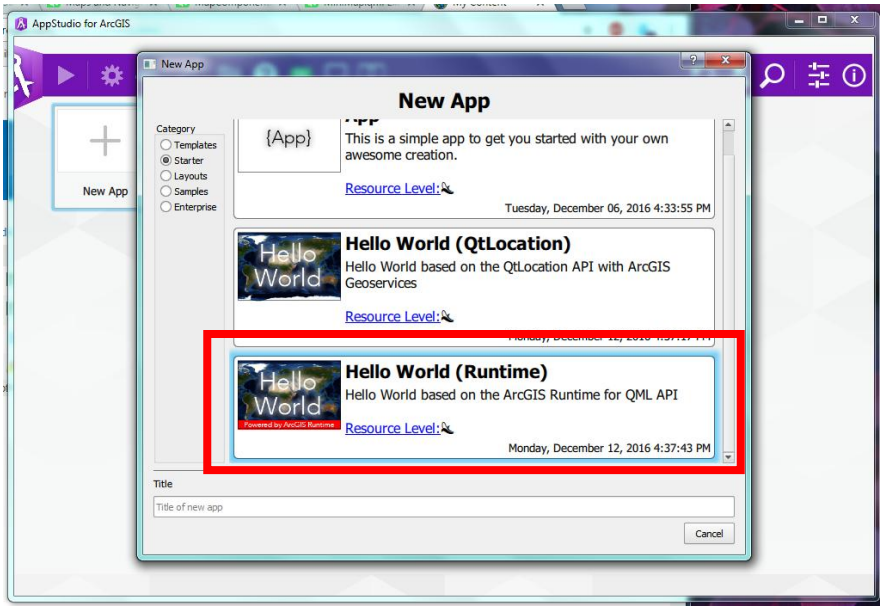
Qt/QML Tutorial: Find Location

The purpose of this tutorial is to create an application that displays a user's location.
This code can be utilized with a Feature Layer to record points and edit the components of recorded points.

Instructions

1. Open App Studio for ArcGIS

2. Create a New App. When prompted, select **Starter** from the Categories on the left, and then choose '**Hello World (Runtime)**. Name your app something creative, like **[lastname]_FindLocation**.



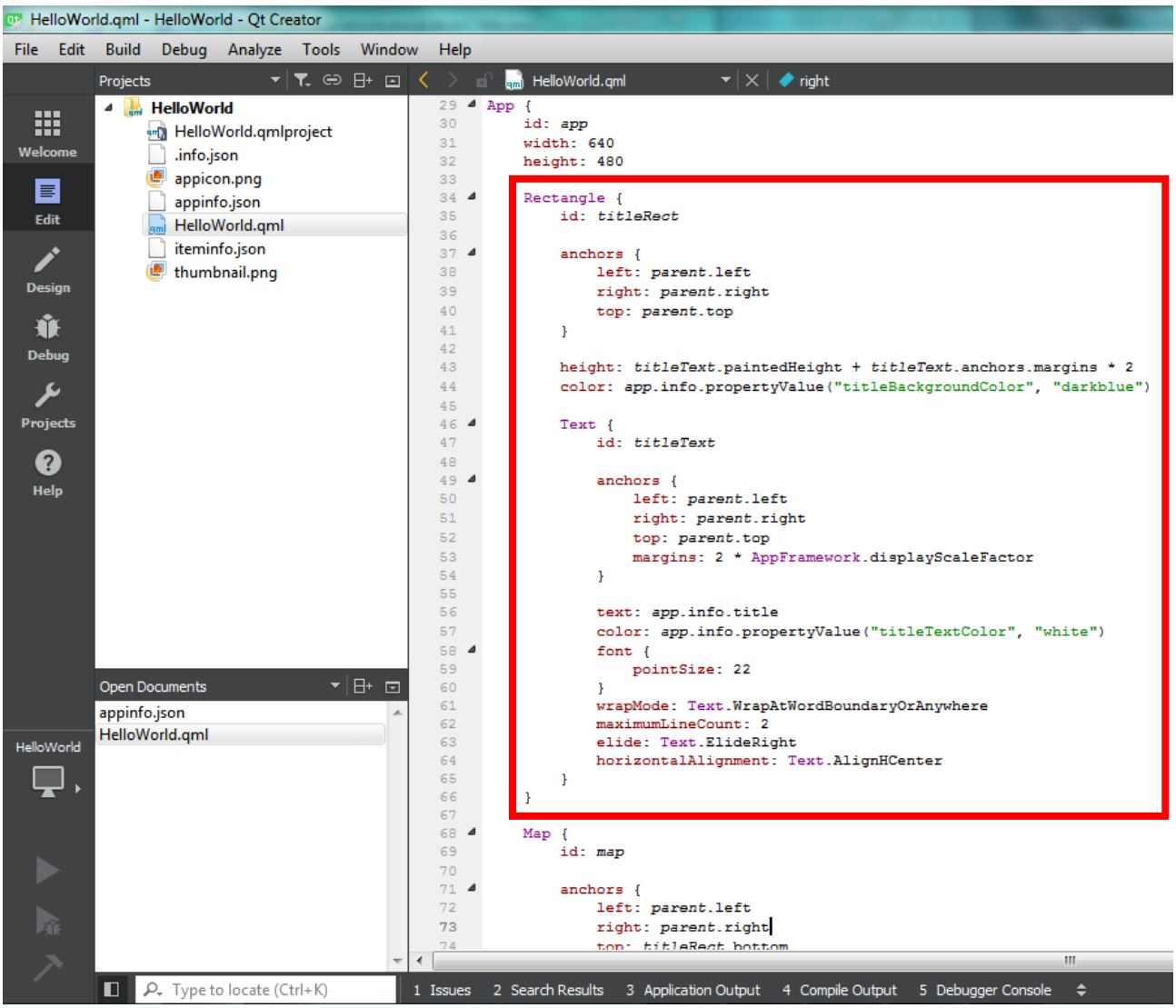
3. Double click your newly created app. Your application will open, and you'll see that you've created a basic application with a world map, a title bar, and some map navigation tools.

To edit this app, we need to open the file in Qt Creator. Select the file in Esri App Studio and click on the Qt icon at the top and the QML file will open in Qt. In QT, you have the ability to read and edit the QML document. Select the '**HelloWorld.qml**' file by double clicking on it.

4. To enable some of the features that we will add later on, we need to import a few modules.
Add the following lines of code at the top of your HelloWorld.qml file:

```
17  import QtQuick 2.3
18  import QtQuick.Controls 1.2
19  import QtQuick.Layouts 1.1
20  import QtPositioning 5.3
21
22  //Added
23  import QtGraphicalEffects 1.0
24  import QtSensors 5.0
25  import QtQuick.Dialogs 1.2
26
27  import ArcGIS.AppFramework 1.0
28  import ArcGIS.AppFramework.Controls 1.0
29  import ArcGIS.AppFramework.Runtime 1.0
30  import ArcGIS.AppFramework.Runtime.Controls 1.0
31
```

5. The user probably knows the name of the application once they've opened it, thus making the title bar redundant. To remove the title bar from the top of the open application, remove the following lines of code:



6. Now that the title bar is removed, the map needs to be edited to fill the entire screen.
Edit the '**anchors**' and '**positionDisplay**' elements:

original *edited*

```
Map {
    id: map

    anchors {
        left: parent.left
        right: parent.right
        top: titleRect.bottom
        bottom: parent.bottom
    }

    wrapAroundEnabled: true
    rotationByPinchingEnabled: true
    magnifierOnPressAndHoldEnabled: true
    mapPanningByMagnifierEnabled: true
    zoomByPinchingEnabled: true

    positionDisplay {
        positionSource: PositionSource {
        }
    }
```

```
Map {
    id: map

    //Altered to utilize full screen without title
    anchors.fill: parent
    extent: envelopeInitalExtent
    wrapAroundEnabled: true
    rotationByPinchingEnabled: true
    magnifierOnPressAndHoldEnabled: true
    mapPanningByMagnifierEnabled: true
    zoomByPinchingEnabled: true
```

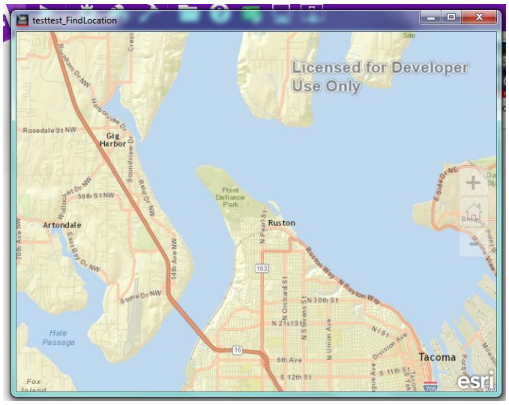
Since the user can move around with their cursor or fingers, the NorthArrow and ZoomButtons are unneccessary. Delete these elements and their components (located at the bottom of the code file.

7. To edit the initial extent of the map, enter the following code following **zoomByPinchingEnabled**.
   For this tutorial, the map is center on northern Tacoma, WA. If you would like to center the map on a different location, edit the *xMax/xMin/yMax/yMin* values to your desired area.

```
//Added and altered to center on N. Tacoma
//INITIAL EXTENT
Envelope {
    id: envelopeInitalExtent
    xMax: -13630134.691272736
    yMax: 6001320.7069897875
    xMin: -13647294.804122735
    yMin: 5982211.44991852
    spatialReference: mainMap.spatialReference
}
```

Following the added lines of code, the **ArcGISTiledMapServiceLayer** element is where the basemap layer can be changed. For the purpose of this tutorial, the default layer is fine.

Save your ongoing Qt file. Navigate back to AppStudio and open the app. You should now have an application that has no title bar and is centered on your desired location.



8. Next is to add a marker that displays the user's position on the map. Add the following lines of code under the **ArcGISTiledMapServiceLayer** element:

```
//Added
//ENABLE POSITION ONCE MAP IS CREATED
onStatusChanged: {
    if (status === Enums.MapStatusReady) {
        positionSource.active = true;
    }
}

//CREATE USER POSITION DISPLAY
positionDisplay {
    id: positionDisplay
    zoomScale: 200000
    mode: Enums.AutoPanModeDefault
    positionSource: PositionSource {
        id: positionSource
    }
}
```

9. Wouldn't it be slick if the user could turn on and off that display marker? To enable this, you need to create a button that allows for the user to turn the marker on and off. Add the following lines of code:

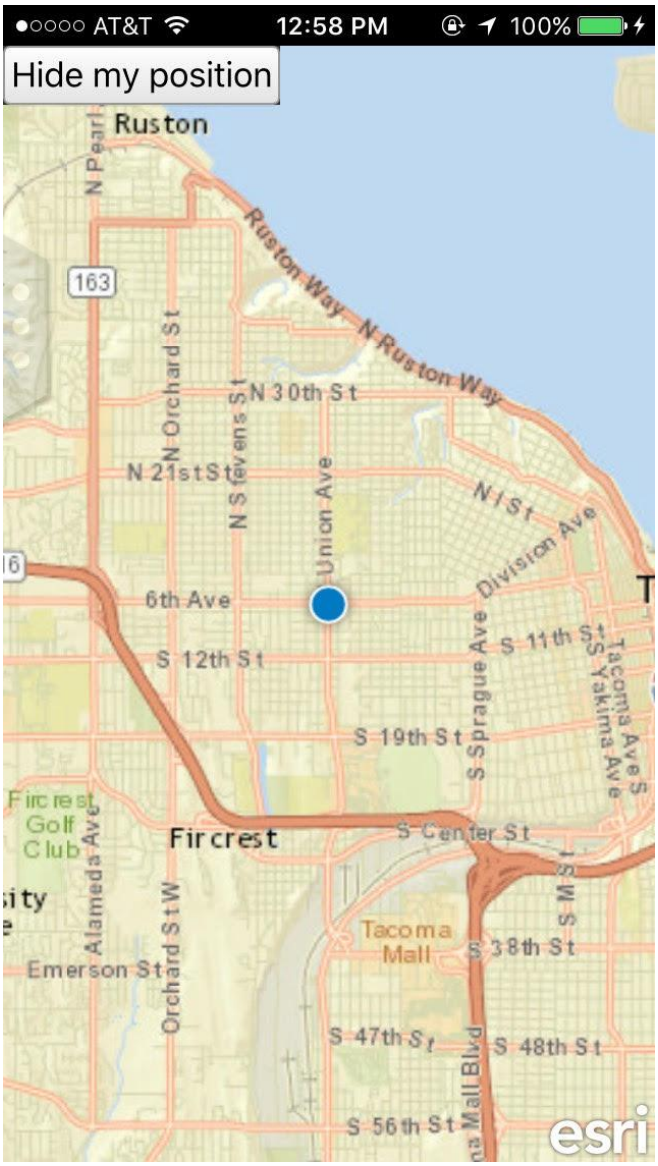
```
//BUTTON BACKGROUND
Rectangle {
    anchors {
        fill: controlsColumn
        margins: -10 * scaleFactor
    }
    color: "lightgrey"
    radius: 5 * scaleFactor
    border.color: "black"
    opacity: 0.77
}

//Added
//BUTTON AREA
Column {
    id: controlsColumn
    anchors {
        left: mainMap.left
        top: mainMap.top
        margins: 20 * scaleFactor
    }
    spacing: 7

    Button {
        id: showPosition
        text: "Hide my position"
        width: addLocationPoint.width
        enabled: mainMap.status === Enums.MapStatusReady

        onClicked: {
            if(positionSource.active === true){
                positionSource.active = false;
                showPosition.text = "Show my position"
            }
            else{
                positionSource.active = true;
                positionDisplay.mode = Enums.AutoPanModeDefault
                showPosition.text = "Hide my position"
            }
        }
    }
```

10. Save your file. Navigate back to AppStudio and double click on your application to ensure that it builds. In this viewer, the marker may not show up. To test this application, I published my application to ArcGIS online and accessed the application via the **AppStudio Player for ArcGIS** application on my iPhone. Below is a screenshot of the created application on my phone:



If you have any questions regarding this tutorial, please contact Emily Pitman at epitman@uw.edu.
Last updated 2/27/2017.