QAugmentedReality - Local based augmented reality component

This article describes QAugmentedReality, an augmented reality (AR) engine that makes it easy for you to create AR applications like Nokia City Lens in QML.

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

Introduction

`QAugmentedReality` is an Augmented Reality engine that provides all functionalities to create your own local based augmented reality application like Nokia City Lens or social applications for example displaying your friends location or your parked car location direction.

The component built entirely from scratch, inherits all camera features adding GPS, Accelerometer and Compass support.

Developers just need to setup for each item the longitude, latitude, description text, image and the engine will do the hard work to manage rendering, gps, compass. The component's strength is the **model** support that allows you to easy manage your data to display. To add a **pin** into your scene you just need to add a `QAugmentedRealityCameraItem` element to `QAugmentedRealityCameraModel`. A video of the component in operation is given below: The media player is loading...

Get Started with QAugmentedReality

- Download the project from here
- Include **QAugmentedReality** directory into your project

.pro

```
include(QAugmentedReality/AugmentedRealityLibrary.pri)
```

main.cpp

```
#include <QtGui/QApplication>
#include "qmlapplicationviewer.h"

#include <QtDeclarative>
#include "qaugmentedrealitycamera.h"

Q_DECL_EXPORT int main(int argc, char *argv[])
{
    QScopedPointer<QApplication> app(createApplication(argc, argv));

    QmlApplicationViewer viewer;

    qmlRegisterType<QAugmentedRealityCamera>("QAugmentedRealityCamera", 1, 0,
"QAugmentedRealityCamera");
    qmlRegisterType<QAugmentedRealityCameraModel>("QAugmentedRealityCamera", 1, 0,
"QAugmentedRealityCameraModel");
    qmlRegisterType<QAugmentedRealityCameraItem>("QAugmentedRealityCamera", 1, 0,
"QAugmentedRealityCameraItem");
    qmlRegisterType<QAugmentedRealityCameraDelegate>("QAugmentedRealityCamera", 1, 0,
"QAugmentedRealityCameraDelegate");

    // Resize the root QML element to view size
    viewer.setResizeMode(QDeclarativeView::SizeRootObjectToView);

    // Performance optimization flags
    viewer.setAttribute(Qt::WA_OpaquePaintEvent);
    viewer.setAttribute(Qt::WA_NoSystemBackground);
    viewer.viewport()->setAttribute(Qt::WA_OpaquePaintEvent);
    viewer.viewport()->setAttribute(Qt::WA_NoSystemBackground);
```


(C) Copyright Nokia 2014. All rights reserved.

```
    viewer.setViewportUpdateMode(QGraphicsView::FullViewportUpdate);


    viewer.setOrientation(QmlApplicationViewer::ScreenOrientationLockLandscape);
    viewer.setMainQmlFile(QLatin1String("qml/QAugmentedRealitySample/main.qml"));
    viewer.showExpanded();

    return app->exec();
}
```

QML

```
import QtQuick 1.1
import QAugmentedRealityCamera 1.0

Rectangle {
    id:home
    width: 640
    height: 360
    color:"white"

    QAugmentedRealityCameraModel {
        id:pinsModel
        pins: [
            QAugmentedRealityCameraItem {
                latitude: 41.891033
                longitude:12.492599
                message:"Rome"
                source: ":/images/colosseo.jpg"
            },
            QAugmentedRealityCameraItem {
                latitude: 45.463502
                longitude:9.187946
                message:"Milan"
                source: ":/images/milan.jpg"
            },
            QAugmentedRealityCameraItem {
                latitude: 48.856358
                longitude:2.352047
                message:"Paris"
                infoBox: "#FFAA23"
                source: ":/images/paris.jpg"
            },
            QAugmentedRealityCameraItem {
                latitude: 51.50746
                longitude:-0.127831
                message:"London"
            },
            QAugmentedRealityCameraItem {
                latitude: 52.518728
                longitude:13.404465
                message:"Berlin"
                source: ":/images/berlin.jpg"
            },
            QAugmentedRealityCameraItem {
                latitude: 40.714997
                longitude:-74.006767
                message:"New York"
                source: ":/images/new-york.jpg"
            },
```


(C) Copyright Nokia 2014. All rights reserved.

```
            QAugmentedRealityCameraItem {
                latitude: 35.689649
                longitude:139.691048
                message:"Tokyo"
            },
            QAugmentedRealityCameraItem {
                latitude: -37.813717
                longitude:144.962955
                message:"Melbourne"
                infoBox: "blue"
            }
        ]
    }

    QAugmentedRealityCamera {
        id:camera
        anchors.fill: parent
        resolution: Qt.size(2592,1456)
        model: pinsModel
    }

    Component.onCompleted:{
        camera.start(camera.availableDevices[0])

        pinsModel.append({
                            "latitude": 45.428094,
                            "longitude": 9.313946,
                            "message": "Peschiera Borromeo",
                            "source": ":/images/bino.jpg"
                        })
        pinsModel.append({
                            "latitude": -34.603541,
                            "longitude": -58.381748,
                            "message": "Buenos Aires",
                            "infoBox" : "white"
                        })

    }
}
```

QAugmentedRealityCamera

`QAugmentedRealityCamera` is the component that provide not only the camera display feature, but includes GPS, Accelerometer, and Compass to display Pins on screen and being able to move them depending on watching direction.

To add a pin you need setup a `QAugmentedRealityCameraModel` for the component.

QAugmentedRealityCameraModel

QAugmentedRealityCameraModel has a `pins` property that have to be filled with `QAugmentedRealityCameraItem` element.

You can add `QAugmentedRealityCameraItem` statically or dynamically.

Adding a pin statically

```
QAugmentedRealityCameraModel {
        id:pinsModel
        pins: [
            QAugmentedRealityCameraItem {
                latitude: 41.891033
                longitude:12.492599
                message:"Rome"
                source: ":/images/colosseo.jpg"
```


(C) Copyright Nokia 2014. All rights reserved.

```
            }]
}
```

Adding a pin dynamically

This approach is preferred when fetching data from a database.

```
Component.onCompleted:{
        pinsModel.append({
                            "latitude": 45.428094,
                            "longitude": 9.313946,
                            "message": "Peschiera Borromeo",
                            "source": ":/images/bino.jpg"
                        })
}
```

QAugmentedRealityCameraItem

`QAugmentedRealityCameraItem` has the following properties:

- `latitude` - latitude of the item
- `longitude` - longitude of the item
- `message` - description text displayed inside the pin
- `infoBox` - the color of the box drawn on the left of the pin. This is because to give you the possibility to have pins of different color for your own purpose
- `source` - As for `infoBox` this feature allow you to set an image for your pin. Transparent images are allowed. The `infoBox` background color will be displayed on transparent areas.

Shooting still images

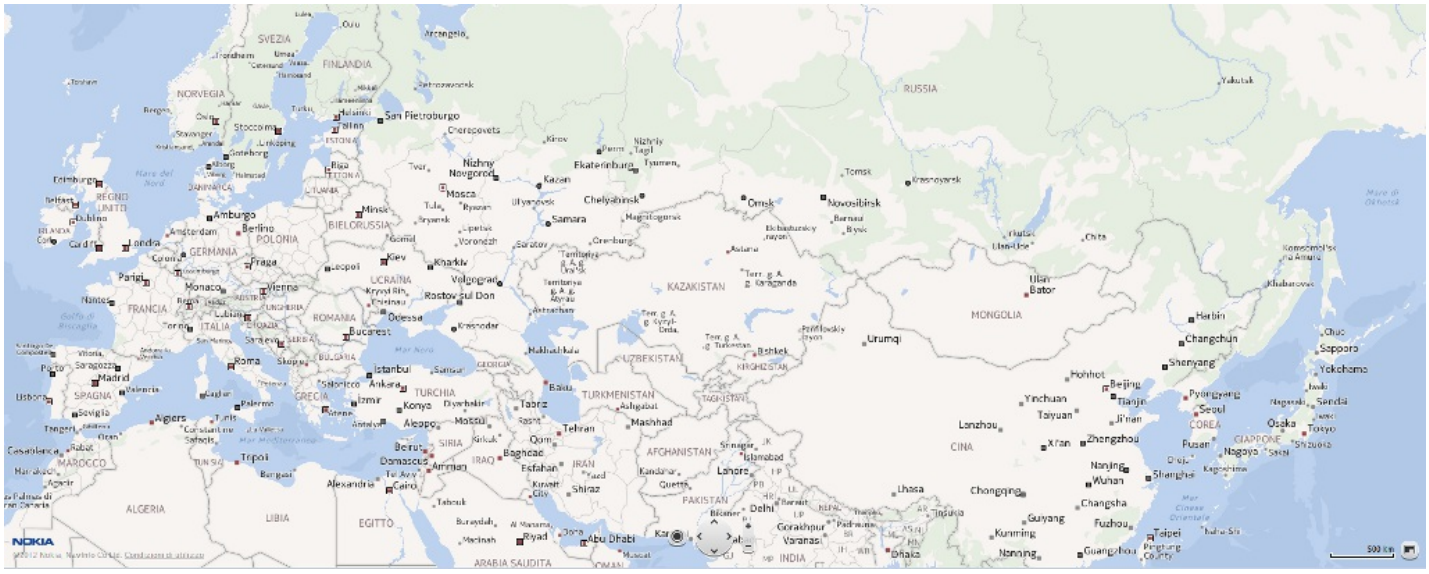
The component inherits from a base camera component so is possible to capture photos as with the default camera component. The result image will include the augmented reality items in the scene.

This feature will be improved as soon as possible to be fitable with all resolutions.

(C) Copyright Nokia 2014. All rights reserved.

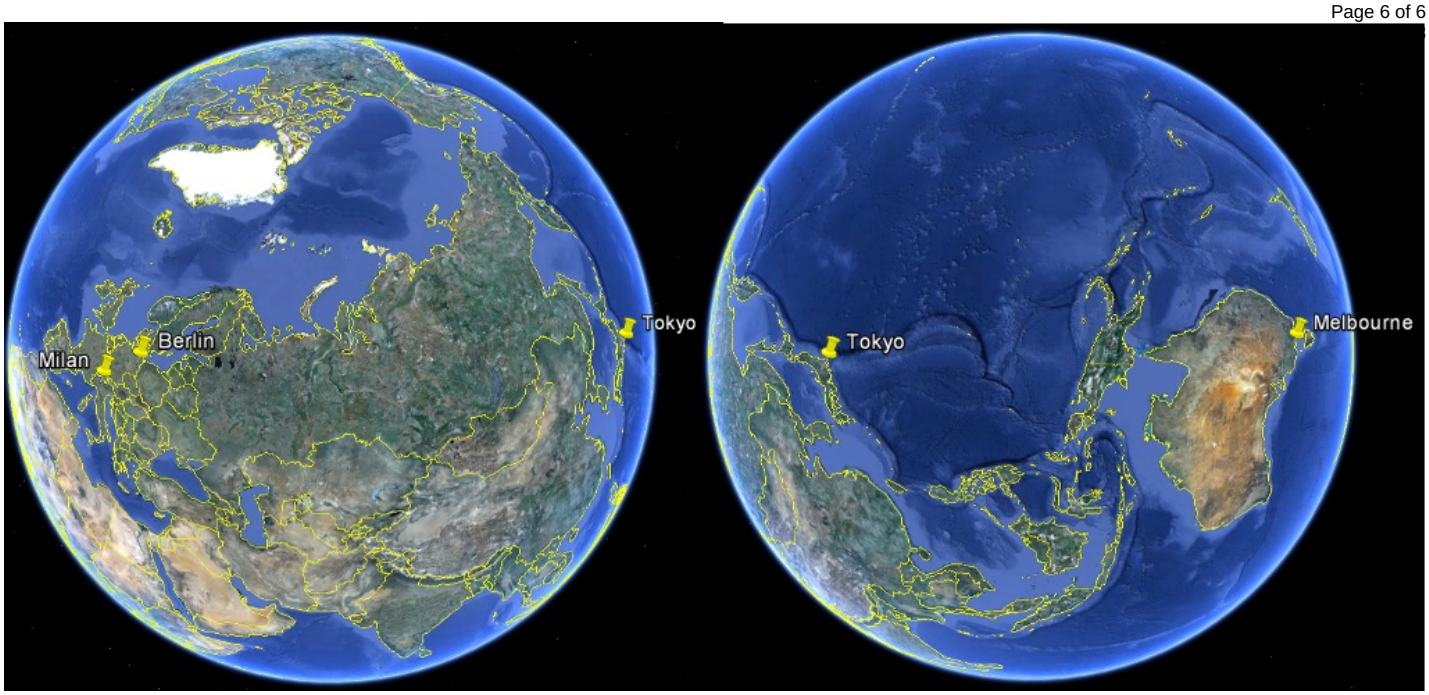
Be careful of large distances - the world is not flat

For pins located a very long way away you may see odd effects. Consider the example showcased on the video above - at certain watching angles Berlin, Tokyo and Melbourne appear close together.



This happens because the earth has a spherical shape and points in a sphere are located using polar coordinate system.

(C) Copyright Nokia 2014. All rights reserved.



In our example what happens is that the bearing between Milan, Berlin, Tokyo and Melbourne is quite small - these are therefore shown as close together at some watching angle. This should not be a problem if you're working with objects that are within tens of kilometres or less.

Sample application

- QAugmentedRealitySample_installer.sis

Note: Don't forget to shake the phone for compass calibration or pins will not be displayed correctly

Further improvements

Improvements that will come as soon as possible:

- filter item based on max distance, for example to display just items far less then 150mt
- At time of writing the **delegate** is static. if possible to allow users to create their own delegate
- Provided a customizable delegate the next step is to provide a customizable model
- **qmldir** usage

Despite improvements that will come the component is easy and powerful enough to start creating your local based augmented reality applications and enjoy with them.

If you have any questions about using the component or suggestions for improvement please add them in the comment section below.

(C) Copyright Nokia 2014. All rights reserved.