

Initiation à la programmation en AS3

Licence Professionnelle Jeux Vidéo

Paris 13



Simon Chauvin

chauvin.simon@gmail.com

2012 - 2013



- ActionScript, Flash, FlashDevelop et Flixel :
 - *Historique, objectifs*
- Programmation AS3 :
 - *Spécificités, principes, syntaxe, exemples*

A decorative graphic consisting of a black hexagonal pattern in the top-left corner and a light blue gradient background for the rest of the slide.

ActionScript, Flash, FlashDevelop et Flixel

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Qu'est ce que l'ActionScript :
 - **Création** en 1998 par Macromedia Inc (racheté ensuite par Adobe)
 - Initialement utilisé pour **contrôler** des **animations** vectorielles 2D sous Macromedia Flash
 - Peu de possibilités : actions associées à des frames
 - Dérivé d'**ECMAScript** (langage de script standardisé) de même que JavaScript

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- ActionScript 2 :
 - 2003
 - **Classes** et héritage
 - Programmation **objet partielle**
 - Les fonctions « onLoad », « onEnterFrame » et autres sont encore présentes

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- ActionScript 3 :
 - 2006
 - **Révision complète** de l'API
 - Programmation **orientée objet**
 - Nouvelle manière de créer des objets avec le mot clé « new »
 - Les fonctions « onLoad », « onEnterFrame » et autres sont maintenant des « Events »
 - ActionScript intègre désormais un système d'« Events »

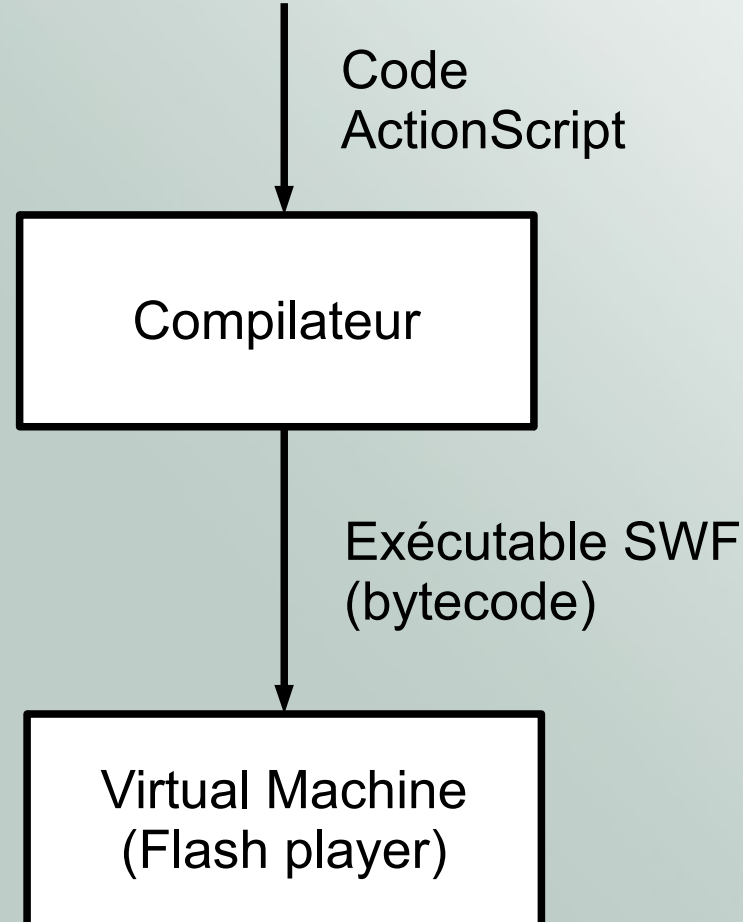
ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash, FlashDevelop et Flixel

- Fonctionnement :

- Le code ActionScript est **compilé** en bytecode
- **Interprétable** par la machine virtuelle du Flash Player



ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- A la compilation sont vérifiés :
 - La syntaxe
 - La correspondance des types avec leurs valeurs
- Documentation :
 - [Erreurs de compilations](#)
 - [Documentation AS3](#)

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Syntaxe :

- Déclaration d'une **variable** :

```
var name:type = valeur;
```

- Déclaration d'une **fonction** :

```
function name (argument:type1, ...):type2  
{  
    var value:type2 ;  
    ...  
    return value ;  
}
```

- Déclaration d'une **classe** :

```
class Name {  
    /* Attributs */  
    ...  
    /* Méthodes */  
}
```

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- FlashDevelop :
 - **Alternative** à Adobe Flash, sortie en 2005
 - Gratuit, open source **IDE** pour MS Windows
 - IDE : environnement de développement intégré
 - **Facilite** la programmation en AS2 et AS3
 - Complétion, génération de code, highlighting, debug, etc.

ActionScript, Flash, FlashDevelop et Flixel

- Présentation

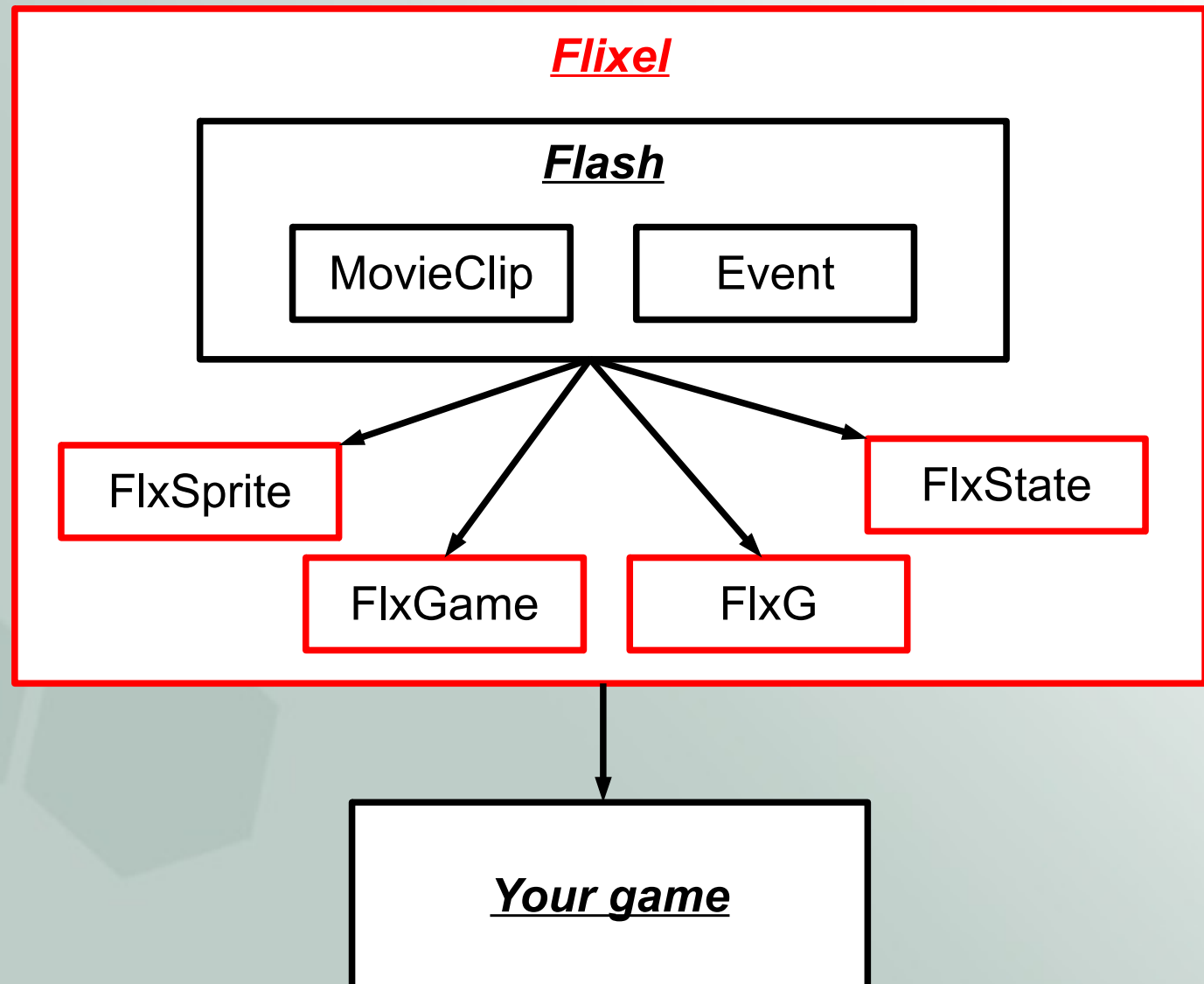
- ActionScript, Flash, FlashDevelop et Flixel

- Flixel :

- **Framework de développement de jeux vidéo** basé sur l'API Flash
- Première version rendu public en 2009
- Développé par **Adam Saltsman** (développeur de **Canabalt**, entre autres)
- Fournit un ensemble de **classes** et **fonctions** facilitant la programmation de jeux :
 - Sprites, states, groups
 - Collisions, scrolling, physique
- Utilise des **Bitmaps**
- Très performant
- Utilisé par beaucoup de développeurs Flash

ActionScript, Flash, FlashDevelop et Flixel

- Flixel vs Flash :



Programmation AS3

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

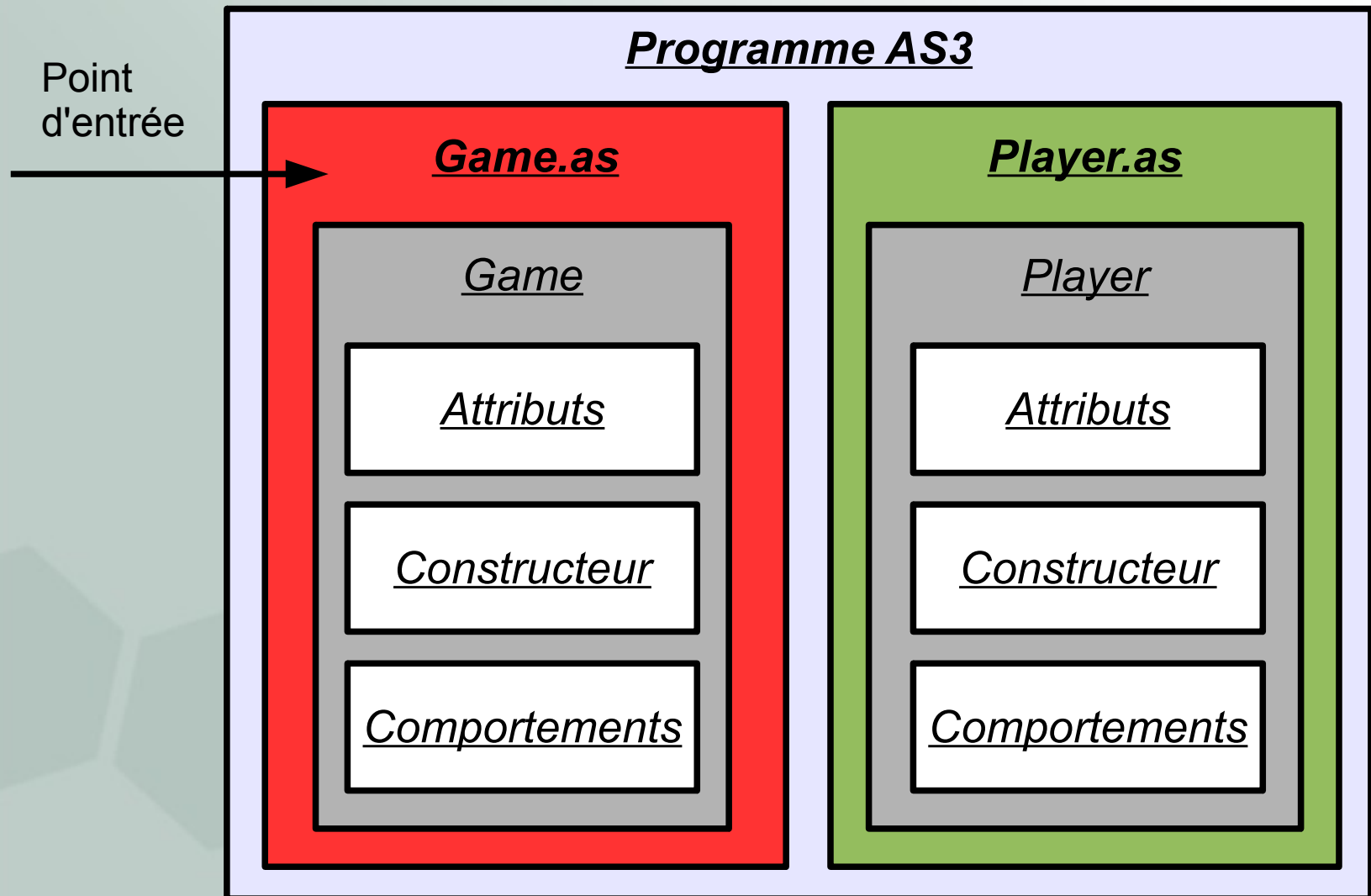
- Un programme AS3 :
 - Un programme AS3 est **composé** de **fichiers .as**
 - Un **seul** fichier sert de **point d'entrée** à l'application
 - Chaque **fichier** contient une **classe** et une seule
 - Chaque **classe** contient des **attributs** et des **comportements** et un seul et même **constructeur**
 - Une **classe** est une **structure**, seul l'objet (l'instance) est en mémoire
 - Les **attributs** sont les **données** de l'objet
 - Les **comportements** sont des **fonctions** applicables à l'objet
 - Une **fonction** comporte des **instructions**
 - Un programme AS3 est **compilé** en un fichier **.swf** **lisible** par le Flash Player

- Un programme AS3 :

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

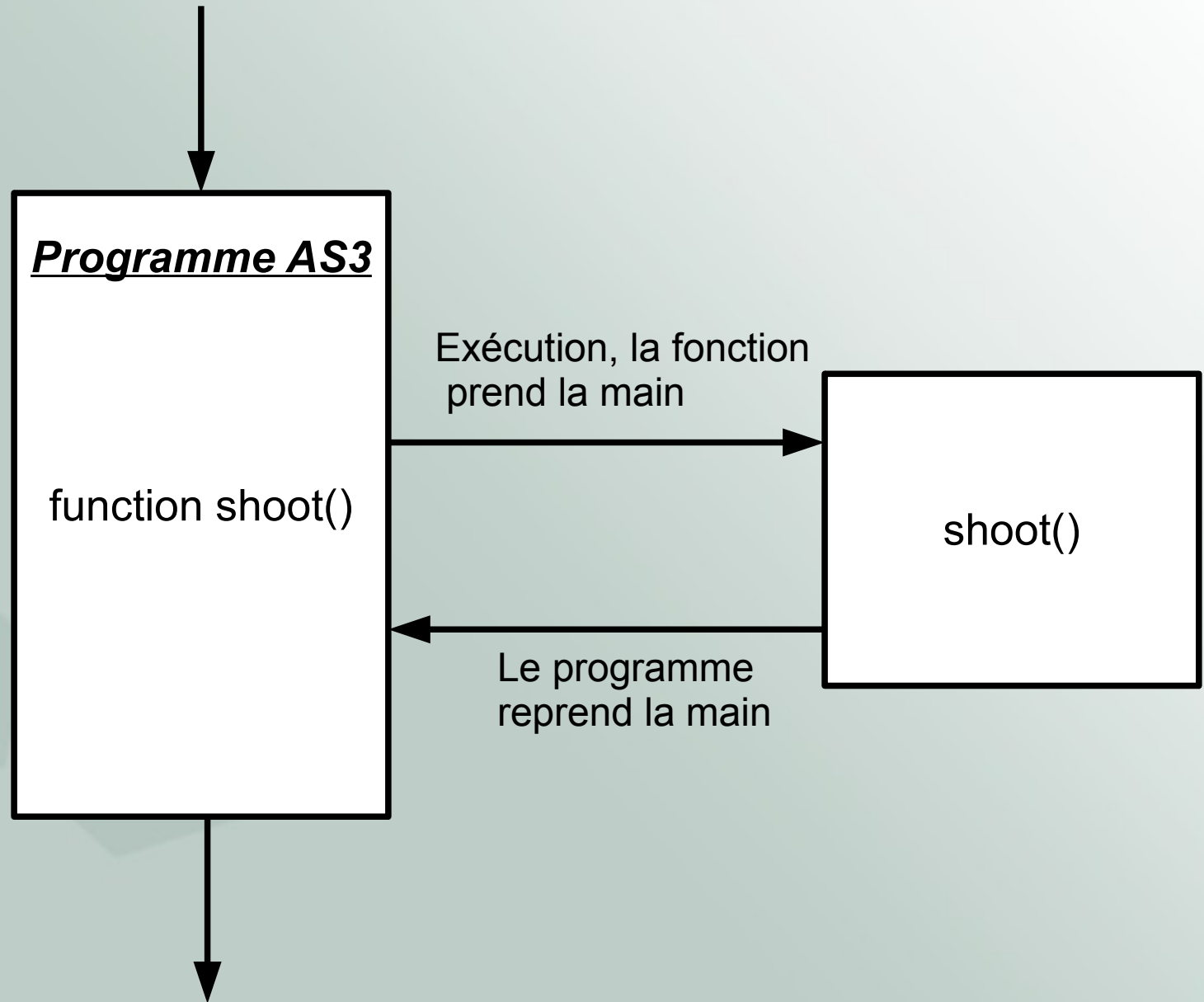


- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Un programme AS3 :



- Présentation
- ActionScript, Flash, FlashDevelop et Flixel
- Programmation AS3

- Un programme AS3 :

Données + Instructions

- Les instructions :

- Sont des suites de mots et caractères clés :

```
var hitPoint:int;  
public class PlayState {}  
if () {} else {}  
function addEnemy() {}
```

- Elles **remplissent des objectifs** différents :
 - Déclarations, tests, boucles, opérations, etc.

- Déclarations de classes :

```
public class Block {  
    //Attributs  
    //Le constructeur  
    //Comportements  
}
```

- Déclarations de sous classes :

```
public class DestructibleBlock extends Block  
{  
    //Attributs  
    //Le constructeur  
    //Comportements  
}
```

- Déclarations de fonctions :

- *Fonction sans arguments :*

```
public function shoot () {  
    //Instructions  
}
```

- *Fonction avec arguments :*

```
public function shoot (weapon:Weapon) {  
    //Instructions  
}
```

- *Fonction avec valeur de retour :*

```
public function shoot (weapon:Weapon):int  
{  
    var damage:int = 10;  
    //Instructions  
    return damage;  
}
```

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Déclarations de fonctions :

- *Fonction redéfinie :*

- La fonction redéfinie doit **exister** dans la classe **mère**

- *Exemple :*

```
override public function shoot () {  
    //Instructions  
}
```

- Déclarations de variables :

```
var score:int;  
var name:String;
```

- Déclarations de constantes :

```
const MAX_HEALTH:int;
```

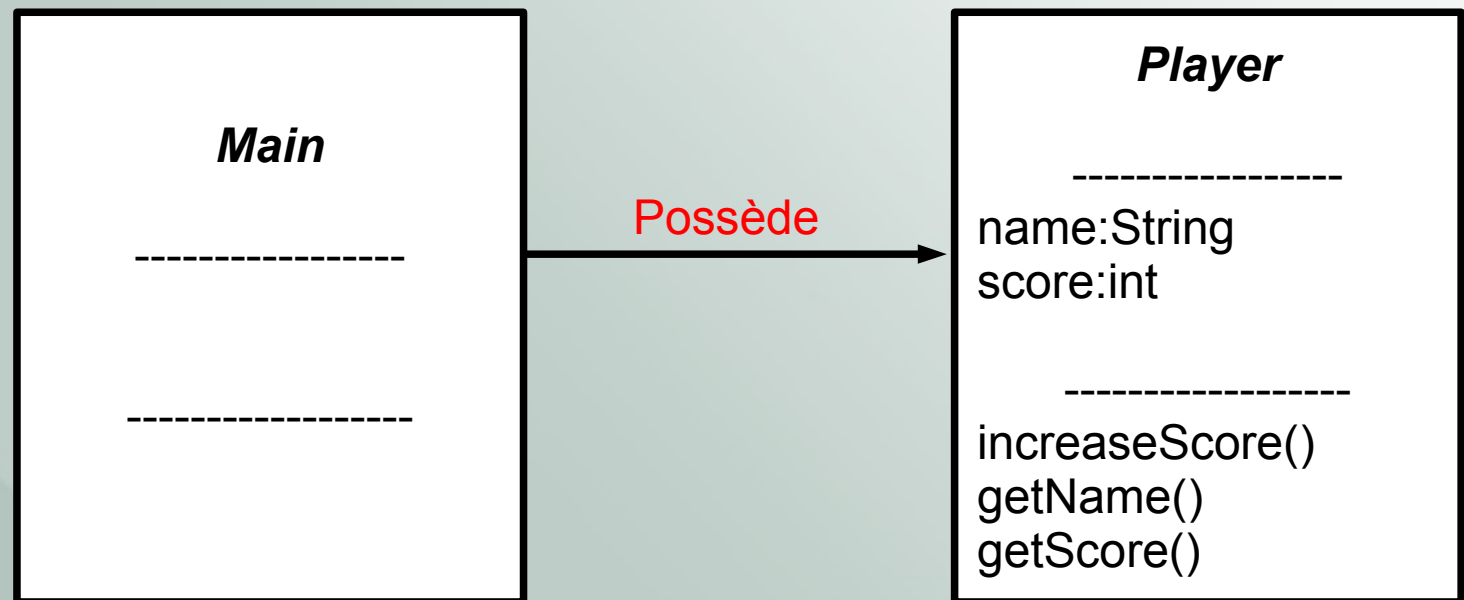
- Déclarations d'attributs :

```
private var score:int;  
public const MAX_HEALTH:int;
```

- Déclaration de constructeurs :

```
public function Player(name:String)  
{  
  
}
```

- Association de classes :



- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- La classe principale :
 - Classe contenant les classes associées
 - Instancie et rend accessible les objets

```
public class Main {  
    //Attributs  
    private var player:Player;  
  
    public function Main () {  
        player = new Player();  
    }  
}
```


- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- La classe Player :

```
public class Player {  
    public const MAX_HEALTH = 100;  
    private var name:String;  
    private var score:int;  
  
    public function Player(name:String) {  
        this.name = name  
    }  
    public function increaseScore(value:int) {  
        score = score + value;  
    }  
    public function getName():String {  
        return name;  
    }  
    public function getScore():int {  
        return score;  
    }  
}
```

- Instancier une classe :

```
var player:Player = new Player("Billy");
```

- Affecte l'instance de "Player" nommé Billy à la variable nommée player

- Accéder aux comportements d'une classe :

```
player.increaseScore();
```

- Appel la méthode increaseScore() sur l'objet nommé player
- Augmente le score de l'objet nommé player de 10

- Accéder aux attributs d'une classe :

```
player.MAX_HEALTH; //Retourne 100
```

```
player.getScore(); //Retourne le score  
courant
```

- Accéder à la classe mère :
 - Utilisation du mot clé `super`
- Accéder au constructeur de la classe mère :
`super () ;`
- Accéder à une méthode de la classe mère :
`super.update () ;`
`super.destroy () ;`

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les structures conditionnelles :

- Simples :

```
if (dead == true) {  
    destroy();  
}
```

```
if (x > 150) {  
    show();  
} else {  
    hide();  
}
```

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les structures conditionnelles :

- Multiples :

```
if (dead == true && x < 150) {  
    destroy();  
}
```

```
if (health <= 0 || y > 600) {  
    gameOver();  
}
```

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les données :
 - Sont de différents types :
 - *Simples :*
 - Les entiers : `int`
 - Les réels : `Number`
 - Les booléens : `Boolean`
 - *Complexes :*
 - Les chaînes de caractères : `String`
 - Les objets : `Player`, `Object`,
`FlxState`
 - Les tableaux : `Array`

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les données simples :
 - `int`, `Number` **et** `Boolean`
 - Sont **initialisées** par défauts à certaines valeurs :
 - `int` sont par défauts égales à `0`
 - `Number` sont par défauts égales à `0.0`
 - `Boolean` sont par défauts égales à `false`
 - Ne contiennent aucune autre donnée

- Les booléens :

- Sont un type de données **binaire**, ils sont **vrais** ou **faux**
- Parfait pour stocker des **etats**
- Exemple :

```
var dead:Boolean = false;  
if (dead == true) {  
    destroy();  
}
```


- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les données complexes :
 - Object, Array, String, vos propres objets
 - Sont des objets contenant **plusieurs types** de données différentes
 - Sont **accessibles** par **références** (un lien vers l'objet en mémoire)
 - Les références sont par défaut **initialisées** à `null`
 - Ce qui signifie qu'il n'y a rien au bout du lien (analogie avec un lien html mort)

- Les chaînes de caractères :

- Sont désignées grâce aux guillemets
- Peuvent être initialisée sans le mot clé `new` :

```
var name:String = "Pete";
```

- Ou avec :

```
var name:String = new String("Pete");
```

- Proposent des attributs :

```
name.length //Retourne 4
```

- Et diverses fonctions :

```
name.split("e"); //Sépare la chaîne de  
caractères par le e
```

- Présentation

- ActionScript, Flash,
FlashDevelop et Flixel

- Programmation AS3

- Les tableaux :
 - Un tableau :
 - [0, 8, 1, 9, 10, 15, 7]
 - Longueur de 7, indice allant de 0 à 6
 - Éléments du tableau :
 - À l'indice 4 se trouve 10
 - À l'indice 0 se trouve 0
 - À l'indice 6 se trouve 7

- Les tableaux :

- Création d'un nouveau tableau avec le mot clé `new` :

```
var tableau:Array = new Array();
```

- Ou sans :

```
var tableau:Array = [];
```

- Ajout, suppression d'éléments :

```
//Ajout en fin de tableau
```

```
tableau.push(élément) ;
```

```
//Suppression en début de tableau
```

```
tableau.pop(élément) ;
```

- D'autres méthodes existent :

```
//Suppression de deux éléments à partir  
de l'index 3
```

```
tableau.splice(3, 2) ;
```

```
//Ajout à l'index 5 du tableau
```

```
tableau[5] = "string"
```

- Les objets :

- Sont définis par **vous** et contiennent les données et les fonctions que **vous** souhaitez

- *Exemple :*

```
public class Player {  
    private var name:String;  
    private var score:int;  
  
    public Player() {  
    }  
    public incScore(value:int) {  
        score = score + value;  
    }  
}  
  
var player:Player = new Player();  
//La variable player contient un lien  
vers l'objet en mémoire
```