

Vous avez créé une animation Flash et vous aimeriez à présent la transformer en une application autonome par le biais d'Adobe Air. Vous allez avoir besoin de certaines fonctionnalités que Quitter une application, personnaliser l'apparence d'une fenêtre si vous ne voulez pas de la barre de fenêtre de votre OS, déplacer une fenêtre si vous n'avez plus de barre de titre dans votre fenêtre (celle de l'OS), etc., voici donc quelques scripts pour vous aider.

### **Conserver une fenêtre au premier plan**

Au début de votre programme saisissez la ligne d'instruction ci-dessous :

```
this.stage.nativeWindow.alwaysInFront=true;
```

Remarque : Attention, n'abusez pas de cette fonctionnalité, forcer une application à rester au premier plan, devant toutes les autres, peut déplaire à certains utilisateurs. N'utilisez cette fonctionnalité que dans les cas de nécessité.

### **Rendre une fenêtre mobile**

Placez simplement une occurrence intitulée fond sur la scène de votre animation et saisissez ce script. Il peut s'agir d'un rectangle de la taille de la scène ou d'un simple élément graphique de la taille d'un bouton ou d'une barre comparable à la barre de titre d'une fenêtre.

```
fond.addEventListener(MouseEvent.MOUSE_DOWN, fenetreMobile);
```

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

```
    this.stage.nativeWindow.startMove();
```

```
}
```

### **Mode d'affichage plein écran**

Au moment où vous double-cliquerez sur une application Air, vous aurez peut-être besoin que celle-ci s'affiche en plein écran. Pour cela, il vous suffit tout simplement de saisir la ligne d'instruction ci-dessous :

```
stage.displayState=StageDisplayState.FULL_SCREEN_INTERACTIVE;
```

Remarque : Pour une application Air, afin que le clavier puisse être utilisé, on utilise FULL\_SCREEN\_INTERACTIVE et non FULL\_SCREEN.

Sur Macintosh, il a été relevé que dans certains cas, le plein écran affiche tout de même la barre des menus. Pour pouvoir la masquer, vous devrez retarder l'instant où la ligne d'instruction ci-dessus s'exécute :

```
var retardateur:Timer=new Timer(10,1);
retardateur.addEventListener(TimerEvent.TIMER,pleinEcran);

function pleinEcran(evt:TimerEvent) {
    stage.displayState=StageDisplayState.FULL_SCREEN_INTERACTIVE;
    stage.scaleMode=StageScaleMode.NO_SCALE;
}

retardateur.start();
```

Si vous utilisez un bouton qui doit basculer le mode fenêtre en mode plein écran, vous n'avez pas besoin de saisir toutes ces lignes, seule la ligne commençant par `stage.displayState...` suffit.

### Quitter l'application

Placez simplement une occurrence intitulée `boutonFermeture` sur la scène de votre animation et saisissez ce script.

```
boutonFermeture.addEventListener(MouseEvent.CLICK,fermerFenetre);
function fermerFenetre(evt:MouseEvent) {
    this.stage.nativeWindow.close();
} Savoir lorsqu'une application perd ou récupère le focus
```

Lorsque vous basculez d'une application à une autre (alors qu'elles sont toutes les deux lancées), vous perdez le focus sur la première pour le passer à la suivante. Une application que

vous créez peut tout à fait gère ce changement d'état, il vous suffit d'utiliser le script ci-dessous :

```
this.stage.nativeWindow.alwaysInFront=true;
```

```
carre.addEventListener(Event.ENTER_FRAME,turner)
```

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

```
    carre.rotation++
```

```
}
```

```
this.addEventListener(Event.DEACTIVATE,sortie)
```

```
this.addEventListener(Event.ACTIVATE,relancer)
```

```
function sortie (evt:Event) {
```

```
    carre.removeEventListener(Event.ENTER_FRAME,turner)
```

```
}
```

```
function relancer (evt:Event) {
```

```
carre.addEventListener(Event.ENTER_FRAME,turner)

}
```

Dans cet exemple, lorsque l'application perd le focus, une occurrence arrête de tourner.

### Fenêtre à deux états

Comme le montre la copie d'écran ci-dessous, certaines applications peuvent occuper différemment l'espace sur votre écran en fonction de leur état (déroulé, déployé, mode fenêtre / contracté, réduit, sous forme de palette).





Placer une occurrence (un objet d'affichage) sur un bord de l'écran

Attention, pour exécuter le script ci-dessous, cela sous entend que la taille de votre scène (celle de l'animation Flash) doit être au moins aussi grande que la résolution de votre écran. Si vous cherchez à positionner un objet d'affichage sur le bord droit ou inférieur de votre écran, cela sous-entend sûrement que vous souhaitez de toutes les façons réaliser une mise en page proportionnelle. Réglez donc les dimensions de votre de votre animation à 2000 pixels de largeur sur 2000 pixels de hauteur (pour être sûr que les grands écrans soient compatibles avec votre application).

Dans l'exemple ci-dessous, l'occurrence **zone** va être positionnée en bas à droite de votre écran.

```
//Positionner la fenêtre
```

```
stage.nativeWindow.x = 0;
```

```
stage.nativeWindow.y = 0;
```

```
//Mémoriser la largeur et la hauteur de l'écran
```

```
var largeurEcran:int = Capabilities.screenResolutionX;
```

```
var hauteurEcran:int = Capabilities.screenResolutionY;
```

```
//Placer l'occurrence en bas à droite de la scène
```

```
zone.x = largeurEcran;
```

```
zone.y = hauteurEcran;
```

Remarque : L'occurrence zone est issue d'un symbole dont le point d'alignement est située en bas à droite (du symbole).

