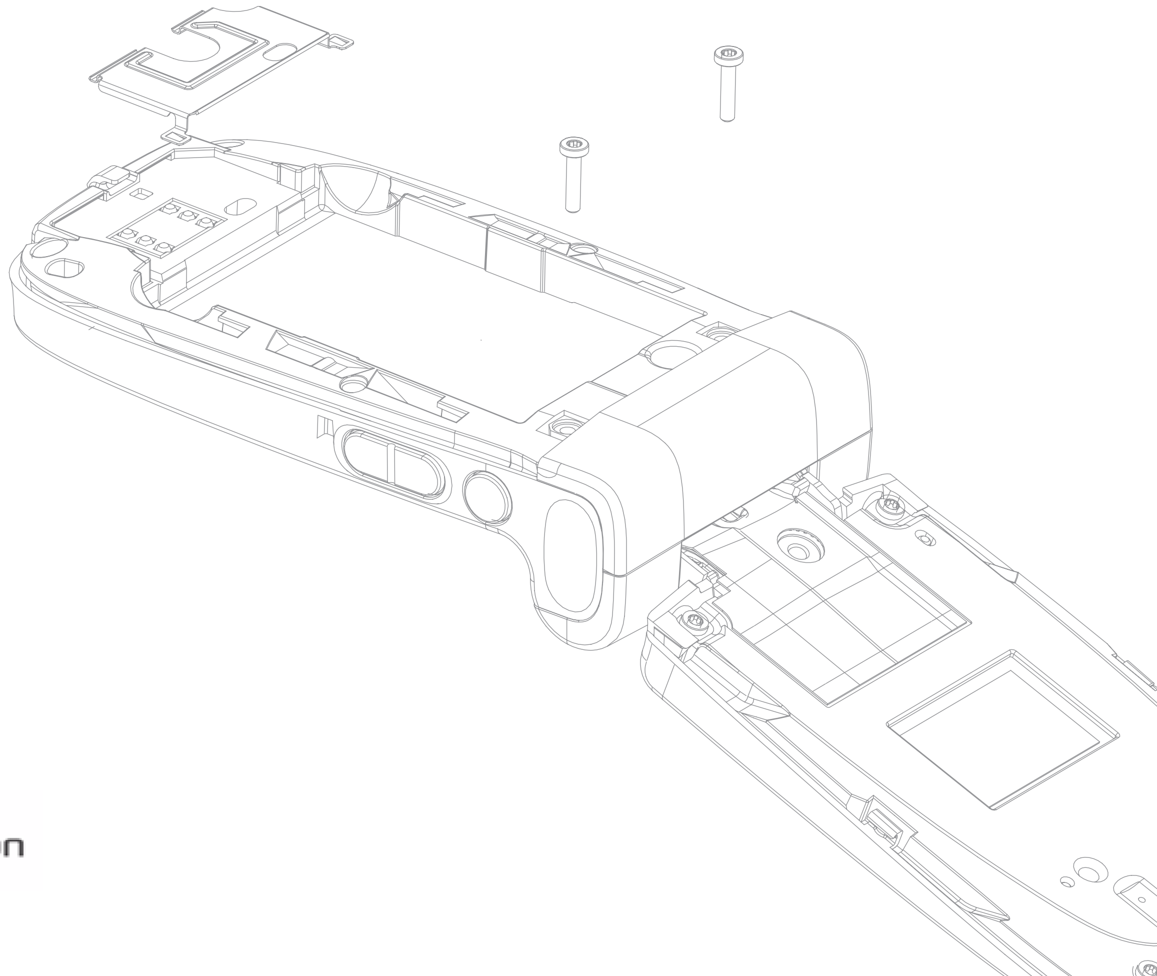
Tutorial

DEVELOPER WORLD THE FAST TRACK FROM MIND TO MARKET

August 2007

Creating dynamic Flash Lite™ animations using phone data

for Sony Ericsson phones



Sony Ericsson

Preface

About this tutorial

This tutorial has been authored by Chris Petty who has been working in multimedia for 6 years before helping to found BlueskyNorth Ltd. a Smashing Ideas company, www.blueskynorth.com, where he is currently Communications Director.

This document is the third part of a series of tutorials, found on Sony Ericsson Developer World, covering different aspects of Adobe™ Flash Lite™ 1.1 development for Sony Ericsson phones.

Prerequisites

To carry out this tutorial you need Adobe Flash™ CS3 Professional with Adobe Device Central™ with the latest Device Profile pack installed. This is available for download at: www.adobe.com/products/creativesuite/devicecentral.

It is also beneficial to have access to the targetted phones, in this case the W850i, and the means of transferring files to your phone, for example, using a Bluetooth™ connection.

The files we create are exported as Flash Lite 1.1. We use ActionScript specific to Flash Lite so access to the Flash Lite 1.1 CDK may prove useful. This is available for download at: www.adobe.com/devnet/devices/development_kits.html.

These Developers guidelines are published by:

Sony Ericsson Mobile Communications AB,
SE-221 88 Lund, Sweden

Phone: +46 46 19 40 00
Fax: +46 46 19 41 00
www.sonyericsson.com/

First edition (August 2007)
Publication number: 1203-4841.1

Sony Ericsson Developer World

On www.sonyericsson.com/developer, developers find documentation and tools such as phone White papers, Developers guidelines for different technologies, SDKs (Software Development Kits) and relevant APIs (Application Programming Interfaces). The Web site also contains discussion forums monitored by the Sony Ericsson Developer Support team, an extensive Knowledge base, Tips and tricks, example code and news.

Sony Ericsson also offers technical support services to professional developers. For more information about these professional services, visit the Sony Ericsson Developer World Web site.

Trademarks and acknowledgements

Document history

Change history		
2007-08-31	Doc. no. 1203-4841.1	Document published on Developer World

Contents

Introduction 5
Tutorial 6
Creating a simple Flash Lite clock 6
  Accessing FS2Commands in Flash 6
Testing the movie 8
Taking this further… 9
Next steps 10
Conclusion 11

Introduction

So far during this tutorial series we have looked at how to create and animate efficient Flash Lite graphics for use on Sony Ericsson phones. We have covered how to optimize our graphics for efficient performance on phone processors and how to test the results in Adobe Device Central to ensure we are delivering the best experience to the end users. In this final tutorial we are looking at how we can use data from the phone to create dynamic animations.

To start, we are going to create a simple clock animation, using the system time from the phone. Following this we look at how these variables can be used to create time reactive content to create animations that change throughout a day. Finally we take a look at some other areas where these principles can be used.

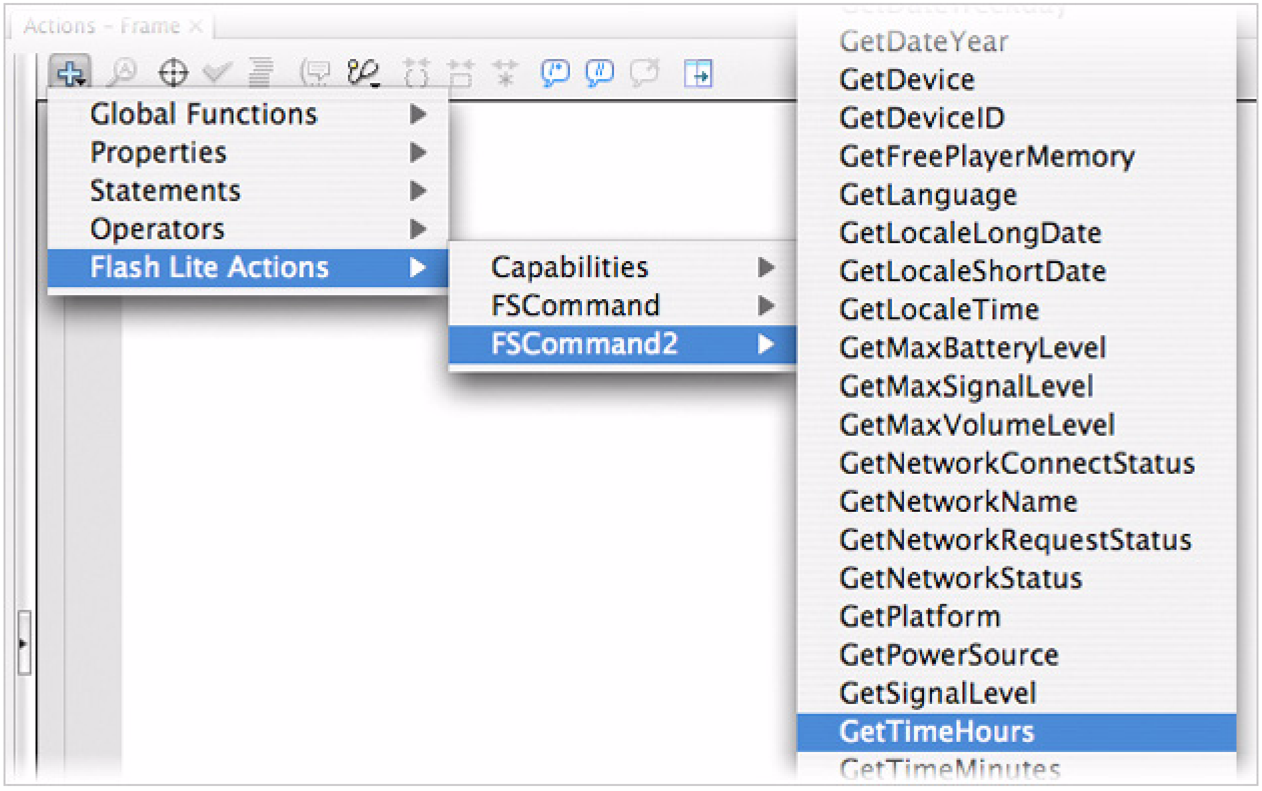
Tutorial

Creating a simple Flash Lite clock

For our first step in this tutorial we create a simple analogue clock face. Using ActionScript, our Flash Lite movie checks the current time on the phone and sets the clock hands in the correct positions.

On opening the `clock_start.fla` in Flash, you see the basic components on the stage. Some graphics are reused from our previous project, with a simple clock interface overlying the wave animation. By clicking on the clock, you see that this consists of a MovieClip named `mc_clock`. By opening this MovieClip you see that there are three layers involved, one for the background, one for the clock hands and finally one for our ActionScript.

Accessing FS2Commands in Flash



To make the clock work we need to obtain values for Hours, Minutes and Seconds from the phone. We do this by using a new set of actions created for Flash Lite called FS2Commands. These commands were specifically introduced to allow us to bring in key pieces of information from the phone, or show the status of a capability. There are a number of commands in this family, all explained in the Flash Lite 1.1 CDK, but the ones we are specifically looking at here are `GetTimeHours`, `GetTimeMinutes` and `GetTimeSeconds`. To use these commands we need to set up variables in our movie where the information obtained resides.

On the actions layer, click in frame 1 and add the following code:

```
hours = fscommand2("GetTimeHours");
```

From now on, our variable `hours` represents the hour value brought in from the phone. We now repeat this process for minutes and seconds:

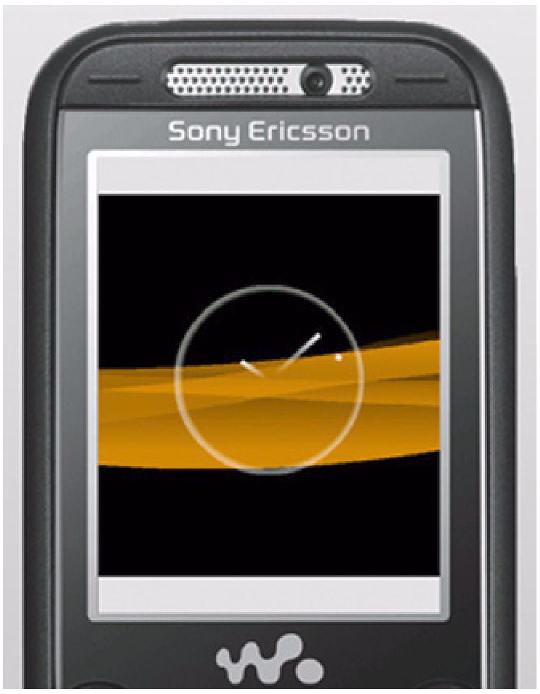
```
minutes = fscommand2("GetTimeMinutes");
seconds = fscommand2("GetTimeSeconds");
```

So we now have our three variables with their associated values. The next step is to use them to move the clock hands. We use the following code to achieve this:

```
if (minutes == 0) {
     setProperty("mc_hour_hand", _rotation, 30*hours);
} else {
     setProperty("mc_hour_hand", _rotation, (0.5*minutes+30*hours));
}
setProperty("mc_hour_hand", _xscale, 100);
setProperty("mc_hour_hand", _yscale, 100);
//
setProperty("mc_min_hand", _rotation, 6*minutes);
setProperty("mc_min_hand", _xscale, 100);
setProperty("mc_min_hand", _yscale, 100);
//
setProperty("mc_sec_hand", _rotation, 6*seconds);
setProperty("mc_sec_hand", _xscale, 100);
setProperty("mc_sec_hand", _yscale, 100);
//
```

Note that there are three code blocks here, one relating to each clock hand. Within each block we have a statement to control the rotation of the hand with each time increment being converted into the required number of degrees of rotation. We also have two further commands concerned with the scaling of the hand MovieClips. Within Flash we often find that the rotation of an object can cause distortion of the original dimensions, in this case the hands can become compressed as time elapses. The simplest way to combat this effect is to enforce an x and y scale of 100% each time this code is run.

Testing the movie

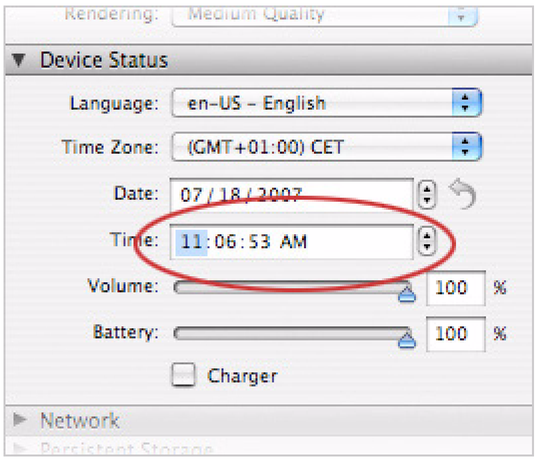


In Device Central, we need to select our target handset, in this case the Sony Ericsson W850i, and ensure the content type is set to Wallpaper. You should see that the clock hands take up positions representing the time shown in the Device Status box. At the moment though, the hands are static, so we need to make them animate. To do this we simply add a frame into our clock MovieClip, setting up a 2-frame loop that constantly requests new data from the device. Test again and you should now see that the clock keeps time.

Taking this further…

So far we have looked at creating a specific interface (a clock) designed to display a specific piece of information (the time) in a very standard way. Using Flash we can go far beyond this simple representation of data and use our variables to control more complex animations. For the next step we are going to create an animation using our time variables to reflect the time of day via environmental change. While retaining our analogue clockface from our first animation, we now add and control the sun/moon as they move throughout the day and the colour of the sky.

On opening the `clock_advanced.fla` you see our clock is on the “clock” layer, our sun/moon graphic is on the “sun/moon” layer and finally on the “bg” layer we have a gradient to illustrate the changing colours of the sky. On testing the movie in Device Central, we see that the clock is working as planned, but the background does not change with the time (Device Central allows us to change the device time in the Device Status panel, so we can rapidly test any time of the day just by increasing/decreasing the hour property).


Rendering: Medium Quality
Device Status
Language: en-US – English
Time Zone: (GMT+01:00) CET
Date: 07 / 18 / 2007
Time: 11 : 06 : 53 AM
Volume: 100 %
Battery: 100 %
Charger
Network
Persistent Storage

We now need to introduce some simple code to the sky and sun MovieClips to allow them to respond to our time variables. Open the `mc_sky MovieClip` and in the second frame of the “actions” layer place the following code:

```
hours = ../clock/:hours;
```

As we have already brought our time variables into the movie within `mc_clock`, we can target these existing variables. In this case we are targeting the `hours` variable to incrementally move our sky graphic every hour during the day. We now need to tell our sky to respond to this variable.

Enter the following script after the previous command:

```
tellTarget ("bg") {
      gotoAndStop(../:hours);
}
```

This script tells our sky graphic (labelled “bg”) to go to and stop on the frame number corresponding to the `hours` variable. By looking within `mc_bg` you see that we have a key-framed animation with the graphic in a different position for each hour. By testing the movie again in Device Central you should now see the background change with the hour value.

Our final step is to get our sun/moon graphic moving across the sky using the same principle we have just applied to the sky. Open `mc_sun` and you can see we have an identical set up to the one we have just seen in `mc_sky`. Again, we need to add some ActionScript to the second frame of the "actions" layer:

```
hours = ../clock/:hours;
tellTarget ("sunMove") {
     gotoAndStop(../:hours);
}
```

Again, this script incrementally moves an animation based on our existing `hours` variable. Test the movie again and you should see the sun/moon now responds to the hour value and corresponds to the sky colour.

Next steps

In this tutorial we have used three properties, in this case the time, to control various graphic elements. Using similar principles it is possible to bring in a variety of variable data using Flash Lite FS2Commands. By looking through the Flash Lite 1.1 CDK we can see that other available data include date, battery level and signal strength.

Conclusion

Currently the animated wallpaper market is dominated by animated Gifs, which are limited in what they can provide. Flash Lite offers us the ability to create really compelling dynamic content in this area. Personalization of User Interface features promises to be a huge growth area, and as we have seen constantly changing content is easy to create, let your creative side run free and see what you can come up with.