# 10

# Карты

Nikolay Moskvin
moskvin@sibext.com

21 ноября 2013 г.

# Содержание

# Мультитач

# Мультитач

sibext company

Что требуется для реализации:

- Определить **View** для которой будет необходим мультитач
- В зависимости от входных данных нажатий **MotionEvent** написать логику
- **MotionEvent** содержит данные, которые указывают на активные нажатия на экран
- **MotionEvent** содержит координаты и совершаемое действие **getAction()**
- Слушать данные о нажатиях можно с помощью методов View: **onTouchEvent** и **onInterceptTouchEvent** либо слушателя: **OnTouchListener**

# Реализуем View с телескопом

sibext company

```
public class TouchView extends View {
  private static final int INVALID_POINTER_ID = -1;

  private Drawable icon;
  private float posX;
  private float posY;

  private float lastTouchX;
  private float lastTouchY;
  private int activePointerId = INVALID_POINTER_ID;

  private ScaleGestureDetector scaleDetector;
  private float scaleFactor = 1.f;

  public TouchView(Context context) {
      this(context, null, 0);
  }

  public TouchView(Context context, AttributeSet attrs) {
      this(context, attrs, 0);
  }

  public TouchView(Context context, AttributeSet attrs, int defStyle) {
      super(context, attrs, defStyle);
      icon = context.getResources().getDrawable(R.drawable.icon);
      icon.setBounds(0, 0, icon.getIntrinsicWidth(), icon.getIntrinsicHeight());
      scaleDetector = new ScaleGestureDetector(context, new ScaleListener());
  }
...
```

# Слушаем данные с экрана

```
...
  @Override
  public boolean onTouchEvent(MotionEvent ev) {
        // Let the ScaleGestureDetector inspect all events.
        scaleDetector.onTouchEvent(ev);

        final int action = ev.getAction();
        switch (action & MotionEvent.ACTION_MASK) {
        case MotionEvent.ACTION_DOWN: {
            ...
            break;
        }
        case MotionEvent.ACTION_MOVE: {
            ...
            break;
        }
        case MotionEvent.ACTION_UP: {
            ...
            break;
        }
        case MotionEvent.ACTION_CANCEL: {
            ...
            break;
        }
        case MotionEvent.ACTION_POINTER_UP: {
            ...
            break;
        }
        }
        return true;
    }
```

# Перерисовываем экран

Также нам нужно определить слушателя для изменений масштаба.

```
...
    @Override
    public void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.save();
        canvas.translate(posX, posY);
        canvas.scale(scaleFactor, scaleFactor);
        icon.draw(canvas);
        canvas.restore();
    }

    private class ScaleListener
      extends ScaleGestureDetector.SimpleOnScaleGestureListener {
        @Override
        public boolean onScale(ScaleGestureDetector detector) {
            scaleFactor *= detector.getScaleFactor();
            // Don't let the object get too small or too large.
            scaleFactor = Math.max(0.1f, Math.min(scaleFactor, 5.0f));

            invalidate();
            return true;
        }
    }
}
```

# Нажимаем и отпускаем пальцем

sibext company

Сохраняем координаты и идентификатор для выбранного объекта

```
...
case MotionEvent.ACTION_DOWN: {
   final float x = ev.getX();
   final float y = ev.getY();

   lastTouchX = x;
   lastTouchY = y;
   activePointerId = ev.getPointerId(0);
...
```

Если палец поднимаем, то сбрасываем активный объект

```
...
case MotionEvent.ACTION_UP:
case MotionEvent.ACTION_CANCEL: {
    activePointerId = INVALID_POINTER_ID;
    break;
}
...
```

# Двигаем палец по экрану

sibext company

Сохраняем актуальные координаты с учетом их смещения.

```
...
case MotionEvent.ACTION_MOVE: {
    final int pointerIndex = ev.findPointerIndex(activePointerId);
    final float x = ev.getX(pointerIndex);
    final float y = ev.getY(pointerIndex);

    // Only move if the ScaleGestureDetector isn't processing a gesture.
    if (!scaleDetector.isInProgress()) {
        final float dx = x - lastTouchX;
        final float dy = y - lastTouchY;

        posX += dx;
        posY += dy;

        invalidate();
    }

    lastTouchX = x;
    lastTouchY = y;

...
```

# Несколько нажатий одновременно

sibext company

Обрабатываем момент, когда мы имеем одновременно несколько зажатых объектов

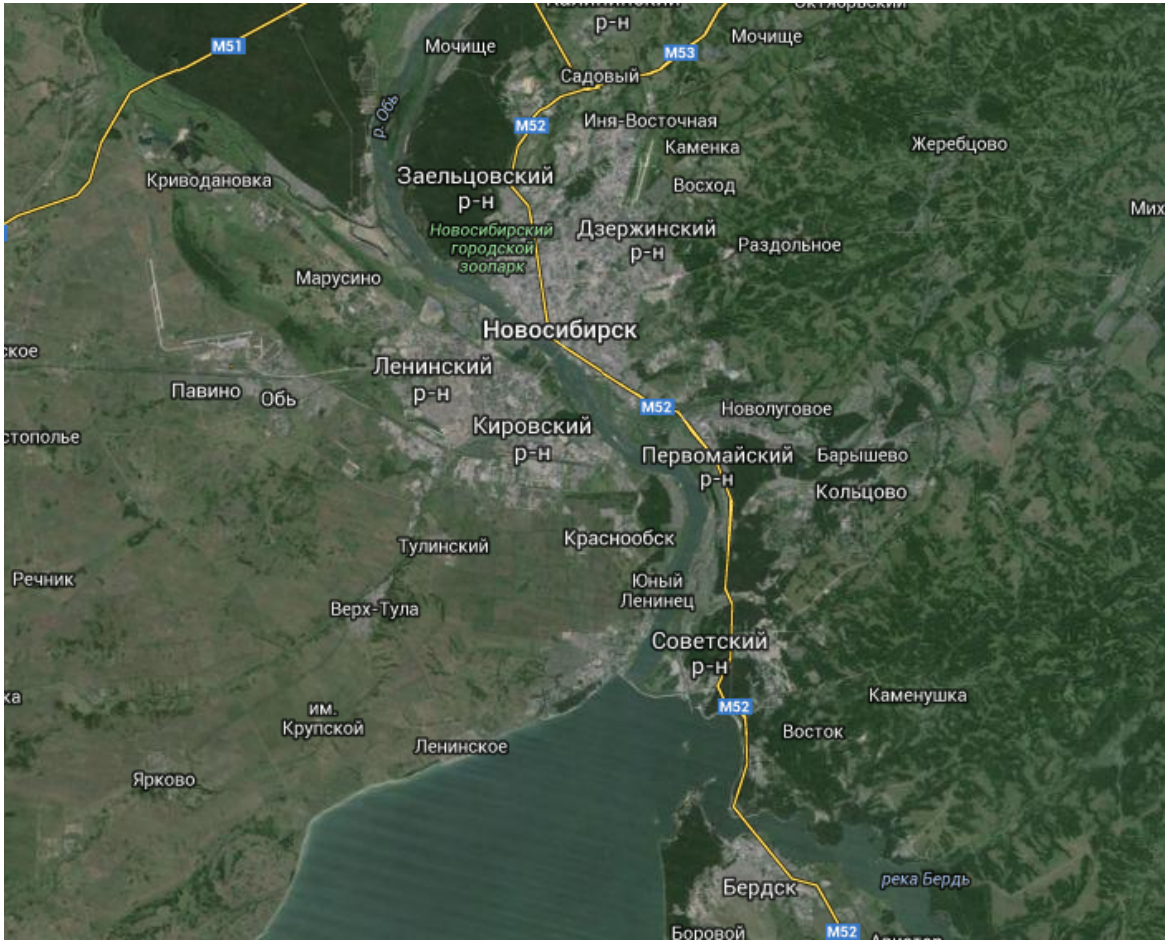
```
...
case MotionEvent.ACTION_POINTER_UP: {
  final int pointerIndex =
        (ev.getAction() & MotionEvent.ACTION_POINTER_INDEX_MASK)
        >> MotionEvent.ACTION_POINTER_INDEX_SHIFT;
  final int pointerId = ev.getPointerId(pointerIndex);
  if (pointerId == activePointerId) {
      // This was our active pointer going up. Choose a new
      // active pointer and adjust accordingly.
      final int newPointerIndex = pointerIndex == 0 ? 1 : 0;
      lastTouchX = ev.getX(newPointerIndex);
      lastTouchY = ev.getY(newPointerIndex);
      activePointerId = ev.getPointerId(newPointerIndex);
  }
...
```

# GestureDetectors

sibext company

- Приложения имеет разные потребности в нажатиях
- Android не произведет сам данные для нажатий, необходимо специально запрашивать
- **GestureDetector** – это небольшой объект фильтра, который способен обрабатывать различные события жестов
- Имеется не большой набор реализаций жестов у Android.
- Но это лишь шаблон, если необходимо то вы можете сами создать свои жесты
- `http://developer.android.com/reference/android/view/GestureDetector.html`

# Google Maps


M51
Мочище
р-н
Мочище
M53
Садовый
р. Обь
M52
Иня-Восточная
Каменка
Жеребцово
Криводановка
Заельцовский
р-н
Восход
Новосибирский
городской
зоопарк
Дзержинский
р-н
Раздольное
Марусино
Новосибирск
Ленинский
р-н
Павино
Обь
M52
Новолуговое
Кировский
р-н
Первомайский
р-н
Барышево
Кольцово
Тулинский
Краснообск
Речник
Юный
Ленинец
Верх-Тула
Советский
р-н
Каменушка
им.
Крупской
M52
Восток
Ленинское
Ярково
Бердск
река Бердь
Боровой
M52

# Google Maps

sibext company

Для API v2 версии карт от Google необходимо сделать

1. Подключить Google Play services SDK `http://developer.android.com/google/play-services/setup.html`
2. Зарегистрировать Google Play сервис у себя в манифесте
3. Создать проект в Google APIs Console
4. Получить и добавить Map API key в манифест
5. Добавить MapFragment в проект

# Верстка

sibext company

Создаем статический фрагмент и указываем путь до фрагмента карт

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
          android:id="@+id/map"
          android:layout_width="match_parent"
          android:layout_height="match_parent"
          android:name="com.google.android.gms.maps.MapFragment"/>
```

# Доступ из Activity

sibext company

```
public class MapPane extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_activity);

        // Get a handle to the Map Fragment
        GoogleMap map = ((MapFragment) getFragmentManager()
                .findFragmentById(R.id.map)).getMap();

        LatLng sib = new LatLng(55.033333, 82.916667);

        map.setMyLocationEnabled(true);
        map.moveCamera(
           CameraUpdateFactory.newLatLngZoom(sydney, 13));

        map.addMarker(new MarkerOptions()
                .title("Novosibirsk")
                .snippet("The center city in Siberia")
                .position(sib));

        // Other supported types include: MAP_TYPE_NORMAL,
        // MAP_TYPE_TERRAIN, MAP_TYPE_HYBRID and MAP_TYPE_NONE
        map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);

    }
}
```

# Маркеры

sibext company

- Указатель местоположение на карте
- Можно заставить плавать по карте – **draggable(true)**
- Можно выставлять цвет, прозрачность и другу картинку
- Можно поворачивать и заставлять маркер указывать направление движения
- `https://developers.google.com/maps/documentation/android/marker`

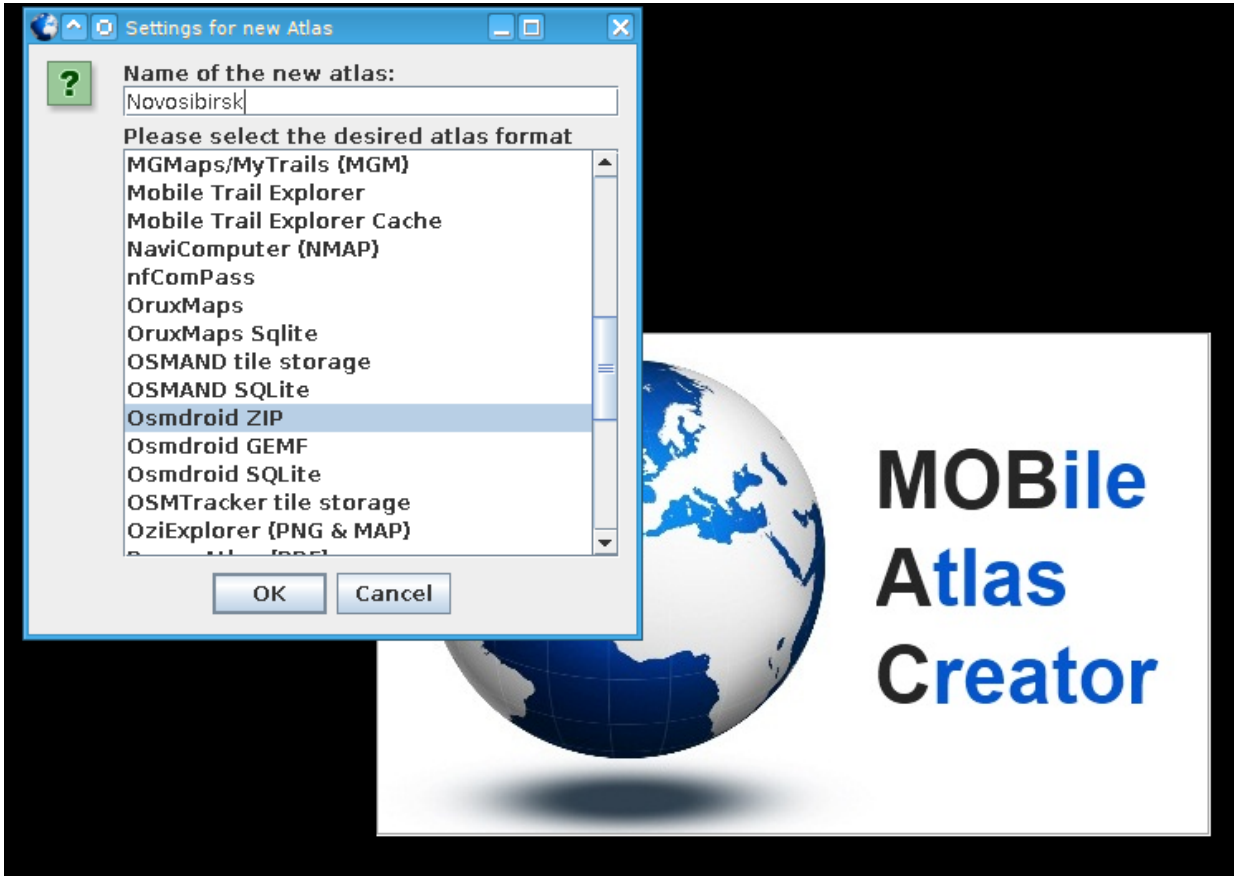
# OSMDroid

sibext company

- Полнофункциональная замена стандартных google карт v1 для Android
- Инструмент с открытым исходным кодом
- Использует карты с OpenStreetMap: http://www.openstreetmap.org/
- Поддерживает online и offline карты
- `https://code.google.com/p/osmdroid/`

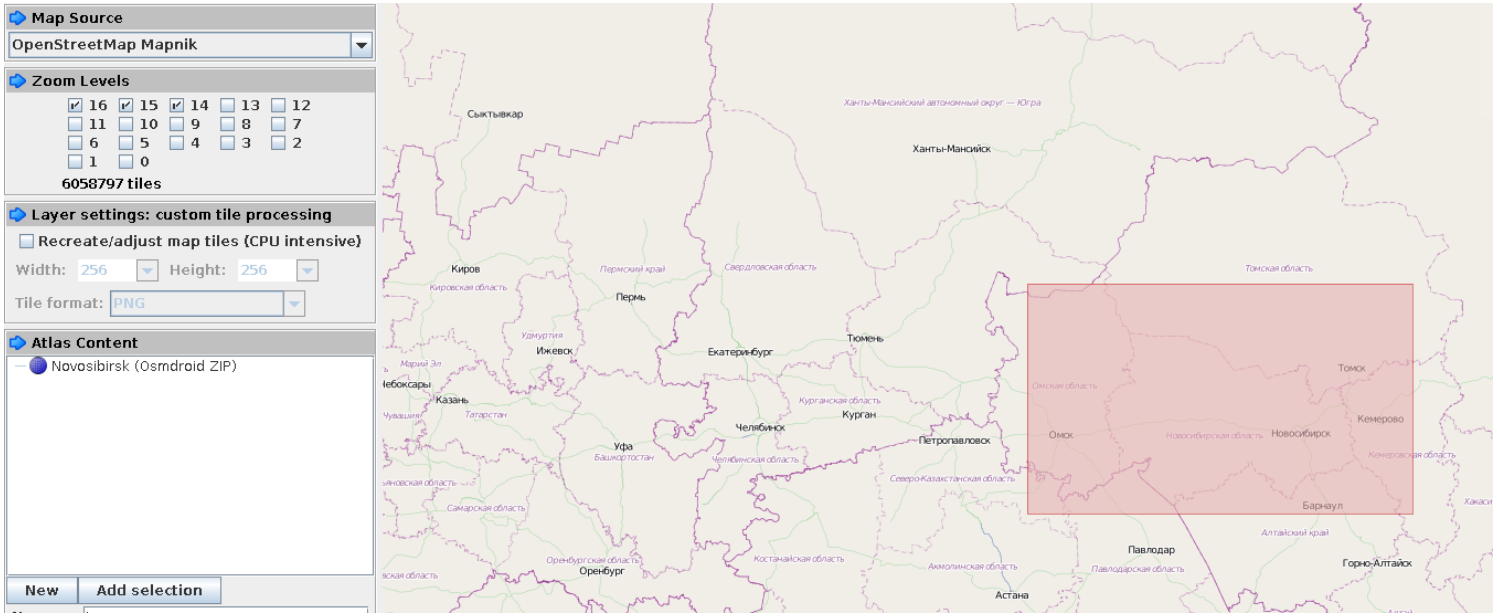
# Карты Offline

- Приготавливаем карту с помощью **Mobile Atlas Creator** – `http://mobac.sourceforge.net/`
- Устанавливаем и запускаем
- Выбираем тип “Osmdroid ZIP” при старте



# Карты Offline

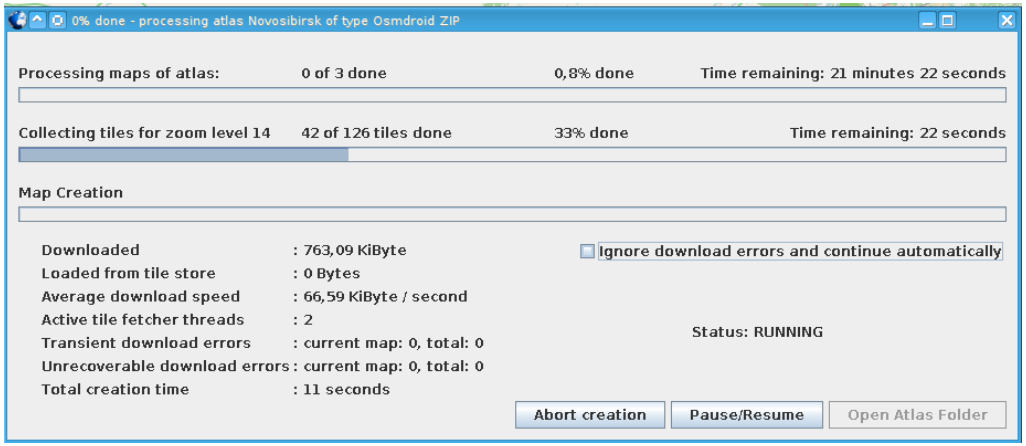
- Выбираем область на карте и выбираем количество zoom уровней




- Необходимо учесть, что каждый слой размером 20 Kb. В примере 6058797*20Kb = 121175 Мб

# Карты Offline

- Тип слоя указываем **Mapnik**
- В **Atlas Content** необходимо сделать “Add selection”
- Делаем “Create atlas”




- Переносим полученный пакет на SD-карту в директорию /mnt/sdcard/osmdroid/

# Подключение osmdroid к проекту

sibext company

- Указываем в зависимости pom.xml

```
  <dependency>
      <groupId>org.osmdroid</groupId>
      <artifactId>osmdroid-android</artifactId>
      <packaging>jar</packaging>
      <version>4.0</version>
  </dependency>
```

- Добавляем карту в компонент

```
public class MapPane extends Activity {
  public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    MapView mapView = new MapView(this, 256);
    setContentView(mapView);
    mapView.setClickable(true);
    mapView.setBuiltInZoomControls(true);
    mapView.getController().setZoom(15);
    mapView.getController().setCenter(
      new GeoPoint(55.033333, 82.916667));
    mapView.setUseDataConnection(false);
    mapView..getOverlays().add(new PinOverlay(this));
  }
}
```

# Overlay

sibext company

- Создаем свой Pin для карты

```
public class PinOverlay extends SafeDrawOverlay {
    private final Point screenCoords;
    private final Point positionIndicatorCoords;
    private final SafePaint paint;

    private Bitmap positionIndicator;
    private GeoPoint location;

    public PinOverlay(Context context) {
        super(context);
        positionIndicator = BitmapFactory.decodeResource(
          context.getResources(), R.drawable.pin);
        screenCoords = new Point();
        paint = new SafePaint();
        positionIndicatorCoords = new Point(
          positionIndicator.getWidth() / 2,
          positionIndicator.getHeight() / 3 * 2);
    }
...
```

# Overlay

sibext company

- Метод onDraw

```
...
    @Override
    protected void drawSafe(ISafeCanvas c, MapView osmv, boolean shadow) {
        if (!shadow && this.location != null) {
            final Projection pj = osmv.getProjection();
            pj.toMapPixels(this.location, screenCoords);
            c.drawBitmap(positionIndicator,
                    screenCoords.x - positionIndicatorCoords.x,
                    screenCoords.y - positionIndicatorCoords.y,
                    paint);
        }
    }
}
```

# Достоинства и недостатки OSMDroid

sibext company

**Достоинства**

- Работает без дополнительных регистраций (Ключ для карты итд)
- Умеет работать с offline картами
- Широкие возможности для рисований на карте
- Большой набор слоёв для карт

**Недостатки**

- Нет обратной совместимости у версий
- Присутствуют проблемы с UI

# Определения азимута

Для того, чтобы вручную определять направление движения относительно верхней части устройства необходимо иметь специальный сенсор на устройстве

```
@Override
public void onCreate(Bundle savedInstanceState) {
  SensorManager sensorManager =
    (SensorManager) getActivity().getSystemService(Context.SENSOR_SERVICE);
  List<Sensor> sensors =
    sensorManager.getSensorList(Sensor.TYPE_ORIENTATION);

  if (sensors != null && !sensors.isEmpty()) {
      sensorManager.registerListener(this,
        sensors.get(0),
        sensorManager.SENSOR_DELAY_UI);
  }
}
...
SensorEventListener listener = new SensorEventListener() {
@Override
public void onSensorChanged(SensorEvent event) {
    float azimuth = event.values[0];
    Log.d(TAG, "onGetAzimuth: " + azimuth);
}
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {}
};
```