

Десятая независимая научно-практическая конференция

«Разработка ПО 2014»

23 - 25 октября, Москва



Базисы разработки геолокационных приложений

Петерсон Дмитрий



Обо мне

- Зам. директора компании СимбирСофт
- Более 12 лет работы в отрасли ИТ
- Технологическая специализация: телекоммуникации, биллинговые системы, разработка мобильных приложений
- В 2009г. - основал мобильное направление в компании SimbirSoft
- Руководил реализацией более 24 мобильных проектов

Что такое ГЕО приложение?

Геолокационное приложение - приложение, логика работы которого зависит от физического местоположения пользователя данного приложения в пространстве

Геолокационные приложения

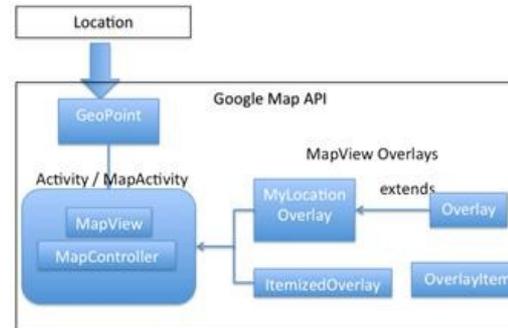
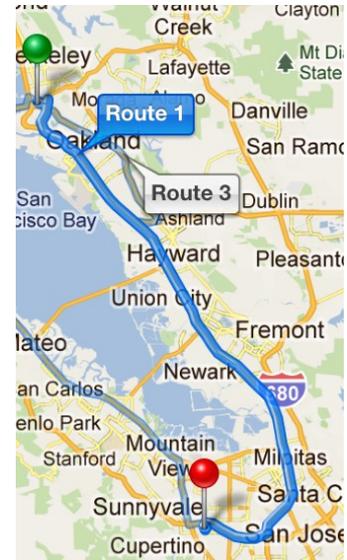
- В 2012 г. мировая аудитория пользователей моб. Приложений достигла 486 млн.
- В 2013 г. Количество скачанных программ данной категории превысило 150 млн.
- По темпам прироста, рынок мобильных гео-приложений и сервисов в период с 2013 по 2016 ежегодно будет расти на 70%

О чем не будем говорить?

- Как использовать стандартный API
- Вопросы которые решаются стандартными API

О чем будем говорить?

- О картах
- Об особенностях работы с аппаратными “ГЕО” датчиками на мобильных устройствах
- Об особенностях построения логики гео-приложений
- Об особенностях размещения информации на картах

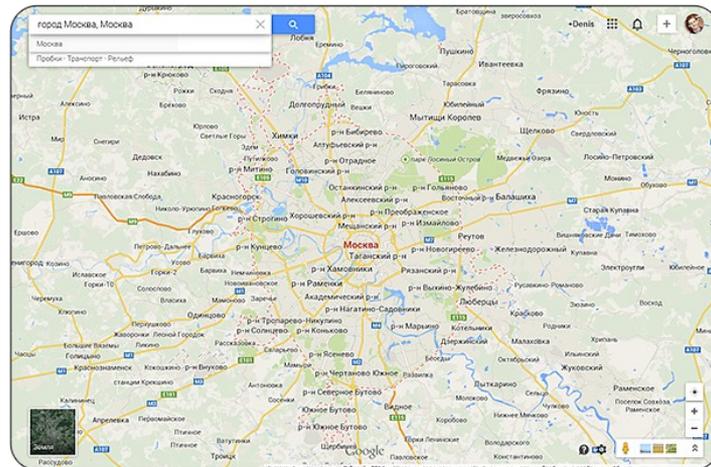
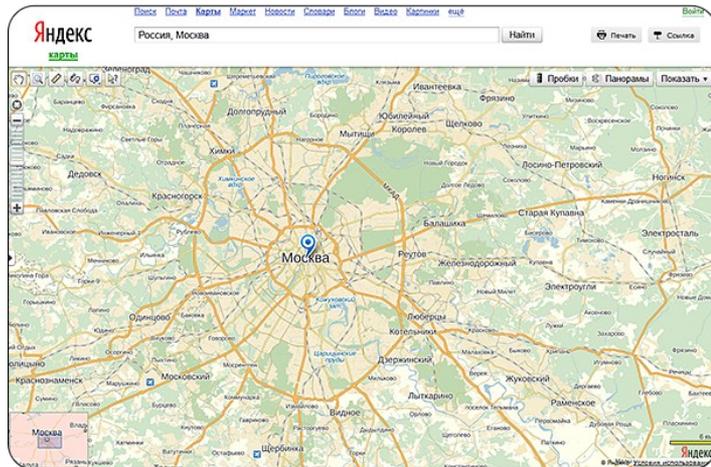


Задача

- Приложение Справочник по Городу
 - Условия работы:
 - Работа без подключения к интернет
 - Карта + текущее местоположение пользователя
 - Кликабельные объекты на карте
 - Информирование пользователя о приближении к точкам интереса (POI)
 - Плавно и быстро



Отображение “картографической” информации в отсутствии подключения к Internet



Проблема, где взять данные*

- Разработать собственную карту
- Приобрести право использования картографических данных
- Использовать открытые данные OpenStreetMap ✓

* нет требования к геометрической точности отображаемых объектов



OpenStreetMap

Формат используемых карт

Векторные

```
<way id="198299632" visible="true" version="1" changeset="1441144114"
<nd ref="2084527206" />
<nd ref="2084527209" />
<nd ref="2084527230" />
<nd ref="2084527224" />
<nd ref="2084527206" />
<tag k="landuse" v="grass" />
</way>
<way id="198299633" visible="true" version="1" changeset="1441144114"
<nd ref="2084527129" />
<nd ref="2084527150" />
<nd ref="2084527186" />
<nd ref="2084527177" />
<nd ref="2084527129" />
<tag k="landuse" v="grass" />
</way>
<way id="198299635" visible="true" version="1" changeset="1441144114"
<nd ref="2084527097" />
<nd ref="2084527098" />
<nd ref="2084527120" />
<nd ref="2084527119" />
<nd ref="2084527097" />
```

Растровые



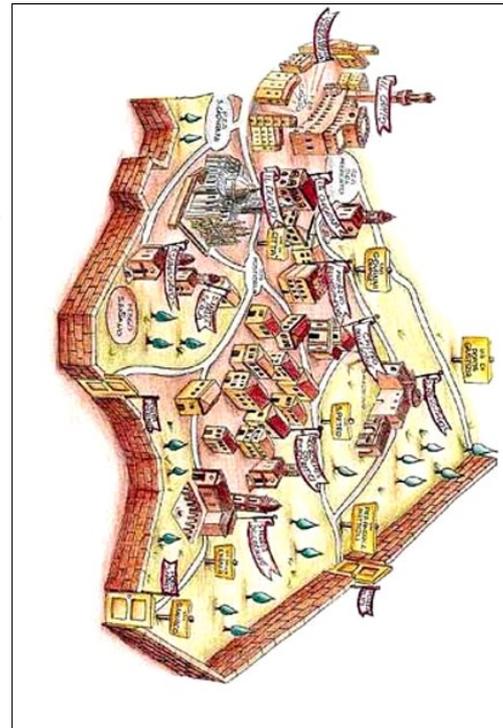
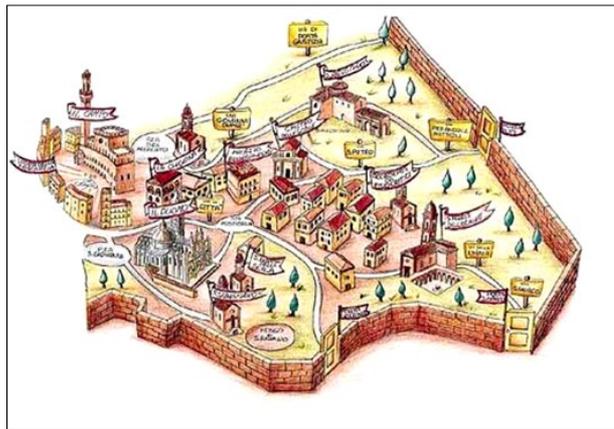
Растровые карты

- + Простота реализации
- + Скорость работы
- + Красочность интерфейса



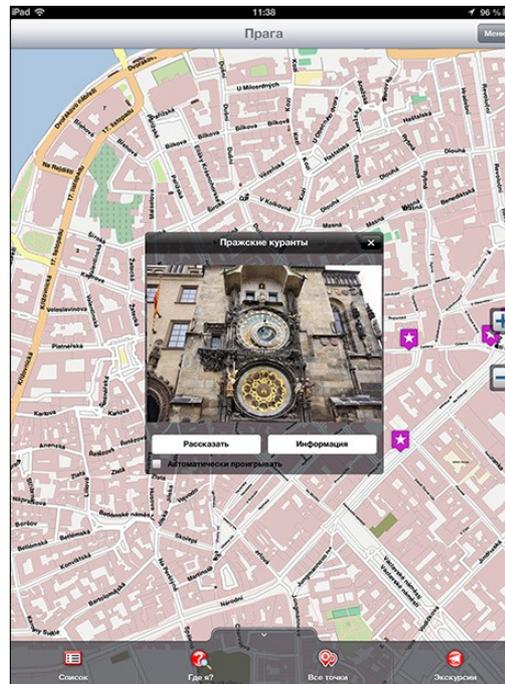
Растровые карты

- Размер данных
- Гибкость, трансформации, вращения

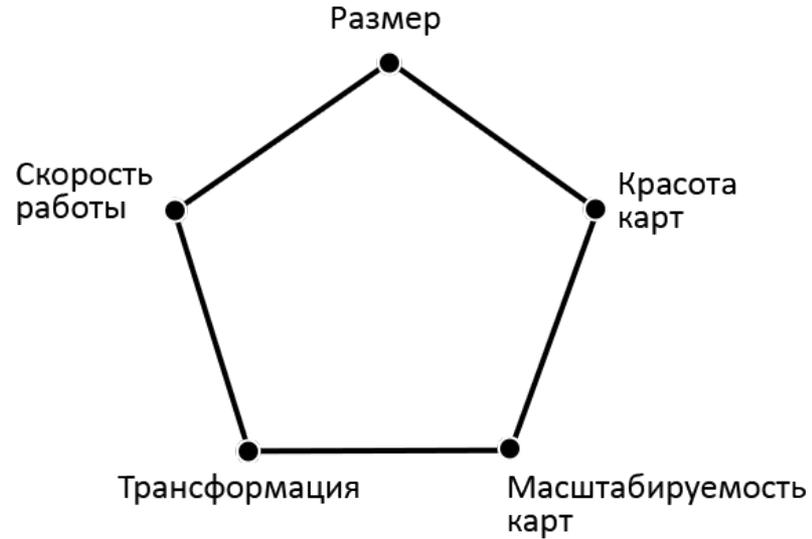


Векторные карты

- + Размер исходных данных
- + Гибкость в работе с картами
- + Масштабируемость
- Сложность реализации
- Скорость работы
- Красочность информации и качество детализации

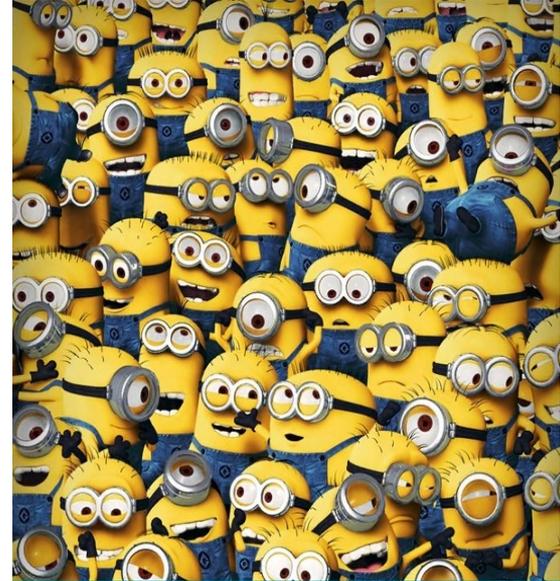


Какой формат используемых карт выбрать?



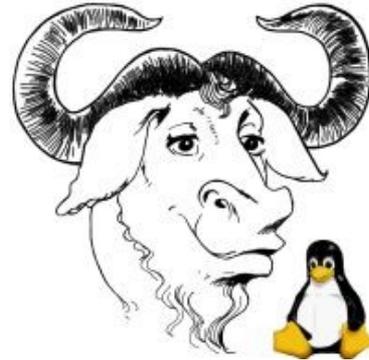
Выбор способа реализации карты

- Собственный рендер карт
 - + Гибкость решения, не типовая логика
 - + Любая лицензия
 - Время разработки
 - Сложность поддержки всех артефактов
 - Поиск ошибок



Выбор способа реализации

- Использование сторонних библиотек
 - + Срок и стоимость внедрения
 - + Возможность доработки
 - Зависимость от заложенной архитектуры
 - Сложность исправлений архитектурных ошибок



Типичные проблемы рендеров

- Отображение береговой линии
- Отображение объектов инфраструктуры
- Отображение сложных геометрических объектов
- Скорость рендеринга и скроллинга карт
- Все решаемо



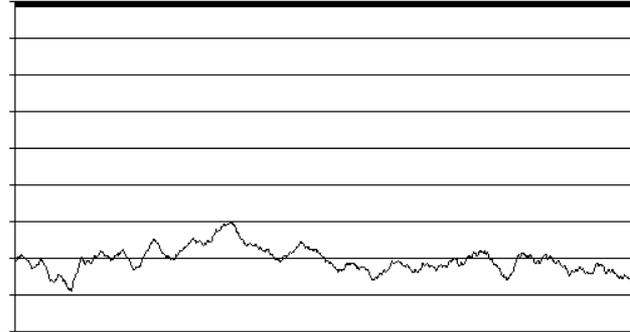
Еще одна задача ГЕО приложений

- Работа с “измерительными” устройствами
- Работа с аппаратными датчиками
 - Акселерометр
 - Гироскоп
 - GPS/Глонасс



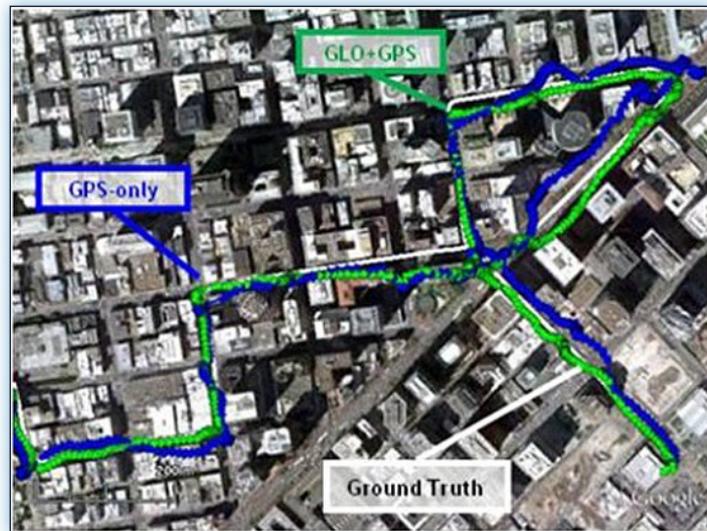
Любой измерительный прибор обладает погрешностью

- В теории
 - Чем сильнее зашумление данных], тем сложнее обработать такую информацию
- В нашем случае
 - Дрожание экрана
 - Резкие перемещения по карте
 - Нервное поведение приложения

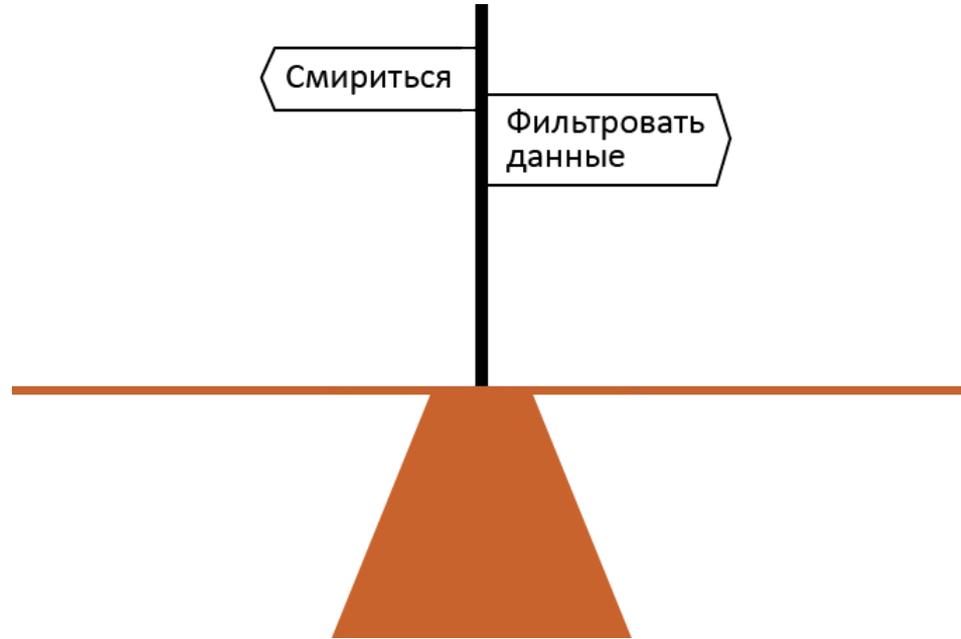


Минусы

- Визуальные дефекты
- Ложное поведение бизнес-логики приложения (точки срабатывают не вовремя)

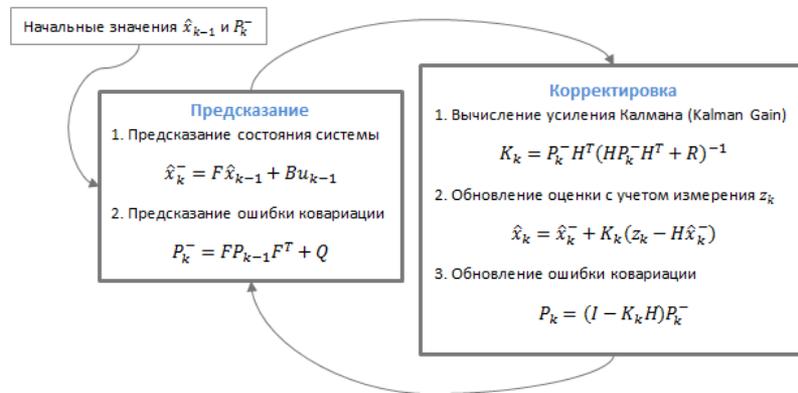


Что делать?



Фильтр Калмана

- Использует динамическую модель системы
- Состоит из двух повторяющихся фаз:
 - Предсказание
 - Корректировка



Упрощенный вариант

- Если используем одну переменную и управляющее воздействие, то матрица превращается в линейное уравнение

Предсказание

$$\hat{x}_k^- = F\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = FP_{k-1}F + Q$$

Корректировка

$$K_k = \frac{P_k^- H}{HP_k^- H + R}$$

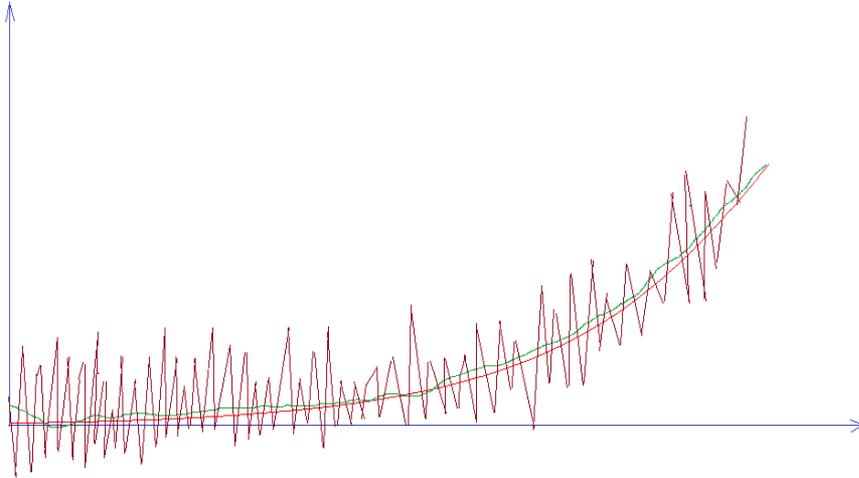
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (1 - K_k H)P_k^-$$

- Настроить фильтр можно через свойства
 - R - ошибка измерения
 - Q- определение шума процесса
- $Q \ll R$ сильное сглаживание и высокая инерционность
- $Q \sim R$ почти без сглаживания

Способы применения Калмана

- Работа с данными GPS, Магнитрометра, Гироскопа и т.п.
- Реализация плавных анимаций
- Работа с изображениями, распознавание



Размещение множества “кликабельных” объектов на карте

- Минимальная область для нажатия 44x44
- Более 1000 объектов
- На большом масштабе



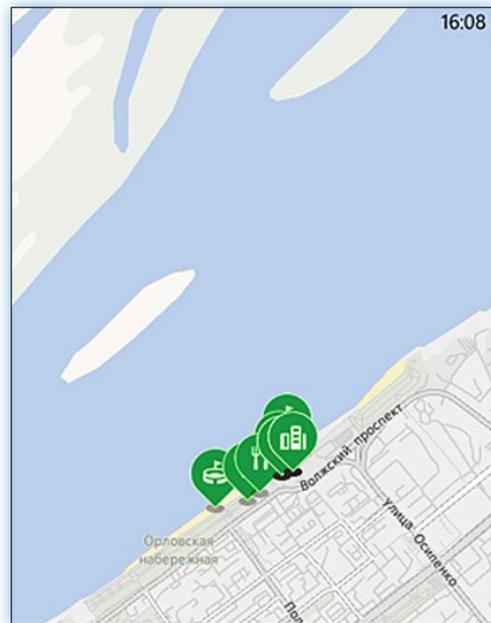
Разместить как есть

+ Проще реализовать

+ Ничего не потеряется

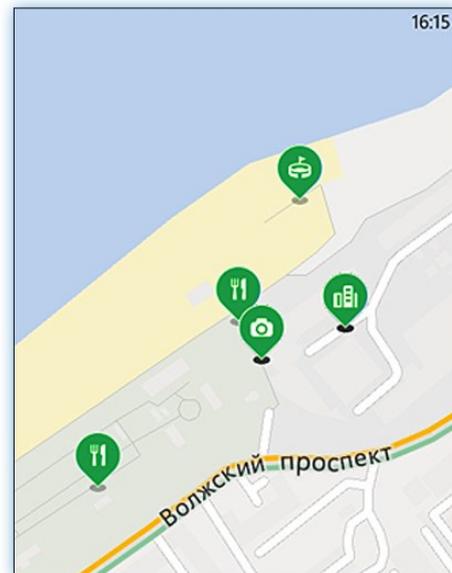
- Недружелюбно к пользователю

- Сложно пользоваться



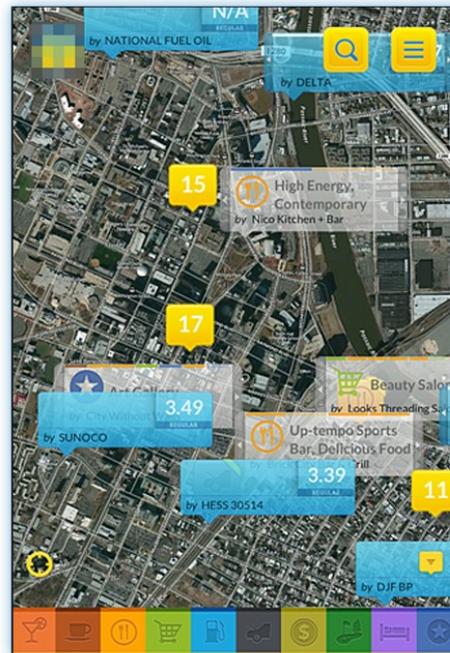
Фильтровать

- Определить приоритеты объектов
- Учитывать расстояние от текущего местоположения
- + Чистота внешнего вида
- + Удобство использования
- Возможность не угадать с желанием пользователя



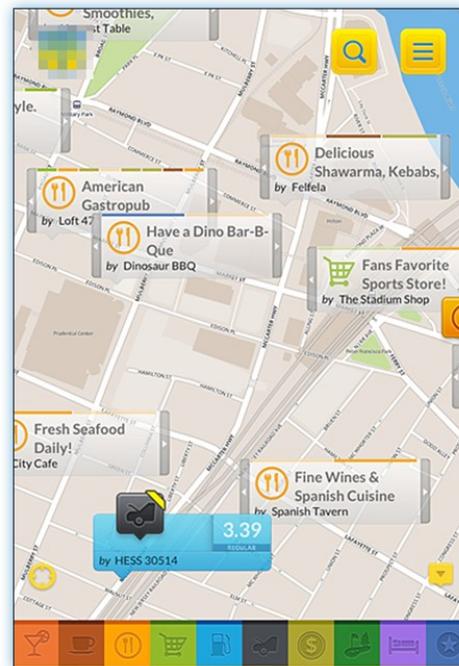
Группировать

- По типам
 - По расстоянию
 - По приоритетам
-
- + Чистота внешнего вида, удобство восприятия
 - + Нет потери информации
 - Сложность поиска требуемой информации

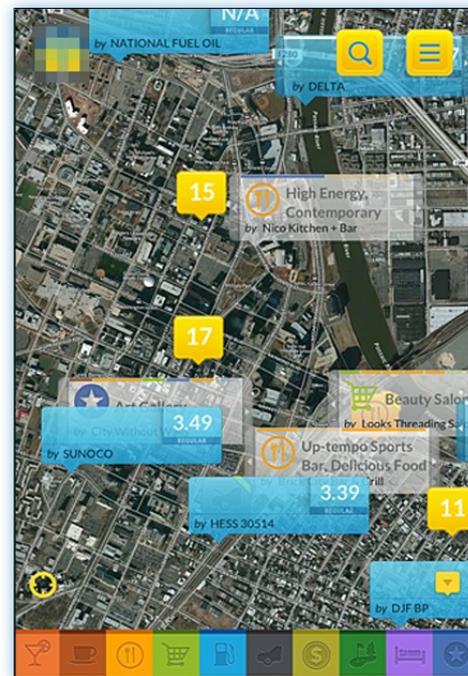


Анимировать

- По типам
 - По расстоянию
 - По приоритетам
- + Большая наглядность
- “живой” интерфейс



- Комбинация этих методов



Влияние множества факторов на логику работы

- ГЕО приложения работают с реальными объектами в реальном мире
 - Радиусы срабатывания
 - Пересекающиеся точки
- POI- это улица или парк или площадь
- Когда считать выход из точки
- Что делать если воспроизводится старый контент, когда подошли к новой точке



Сложность тестирования и разработки

- Погодные условия
- Различные регионы эксплуатации и разработки приложения
- Выход:
 - Моделировать
 - Описывать все пользовательские сценарии
 - Реально побывать в месте эксплуатации данной системы.



Москва



Нью-Йорк



Спасибо за внимание !

Петерсон Дмитрий

dmitry.peterson@simbirsoft.com

