Jigsaw Puzzle for QML Beginners

This article uses a simple jigsaw puzzle game to illustrate how to make Qt Quick items *draggable*. It is targeted towards new Qt Quick developers.

20 Feb 2011

How it works

The example uses square image cutouts that you can create from your own pictures using online image splitters (for example). The code defines elements for each image that contain a MouseArea with drag properties defined - these make the item draggable.

Pre-Conditions

Assumption is that you have 3X3 (9) cutouts of you image for puzzle.

QML Code

```
import QtQuick 1.0

Rectangle {

    id: screen

    Image {
        source: "11.JPG"

        MouseArea {
            // Mouse Area
            anchors.fill: parent
            // specifies the id of the item to drag.
            drag.target : parent
            // minimumX & minimumY limits how far the target can be dragged along the axis ( X1, Y1)
            drag.minimumX: 0
            drag.minimumY: 0
            // maximumX & maximumY limit how far the target can be dragged along along the axis ( X2, Y2)
            // Lets force the image to be within the visible screen area
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    // Same properties apply for other image elements
    Image {
        source: "12.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "13.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "14.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "15.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "16.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
```

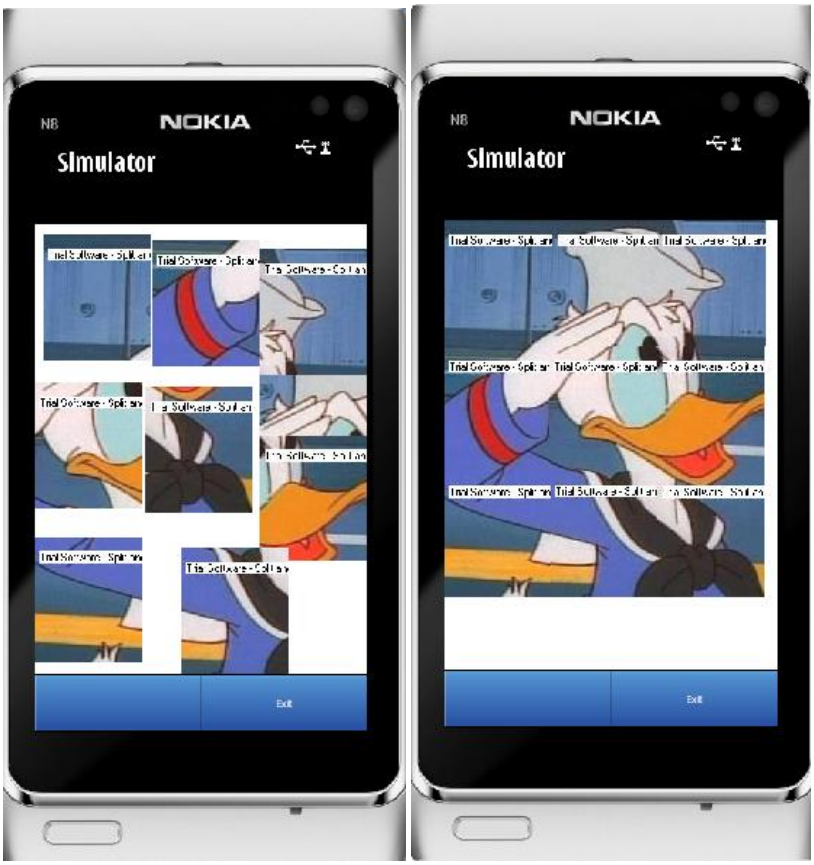

(C) Copyright Nokia 2013. All rights reserved.

```
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "17.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "18.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

    Image {
        source: "19.JPG"
        MouseArea {
            anchors.fill: parent
            drag.target : parent
            drag.minimumX: 0
            drag.minimumY: 0
            drag.maximumX: screen.width - parent.width
            drag.maximumY: screen.height - parent.height
        }
    }

}
```

Limitations

Since this is very basic version for beginners, example or code doesn't have game specific logic involved, only drag for each items is defined, there is no puzzle completion logic written, its up to user to visually check if he is able to solve the puzzle, game logic will be continued in advanced topic.

Screen Shots



(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.