Introduction to

ActionScript 3.0 programming

Introduction

Thomas Lövgren
Flash developer, designer & programmer
thomas.lovgren@humlab.umu.se

Umeå Institute of Design, 2011-09-20
Introduction to Flash ActionScript 3.0 programming

Lecture outline

In this lecture we’ll discuss the following topics:

- Programming introduction:
  - Introduction
  - Workspace
  - Coding tips
  - Optimizing
  - Publishing
  - Links & Resources
Adobe Flash/ActionScript ’Today’: Techniques, possibilities & usage overview

- Sound Visualization
- "Custom packages" (worldwide)
- Concept Design (Basic interactivity)
- AIR (Mobile, desktop & web applications)
- Prototype Development
- Virtual worlds
- AS3 Kinect
- Interaction interfaces
- Multiplayer games
- Mobile Systems
- iOS Games & apps
- Projection
- Box2D Physics
- Adobe Catalyst (Flex integration)

- Web applications (Multimedia presentations)
- Interactive-videos
- Game Development
- Motion Tracking
- Flex (Flash Builder)
- Android games & apps
- Wii - Flash (Nintendo)
- Google apps
- Digital Art (Exhibitions)
- Augmented Reality
- Sensors (Phidgets)
- Game frameworks
- Flash 3D (Real-time rendering, FP11)
- Video apps (streaming & prog)
- TUIO

- Social Network: games & apps
- Complex Animations (AS3)
- Video/console games (concept, menus, online versions etc)
- Navigation-systems
- Physics and inverse kinematics
- GestureWorks
- Storytelling
- Animated banners, logos & graphs

- Multi-touch applications
- Database applications (with: XML, php, asp, .NET)
- SION sound apps (Google) TV
- Socket Server apps
- Timeline animations (Cartoon)

- Machinarium: The best selling iPad app on the App Store was created with Adobe Flash (Aug/Sept 2011)

Introduction
Why learn Adobe Flash & programming? As an Interaction Designer

- Adobe Flash with ActionScript 3 is the leading tool/technique for creating multimedia, applications, presentations, games on Internet – a powerful platform for developing systems, prototypes, desktop and mobile-applications

Examples of general usage for a IxD designer:

- Presentations / online portfolio
- Video applications
- Concept design (with basic interactivity/navigation)
- Interface-, Interaction- & User experience design
- Prototype development (usability tests), ex. Google-, webb-, iOS- & Android apps
- Working in project-teams (communicate with/understand programmers)
- Adobe CS: Photoshop, Flash, InDesign, Illustrator, Catalyst and more...(all integrated in the same suite)
- Physical computing: Sensors (Phidgets), Webcam, Wii-remote, Multi-touch, Motion Tracking etc.
- Future: Adobe constantly developing the tools & techniques!
- and more....

ActionScript 3 skills can be useful for other programming-languages!
Flash/ActionScript–based tools & technologies

- **Adobe Flash CS software**: The industry-leading authoring environment for producing expressive interactive content across desktops, smartphones, tablets, and televisions (ActionScript 3)

- **Adobe AIR**: A powerful and flexible technique for building web, mobile, desktop (standalone) & TV-applications (ActionScript 3)

- **Adobe Flex**: Open source framework that allows you to build applications for mobile-devices, browsers and desktops (ActionScript 3)

- **Adobe Flash Builder**: Development tool for building expressive mobile, web, and desktop applications using the Flex framework (ActionScript 3)

- **FlashDevelop**: is a free and open source (MIT license) source code editor (ActionScript 3)

- **Adobe Flash Catalyst**: Interaction design software, integrate content with Flex and write code in Flash Builder (ActionScript 3)

✓ Note! Some of these techniques & tools also support HTML, HTML5, JavaScript, CSS besides AS3
ActionScript versions

- **ActionScript 2.0 (Flash 7 MX 2004)**
  - Object oriented, classes, objects, inheritance, components etc
  - Based on ECMAScript

- **ActionScript 3.0 (2006)**
  - Faster, cleaner, more powerful, easier to debug, more feature rich, strict and secure
  - Object-oriented structure: Built by classes, libraries, objects, functions, and properties
  - New AVM2 (ActionScript Virtual Machine)
  - The syntax, structure & ’robustness’ is quite similar to Java
  - Expands the Flash-based technology...
New features in ActionScript 3.0

- int and uint data-types
- New display list model
- New event model
- E4X (XML)
- Runtime exceptions
- Runtime variable typing
- Sealed classes
- Method closures
- Regular expressions
- Namespaces
- Improved Error handling
Flash Workspace

1. Main Menu
2. Tool panel
3. ActionScript Editor/Panel
4. Timeline (main)
5. Properties & Publishing settings
6. Main toolbar
7. Align/Info/Transform/History
8. Color/swatches
9. Components
10. Library
11. Main stage
12. Document Properties
13. Frames/tween
14. Object on stage
15. Code Snippets
16. Errors & Output
17. Help (Tip: use this!)

Set up/arrange the workspace for your own needs
So you feel comfortable working!
Introduction

ActionScript Panel (Editor)

1. Packages/Script Library
2. Script/Symbol(s)/quick navigation
3. Topmenu
4. Code/Composing section
5. Code Snippets
6. Right menu
7. Find and Replace
8. Help menu
AS3 Language package/structure

- The concept of Classes and Objects is an important part in AS3, basically the whole language package is based on various classes with special tasks
  - Flash built-in classes (language package)
  - User defined classes

- What is a Class? What is an object? ....
Classes & Objects

- A class is a self-contained description for a set of services and data

Example

- Think of a blueprint for a House
- If the blueprint is the *Class*, then the House is the *Object*
- From the *House Class*, we can create as many *House Objects* we want (with unique properties)
- Properties are parameters that describes the Object
- Every new House Object is a *new Instance* of the House Class
Comments

- Line comments and Block comments
- Make the code more understandable
- *Tip*: use comments!
Trace - command

- A testing and debugging command for quick feedback
- Places any relevant text into the Flash *Output panel*
- Only available at author-time, and has no use at runtime
- For example, the trace command could be useful for testing: a loop, function, or a loading process
Suffix and Code-hinting (1/2)

- By using suffix we get a codehint for that specific object
- A suffix is written with an underscore character [ _ ]
- For example, for a MovieClip it looks like: myMovieclip_mc
Suffix and Code-hinting (2/2)

- A list of some useful suffixes in AS3:

<table>
<thead>
<tr>
<th>Object type</th>
<th>Variable suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>_array</td>
</tr>
<tr>
<td>Button</td>
<td>_btn</td>
</tr>
<tr>
<td>Camera</td>
<td>_cam</td>
</tr>
<tr>
<td>Color</td>
<td>_color</td>
</tr>
<tr>
<td>ContextMenu</td>
<td>_cm</td>
</tr>
<tr>
<td>ContextMenuItem</td>
<td>_cmi</td>
</tr>
<tr>
<td>Date</td>
<td>_date</td>
</tr>
<tr>
<td>Error</td>
<td>_err</td>
</tr>
<tr>
<td>LoadVars</td>
<td>_lv</td>
</tr>
<tr>
<td>LocalConnection</td>
<td>_lc</td>
</tr>
<tr>
<td>Microphone</td>
<td>_mic</td>
</tr>
<tr>
<td>MovieClip</td>
<td>_mc</td>
</tr>
<tr>
<td>MovieClipLoader</td>
<td>_mcl</td>
</tr>
<tr>
<td>PrintJob</td>
<td>_pj</td>
</tr>
<tr>
<td>NetConnection</td>
<td>_nc</td>
</tr>
<tr>
<td>NetStream</td>
<td>_ns</td>
</tr>
<tr>
<td>SharedObject</td>
<td>_so</td>
</tr>
<tr>
<td>Sound</td>
<td>_sound</td>
</tr>
<tr>
<td>String</td>
<td>_str</td>
</tr>
<tr>
<td>TextField</td>
<td>_txt</td>
</tr>
<tr>
<td>TextFormat</td>
<td>_fmt</td>
</tr>
<tr>
<td>Video</td>
<td>_video</td>
</tr>
<tr>
<td>XML</td>
<td>_xml</td>
</tr>
<tr>
<td>XMLNode</td>
<td>_xmlnode</td>
</tr>
<tr>
<td>XMLSocket</td>
<td>_xmlsocket</td>
</tr>
</tbody>
</table>
Introduction

Code Snippets panel

- The **Code Snippet panel** contains a couple of pre-define code-blocks; This can be specially helpful for commonly used functions/parts
- It’s possible to write/define your own Code snippets
- Code snippets can be imported and exported
- To use a Code Snippet, for example the 'Mouse Click Event':
  - Select the 'Event Object' on stage
  - Press the "add" button in the Snippet-panel (top right)
  - All code + comments for the 'Mouse Click Event' automatically *embeds* in a separate 'Action frame layer'
Pseudo-code (algorithms)

Pseudo-code (algorithms) are very useful when you are dealing with all kinds of programming (breaking down, structure & solving problems)

- Pseudo-code is a structured list for describing tasks by algorithms
- It allows the designer to focus on the logic of the algorithm, without being distracted by details of language syntax
- Organize your thoughts, and break the main tasks into smaller ones
- Useful in all phases (Planning, design, construction, test & documentation)
- Writing pseudo-code WILL save you time in the end!

```plaintext
if credit card number is valid
    execute transaction
else
    show a generic failure message
end if
```

```plaintext
I get up
Get in the shower
Find something to wear
Eat
Look at tv
Eat
Brush teeth
Go to skool
```
Syntax structure

- It’s highly recommended to use a good programming/syntax structure (that feels comfortable)
- The program reads/executes the code from top to bottom

Tip! Use the same “your” syntax structure in every project
Flash Player

- Adobe Flash Player, is a free software application (browser plug-in) which allows the playing of standalone Adobe Flash (SWF) content
- The Flash Player API is the set of classes, objects, functions, and properties that come built into Flash Player
- There are several key-benefits in terms of: Reach, consistency, and expressiveness
- Positive: Over 99% of all internet-enabled computers have Flash Player today
- Negative: It’s a plug-in that has to be installed

- Supported Web-browsers: Internet Explorer, Mozilla Firefox, Opera, Safari etc - on selected platforms

- Available for Windows, Linux, Solaris, Mac OS X and various Mobile operating systems (Not iOS today!)

- Latest version: Adobe Flash Player 10 (version 10.3.183.7)

Optimizing & tests (for web)

- By loading *Assets/Media/data* into the application we can reduce the file-size and optimize the system.
- When the application is exported it’s possible to *Simulate Downloading*.
- In the ’View dialogue’ we can set up different *Download speeds* and test our application (simulated on the web).
Publishing (Player version & HTML-parameters)

- In the *Publish Settings-dialogue*, we can select Flash Player version, Audio formats, publishing formats, ActionScript version etc.

- Select Flash Player version on export; depending on 'target group & project needs'

- It’s also possible to select HTML-parameters for the embedded Flash-movie (swf-file)
Links & Resources

- **ActionScript Language Reference**

- **Flash Professional glossary (A–Z)**

- **Flashkit.com - movies, tutorials, forum etc**

- **Actionsrcript.org - movies, tutorials, forum etc**
  [http://www.actionscript.org/](http://www.actionscript.org/)

- **Kirupa.com - movies, tutorials, forum etc**

- **Adobe Flash Security**

- **Flash testing and usability**

- **Accessibility**

- **Adobe Flash Player**

---

**Help & Support**

- **Flash Help file: Local on computer (if installed)**
- **Flash/ActionScript Websites & Forums**
- **Download and study exemples**
- **Blogs**
- **Internet (googling)**