

# Séquence 4 : Instructions conditionnelles

## *if else* et *switch case*

### I. Instruction *if else*

#### I.1. Structure de *if else*

L'instruction *if* fonctionne en AS3 comme elle le fait dans de nombreux langages de programmation :

```
if (maValeur == 1) {
    faireQuelqueChose();
}
else {
    faireAutreChose();
}
```

1. Si la condition située après le mot-clé *if* et placée **obligatoirement entre parenthèses** est vraie, alors les instructions placées dans le bloc défini entre les accolades sont exécutées.
2. Si la condition est fausse, alors les instructions définies dans le bloc situé après le mot-clé *else* sont exécutées.

De cette façon, un seul des deux blocs peut être exécuté à la fois, selon que la condition est vérifiée ou non.

Il est aussi possible d'ajouter le mot-clé *else if* pour étendre la structure *if* :

```
if (maValeur == 1) {
    faireQuelqueChose();
} else if (maValeur == 2) {
    faireAutreChose();
} else {
    faireEncoreAutreChose();
}
```

Vous pouvez également inclure des conditions plus complexes avec *&&* et *|*. Ces signes représentent les opérateurs de comparaison AND (ET) et OR (OU).

```
if ((maValeur == 1) && (maChaine == "Cela")) {
    faireQuelqueChose();
}
```

#### I.2. Comment écrire une condition ?

En programmation pour écrire une condition, on utilise les opérateurs de comparaison suivant :

Opérateur	Signification pour des valeurs numériques	Signification pour des valeurs de type caractère
==	égal ( 2 == 2 )	identique ( "Sti2d" == "Sti2d" )
<	inférieur ( 2 < 3 )	plus petit dans l'ordre alphabétique
<=	inférieur ou égal ( 2 <= 3 ou 3 <= 3 )	plus petit ou identique dans l'ordre alphabétique
>	supérieur ( 3 > 2 )	plus grand dans l'ordre alphabétique
>=	supérieur ou égal ( 3 >= 2 ou 3 >= 3 )	plus grand ou identique dans l'ordre alphabétique
!=	différent ( 2 != 3 )	différent ( "ISN" != "isn" )

#### I.3. Application

Vous allez réaliser un application qui permet de déplacer un objet graphique monSprite sur la scène à l'aide des touches de direction.

Pour cela, vous allez créer un projet Sequence 4.

Ensuite compléter le code ci-dessous pour placer l'objet sur la scène.

```
[Embed (source="images/applaudissement.mp3")]
private var applaudissement : Class;
private var son : Sound = new applaudissement;

[Embed (source="images/ballon.png")]
private var dessinBallon : Class;
private var ballon : CreerSprite = new CreerSprite(new dessinBallon);

[Embed (source="images/panier.jpg")]
private var dessinPanier : Class;
private var panier : CreerSprite = new CreerSprite(new dessinPanier);

//declaration d'une variable timer
private var monTimer : Timer = new Timer(500, 0);

//declaration d'une variable entier
private var nombreEntier : int = 3;

private var largeurEcran : int ;
private var hauteurEcran : int ;

//declarer ici les nouvelle variable:

public function Main():void
{
    //creer l'evenement lié au timer
    monTimer.addEventListener(TimerEvent.TIMER, appelleCyclique);
    //lancer le timer
    monTimer.start();

    //point d'entrée :

    // récupérer la largeur et la hauteur de la scène
    largeurEcran = stage.stageWidth ;
    hauteurEcran = stage.stageHeight ;

    ///dessin de l'objet graphique sprite
    addChild(ballon);
    //tester
    //centrer le ballon sur la scène en utilisant les variables largeurEcran et hauteurEcran

    //tester

    // diminuer la taille du ballon
```

```

//Création de l'écouteur d'évènement sur les touches
stage.addEventListener(KeyboardEvent.KEY_DOWN , toucheTaper);
}

//Lorsque l'on appuie sur une touche, cette méthode est appelée
private function toucheTaper(e:KeyboardEvent):void
{
    var numeroTouche : int ;
    //recupère le numéro de la touche actionnée :
    numeroTouche = e.keyCode;

    //affiche dans l'aire Output le numéro de la touche actionnée :
    trace(e.keyCode);

    if (numeroTouche == 37) // touche flèche gauche tapée
        { ballon.x = ballon.x - 5 ; } //déplacement de 5 pixel vers la gauche de
        //l'objet monSprite

    //tester
    //completer ici
}

//Cette fonction est appelée tous les 0,5 secondes.
private function appelleCyclique(e:TimerEvent):void
{
}

```

Enfin, compléter ce code avec les instructions `if` et `else` pour que l'objet `monSprite` se déplace suivant les 4 directions sur la scène.

## II.L'instruction *switch* (choix multiples)

Lorsque le nombre de choix possibles est plus grand que deux, l'utilisation de la structure *if-else* devient très lourde à gérer.

C'est pour cela que l'instruction *switch* (traduire par « selon » ou « suivant »), qui permet de programmer des choix multiples selon une syntaxe plus claire.

### II.1.Construction du *switch*

L'écriture de l'instruction *switch* obéit aux règles de syntaxe suivantes :

```

switch (maVariable) //appuyer Ctrl+B pour qu'il
{
    case valeur1 :
        //Une ou plusieurs instructions
        break;
    case valeur2 :
        //Une ou plusieurs instructions
        break;
    case valeur3 :
        //Une ou plusieurs instructions
        break;
}

```

```

default :
    //Une ou plusieurs instructions
}

```

La variable *maVariable* est évaluée. Suivant celle-ci, le programme recherche la valeur correspondant à la valeur obtenue et définie à partir des instructions *case valeur*.

1. Si le programme trouve une valeur correspondant au contenu de la variable *maVariable*, il exécute la ou les instructions qui suivent *valeur*, jusqu'à rencontrer le mot-clé *break*.
2. S'il n'existe pas de *valeur* correspondant à *valeur* alors le programme exécute les instructions de l'étiquette *default*

## II.2.Application

Vous allez modifier l'application précédente :

- Remplacer les instructions *if else* par l'instruction *switch* et *case* pour que le programme soit plus claire.
- Tester , si on a le même comportement que l'application précédente.
- Si tout fonctionne correctement, ajouter un nouvelle objet graphique *panier* (Sprite) sur la scène.
- L'*objetCible* sera positionné au hasard sur la scène à l'aide des instructions suivantes:
 

```

panier.x = largeurEcran * Math.random() ;
panier.y = hauteurEcran * Math.random() ;

```

L'instruction *Math.random()* renvoie un nombre aléatoire n, où  $0 \leq n < 1$ .

- Tester.
- Maintenant, vous allez créer une instruction qui teste si l'objet *ballon* est en contact avec *panier*. Pour cela, vous utiliserez l'instruction suivante :
 

```

ballon.hitTestObject(panier)

```

L'instruction *hitTestObject(obj:DisplayObject):Boolean* évalue l'objet *ballon* pour savoir s'il recouvre l'objet d'affichage *panier* en rendant un Booléen (*true* ou *false*).
- Lorsque les 2 objets seront en contact le programme devra renvoyer un son.

## III.Mémento

Les structures de test permettent d'exécuter ou non un bloc d'instructions en fonction de la véracité d'une condition.

Vérifier si un objet est sur la scène ou non

```

if ((objet.x >= 0) && (objet.x <= largeurEcran)){
    trace("L'objet se situe sur la scène");
}
else {
    if (objet.x < 0){
        trace ("L'objet se trouve à gauche de la scène");
    }
    else {
        trace ("L'objet se trouve à droite de la scène");
    }
}

```

Ici, deux structures *if-else* sont imbriquées. La première vérifie si la position en x d'un objet est comprise entre 0 et largeur. Dans ce cas, l'objet se situe bien à l'intérieur de la scène (en x). Sinon, deux autres solutions sont possibles : soit la position est inférieure à 0, l'objet se trouve à gauche de la scène, soit elle est supérieure à largeur (le cas inférieur à largeur étant traité par la toute première condition). L'objet se trouve alors à droite de la scène.

Choisir une option parmi d'autres

```
switch (option) {  
    case "MenuFichier" :  
        //Afficher les options du menu Fichier  
    break;  
    case "MenuEdition" :  
        //Afficher les options du menu Édition  
    break;  
    case "MenuAffichage" :  
        //Afficher les options du menu Affichage  
    break;  
    default :  
        //Traiter les autres cas  
}
```

La structure *switch* simplifie l'écriture des tests à choix multiples. Suivant la valeur contenue dans la variable *option*, l'une des 4 étiquettes proposées (MenuFichier, MenuEdition, MenuAffichage ou default) est exécutée.

## IV.Conclusion

---

---

---

---

---