Real-estate sales camera app

This article describes a method to create a really useful app for a real-estate agent. The user points the camera to a house on sale and the adaptive list gives the details of various rooms. The adaptive list could include the sizes of rooms, the type of material used for various rooms and whatever is felt necessary by the agent.

Note: This is an entry in the PureView Imaging Competition 2012Q2

Introduction

This article describes a method to create a really useful app for Realty. The user points the camera to a house on sale and the adaptive list gives the details of various rooms. The adaptive list could include the sizes of rooms, the type of material used for various rooms and whatever is felt necessary by the Realty agent. The list is based on longitude of the place where the camera is in use. The app can be extended to include nearby houses on sale too.

GPS location detecting

The location API is used for getting the position of Cam. The Location API provides a library for location positioning, landmark management and mapping and navigation. Full documentation can be seen here GPS This is part of Qt Mobility 1.2.

A brief description is below

- NameSpace : The QtMobility APIs are placed into the QtMobility namespace. This is done to facilitate the future migration of QtMobility APIs into Qt. See the Quickstart guide for an example on how the namespace impacts on application development.
- Location Position: Location data involves a precisely specified position on the Earth's surface — as provided by a latitude-longitude coordinate — along with associated data, such as:
  - The date and time at which the position was reported
  - The velocity of the device that reported the position
  - The altitude of the reported position (height above sea level)
  - The bearing of the device in degrees, relative to true north

This data can be extracted through a variety of methods. One of the most well known methods of positioning is GPS (Global Positioning System), a publicly available system that uses radiowave signals received from Earth-orbiting satellites to calculate the precise position and time of the receiver. Another popular method is Cell ID positioning, which uses the cell ID of the cell site that is currently serving the receiving device to calculate its approximate location. These and other positioning methods can all be used with the Location API; the only requirement for a location data source within the API is that it provides a latitude-longitude coordinate with a date/time value, with the option of providing the other attributes listed above.

Here we just get the longitude data

List Algorithm

We use a javascript file to do the list filtering : There are 2 ways to filter the list either by getting the longitude or by reverse geocoding to get the location name . Here we get the location name and convert it into lower case. The javascript file can be added to QML project. Please see the details of how to add java script file here

```
function checkFilter(filter, string) {
    if (filter.length > string.length) return false;
    var filterlower = filter.toLowerCase();
    var stringlower = string.toLowerCase();
    for (var i = 0; i < filter.length; i++) {
        if (filterlower[i] != stringlower[i])
            return false;
    }
    return true;
}
```

Updates the filtermodel based on the current filter

(C) Copyright Nokia 2014. All rights reserved.

```
function updateFilterModel(title) {
    if (filter.length > 0) {
        for (var i = filtermodel.count-1; i >= 0; i--) {
            if (!checkFilter(filter, filtermodel.get(i).title))
                filtermodel.remove(i)
        }
    }
    updateUniqueChars()
}
```

The database is created using ListModel. An example list model can be:

```
ListModel {
    ListElement {
        title: "Naperville John Does House"
        picture: "content/pics/Johndoehouse.jpg"
        material: "<html>
                       <ul>
                     <li> cherry wood floors in living room
                     <li> Vinyl flooring in bedrooms
                     <li> full carpeting

                       </ul>
                      </html>"
        description: "<html>
                  <ol>
                <li> Fabulous 4 BR home with over 3200 square feet. Home has mature
landscaping and is located across from a park. Easy access to both I-55 and the
Naperville train station. Special features include 2-story open foyer, 9' 1st floor
ceilings, 1st floor den, gleaming Brazilian cherry hardwood floors, gourmet kitchen,
generous family room with fireplace, 2nd floor loft and luxury master bedroom suite make
this home a must see.
                  </ol>
                 </html>"
    }
```

The above List Elements consists of

1. **Title**: It is used for GPS filtering. Thus if the reverse geocode determines that you are in Naperville, data with 'title' incorporating Naperville shows up on the mobile canvas.
2. **Picture**: It is the picture of the house on sale.
3. **Material**: The flooring material in each room is given here
4. **Description**: This is the general description of the house on sale.

In essence, any number of list elements can be added depending on the requirement of realty agency.

Images

The camera points to a house in koramangalam, Bangalore. The cam and GPS filters the List view and the filtered list appears on screen showing the various house on sale in Koramangalam, Bangalore, India (Back Img simulated)

(C) Copyright Nokia 2014. All rights reserved.

When one of the houses thrown up by GPS filtered list is clicked, we get the description of the house on sale. Depending on realty agent requirements, many more options can be added to the delegates

Summary

We can use GPS and List Element to show relevant House on sale data. The GPS filters the list depending on where the camera points. We can thus use GPS, ListView and Mobility to create an effective Realty app.

Please feel free to edit for clarity or more material

(C) Copyright Nokia 2014. All rights reserved.