Experimental Interactivity
J. Adrian Herbez - aherbez@gmail.com

Class 1: Getting started with OOP

Moving away from the timeline

There is a distinct evolution that most Flash users pass through. Generally, one starts by creating animations on the timeline then, as the projects become more complex, they gradually start folding in bits of actionscript here and there, often on the timeline.

That can work great for small projects, but in order to tackle really big and ambitious projects, you'll need a more structured approach, and that means transitioning over to a more Object-Oriented approach, creating separate .as source files for each of your objects.

Anatomy of an Actionscript 3.0 source file

Here's an example of a simple class file, Dot.as

```
package
{
     // import statements
     import flash.display.MovieClip;

     // class definition
     public class Dot extends MovieClip
     {
          // class variables
          public var myName:String;

          // constructor
          public function Dot(n:String)
          {
               myName = n;
               trace('new dot!');
          }

          // simple method
          public function sayHi():void
          {
               trace('Hello ' + myName);
          }
     }

}
```

Even though the above is simple, it still demonstrates all the components of a AS 3.0 class file:

The package statement

packages make it easy to organize your code- more on that later- for now, just use the default package

Import statements
you'll generally need to access functionality defined in other classes, either those built-in to Flash (such as the MovieClip class), or classes you've written yourself. In order to do that, you'll need to use the import statement to make a given class accessible to the one that you're writing.

The class definition
the class definition itself. This will generally take the form of:

public class *className* extends *baseClass*

...where *className* is the name of your class and *baseClass* is the class that it builds on (if you're not sure what to use, MovieClip is a pretty safe bet)

Declaration of class variables
Almost every class will need to keep track of some information for each copy of itself, such as a name or a color. In order to add information to the class, you'll need to list it at the beginning of the class. All such statements are formatted like the following:

public/private variableName:variableType

public variables are for information that should be accessible from outside the class, and private variables are for information that will only be used within the class. It is generally considered safer to use private variables, as it prevents the data from being modified outside of the class.

Class constructor
This function handles creating a new instance of your class. You will often need to perform some steps to initialize your classes, and this is the place to do it. Class constructors have the following format:

public function *className*()

Class methods
Finally, you'll generally want to add functionality to your class by giving it methods. If the class variables of a class are the nouns that it has access to, then the methods are the verbs. This is how one gets classes to do something useful, beyond just storing data. Class methods should be defined as follows:

public/private function *functionName*(*arguments*):*returnType*

The public keyword should only be used if another class will need to invoke the function. For many functions, that isn't actually necessary- only the class itself will need to invoke it. For such functions, use the private keyword instead.