Enabling pinch zooming in QGraphicsWebView with QPinchGesture

This article shows how to use QPinchGesture to add support for pinch-zooming to a QGraphicsWebView.

Code overview

Our first step is to enable touch events. This enables us to receive `TouchBegin` event.

```
MyWebView::MyWebView(QGraphicsItem *parent) :
    QGraphicsWebView(parent)
{
    setAcceptTouchEvents(true); //set this to receive touch events
    grabGesture(Qt::PinchGesture);
}
```

Pinch gesture is enabled by calling `grabGesture(Qt::PinchGesture) ;` . This will make `MyWebView` as the target for `QGestureEvents`. Standard event handling needs to bypassed by reimplementing QGraphicsWebView's `sceneEvent(QEvent *event)`.

```
bool MyWebView::sceneEvent(QEvent *event)
{
    switch(event->type()){

    case QEvent::TouchBegin:
        //accepting touch begin allows us to get touch updates
        return true;
        break;

    case QEvent::Gesture:
        return gestureEvent(static_cast<QGestureEvent*>(event));
        break;

    default:
        return QGraphicsWebView::sceneEvent(event);
    }
}
```

`TouchBegin` event must be accepted before you will receive additional `TouchUpdate` and `TouchEnd` events, which are needed by gesture recognizer to work.

When Gesture -event is received we pass it to gestureEvent- handler that inspects type of the gesture and forwards it to pinchGesture handler. Using `gestureEvent` handler is useful when you want to play with multiple gestures.

```
bool MyWebView::gestureEvent(QGestureEvent *event)
{
    if (QGesture *pinch = event->gesture(Qt::PinchGesture)){
         pinchTriggered(static_cast<QPinchGesture *>(pinch));
    }
    return true;
}
```

The last step is to zoom the current page according to user input. It is a good idea to limit the possible zoom factor to a certain interval as very big and small zoom levels are seldom useful.

```
void MyWebView::pinchTriggered(QPinchGesture *gesture)
{
    QPinchGesture::ChangeFlags changeFlags = gesture->changeFlags();
    if (changeFlags & QPinchGesture::ScaleFactorChanged) {
   qreal value = gesture->scaleFactor();
                qreal zoom = value*zoomFactor();

 if(zoom < 2 && zoom > 0.5){
                    qDebug()<< "zooming";
                    setZoomFactor(zoom);
            }
    }
}
```

More information

Check your SDK gesture example for how to use pan and swipe events:

- \\NokiaQtSDK\Examples\4.6\gestures
- QPinchGesture Class Reference

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.