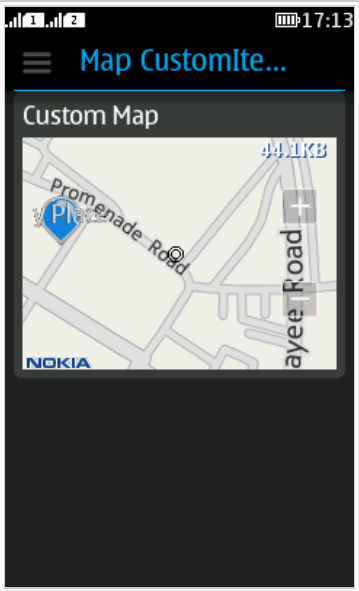
Displaying Nokia Maps in a Form in Java ME

This article shows how to use a map inside an LCDUI Form in Java ME.

Note: This is an entry in the Asha Touch Competition 2012Q3

Introduction

Applications that need to show maps within a Java ME Form cannot do so directly. This article shows how you can define a CustomItem (called `MapCustomItem`) which can be used to display a Nokia Map within a `Form`.



Map used inside a `Form`

Note: Compatible with all devices which supports Nokia Maps API.

The article Creating a custom Map Item with Maps API for Java ME covers similar ground to this article, but uses a `MapDisplay` and `MapFactory` for the map. This article uses a `MapCanvas`, which saves you from having to implement methods for zooming, pan etc. You can create interactive map inside a `Form` with very little code.

Development

Nokia Maps for Java ME are displayed on a MapCanvas object. Unfortunately `MapCanvas` has some protected methods, which prevent it being used directly within a `CustomItem`. We therefore first create `MapXtention`, extending `MapCanvas` to provide `public` accessors for the required methods.

Finally, we then create a "Form ready" map item (`MapCustomItem`) by extending `CustomItem` (and using `MapXtention`).

MapXtention

`MapXtention` extends `MapCanvas`, creating *public* methods to access `paint()`, `sizeChanged()` and few other required methods of `MapCanvas`.

The new class contains these functions:

- `public void paint(Graphics g);`
  The definition of this function will be:
  ```
  public void paint(Graphics g){
   super.paint(g);
  }
  ```
- `public void size(int w, int h);`
  This method is to re-size the `MapCanvas` to `CustomItem` size or desired size.
  The definition of this function will be:
  ```
  public void size(int w, int h){
   super.sizeChanged(w, h);
  }
  ```


(C) Copyright Nokia 2013. All rights reserved.

- `public void onPressed(int x, int y);`
  This method inter calls the `MapCanvas pointerPressed()` event method.
  The definition of this function will be:

```
public void onPressed(int x, int y){
  super.pointerPressed(x, y);
}
```

- `public void onDrag(int x, int y);`
  This method inter calls the `MapCanvas pointerDragged()` event method.
  The definition of this function will be:

```
public void onDrag(int x, int y){
  super.pointerDragged(x, y);
}
```

- `public void onRelease(int x, int y);`
  This method inter calls the `MapCanvas pointerReleased()` event method.
  The definition of this function will be:

```
public void onRelease(int x, int y){
  super.pointerReleased(x, y);
}
```

The complete `MapXtention` is listed below:

```
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Graphics;

import com.nokia.maps.map.MapCanvas;


public class MapXtention extends MapCanvas{

 public MapXtention(Display arg0) {
  super(arg0);
 }

 public void onMapContentComplete() {
 }

 public void onMapUpdateError(String arg0, Throwable arg1, boolean arg2) {
 }

 public void paint(Graphics g){
  super.paint(g);
 }

 public void size(int w, int h){
  super.sizeChanged(w, h);
 }

 public void onPressed(int x, int y){
  super.pointerPressed(x, y);
 }

 public void onDrag(int x, int y){
  super.pointerDragged(x, y);
 }

 public void onRelease(int x, int y){
  super.pointerReleased(x, y);
 }
}
```

MapCustomItem

`MapCustomItem` extends `CustomItem` and this is the class which performs necessary operations to make `MapCanvas` as `CustomItem`. The important methods that make `MapCustomItem` possible are:

- `protected void paint(Graphics g, int w, int h);`
  Abstract method from `CustomItem` class.
  It definition should be:

```
protected void paint(Graphics g, int w, int h) {
 map.paint(g);
        repaint();
}
```

> **Note:** `map` is an object of `MapXtention` class.

- `public void setMap(double lat, double lon, String title);`
  This method can be used to set Map location, Zoom level and Pan. `String title` is an extra field, required only for creating `StandardMarker`, that use a `String` argument for displaying text with Marker.
  The definition of this method will be:

```
public void setMap(double lat, double lon, String title){
         map.getMapDisplay().removeAllMapObjects();
  map.getMapDisplay().reconnect();
  //map.getMapDisplay().setCenter(new GeoCoordinate(lat, lon, 0));
  map.getMapDisplay().setCenter(new GeoCoordinate(13.93436, 75.59581, 0));
  map.getMapDisplay().setZoomLevel(17, 0, 0);
  //map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(lat, lon, 0), 1, title, 2));
  map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(13.93436, 75.59581, 0), 1, title
}
```


(C) Copyright Nokia 2013. All rights reserved.

- `double lat`, `double lon` are the latitude and longitude value respectively.
  Use `MapCustomItem_Object.setMap( latitude_value, longitude_value, "Some text" );` in MIDlet/method to update/set map/map position and marker text. Frequent call to this method with changing position/location can be used for navigation/to map current position on map.
- Constructor should contain the following statements:

```
protected MapCustomItem(String title, Display d) {
 super(title);
 map = new MapXtention(d);
 map.getMapDisplay().setBaseMapType(4);
 map.size(getMinContentWidth(), getMinContentHeight());
}
```

  - `title`: String representing title of the `CustomItem` (i.e title of `MapCustomItem` object in form).
  - `d`: Display object, required by `MapXtention` object. However, `MapXtention` class extends `MapCanvas` class. `MapXtention` class also behaves exactly as `MapCanvas` class.
  - `setBaseMapType(4)`: 4 is the base type used for*Transits* as map display type. More.
  - Set the size of `map` in the constructor only. Do not use this in `paint` method.
- Finally override `pointerPressed()`, `pointerDragged()` and `pointerReleased()` as shown bellow to perform Zooming, pan and other map operations.

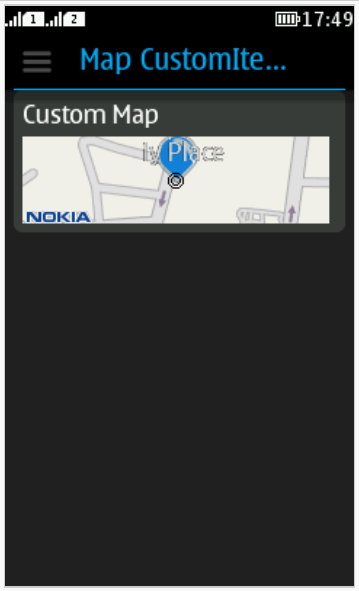
```
protected void pointerPressed(int x, int y){
 map.onPressed(x, y);
}

protected void pointerDragged(int x, int y){
 map.onDrag(x, y);
}

protected void pointerReleased(int x, int y){
 map.onRelease(x, y);
}
```

The complete `MapCustomItem` class is listed below:

```
import javax.microedition.lcdui.CustomItem;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Graphics;

import com.nokia.maps.common.GeoCoordinate;


public class MapCustomItem extends CustomItem{

 MapXtention map;
 protected MapCustomItem(String title, Display d) {
  super(title);
  map = new MapXtention(d);
  map.getMapDisplay().setBaseMapType(4);
  map.size(getMinContentWidth(), getMinContentHeight());
 }

 protected int getMinContentHeight() {

  return 160;
 }

 protected int getMinContentWidth() {

  return 210;
 }

 protected int getPrefContentHeight(int arg0) {

  return getMinContentHeight();
 }

 protected int getPrefContentWidth(int arg0) {

  return getMinContentWidth();
 }

 protected void paint(Graphics g, int w, int h) {
  map.paint(g);
                repaint();
 }

 public void setMap(double lat, double lon, String title){
  map.getMapDisplay().removeAllMapObjects();
  map.getMapDisplay().reconnect();
  //map.getMapDisplay().setCenter(new GeoCoordinate(lat, lon, 0));
  map.getMapDisplay().setCenter(new GeoCoordinate(13.93436, 75.59581, 0));
  map.getMapDisplay().setZoomLevel(17, 0, 0);
  //map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(lat, lon, 0), 1, title, 2));
  map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(13.93436, 75.59581, 0), 1, title, 2));
 }

        protected void pointerPressed(int x, int y){
  map.onPressed(x, y);
 }

 protected void pointerDragged(int x, int y){
  map.onDrag(x, y);
 }

 protected void pointerReleased(int x, int y){
  map.onRelease(x, y);
 }

}
```


(C) Copyright Nokia 2013. All rights reserved.

Tiny Map



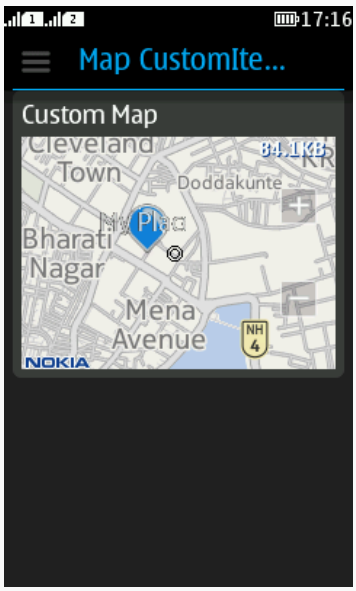
Tiny Map in a form

You can create "tiny map" in form (these type of maps have zoom level set constant and user can pan the map). To create "tiny map", modify the following statement to these functions of `MapCustomItem` class.

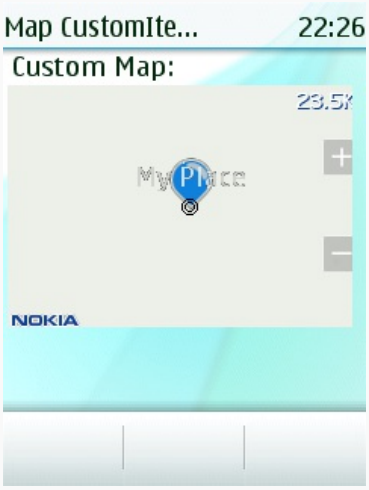
```
protected int getMinContentHeight() {
 return 60;
}
```

```
public void setMap(double lat, double lon, String title){
 map.getMapDisplay().removeAllMapObjects();
 map.getMapDisplay().reconnect();

 map.getMapDisplay().removeMapComponent(map.getMapDisplay().getMapComponent("ZoomImgComponent")); //to remove Zoom Buttons
 map.getMapDisplay().removeMapComponent(map.getMapDisplay().getMapComponent("DownloadIndicator")); //to remove download texts
 // map.getMapDisplay().setCenter(new GeoCoordinate(lat, lon, 0));
        map.getMapDisplay().setCenter(new GeoCoordinate(13.93436, 75.59581, 0));
 map.getMapDisplay().setZoomLevel(17, 0, 0);
       //map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(lat, lon, 0), 1, title, 2));
 map.getMapDisplay().addMapObject(map.getMapFactory().createStandardMarker(new GeoCoordinate(13.93436, 75.59581, 0), 1, title, 2));
}
```

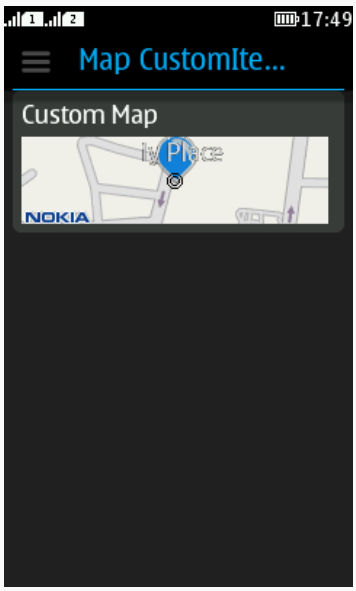
Snapshots



Series 40 full-touch device:Asha 305



S40 Touch and Type device:Asha 303



Tiny Map in a form

Source

media:Custommap.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.