

## **PopupMenu**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> PopupMenu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 23, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>PopupMenu</b>	<b>1</b>
1.1	PopupMenu . . . . .	1
1.2	attachpopupmenu . . . . .	2
1.3	disablepopupmenuitem . . . . .	2
1.4	freepopupmenu . . . . .	2
1.5	initpopupmenu . . . . .	2
1.6	popupmenubar . . . . .	3
1.7	popupmenuchecked . . . . .	3
1.8	popupmenucheckitem . . . . .	3
1.9	popupmenuchecksubitem . . . . .	4
1.10	popupmenuevent . . . . .	4
1.11	popupmenuinfo . . . . .	5
1.12	popupmenuitem . . . . .	5
1.13	popupmenusubbar . . . . .	5
1.14	popupmenusubinfo . . . . .	5
1.15	popupmenusubitem . . . . .	6
1.16	popupmenusubmenuitem . . . . .	6
1.17	popupmenutitle . . . . .	6
1.18	waitpopupmenuevent . . . . .	7

## Chapter 1

# PopupMenu

### 1.1 PopupMenu

PureBasic - PopupMenu V1.00

Un 'PopupMenu', appelé aussi menu contextuel, à pour but d'apparaître lors d'un click droit en dessous du pointeur de la souris. Ce menu change en fonction de la zone qui est survolée, ce qui permet d'avoir des menus spécifique pour chaque objet différent. Etant donné que l'AmigaOS ne supporte pas les menus contextuels en standard, l'excellente 'popupmenu.library' est utilisée. N'oubliez pas de la fournir avec votre programme si vous l'utilisez. Le paramétrage de la popupmenu.library s'effectue par le biais de son programme de préférence.

Commandes disponibles:

- AttachPopupMenu
- DisablePopupMenuItem
- FreePopupMenu
- InitPopupMenu
- PopupMenuBar
- PopupMenuChecked
- PopupMenuCheckItem
- PopupMenuCheckSubItem
- PopupMenuEvent
- PopupMenuInfo
- PopupMenuItem
- PopupMenuSubBar
- PopupMenuSubInfo
- PopupMenuSubItem
- PopupMenuSubMenuItem
- PopupMenuTitle
- WaitPopupMenuEvent

Exemple:

PopupMenu demo

## 1.2 attachpopupmenu

### Syntaxe

```
Résultat.w = AttachPopupMenu(#Menu.w, Window.l)
```

### Résumé

Ce commande doit être appelée pour terminer chaque menus créé avec une séquence de commandes comme `PopupMenuTitle()`, `PopupMenuItem()`, `PopupMenuSubItem()`, etc... sinon le menu ne sera construit ni utilisé. Si le résultat est NUL, le menu a pu être créer et attaché à la fenetre. Sinon, une erreur est survenue.

#Menu : l'identifiant numérique du nouveau menu à créer

Window: l'identifiant de la fenetre sur laquelle le menu contextuel doit être rattaché (il est conseillé d'utilisé `WindowID()` pour obtenir cette valeur)

## 1.3 disablepopupmenuitem

### Syntaxe

```
DisablePopupMenuItem(#Menu.w, Item.w, State.w)
```

### Résumé

Active ou désactive n'importe quel élément d'un menu contextuel sélectionnable.

#Menu: l'identifiant numérique du menu dans lequel se trouve l'élément à désactiver ou activer.

Item: l'identifiant numérique de l'élément à activer/désactiver.

State: FALSE: active l'élément, TRUE: désactive l'élément.

## 1.4 freepopupmenu

### Syntaxe

```
FreePopupMenu(#Menu.w)
```

### Résumé

Détruit le menu contextuel spécifié. Il n'est plus possible d'utiliser ce menu (à moins d'en créer un nouveau).

#Menu: l'identifiant numérique du menu à détruire

## 1.5 initpopupmenu

---

#### Syntaxe

```
Résultat.l = InitPopupMenu(#Menus.l)
```

#### Résumé

Initialise l'environnement nécessaire à la gestion des menus contextuels. Cette commande doit être appelée avant n'importe quelle autre commande de cette bibliothèque. Il est impératif de tester la valeur retournée par cette commande car elle permet de savoir si la bibliothèque 'popupmenu.library' version 9.0 ou supérieure est disponible. Si le résultat est NUL, cette bibliothèque est introuvable et vous devez désactiver tous les appels aux commandes de cette bibliothèque ou quitter le programme.

#Menus: Nombre maximum de menus contextuels à gérer.

## 1.6 popupmenubar

#### Syntaxe

```
PopupMenuBar()
```

#### Résumé

Permet d'insérer une barre de séparation au menu contextuel en cours de construction.

## 1.7 popupmenuchecked

#### Syntaxe

```
Résultat.w = PopupMenuChecked(#Menu.w, Item.w)
```

#### Résumé

Permet de savoir si un 'CheckItem' est coché ou non. Si le résultat est NUL, alors le 'CheckItem' n'est pas coché, sinon il est coché.

#Menu: l'identifiant numérique du menu dans lequel se trouve l'élément.

Item: l'identifiant numérique de l'élément.

## 1.8 popupmenucheckitem

#### Syntaxe

```
PopupMenuCheckItem(Item.w, Text$, ShortCut$)
```

#### Résumé

Ajoute un élément 'CheckItem' (de type case à cocher) au menu contextuel en cours de construction. L'état initial de l'élément sera décoché.

Item: Identifiant numérique de l'élément (qui servira à identifier cet élément plus tard grâce à la fonction `WaitPopupMenuEvent()` ou `PopupMenuEvent()`).

Text\$: Le texte qui sera affiché dans le menu contextuel

Shortcut\$: Raccourci clavier pour cet élément. Si aucun raccourci n'est nécessaire, ←  
 passez une chaîne nulle en paramètre ("").

## 1.9 popupmenuchecksubitem

### Syntaxe

`PopupMenuCheckSubItem(SubItem.w, Text$, Shortcut$)`

### Résumé

Ajoute un sous menu de type 'CheckItem' (case à cocher) au menu en cours contextuel de construction. L'état initial de l'élément sera décoché.

SubItem: Identifiant numérique de l'élément (qui servira à identifier cet élément plus tard grâce à la fonction `WaitPopupMenuEvent()` ou `PopupMenuEvent()`).

Text\$: Le texte qui sera affiché dans le menu contextuel

Shortcut\$: Raccourci clavier pour cet élément. Si aucun raccourci n'est nécessaire, ←  
 passez une chaîne nulle en paramètre ("").

## 1.10 popupmenuevent

### Syntaxe

`Résultat.l = PopupMenuEvent(#Menu.w)`

### Résumé

Permet de savoir si un événement relatif aux menus contextuels est survenu. Si oui, 'Résultat' sera différent de NUL. Cette commande n'est pas bloquante et retourne immédiatement. `WaitPopupMenuEvent()`, au contraire, permet d'attendre jusqu'à ce qu'un événement survienne.

#Menu: Le menu contextuel à ouvrir si le bouton droit de la souris est pressé.

Résultat: l'identifiant de l'élément qui a été choisi dans le menu contextuel.

## 1.11 popupmenuinfo

### Syntaxe

```
PopupMenuInfo(Text$)
```

### Résumé

Ajoute une ligne d'information au menu contextuel en cours de création. Il ne sera pas possible de choisir cet élément ultérieurement.

Il est nécessaire de mettre une barre de séparation après cet élément via la commande `PopupMenuBar()` sinon le texte ne sera pas correctement rendu.

**Text\$:** Texte à afficher dans le menu contextuel

## 1.12 popupmenuitem

### Syntaxe

```
PopupMenuItem(Item.w, Text$, ShortCut$)
```

### Résumé

Permet d'ajouter un élément basique (type texte) dans le menu contextuel en cours de construction.

**Item:** Identifiant numérique de l'élément (qui servira à identifier cet élément plus tard grâce à la fonction `WaitPopupMenuEvent()` ou `PopupMenuEvent()`).

**Text\$:** Le texte qui sera affiché dans le menu contextuel

**ShortCut\$:** Raccourci clavier pour cet élément. Si aucun raccourci n'est nécessaire, ←  
passez une chaîne nulle en paramètre ("").

## 1.13 popupmenusubbar

### Syntaxe

```
PopupMenuSubBar()
```

### Résumé

Ajoute une barre de séparation dans le sous menu en cours.

## 1.14 popupmenusubinfo

### Syntaxe

```
PopupMenuSubInfoItem(Text$)
```

### Résumé



Ajoute une ligne d'information au sous menu en cours de création. Il ne sera pas possible de choisir cet élément ultérieurement.

Il est nécessaire de mettre une barre de séparation après cet élément via la commande `PopupMenuBar()` sinon le texte ne sera pas correctement rendu.

**Text\$:** Texte à afficher dans le menu contextuel

## 1.15 `popupmenusubitem`

### Syntaxe

```
PopupMenuSubItem(SubItem.w, Text$, ShortCut$)
```

### Résumé

Permet d'ajouter un élément basique (type texte) dans le sous menu en cours de construction.

**Item:** Identifiant numérique de l'élément (qui servira à identifier cet élément plus tard grâce à la fonction `WaitPopupMenuEvent()` ou `PopupMenuEvent()`).

**Text\$:** Le texte qui sera affiché dans le menu contextuel

**ShortCut\$:** Raccourci clavier pour cet élément. Si aucun raccourci n'est nécessaire, passez une chaîne nulle en paramètre (`""`).

## 1.16 `popupmenusubmenuitem`

### Syntaxe

```
PopupMenuSubMenuItem(Text$)
```

### Résumé

Permet de créer une entrée pour un sous-menu dans le menu contextuel en cours de création. Tous les éléments qui suivent peuvent être du type `PopupMenuSubxxxx()`.

**Text\$:** Le texte pour cet élément

## 1.17 `popupmenutitle`

### Syntaxe

```
PopupMenuTitle(Text$)
```

### Résumé

Permet de créer un titre pour le menu contextuel. C'est la première commande qui doit être appelée lors de la création

---

d'un menu contextuel.

Text\$: Le titre du menu contextuel

## 1.18 waitpopupmenuevent

### Syntaxe

```
Résultat.l = WaitPopupMenuEvent (#Menu.w)
```

### Résumé

Permet de savoir si un événement relatif aux menus contextuels est survenu. Si oui, 'Résultat' sera différent de NUL. Cette commande bloque le programme jusqu'à ce qu'un événement arrive. `PopupMenuEvent()`, au contraire, n'attend pas et retourne immédiatement.

#Menu: Le menu contextuel à ouvrir si le bouton droit de la souris est pressé.

Résultat: l'identifiant de l'élément qui a été choisi dans le menu contextuel.