

Introducción a QML – Primer ejemplo III

En esta parte del ejercicio empezaremos a dotar de dinamismo a nuestro formulario QML. Para ello aplicaremos alguno de los conceptos adquiridos en los últimos post relacionados.

Una de las animaciones que implementaremos mostrará un icono asociado al botón “Enviar”. Cuando pulsemos este botón reduciremos el tamaño del icono.

La segunda animación será para cambiar progresivamente el tamaño de los iconos de la toolbar.

Código fuente del botón “Enviar”

Nótese que para destacar el botón cambiamos el valor de dos propiedades al pulsar sobre el mismo mediante “State”.

```
2  # Rectangle {
3      id: container
4      property string colorString: "#343434"
5      property alias text: label.text
6      property alias imageString: image1.source
7
8      color: colorString
9      smooth: true
10
11     signal clicked
12
13     # Background {
14         id: background1
15     }
16     # MouseArea {
17         id: mouseArea
18         anchors.fill: parent
19         onClicked:{
20             container.clicked()
21         }
22     }
23     # states: State {
24         name: "pressed"; when: mouseArea.pressed == true
25         PropertyChanges { target: image1; scale: 0.7; }
26         PropertyChanges { target: label; color: "#aaafff"; }
27     }
28     # BorderImage {
29         id: buttonImage
30         source: "images/toolbutton.sci"
31         width: container.width
32         height: container.height
33         anchors.centerIn: parent
34         # Image {
35             id: image1
36             smooth: true
37             scale: 0.8
38             width: 36
39             height: 36
40             source: imageString
41         }
42     }
43     # Text {
44         id: label
45         width: 0
46         font.bold: true
47         style: Text.Raised
48         styleColor: "black"
49         font.pixelSize: 12
50         color: "#ffffff"
51         transformOrigin: "Center"
52         anchors.verticalCenterOffset: 0
53         anchors.horizontalCenterOffset: 10
54         anchors.centerIn: parent
55         smooth: true
56         verticalAlignment: "AlignVCenter"
57         horizontalAlignment: "AlignHCenter"
58     }
59 }
60
```

Código fuente del elemento botón de la toolbar

En este caso además del cambio de estado gestionaremos la “Transition” del mismo sobre la propiedad scale.

```
3  Rectangle {
4      id: rootArea
5      property string colorString: "#343434"
6      property alias imageString: imagel.source
7      property Item rootRect
8
9      signal clicked
10
11     color: colorString
12     radius: 16
13
14     border.color: "#FFFFFF"; border.width: 2
15
16     width: 96; height: 96;
17
18     MouseArea {
19         id: mouseArea
20         anchors.fill: parent
21
22         onClicked:
23             rootArea.clicked()
24
25         states: State {
26             name: "pressed"; when: mouseArea.pressed == true
27             PropertyChanges { target: imagel; scale: 0.6; }
28         }
29
30         transitions: Transition {
31             PropertyAnimation{ properties: "scale"; duration: 50}
32         }
33
34         BorderImage {
35             id: buttonImage
36             width: parent.width
37             height: parent.height
38             Image {
39                 id: imagel
40                 smooth: true
41                 scale: 0.7
42                 width: parent.width
43                 height: parent.height
44                 source: imageString
45             }
46         }
47     }
48 }
49
```