

Psychological Task Design & Development

A Programming Workshop Part II_A – The Flash IDE

Wouter Boendermaker, M.Sc.

Johanna Quist, M.Sc.

Soraya Sanchez Maceiras, B.Sc.

University of Amsterdam

EPP Programming Workshop – February 12-13, 2015

Some notes about Flash

- Flash wasn't made to present stimuli
 - Create the basics ourselves
 - Way more is possible once you get the hang of it 😊
- Flash was originally based on animation software
 - ActionScript was added later
 - Based on EcmaScript standards
- Unlimited number of Objects available
 - Can make your own Objects (not in e.g., Presentation)
 - Object Oriented Programming (OOP)
- Alternative to Adobe Flash Pro:
 - **FlashDevelop** (free environment, less beginner friendly)

Three important file types

.as files

- The ActionScript3 syntax file (like **.sce** files in Presentation)
- Main focus of the workshop

.fla files

- The configuration file (like **.exp** file in Presentation)
- Adjust several general settings
- Create and store assets (pictures, sounds, fonts, etc.) in **Library**
- Possible to attach scripts to **timeline**; less suitable for elaborate designs

.swf files

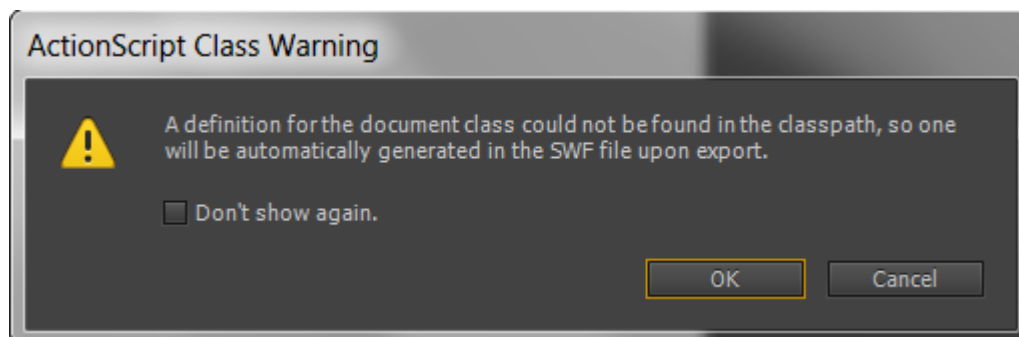
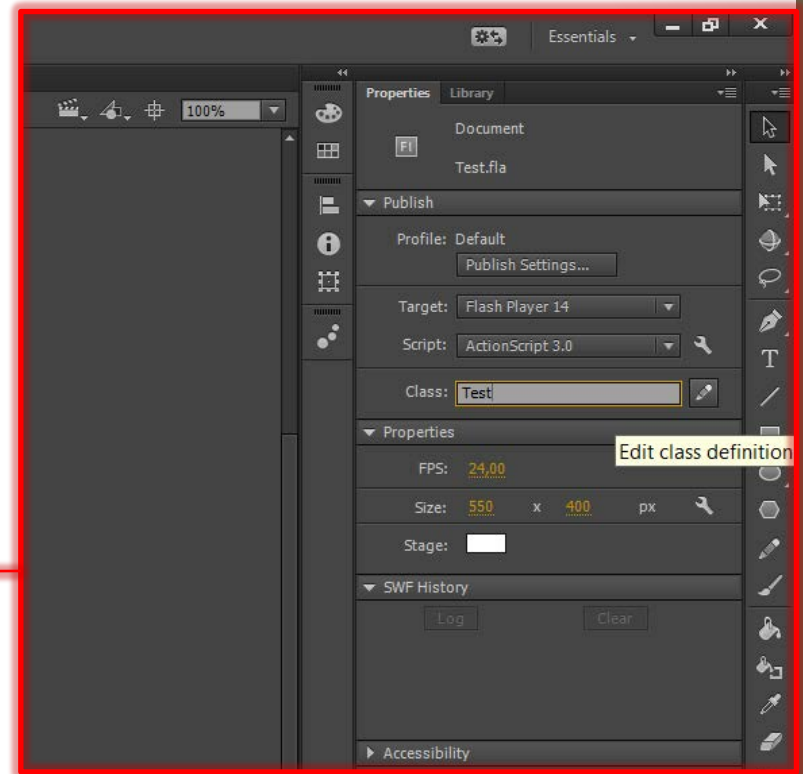
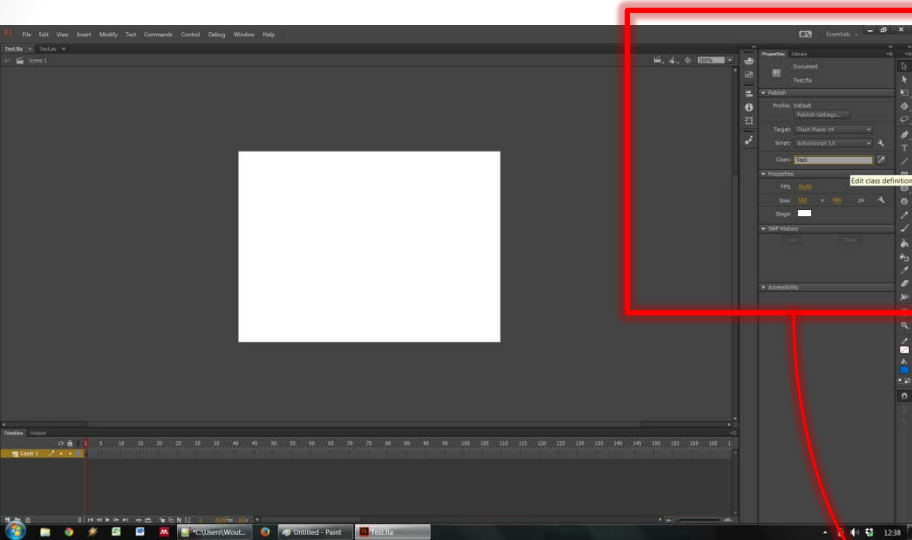
- When we run a piece of ActionScript code, this file type will be created
- This is the actual program that people can run
 - Using **Flash Player** on websites
 - Using **AIR Player** on desktop or mobile (Android, iOS)

Exercise 1 – Getting Started

1. Make a (sub)folder, e.g. `C:\Users\Wouter\desktop\workshop`
2. Under **Create new**, or from the File > New menu, select:
 - ActionScript 3.0 (upper half of the list) to open a new *Untitled-1.fla* file
 - ActionScript File (lower half of the list) to open an empty *Untitled-1.as* file
3. Save these files as **Test.as** and **Test.fla**
 - NB: Case Sensitive! Start filenames (Classes) with Capitals (and no spaces!)
4. Link **.fla** to **.as**
 - In **.fla** file, select the **properties tab** in the **side panel** (see next slide)
 - Put the Class(=file)name into the **Class** field
 - Click **Edit class definition** button

You can download these files from www.wouboe.nl (epw_ex1.zip)

Linking .fla to .as



First piece of code

You can now copy the following code to the empty `Test.as` file:

```
package {
    import flash.display.MovieClip;
    public class Test extends MovieClip {
        public function Test() :void {
            // this is the constructor function
            // put your code here
            // between the inner {curly brackets}
        }
    }
}
```

Note:

- Run with [CTRL] + [ENTER] (PC) / [CMD] + [RETURN] (MAC)
- This should create **Test.swf**
- Notice the structured area's (**levels**) delimited by the {curly brackets}.
- Use **TABs** whenever you use curly brackets.
- NB: only change stuff inside the **constructor function {...}** (for now)!

Basic programming principles

- A text-based script is **interpreted** by the programming environment and then **compiled** into a program that can be **run** on a machine.
- Commands are usually executed through a top-down route, which may be interrupted or rerouted throughout your script to make it more dynamic.
- Every programming language has its own syntax, but there are many similarities between languages.

Be prepared, expect trouble!

- When programming (complex) designs, you **will** run in to different kinds of problems ('bugs').
- This is actually normal and part of the puzzle 😊

Problems I - Compile Time Errors

- **Syntax:** You've made a typing error somewhere.
 - Easily correctable: Flash will show you where most of the time. 😊
- **Semantics:** You're using a valid piece of code in an invalid way: The syntax is correct, but what you want makes no sense (to Flash).
 - Check your semantics, or look up examples in the (elaborate) help file online (or google).

Problems II – Runtime Errors

- **Runtime:** Syntax and semantics are ok, and *compiling* the program succeeds. However, while *running* the program, an error occurs.

→ Flash can tell you roughly where and why these errors occurred.

Example: Trying to look up the fifth element of an array (a list of items) that holds only 4 elements.

Problems III – Logic / Design Errors

- **No error message (nasty!):** According to Flash, your code works fine, and no error seems to occur. However your code doesn't do exactly what you think it should do.

Example: Your randomization code works fine, technically, and the program runs, but on closer inspection of the generated data, you find out that it allows too many repetitions.

Or: While you think your subliminal cue is neatly shown for the requested 25 ms, your display only shows it for 16.6 ms (60 Hz)!

→ Flash cannot help you with this, you'll need to work neatly and check your outcomes.

→ Experience helps a bit, but piloting all the more 😊