

Vous ne souhaitez pas utiliser le composant dédié à cette technique, mais plutôt réaliser votre propre scroll. Le script ci-dessous vous présente le code utilisé pour gérer le défilement horizontal d'une image en fonction du déplacement d'un curseur.

Dans cette page, vous trouverez 2 cas. Le premier vous présente un script dans lequel vous pouvez choisir une image avant de la scroller, le deuxième est une simplification du premier avec une seule image (sans sélection préalable de l'image à scroller).

Premier cas

Placez simplement sur la scène 4 occurrences que vous nommez bt1, bt2, bt3 et curseur. Ce dernier correspond à l'ascenseur que vous allez déplacer alors que les 3 premiers sont de simples occurrences de CLIP (et non de Bouton).

Explications

Nous déplaçons horizontalement un curseur entre le 100e pixel du bord gauche de la scène, jusqu'au 700e (soit un déplacement de 600 pixels. Nous bloquons ce déplacement sur le 550e pixel du haut de la scène.

Dès que nous bougeons l'occurrence intitulée **curseur**, nous cherchons à calculer la valeur de déplacement de l'image.

Nous nous sommes limités à 3 images dans cet exemple, mais vous pourriez en ajouter tant que vous voulez. Si tel était le cas, je vous invite tout de même à utiliser une boucle for().

Remarque : Le code ci-dessous aurait pu être optimisé, mais il deviendrait plus compliqué à comprendre et adapté pour les plus novices d'entre vous.

```
var zoneDeplacement = new Rectangle(100,550,600,0)
var limite;
var cadre = new Loader()
var chemin = new URLRequest("image1.png")

addChild(cadre)
cadre.x = 100
cadre.y = 50
cadre.load(chemin)
cadre.contentLoaderInfo.addEventListener(Event.COMPLETE,imageChargee)

 curseur.addEventListener(MouseEvent.MOUSE_DOWN,deplacer)

function deplacer (evt:MouseEvent) {
    curseur.startDrag(false,zoneDeplacement)
    stage.addEventListener(MouseEvent.MOUSE_UP,arreterDeplacement)
    stage.addEventListener(MouseEvent.MOUSE_MOVE,deplacerImage)
}

function arreterDeplacement (evt:MouseEvent) {
    stopDrag()
    stage.removeEventListener(MouseEvent.MOUSE_UP,arreterDeplacement)
    stage.removeEventListener(MouseEvent.MOUSE_MOVE,deplacerImage)
}

function deplacerImage (evt:MouseEvent) {
    cadre.x = 100+((curseur.x-100)/600)*-limite;
}
```

```
bt1.sonImage = "image1.png"
bt2.sonImage = "image2.png"
bt3.sonImage = "image3.png"
```

```
bt1.addEventListener(MouseEvent.CLICK, chargerImage)
bt2.addEventListener(MouseEvent.CLICK, chargerImage)
bt3.addEventListener(MouseEvent.CLICK, chargerImage)
```

```
function chargerImage (evt:MouseEvent) {
    chemin.url = evt.currentTarget.sonImage
    cadre.load(chemin)
    cadre.contentLoaderInfo.addEventListener(Event.COMPLETE, imageChargee)
}
```

```
function imageChargee (evt:Event) {
    limite = (cadre.width-650)
}
```

Deuxième cas

Remarque : Vous n'avez plus besoin sur la scène, des occurrences intitulées **bt1**, **bt2** et **bt3**.

```
var zoneDeplacement = new Rectangle(100,550,600,0)
var limite;
var cadre = new Loader()
var chemin = new URLRequest("image1.png")
```

```
addChild(cadre)
cadre.x = 100
```

```
cadre.y = 50
cadre.load(chemin)
cadre.contentLoaderInfo.addEventListener(Event.COMPLETE,imageChargee)

curseur.addEventListener(MouseEvent.MOUSE_DOWN,deplacer)

function deplacer (evt:MouseEvent) {
    curseur.startDrag(false,zoneDeplacement)
    stage.addEventListener(MouseEvent.MOUSE_UP,arreterDeplacement)
    stage.addEventListener(MouseEvent.MOUSE_MOVE,deplacerImage)
}

function arreterDeplacement (evt:MouseEvent) {
    stopDrag()
    stage.removeEventListener(MouseEvent.MOUSE_UP,arreterDeplacement)
    stage.removeEventListener(MouseEvent.MOUSE_MOVE,deplacerImage)
}

function deplacerImage (evt:MouseEvent) {
    cadre.x = 100+((curseur.x-100)/600)*-limite;
}

function imageChargee (evt:Event) {
    limite = (cadre.width-650)
}
```