QTLocation in practice

Guillaume Charbonnier
gcharbonnier@a-team.fr

Développement mobile avec Qt/QML
La Cantine Numérique the 2th of january 2016

Workshop objectives

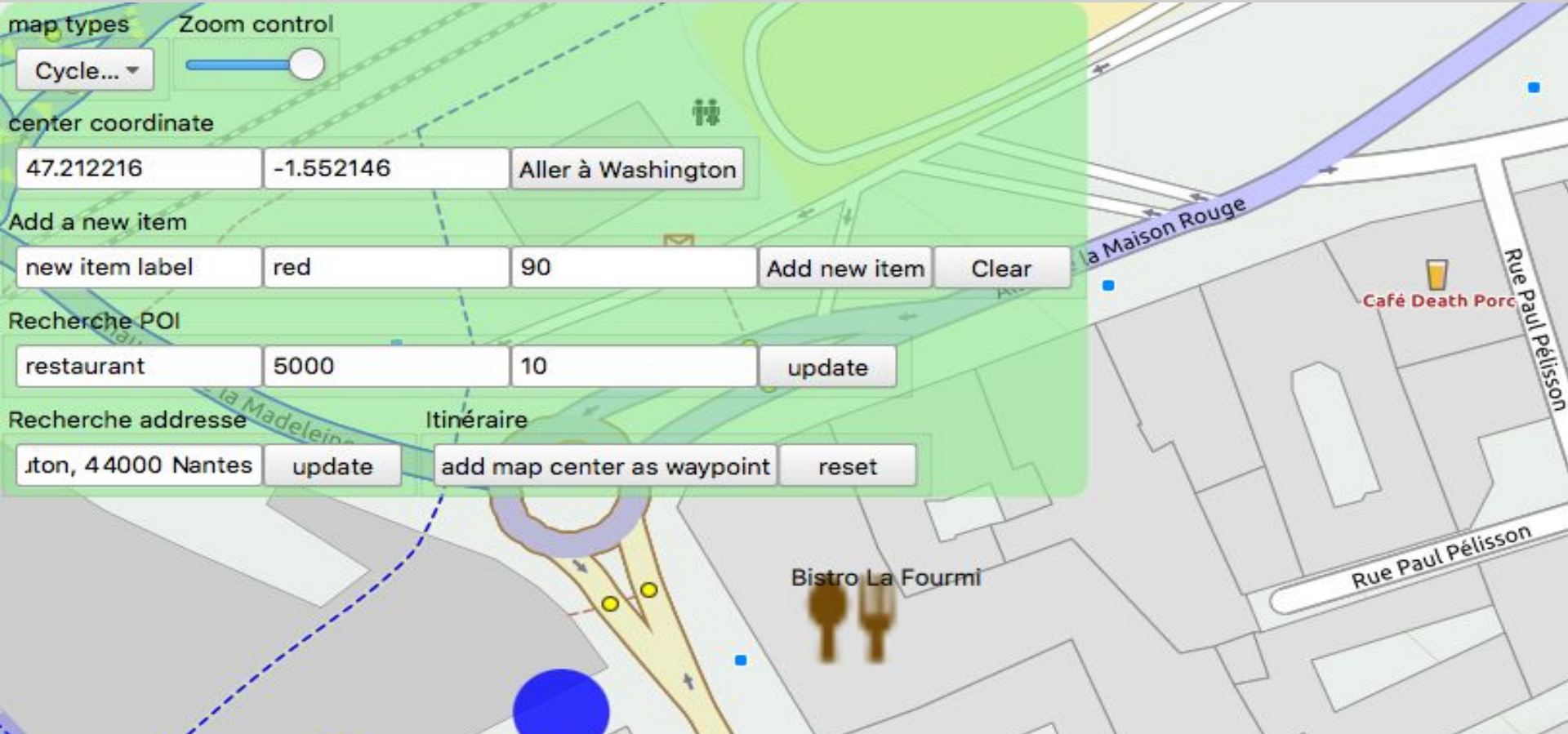
Get familiar with the QtPositioning and QtLocation modules in QML.

Steps :

1. Create project - (time box : 10mn)
2. Add a map - (time box : 15mn)
3. Map customization - (time box : 15mn)
4. Adding visual items - (time box : 30mn)
5. geocoding - (time box : 5mn)
6. route (extra)

Next slide is a screenshot of the resulting demonstrator.

Example


map types
Zoom control
Cycle...
center coordinate
47.212216
-1.552146
Aller à Washington
Add a new item
new item label
red
90
Add new item
Clear
Recherche POI
restaurant
5000
10
update
Recherche addresse
Itinéraire
ıton, 44000 Nantes
update
add map center as waypoint
reset
Bistro La Fourmi
Café Death Porc
Rue Paul Pélisson
Rue Paul Pélisson
la Maison Rouge

Step 1

Project creation

- Create a new project
- Create the window
- Activate and load location and positioning modules

Create a new project

1. Create a new project of the type Application (Qt Quick Application)
2. Modify the .pro file to activate modules

```
QT += qml quick positioning location
```

3. Modify main.qml to load modules

```
import QtPositioning 5.3
import QtLocation 5.5
import QtQuick.Controls 1.4
import QtQuick.XmlListModel 2.0
```

Step 2

Add a map

- Select the geolocalization service
- Add a map
- going further :
  - using another plugin

```
Plugin {
    id:myPlugin
    name:  "osm"
}
```

```
Map{
    id: myMap
    anchors.fill: parent
    plugin: myPlugin
}
```

```
Plugin{
    id:mapPlugin
    name: "mapbox"
    PluginParameter { name: "useragent"; value: "%YOUR_APP_NAME%" }
    PluginParameter { name: "mapbox.access_token"; value: "%YOUR_TOKEN%" }
    PluginParameter { name: "mapbox.map_id"; value: "mapbox.pencil"  }//"wheatpaste/comic/pencil/dark"
  }
```

Sélectionner un autre service

Step 3

Customize the map

- control map position
  - arbitrary location
  - using device position (network/GPS)
- control zoom level
- modify gestures
- Change map type

```
Location{
    id: myLocation
    coordinate {
        latitude : 47.212047;
        longitude : -1.551647
    }
}
Map{
    id : myMap
    //...
    center : myLocation.coordinate
}
```

```
PositionSource{
    id : myPos
    updateInterval : 1000
    active : true
    preferredPositioningMethods :  PositionSource.AllPositioningMethods
}

Map{
    id : myMap
    //...
    center : myPos.position.coordinate
}
```

Modify the map position

```
Item{
    id:controlPanel
    width:controls.childrenRect.width
    height:controls.childrenRect.height
    Rectangle{
        anchors.fill: parent
        color:"lightgreen"
        opacity:0.6
        radius:10
    }
    Flow{
        id:controls
        width:parent.width
        //Place your controls there
    }
}
```

This is not related to QtLocation.
But this will be needed later on to add controls.

```
Item{ id:controlPanel  /* ... */
    Flow{ id:controls /* ... */
        GroupBox{
            title:"map types"
            ComboBox{
                model:myMap.supportedMapTypes
                textRole:"description"
                onCurrentIndexChanged: myMap.activeMapType = myMap.supportedMapTypes[currentIndex]
            }        }

        GroupBox{
            title:"Zoom control"
            Slider{
                value:myMap.zoomLevel
                minimumValue: myMap.minimumZoomLevel
                maximumValue: myMap.maximumZoomLevel
            }        }     }  }
```

Change zoom and map type

```
GroupBox{
  title:"center coordinate"
  Row{

  TextField{
     text:myMap.center.latitude.toFixed(6)
     onEditingFinished: myMap.center.latitude = text
  }
  TextField{
     text:myMap.center.longitude.toFixed(6)
     onEditingFinished: myMap.center.longitude = text
  }
}
```

Step 4

Add visual items

- add visual item
  - screen size
  - real life size
- add items from a model
- add items using a xml model
- use search service

```
Map{
  id:myMap
  /* ... */
  MapQuickItem {
    coordinate: myMap.center.atDistanceAndAzimuth(50,270)
    sourceItem:  Rectangle{
       width:50; height:50; radius:25; color:"blue"; opacity:0.8
    }
  }
  MapCircle {
    center: myMap.center.atDistanceAndAzimuth(100,90)
    opacity:0.8
    color:"red"
    radius:10
  }
}
```

Data definition...

```
ListModel{
    id:dummyModel
    ListElement {
        Latitude: 47.212047
        Longitude: -1.551647
        Label: "something"
        Orientation: 0
        Color:"green"
    }
    ListElement {
        /* idem */
    }
}
```

...and display !

```
Map{
  id:myMap
  /* ... */
  MapItemView{
     model: dummyModel
     delegate: MapQuickItem {
       coordinate: QtPositioning.coordinate(Latitude,
Longitude)
        sourceItem:  Text{
           width:100; height:50
           text:model.Label
           rotation: model.Orientation
           opacity: 0.6
           color:model.Color
      }    }  }}
```

```
GroupBox{
  title:"Add a new item"
  Row{
    TextField{ id:newItemName;text:"new item label";}
    TextField{ id:newItemColor;text:"red";}
    TextField{ id:newItemOrientation;text:"90";}
    Button{
      text:"Add new item"
      onClicked: {
        dummyModel.append({
          "Latitude": myMap.center.latitude,"Longitude":myMap.center.longitude,"Label":newItemName.text , "Color":
newItemColor.text, "Orientation":Number(newItemOrientation.text), })
      }
    }
    Button{
      text:"Clear"
      onClicked: {
        dummyModel.clear();
      }    }  }}
```

Data definition...

```
XmlListModel
  {
    id: bikeModelXml
    source: "https://www.capitalbikeshare.
com/data/stations/bikeStations.xml"
    query: "/stations/station"
    XmlRole { name: "latitude"; query: "lat/string()"; isKey:
true }
    XmlRole { name: "longitude"; query: "long/string()";
isKey: true }
  }
```

...and display !

```
Map{
  id:myMap
  /* ... */
  MapItemView{
    model: bikeModelXml
    delegate: MapQuickItem {
      coordinate: QtPositioning.coordinate(model.latitude,
model.longitude)
      sourceItem:  Image{ width:50;height:50;source:"qrc:/bike.
png"
    }
  }
}
```

Data definition...

```
PlaceSearchModel {
    id: searchModel
    plugin: myPlugin
    searchTerm: searchFieldName.text
    searchArea: QtPositioning.circle(myMap.center, Number
(searchFieldRadius.text));
    Component.onCompleted: update()
    limit:100
  }
```

UI...

```
GroupBox{
  title:"Recherche POI"
  Row{
    TextField{ id:searchFieldName;text:"restaurant";}
    TextField{ id:searchFieldRadius;text:"5000";}
    Button{ text:"update"; onClicked: searchModel.update()}
  }
}
```

...and display !

```
Map{
  id:myMap
  /* ... */
  MapItemView{
    model: searchModel
    delegate: MapQuickItem {
      coordinate: model.place.location.coordinate
      sourceItem:  Image{ width:64;height:64;
        source:model.place.icon.url(Qt.size(64,64));
        Text{
          anchors.fill:parent
          text:model.title
        }
      }
    }
  }
}
```

Step 5

Geocoding

- Get an adress from a coordinate
- and the other way around

Data definition...

```
GeocodeModel{
      id:geocodeModel
      plugin:myPlugin
      autoUpdate:false
      onLocationsChanged: {
          if (count>0)
              geocodeResult.coordinate = get(0).coordinate;
      }
  }
```

...and display !

```
MapQuickItem {
 id:geocodeResult
 sourceItem:  Image{
     width:50;height:50;
     source:"qrc:/poi.png" }
}
```

UI...

```
GroupBox{
   title:"Recherche addresse"
   Row{
       TextField{ id:searchAddress;text:"11 rue juton, 44000
Nantes";}
       Button{ text:"update"; onClicked: {
               geocodeModel.query =  searchAddress.text
               geocodeModel.update()
           }}
   }
}
```

Step 6

Route

- query parameters
- model
- modify route
- display results as a list

Setting the route query

```
RouteQuery{
   id:routeQuery
   travelModes:RouteQuery.CarTravel
   routeOptimizations : RouteQuery.FastestRoute
}
```

Route mode

```
RouteModel {
   id: routeModel
   plugin:myPlugin
   query: routeQuery
   autoUpdate: false
}
```

```
ListView {
  id: listview
  anchors.fill: parent
  spacing: 10
  model: routeModel.status == RouteModel.Ready ? routeModel.get(0).segments : null
  visible: model ? true : false
  delegate: Rectangle{
     width: parent.width
     height:50
     color:"lightgrey"
     Row {
     anchors.fill: parent
     spacing: 10
     property bool hasManeuver : modelData.maneuver && modelData.maneuver.valid
     visible: hasManeuver
     Text { text: (1 + index) + "." }
     Text { text: hasManeuver ? modelData.maneuver.instructionText : "" }
     }
  }
}
```

Display as a list

```
GroupBox{
  title:"Itinéraire"
  Row{
    Button{ text:"add map center as waypoint"; onClicked: {
        routeQuery.addWaypoint(myMap.center);
        routeModel.update();
    }}
    Button{ text:"reset"; onClicked: {
        routeQuery.clearWaypoints();
        routeModel.update();
    }}
  }
}
```

Guillaume Charbonnier
gcharbonnier@a-team.fr