Archived:GridMenu Component for Flash Lite

Introduction

23 Aug 2009

This article explains the use of the Flash Lite gridMenu component and API. The gridMenu component is scalable and skinable user interface Flash Lite component. Through inspectable parameters and/or component API it is easy to customize the component to developer's needs. Change icon graphics, button graphics, labels style and selector design in the library. The component is touch screen and key press compatible and has an API function to adjust the component for screen rotations.

Requirements

- Adobe Flash Professional CS3 or CS4
- Flash Lite 2.0 Player and above

Download

Download gridMenu component installation, source files and examples here:
gridMenu_v1.0.zip CS4
gridMenu_v1.0CS3.zip CS3

Installation

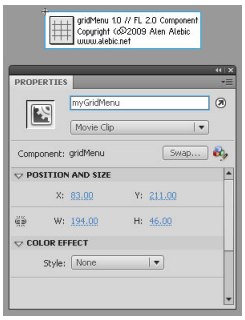
Installation of the gridMenu is easy. Execute the components MXP file and follow the simple instructions in Adobe Extension Manager to complete the installation process. Restart Adobe Flash CS4 after installation.

Note: Component FLA and AS files should appear in the directories *[INSTALL PATH]\Flash CS4\en\Configuration\Components\alebic and [INSTALL PATH]\Flash CS4\en\Configuration\Classes\com\alebic*. (where *[INSTALL PATH]* in WinXP is ie. *C:\Documents and Settings\User\Local Settings\Application Data\Adobe\Flash CS4\en\Configuration\Classes\com\alebic*) In some setup environments, the files may appear under the wrong language directories (for example, fi instead of en). In such cases, the files should be manually moved to the correct directories as specified above.

Preparations

1. Create a Flash Lite Project. Open the Components panel (Ctrl/Apple key + F7) and drag the gridMenu component onto the stage.

2. Assign a unique instance name for the object using the Properties panel (Ctrl/Apple key + F3) (ie. myGridMenu).



3. This component requires the Flash Lite movie to be set to align to the top left corner of the device screen and not to scale. Some ActionScript code is needed to achieve this. The following code must be added to the first frame of the project:

```
fscommand2("DisableKeypadCompatibilityMode");
fscommand2("FullScreen", true);
fscommand2("SetQuality", "high");
Stage.scaleMode = "noScale";
Stage.align = "TL";
_focusrect = false;
```


(C) Copyright Nokia 2014. All rights reserved.

4. For scalability reasons, the gridMenu needs updates from the application. For example, when the device orientation changes, the height and position of the component need to be updated. Changes can be detected by listening for the change of the stage size:

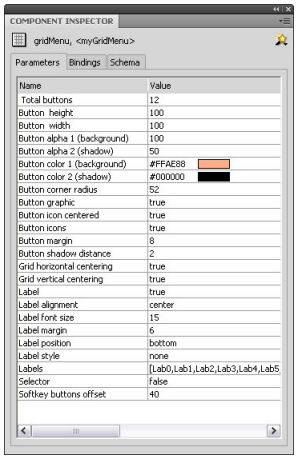
```
// listener for stage resize
var stageSizeListener:Object = new Object()
Stage.addListener(stageSizeListener);

// event handler for stage resize
stageSizeListener.onResize=function(){
myGridMenu.adjustLayout();
}
```

5. In order to catch the interaction events, you need to add listeners for key press events and for on touch dispatched events from the component:

```
// listener for key press
var gridKeyListener:Object = new Object();
Key.addListener(gridKeyListener);

// event handler for key press
gridKeyListener.onKeyDown = function() {
    switch (Key.getCode()) {
        case Key.UP :
            myGridMenu.selectionUp();
            break;
        case Key.DOWN :
            myGridMenu.selectionDown();
            break;
  case Key.LEFT :
            myGridMenu.selectionLeft();
            break;
        case Key.RIGHT :
            myGridMenu.selectionRight();
            break;
        case Key.ENTER :
            myGridMenu.selectionEnter();
            break;
    }
};

// listener for eventDispatch on button release
var event_gridButtonSelected = myGridMenu.onReleaseEvent();
myGridMenu.addEventListener(event_gridButtonSelected, gridEventHandler);

// event handler for button release
function gridEventHandler(evt:Object){
 trace( myGridMenu.getActiveButton());
}
```

Setting the inspectable parameters

All parameters can be managed trough Component Inspector (Window > Component Inspector) ..



.. or can be set dynamically trough a setter function :

```
public function setInspectables(...):Void
```

Inspectable parameters table

Name	Description	Type
Total buttons	Desired number of buttons	(Number)
Button height	Height in pixels	(Number)
Button width	Width in pixels	(Number)
Button alpha 1 (background)	If button graphic is not used, background is visible, this sets _alpha from 0 to 100	(Number)
Button alpha 2 (shadow)	This sets _alpha from 0 to 100 for the button shadow	(Number)
Button color 1 (background)	If button graphic is not used, background is visible, this sets the color with color picker	(Color)
Button color 2 (shadow)	This sets shadow color with color picker	(Color)
Button corner radius	Corner radius, set to 0 for squares or higher for rounder corners	(Number)
Button graphic	Button graphic is used or not (edit "Button graphic" MovieClip in Library)	(Boolean)
Button icon centered	Button icon should be centered or not	(Boolean)
Button icons	Use icons or not	(Boolean)

(C) Copyright Nokia 2014. All rights reserved.

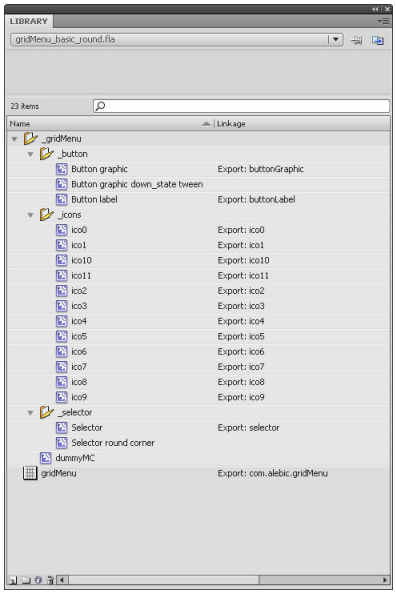
Button margin	Margin between buttons in pixels	(Number)
Button shadow distance	Depth of the shadow in pixels	(Number)
Grid horizontal centering	Center the menu horizontally or not	(Boolean)
Grid vertical centering	Center the menu vertically or not	(Boolean)
Label	Use button labels or not	(Boolean)
Label alignment	Alignment of the label (left, center, right)	(String)
Label font size	Font size of the labels in pixels	(Number)
Label margin	Margin of the label from top or bottom	(Number)
Label position	Label position (top, bottom)	(String)
Label style	Style of the label (none, bold, italic, bold & italic)	(String)
Labels	Array of labels	(Array)
Selector	Use selector or not (edit "Selector" MovieClip in Library)	(Boolean)
Softkey buttons offset	Offset the grid to make room for softkey buttons	(Number)

Pass these variables to *setInspectables* function in this order and in correct variable type. If you wish to skip a parameter, type *undefined* in it's place. Example of *setInspectables* function usage:

```
myGridMenu.setInspectables(3,100,100,undefined,undefined,"#00FF00","#000000",12,true,true,true,5,2,true,true,true,"center",15,5,"bottom","none",["Lab1","Lab2","Lab3"],false,80);
```

Assets library

Change icon graphics, button graphics, labels style and selector design in the library.



By editing icon graphics in the library you can get any design you like. If you need more buttons with icons, add MovieClips to the library and export with the name icoX where X is the next available index.

Transparent mode (bare model)
To fully customize the look of your gridMenu, set these parameters accordingly:
- set background and shadow alpha parameters to 0
- set Button graphic to false
- set Selector to true (to highlight currently selected button)
- edit icon graphics in the library

gridMenu component API

Public function for adjusting layout after screen rotation

```
public function adjustLayout():Void
```

Public functions for key press events:

```
public function selectionDown():Void
public function selectionUp():Void
public function selectionLeft():Void
public function selectionRight():Void
public function selectionEnter():Void
```

Public function to get the release event name

```
public function onReleaseEvent():String
```

Public function to get the current active button's index number

```
public function getActiveButton():Number
```

Examples

There are four examples zipped with the component files. Code is well commented so it should be easy to understand them.
In *gridMenu_basic* and *gridMenu_basic_round* examples there is an *adjustStageSize()* function that adjusts the grid for smaller screen resolutions (Stage.width < 360).

Note: Used in the examples are free-ware icons found on-line and inside every FLA file there is a comment mentioning the source and copyrights of the art work owner. These examples are only for non-commercial purposes and should not be used in commercial projects.

(C) Copyright Nokia 2014. All rights reserved.

Examples: *gridMenu_basic, gridMenu_basic_round, gridMenu_favorites, gridMenu_nokia*

Thanks..

..to Nokia Developer for well documented guide and examples on how to make Flash Lite components.

Conclusion

I hope I managed to create something useful and helpful. I tried to make it as customizable as I could while keeping it simple. Of course, there is space for improvement and upgrades...leave comments with suggestions for gridMenu v2.0

(C) Copyright Nokia 2014. All rights reserved.