Archived:Creating a customized Ticker Component in Flash Lite

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

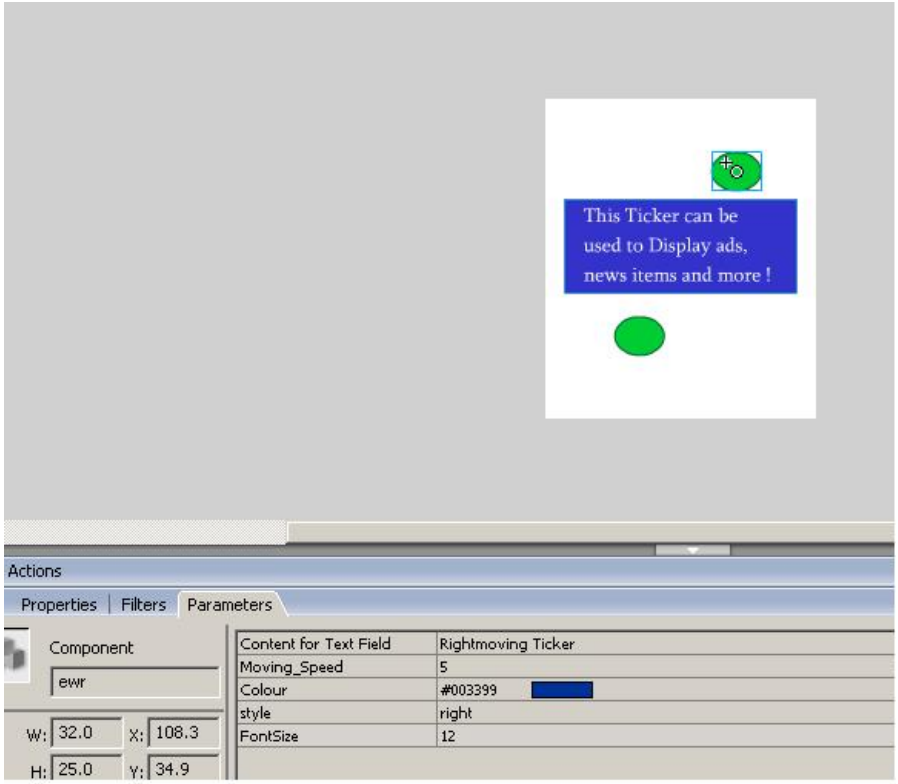
We do not recommend Flash Lite development on current Nokia devices, and all Flash Lite articles on this wiki have been archived. Flash Lite has been removed from all Nokia Asha and recent Series 40 devices and has limited support on Symbian. Specific information for Nokia Belle is available in Flash Lite on Nokia Browser for Symbian. Specific information for OLD Series 40 and Symbian devices is available in the Flash Lite Developers Library.

Introduction

The Ticker is essentially a scrolling text that can be used to display advertisements, news headlines, stock quotes and many more. A ticker is an object that provides scrolling text across the top/bottom of the display. Ticker component is one of the simplest components to create on Flash Lite. Such a component is however, not available in the standard components set. Hence, in this article, we discuss how to create such a component.

How to use

Download the source file(FLA) given below. When you wish to use this component in your FLA, import this from the downloaded file, using the FILE->IMPORT->OPEN EXTERNAL LIBRARY option.Drag and drop this component onto the Stage and set the Parameters in the Parameters Tab.



Note: The green circle is just a placeholder to indicate the position of the ticker and will not be visible during execution.

Properties

Standard parameters are

- The Content String: which is the text to be displayed
- Moving_Speed: controls the speed of motion of the Content
- Color: denotes the Color of the Scrolling Text
- Style: denotes the Direction of Motion. Can be left or Right

(C) Copyright Nokia 2014. All rights reserved.

- Last input parameter is the TextFontSize

You can introduce further stylizing parameters like BOLD, ITALICS by appending to the Parameters list.

Implementation

1. To start coding such a component, begin with a New Movie Clip.
2. Create a dummy shape on the stage. This will serve as placeholder, to denote where you have placed it on the Stage. As the ticker's textfield is created dynamically. This is only for your reference and should not appear during the execution.
3. To not display this dummy shape created previously, we need to add some code to the first frame of the topmost layer(the only layer)

```
this._visible = false;
```

4. Now, on the same frame we code to create an EmptyMovieClip on the parent stage and insert a TextField into it. We also create a reference to the TextField by the name textf, this is just for convenience.

```
main.createEmptyMovieClip("ti", main.getNextHighestDepth());
main.ti.createTextField("contentStr", 0, 0, this._y, 50, 50);
var textf = main.ti.contentStr;
```

5. To create the scrolling effect, we need to move its _x and _y properties at regular intervals. This we achieve by including it in a setInterval function.

```
function runthis() {
 if (Style == "left") {
  if (textf._x+textf._width<0) {
   textf._x = Stage.width;
  } else {
   textf._x -= speed;
  }
 } else {
  if (textf._x>Stage.width) {
   textf._x = -textf._width;
  } else {
   textf._x += speed;
  }
 }
}
setInterval(runthis, 0);
```

6. The ticker can be either left moving or right moving and this is decided by the Style variable in the above function.
7. You can also include other stylizing parameters like TextColour, FontSize. This can be best done by creating a TextFormat instance and then binding it to the TextField using setTextFormat().

```
var tf = new TextFormat();
if (TextColor == undefined) {
 tf.color = 0x000000;
} else {
 tf.color = TextColor;
}
// Assure FontSize is non-negative, or take the Default Value
if (FontSize>0) {
 tf.size = FontSize;
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
textf.autoSize = "left";
textf.setTextFormat(tf);
```

8. Now on the library panel, right click on this MovieClip and select Component Definition. Here, add the input parameters (the previously discussed TickerContent, Style, Speed or even add new parameters like BOLD, ITALICS etc) to the list.
9. Now the component is ready and can be dragged onto the stage. You shall see the dummy shape, but this does not appear during run-time.
10. To enable placement of multiple instances of this component, we need to generalize the emptyMovieClip name. To do that, we create a quasi-static variable inst. This denotes the number of instances of the component we have on stage. Modifying the code in step 4 as we enable multiple instance to be placed on the stage.

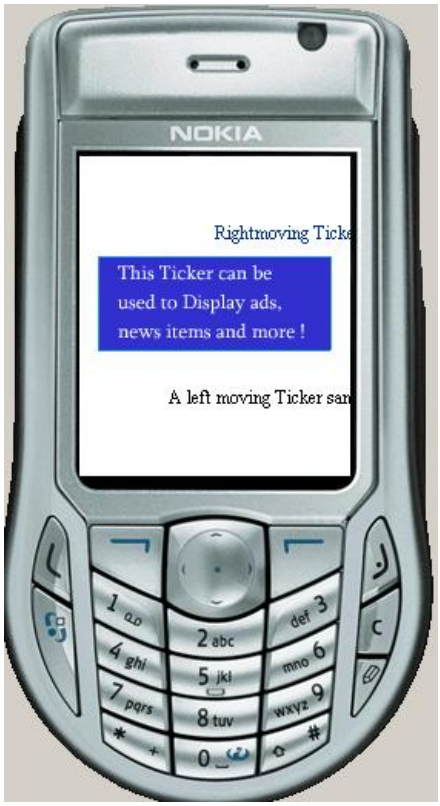
```
if(_parent.inst == undefined) {
 _parent.inst =0;
}
_parent.inst ++;
main.createEmptyMovieClip("ti"+_parent.inst, main.getNextHighestDepth());
eval("main.ti"+_parent.inst).createTextField("contentStr", 0, 0, this._y, 50,
50);
var textf = eval("main.ti"+_parent.inst).contentStr;
```

11. Having the component selected, enter the values to the variables in the Parameters panel.

Source Code

The complete FLA is available for download here.Media:Ticker.zip

Screenshot



(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.