

Abordons ensemble un premier exemple, celui d'une carte à jouer sur laquelle nous allons cliquer pour pouvoir la faire disparaître.

1. Créez une nouvelle animation que vous enregistrez.
2. Créez un symbole de type Clip.
3. Gardez sélectionnée sur la scène, l'occurrence du clip que vous venez de créer et nommez la. Exemple : carte
4. Cliquez ensuite sur la scène (pas sur l'occurrence) ou sur l'image clé 1 de la fenêtre Scénario.
5. Cliquez dans la fenêtre Actions.
6. Saisissez le script ci-dessous.

```
function masquer(evt:MouseEvent) {  
  
    carte.visible=false;  
  
}  
  
carte.addEventListener(MouseEvent.CLICK,masquer)
```

Dans l'exemple suivant (ci-dessous), deux occurrences ont été placées sur la scène et nommées bouton1 et carte. En cliquant sur bouton1, nous souhaitons masquer l'occurrence intitulée carte, voici le script :

(Cliquez préalablement sur l'image-clé de la timeline puis dans la fenêtre Actions avant de modifier script actuel ou de resaisir celui ci-dessous)

```
function masquer(evt:MouseEvent) {
```

```
carte.visible=false;
```

```
}
```

```
bouton1.addEventListener(MouseEvent.CLICK,masquer)
```

Dans les deux exemples que nous venons d'aborder, nous avons fait appel à la constante `MOUSE_DOWN`, mais nous aurions pu utiliser d'autres évènements :

- `MOUSE_DOWN`
- `MOUSE_UP`
- `MOUSE_OVER`
- `MOUSE_OUT`
- `MOUSE_MOVE`
- `DOUBLE_CLICK`

N.B. : Pour utiliser l'évènement `DOUBLE_CLICK`, vous devez également régler la propriété `doubleClickEnabled` de l'occurrence concernée sur `true`.

```
carre.addEventListener(MouseEvent.CLICK,turner);
```

```
carre.doubleClickEnabled = true;
```

```
function tourner(evt:MouseEvent) {
```

```
    carre.rotation = 45;
```

```
}
```

### La propriété `currentTarget`

Afin qu'une fonction puisse "servir" à plusieurs occurrences, vous pouvez "cibler" une occurrence avec la propriété `currentTarget`.

1. Créez une nouvelle animation que vous enregistrez.
2. Créez un symbole de type `Clip`.
3. Dupliquez deux fois l'occurrence du clip que vous venez de créer.
4. Nommez les trois occurrences. Exemple : `carre1`, `carre2`, `carre3`
5. Cliquez ensuite sur la scène (pas sur l'occurrence) ou sur l'image clé 1 de la fenêtre Scénario.
6. Saisissez le script ci-dessous.

```
carre1.addEventListener(MouseEvent.CLICK,tourner);
```

```
carre2.addEventListener(MouseEvent.CLICK,tourner);
```

```
carre3.addEventListener(MouseEvent.CLICK,tourner);
```

```
function tourner(evt:MouseEvent) {
```

```
    evt.currentTarget.rotation = 45;
```

```
}
```

En ayant remplacé `carre` de l'exemple précédent par `evt.currentTarget`, nous pouvons utiliser la même fonction avec différentes occurrences.

Quelle est la différence entre **currentTarget** et **target** ?

La réponse à cette question va vous sembler évidente dès lors que vous aurez pris connaissance de l'exemple suivant :

Vous venez de placer sur la scène, 3 occurrences intitulées `carte1`, `carte2` et `carte3`. Vous les sélectionnez toutes les 3 et appuyez sur la touche de fonction F8 pour créer un nouveau symbole. Vous avez à présent sur la scène, non plus trois occurrences, mais une seule, celle de votre nouveau clip. Nommez la **ensemble**. Afin de bien comprendre la différence entre **currentTarget**

et **target**, cliquez sur l'image clé de la timeline ou sur la scène (en dehors de l'occurrence, puis cliquez dans la fenêtre Actions. Saisissez à présent ce script :

```
ensemble.addEventListener(MouseEvent.CLICK,deplacer);
```

```
ensemble.addEventListener(MouseEvent.CLICK,relacher);
```

```
function deplacer(evt:MouseEvent) {
```

```
    evt.currentTarget.startDrag();
```

```
}
```

```
function relacher(evt:MouseEvent) {
```

```
    stopDrag();
```

```
}
```

Modifiez à présent simplement le paramètre **currentTarget** par **target**.

```
ensemble.addEventListener(MouseEvent.CLICK,deplacer);
```

```
ensemble.addEventListener(MouseEvent.CLICK,relacher);
```

```
function deplacer(evt:MouseEvent) {
```

```
    evt.target.startDrag();
```

```
}
```

```
function relacher(evt:MouseEvent) {
```

```
    stopDrag();
```

```
}
```

Qu'observez-vous ? Lorsque vous utilisez la propriété `currentTarget`, cette dernière fait référence à l'occurrence sur laquelle vous avez posé un écouteur, ne tenant pas compte des éventuelles occurrences contenues dans l'occurrence ensemble. Avec la propriété `target`, le programme traverse l'occurrence ensemble et va réellement cibler l'occurrence cliquée et contenue dans l'occurrence ensemble. Conclusion, si vous souhaitez programmer l'interactivité des occurrences contenues dans une autre occurrence, vous pouvez utiliser `target` alors qu'avec la propriété `currentTarget`, vous ciblez celle que vous avez désigné dans la ligne d'instruction ci-dessous.

```
ensemble.addEventListener(MouseEvent.CLICK,deplacer);
```