Partie 7

Autour de la syntaxe as3

Maitriser l'utilisation du mot clé « is »

Lors du chapitre sur le modèle évènementiel, nous avons abordé les phases de captures et de bouillonnement. Nous avons vu que, la diffusion d'évènements était déclenchée par l'ensemble des objets enfants de la liste d'affichage du container écouté.
Nous nous sommes retrouvés devant l'obligation de désactiver la réactivité souris des DisplayObject dont nous ne souhaitions pas voir déclencher notre action.
Nous avons une autre solution c'est de filter les objets. Dans la fonction associée à l'évènement souris, nous allons effecteur un test en utilisant le mot clé is et de traiter les clicks souris en fonction de l'objet cliqué.

```
if(mon_objet is MovieClip)
{
	//Effectuer une action
}
```

Lorsque l'on formule un test qui utilise cette syntaxe. Le processus va nous renvoyer un booléen qui nous indiquera si l'occurrence est bien du type que l'on recherche.
Nous pouvons bien sur rentrer toutes les classes possibles, qu'elles soient intrinsèques à flash ou qu'elles soient des classes personnalisées.
Le test renverra true sur l'ensemble des types dont hérite la classe.

Admettons que j'ai une classe personnalisée se nommant Vignette héritant de MovieClip

```
var ma_vignette :Vignette=new Vignette() ;

if(ma_vignette is MovieClip)
{
	//Effectue l'action car ma classe Vignette hérite de MovieClip.
}

if(ma_vignette is Vignette)
{
	//Effectue l'action il s'agit bien de ma classe Vignette.
}
if(ma_vignette is DisplayObject)
{
	//Effectue l'action car ma classe Vignette hérite de MovieClip qui
	 // hérite de DisplayObject.
}
if(ma_vignette is Object)
{
	//Effectue l'action car ma classe Vignette hérite de MovieClip qui
	herite à la base de la classe Object().
}

if(ma_vignette is TextField)
{
	//N'effectue pas l'action car vignette n'hérite pas de TextField.
}
```

L'utilisation de la syntaxe crochet dans AS3 .

La classe Object est la brique de base en AS3. sur elle reposent l'ensemble des fonctionnalités du langage. Cela veut dire que tous les types complexes et primitifs héritent de la classe Object().

Quand en as2 il s'agissait de créer un objet, nous pouvions simplement implémenter ses propriétés à l'aide de la syntaxe pointée.

```
var mon_objet =new Object() ;
mon_objet.propriete_1= « une chaine » ;
```

En as 3 cette syntaxe provoquera une erreur car la syntaxe pointée ne fonctionne qu'avec des méthodes et des propriétés déjà existantes dans l'objet créé. Or dans un objet rien ne préexiste. Je devrais donc pour assigner une propriété ou une méthode sur un objet générique ou une classe dynamique utiliser la syntaxe suivante :

```
var mon_objet : Object =new Object() ;
mon_objet [«propriete_1»]=  «chaine de caractères»;
mon_objet [«propriete_2»]= {identifiant : 1 , valeur : «perdu» };
```

pour récupérer les données stockées nous disposons alors de deux méthodes possibles :
Soit en utilisant la syntaxe pointée :

```
mon_objet. propriete_1 ; // trace renvoie : [object Object]
```

soit en utilisant de nouveau la syntaxe à crochet :

```
mon_objet [«propriete_1»] ; // trace renvoie : [object Object]
```

Il est possible de parcourir l'architecture de l'objet entièrement avec cette syntaxe :

```
mon_objet [«propriete_2»] [«valeur»]; // trace renvoie : « perdu »
```

La première remarque à formuler c'est que je peux récupérer une propriété de n'importe que objet héritant d'Objet() (soit TOUS les types primitifs et composites) qui utilisent la syntaxe pointée.
Il me suffit d'utiliser cette syntaxe en lieu et place de la syntaxe pointée.

```
var clip :MovieClip = new MovieClip();

clip.x = 200 ; // place le clip à 200 pixel du bord du container
parent.
Trace(clip[«x»])// renvoie 200 ;
```

La deuxième chose remarquable, c'est qu'entre les crochets nous n'utilisons pas la propriété en elle-même mais une chaine de caractère correspondant à sa valeur. C'est l'aspect de chaine qui va nous intéresser, car cette façon d'ecrire est en mesure de nous apporter une souplesse énorme dans notre code. En effet la valeur de la chaine peut être assignée à une variable et nous pouvons utiliser cette variable pour modifier les propriétés sélectionnées

```
var clip :MovieClip = new MovieClip();

//je crée un tableau avec quelques une des propriétés de mon clip sous forme
//de chaines de caractère
var tableau_choix:Array = ["x","y", "rotation" ] ;
// Je crée un chiffre au hasard entre 0 et le nombre d'éléments du tableau
moins un ;
```

```
        var chiffre_hasard :Number = Math.round( Math.random()*(tableau_choix.length-
        1));

        clip.graphics.beginFill(0xCC33FF,.6)
        clip.graphics.drawRect(0,0,200,100);
        this.addChild(clip);
        clip.x=stage.stageWidth/2-clip.width/2;
        clip.y=stage.stageHeight/2-clip.height/2;

        // J'affecte une valeur de 200 à la propriété du clip qui sera sélectionnée
        //dans le tableau
        clip[tableau_choix[chiffre_hasard]] = 200 ;

        trace(chiffre_hasard+" :  propriété : "+tableau_choix[chiffre_hasard]+" :
"+clip[tableau_choix[chiffre_hasard]]);
```

Nous pouvons aussi appliquer cette syntaxe à travers l'arbre d'un XML. Dont nous nous servirons pour stocker les différents états d'un Sprite que nous aurions configuré en bouton :

```
var donnees_xml:XML= <main>
             <config largeur="200" hauteur="30"></config>
             <normal couleur="0x333333" alpha_fond="0.5"></normal>
             <dessus couleur="0xDD00EE" alpha_fond="0.8"></dessus>
                          </main>;


var fond:Sprite = new Sprite();
fond.mouseEnabled = false;
decorer("normal");

var bouton:Sprite = new Sprite();
bouton.buttonMode=true;

bouton.addChild(fond);
this.addChild(bouton);

bouton.addEventListener(MouseEvent.MOUSE_OVER,dessus);
bouton.addEventListener(MouseEvent.MOUSE_OUT,dehors);

function dessus(even:MouseEvent):void
{decorer("normal");}

function dehors(even:MouseEvent):void
{decorer("dessus");}

function decorer(etat:String):void
{
fond.graphics.clear();
fond.graphics.beginFill(donnees_xml[etat].@couleur,donnees_xml[etat].@alpha_fond);
fond.graphics.drawRoundRect( 0 , 0 , donnees_xml.config.@largeur,
donnees_xml.config.@hauteur , 4 , 4);
}
```