в отличии от последних не использует клиент-серверную архитектуру, то есть все операции выполняет само приложение. Придерживаясь архитектуры SQLite, база данных представляет собой один файл, в котором хранятся все данные (исключением является не монопольный режим, когда ещё создается файл журнала), который может быть свободно скопирован, или удален.

SpatiaLite имеет свою собственную реализацию индексов (R-деревья) и геометрических типов, которые позволяют делать пространственные запросы.

SpatiaLite, доступен для Linux и Windows, в виде библиотеки. В комплекте идут несколько важных утилит, включая GUI для манипулирования SpatiaLite. [22]

Spatialite включает в себя программу импорта дорожной сети в формате файла OSM, после чего позволяет построить на базе имеющейся информации граф дорожной сети.

**Слайд 8**

GTOPO30 является глобальной цифровой моделью рельефа (ЦМР) с горизонтальным шагом сетки 30 угловых секунд (примерно 1 км). GTOPO30 была получена из нескольких растровых и векторных источников топографической информации. Данные по рельефу поверхности Земли доступны для свободной загрузки на сайте NOAA. National Centers for Environmental Information (Национальные центры информации об окружающей среде, ранее назывался National Center Geophisical Data - Национальный центр геофизических данных). Данные доступны в различных форматах, включая формат **xyz**, это формат такого вида: долгота<пробел> широта<пробел> высота<конец строки>. Высота считается от уровня моря

**Слайд 9**

Для использования в игре реальных погодных данных (мы будем использовать не данные реального времени, а архивные данные) нужно иметь базу, содержащую климатические данные, измеренные на метеорологических станциях по всему миру, местоположение которых известно. National Center for Environmental Information (USA) предоставляет свободный доступе к файлам с климатическими данными с метеорологических станций, входящих в Всемирную Метеорологическую Организацию.

Всемирная метеорологическая организация (ВМО, англ. World Meteorological Organization, WMO, фр. Organisation météorologique mondiale, OMM) — специализированное межправительственное учреждение Организации Объединённых Наций в области метеорологии. Основано в 1950 году. Является компетентным органом ООН по вопросам наблюдения за состоянием атмосферы Земли и её взаимодействия с океанами.[26]

Нас интересуют данные на поверхности Земли, они доступны по FTP по адресу: <ftp://ftp.ncdc.noaa.gov/pub/data/gsod>. Данные расположены по каталогам в соответствии с годом. Например, данные за 2014 год находятся по адресу: <ftp://ftp.ncdc.noaa.gov/pub/data/gsod/2014/>. В каталоге расположены заархивированные файлы, а каждом файле содержатся данные по одной станции за год.

**Слайд 10**

Так как игра браузерная, то на клиентской стороне использован язык JavaScript. JavaScript — прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами[9]. Ознакомиться с основными конструкциями языка можно обратившись к одному из многочисленных руководств.[9]

Node или Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи node-webkit и AppJS для Linux, Windows и Mac OS) и даже программировать микроконтроллеры (например, tessel и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом

Express - минималистичный и гибкий фреймворк для nodejs, предоставляющий широкий набор функций для веб и мобильных приложений

Socket.IO — JavaScript библиотека для веб-приложений и обмена данными в реальном времени. Состоит из двух частей: клиентской, которая запускается в браузере и серверной для node.js. Оба компонента имеют похожее API. Подобно node.js, Socket.IO событийно-ориентированная, поэтому удобна для создания игр. Клиент и сервер могут эмитировать события и обрабатывать события. При эмитировании событий могут посылаться данные, которые будут доступны в обработчике события.

Socket.IO главным образом использует протокол WebSocket, но если нужно, использует другие методы, например Adobe Flash сокеты, JSONP запросы или AJAX запросы[4], предоставляя тот же самый интерфейс. Помимо того, что Socket.IO может быть использована, как оболочка для WebSocket, она содержит много других функций, включая вещание на несколько сокетов, хранение данных, связанных с каждым клиентом, и асинхронный ввод/вывод

Для построении клиентской части игры работы с картами была выбрана open source JavaScript библиотека Leaflet. (http://leafletjs.com/), разработанная командой разработчиков во главе с Владимиром Агафонкиным.

Библиотека имеет богатый функционал, простой синтаксис, высокую производительность. Она работает эффективно на всех основных настольных и мобильных платформах используя HTML5 и CSS3 на современных браузерах, но в то же время способна работать на старых. Она может быть расширена с помощью огромного количества плагинов, имеет красивый, простой в использовании и хорошо документированный API.

SQLite является библиотека в процессе, который реализует автономные, без серверов, нулевой конфигурации, транзакционной базы данных двигателя SQL. Код для SQLite находится в общественном достоянии, и, таким образом, бесплатно для использования в любых целях, коммерческих или частных. SQLite является наиболее широко распространенным в мире база данных с несколькими приложениями, чем мы можем рассчитывать, в том числе несколько громких проектов.

SQLite является встроенный движок базы данных SQL. В отличие от большинства других баз данных SQL, SQLite не имеют отдельный серверный процесс. SQLite читает и пишет прямо на обычных дисковых файлов. Полная база данных SQL с несколькими столами, индексы, триггеры и представления, содержится в одном файле на диске. Формат файла базы данных является кросс-платформенным - вы можете свободно копировать базу данных между 32-разрядных и 64-разрядных систем или между тупоконечника и мало-обратным порядком байтов архитектуры. Эти особенности делают SQLite популярным выбором в качестве формата файла приложения. Подумайте SQLite не в качестве замены для Oracle, но в качестве замены для FOPEN ()

SQLite является компактная библиотека. При включении все функции, библиотека размер может быть меньше, чем 500KiB, в зависимости от целевой платформы и оптимизации компилятора настроек. (64-битный код больше. А некоторые оптимизации компилятора, такие как агрессивной функции встраивания и разворачивания цикла может привести объект код будет гораздо больше.) Если дополнительные функции опущены, размер SQLite библиотеки могут быть меньше 300KiB. SQLite также может быть сделано, чтобы работать в минимальном пространстве стека (4KiB) и очень мало кучи (100KiB), что делает SQLite популярным выбором двигателя базы данных по памяти ограничен устройств, таких как сотовые телефоны, КПК и MP3-плееров. Существует компромисс между использованием памяти и скорости. SQLite обычно работает быстрее, тем больше памяти вы даете его. Тем не менее, производительность, как правило, довольно хорошо, даже в среде с низким памяти.

**Слайд 11**

Игрок выбирает для игры миссию, в которой уже заранее предопределены для обоих противников их юниты, раположение, тип. Юниты делятся на боевые подразделения и базы снабжения. В ходе игры юниты расходуют часть ресурсов, и восполняют их за счет баз снабжения. Как только юнит попадает в окружение(это происходит если не существует не перекрытых дорог до хотя бы одной базы снабжения), снабжение прерывается и ресурсы юнита уменьшаются. Если в течении определенного времени окружении не будет снято, ресурсы юнита будут исчерпаны и он погибнет. При сближении юнитов противных сторон возникает бой. В процессе боя юниты наносят урон друг другу в соответствии со своей боевой мощью, которая зависит от ряда факторов. Бой может длиться до уничтожения одного из юнитов. Игрокам требуется управлять своими юнитами для того чтобы уничтожить всех юнитов противника. В этом случае наступает победа и игра заканчивается.

**Слайд 12**

по картинке

**Слайд 13**

по картинке

**Слайд 14**

по картинке

**Слайд 15**

- помечаем все узлы графа как непосещенные;

- определяем узлы относящиеся к своим базам снабжения;

- определяем узлы(дуги), перекрытые вражескими юнитами;

- «запускаем волну» со всех узлов, принажлежащим к нашим базам, при этом узлы куда «пришла волна» помечаются как посещенные;

- для каждого нашего юнита определяем узлы графа, которые он занимает, и если среди них нет посещенных, то юнит считается окруженным.

Игровой сервер периодически посылает на сервер расчета окружения данные о местоположении юнитов и баз, тот производит расчет и возвращает ответ, в котором указано, юниты с каким id окружены. На основании этого ответа игровой сервер инициализирует соответствующие свойства в объектах юнитов серверного объекта игры, с которым синхронизируются клиентские объекты игры. Это отражается в пользовательском интерфейсе

**Слайд 16**

Сервис принимает POST-запросы от игрового сервера, в которых содержатся координаты местоположения юнитов. Модуль высотных данных на основании координат, заданных как массив точек такого вида: [[lat1,lng1], [lat2,lng2],...] , формирует SQL запрос к базе *sqlite,* содержащей данные по высотам точек на планете. База высотных данных создается предварительно, отдельно написанным модулем, который производит разбор файла с высотными данными в формате XYZ, о котором было написано в главе 4.4. и записывает их в базу данных формата sqlite3. Так как данные в базе содержатся по ограниченному набору точек с шагом примерно 1 угловую минуту, то при запросе высоты точки производится поиск одной из ближайших точек. Так как объем данных достаточно велик (размер файла базы составляет порядка 6,5 Гб) для ускорения выполнения запроса данные разбиты по градусам широты в отдельные таблицы. В результате запроса получаем наборы координат точек и соответствующих им высот, который сервер высотных данных возвращает игровому серверу. Ответ представляет собой массив объектов вида: {lat:lat, lng:lng, elevation: elevation}.

Игровой сервер модифицирует объект игры: в соответствующие свойства обьектов юнитов записываются значения высоты ближайшей точки из ответа сервера высотных данных.

**Слайд 17**

Игровой сервер подготавливает массив координат юнитов игры, дату(берется текущая дата и год указанный в описании миссии) и отправляет запрос на сервер погодных данных. На основании координат и даты формируются SQL запросы к базе данных. Сначала по координатам находятся ближайшие к каждой заданной точке метеостанции, далее идет выборка метеоданных для этих станций на указанную в запросе дату. По результатам запроса формируется результат и отправляется игровому серверу.

**Слайд 18**

по картинке

**Слайд 19**

* рассмотрены существующие онлайн игры, использующие реальные карты;
* произведен анализ возможностей и ограничений существующих популярных картографических сервисов в части получения маршрутов и данных рельефа местности;
* проведен поиск open source программного обеспечения для построения своих сервисов маршрутов и высотных данных на основе общедоступных бесплатных географических данных;
* показана возможность использования OSRM как сервиса маршрутов для использования в игре;
* показана возможность получения графа дорожной сети из данных OpenStreetMaps и использования его для построения маршрутов юнитов в игре и определения окружения юниов.
* реализованы сервисы маршрутов, определения окружения, высотных данных и погодных данных;
* реализован прототип военной online стратегии, использующей реальные карты и учитывающей рельеф местности и определение окружения юнитов.