**Using ArcGIS App Studio, QT and QML:**

**How to Display Mouse Coordinates in a Mobile Mapping Application**

**Author:** Chris Vandenberg, MS Student at the University of Washington – Tacoma

**Purpose:** This tutorial will introduce you to the basics of working with ArcGIS App Studio and the QT interactive development environment (IDE) to create and customize mobile applications using the declarative programming language QML. By utilizing the examples provided by ESRI, users can start to breakdown and understand the many components of QML and how they interact to create interactive and user-friendly mobile applications. More specifically, this tutorial will customize the Hello World Runtime starter application to display the location of the user’s mouse in coordinates. The software components needed for this tutorial can be found in the links below.

ArcGIS App Studio = <https://appstudio.arcgis.com/>

GitHub Mouse Coordinates Example =

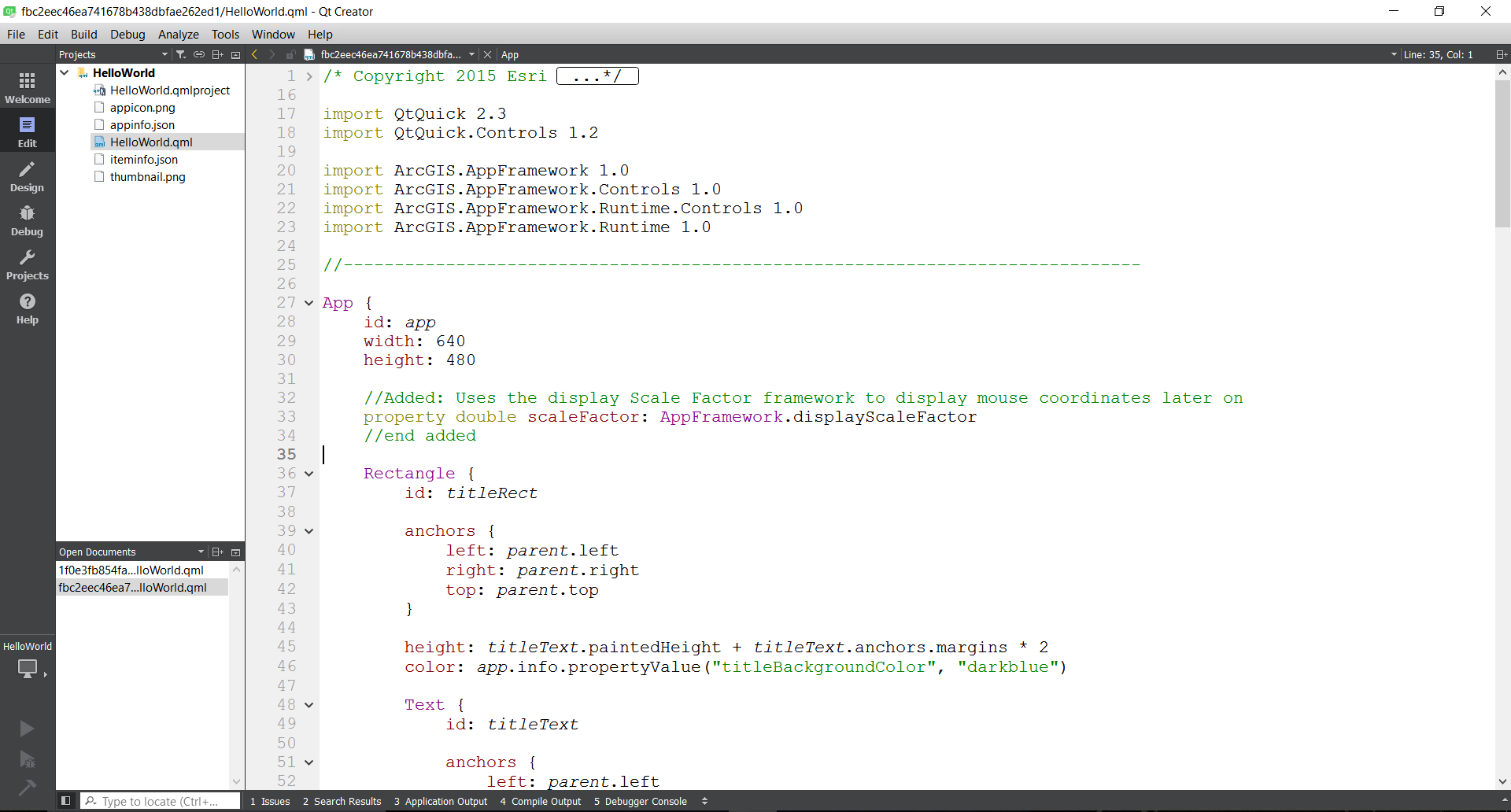
## (Follow the Instructions to run this sample in AppStudio Desktop

## Open ArcGIS App Studio and create a new Hello World (Runtime) Application by selecting the “new app“button, then click on the “starter” category and give your application a unique name.

## 

1. Add the following code to your HelloWorld.qml. It should be added below the App { statement, as shown below. This code declares the display scale factor framework which will be used to display mouse coordinates later on.

## property double scaleFactor: AppFramework.displayScaleFactor



## 

1. Comment out the position display statement, which displays the user location. This throws an error later on when we build our app. Do this by selecting the code, holding down ctrl and entering a single backslash ( / ).



1. Next, scroll down in your HelloWorld.qml code and paste the following code below the url statement in the ArcGISTiledMaServiceLayer { section (see screenshot). This code uses the mouse area position and converts these coordinates to text, displaying them in the DegreesMinutesSeconds format.

MouseArea {

anchors.fill: *parent*

hoverEnabled: true

onPositionChanged: {

var *mapPoint* = *map*.toMapGeometry(*mapToItem*(*map*, *mouseX*, *mouseY*)); //changed mainMap to map

coordsText.text = *mapPoint*.toDegreesMinutesSeconds(2);

mouseText.text = "Mouse: X=" + *mouseX*.toString() + " Y=" + *mouseY*.toString();

}

}



1. Lastly, scroll to the bottom of your HelloWorld.qml code and add the following code beneath the Zoom object statement. This code creates envelope, rectangle and column containers which are used to display the mouse coordinates that are retrieved by referencing the scale factor variable within those containers**.**

//Added: Creates an envelope and rectangle in which the mouse coordinates are retrieved via the scale factor variable.

Envelope {

id: *initialExtent*

xMax: -15000000

yMax: 2000000

xMin: -7000000

yMin: 8000000

}

Rectangle {

id: *rectangleControls*

color: "lightgrey"

radius: 5

border.color: "black"

opacity: 0.77

anchors {

fill: *columnControls*

margins: -10 \* *scaleFactor*

}

}

Column {

id: *columnControls*

anchors {

top: *parent*.top

left: *parent*.left

margins: 20 \* *scaleFactor*

}

spacing: 5 \* *scaleFactor*

Text {

id: *coordsText*

}

Text {

id: *mouseText*

}

}

Rectangle {

anchors.fill: *parent*

color: "transparent"

border {

width: 0.5 \* *scaleFactor*

color: "black"

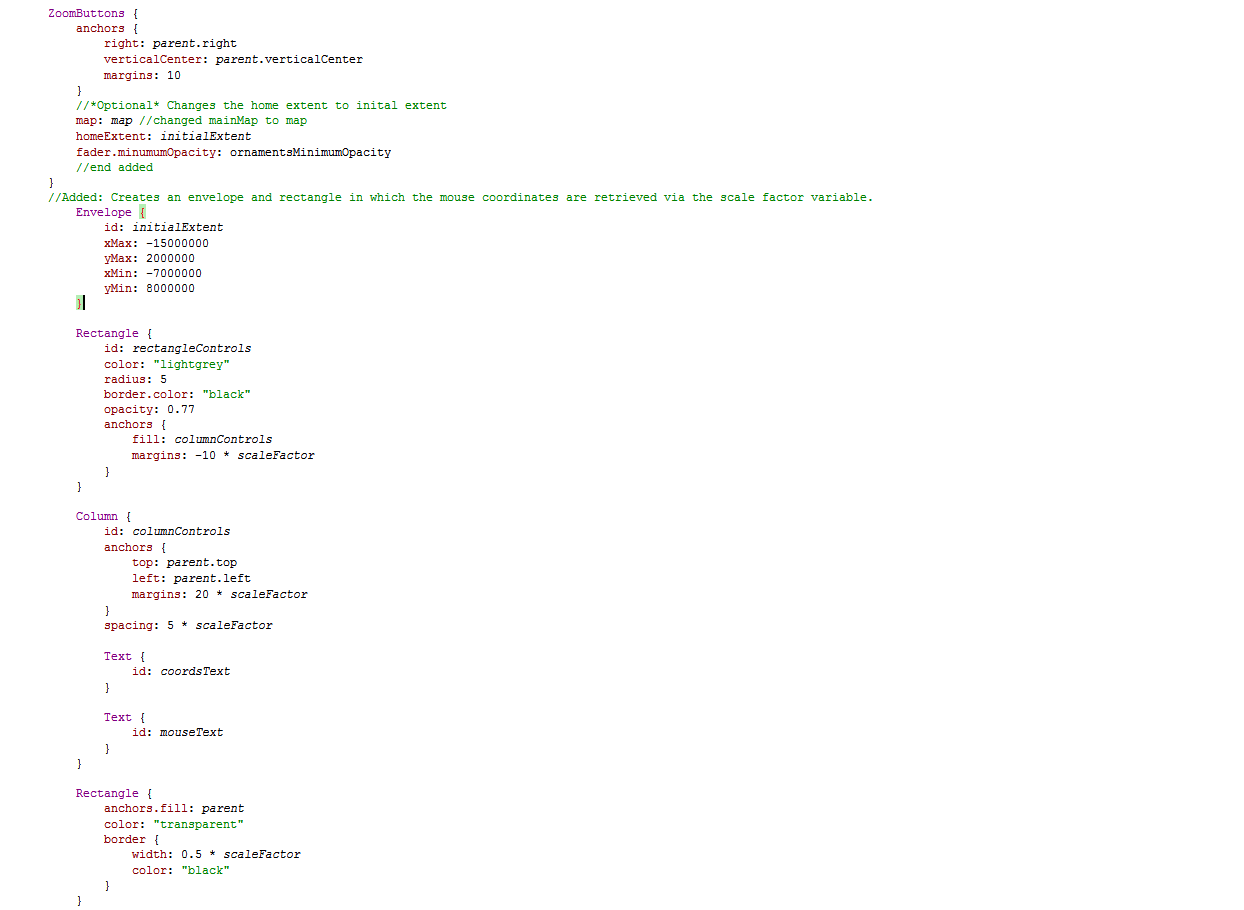
}

}

//end added

}

}



**TIP:** Don’t forget to close your brackets at the bottom of your QML code!

1. You’re almost done! Click File and Save All to save your QML project in the QT creator window. Then go back to AppStudio for ArcGIS and click your new application. The application should successfully build and display the coordinates of your mouse in the upperleft corner.

