

## Table des matières

|                                 |   |
|---------------------------------|---|
| I – Définition                  | 2 |
| II – Les propriétés disponibles | 2 |
| III – Les méthodes disponibles  | 3 |
| IV – Les événements disponibles | 5 |

## I – Définition

**MenuFx** est un module conçu pour la gestion des menu dans un jeu vidéo. Vous pouvez réaliser toute sorte de menu rien qu'en utilisant ce module. Le module implémente les règles de bases de réalisation d'un menu.

**NB :** Ce module est compatible à un jeu 2D, 3D et est sauvegardable.

## II – Les propriétés disponibles

+ **int ColumnCount** = **0** : Contient le nombre de colonne sur l'horizontal. La valeur de cette propriété est dans l'intervalle [0; ++].

+ **int RowCount** = **0** : Contient le nombre de ligne sur la vertical. La valeur de cette propriété est dans l'intervalle [0; ++].

+ **bool InfiniteScrolling** = **false** : Voulez-vous mettre un effet de sélection infinie au cours des sélections ?

+ **bool MouseCapture** = **true** : Désirez-vous synchroniser la souris au menu ?

+ **float VolatilTime** = **0.0** : Souhaitez-vous faire un menu volatil ? Toute valeur inférieur ou égale à **0.0** désactive l'activité de cette propriété. Pour exploiter cette option, il faut écouté les événements *menu\_begin* et *menu\_end*. La valeur de cette propriété est dans l'intervalle [0.0; ++].

+ **bool ExternalSelector** = **false** : Devrons nous mettre un sélecteur externe au menu ? Le sélecteur externe fait référence à un objet externe utilisé pour matérialiser la sélection des différentes options du menu.

+ **Vector3 SelectorTransition** = **Vector3 (0.0, 0, 2)** : Devrons nous appliquer une quelconque animation vectorielle à votre sélecteur externe ? La première valeur contient le temps de la transition ; la seconde l'assouplissement et la dernière, le type de transition à utilisé. Notez les valeurs possibles des deux dernières entrées de cette propriété sont celles de Godot.

+ **NodePath ListenModule** : Contient l'instance d'un module du même trampe de ce module. Exemple : *AnimatorFx*, *MenuFx*, *AudioTrackFx* etc... En d'autres termes, contient la référence d'un module possédant également un curseur. Le but de cette fonctionnalité est de synchroniser la valeur du curseur du module référé à celle de ce module.

+ **NodePath HozScrollBar** : Contient l'instance d'un noeud de type *HScrollBar*.

+ **NodePath VerScrollBar** : Contient l'instance d'un noeud de type *VScrollBar*.

+ **int ScrollStep = 0** : Quel est le pas à ajouté pour défiler le défileur du menu ? La valeur de cette propriété est dans l'intervalle [0 ++].

+ **Array Options** : Tableau de dictionnaires contenant toutes les différentes configurations sur chaque option du menu que le développeur souhaite implémentée. Les dictionnaires issus de ce tableau supportent les clés suivantes :

- » **NodePath | String option** : Quel est le chemin d'accès l'option que vous voulez ajouter ?  
L'utilisation de cette clé est obligatoire.
- » **String name** : Quel nom désirez-vous donner à votre option ?
- » **bool ignored = false** : Voulez-vous que le module ignore cette option à chaque fois que l'on mettra le focus dessus ?
- » **bool locked = false** : Voulez-vous désactiver le(s) action(s) que génère cette option ?
- » **bool tabcontrol = false** : Souhaitez-vous exécuter le(s) action(s) prévue(s) sur l'option lorsqu'on la sélectionne directement ?
- » **Array | Dictionary actions** : Que se passera t-il lorsqu'on exécutera le(s) action(s) que cache l'option ? Cette clé contient toutes les configurations relatives à un ou plusieurs flux d'exécution(s). L'utilisation de cette clé est déjà décrite au niveau des bases du framework. Précisément le sujet portant sur l'utilisation de la propriété *EventsBindings* (la section des actions d'un événement).

**NB** : Les répétitions au niveau des noms des options ne seront pas tolérées.

### III – Les méthodes disponibles

- + **void next (delay = 0.0)** : Passe à l'option suivante dans le menu établi.
  - » **float delay** : Quel est le temps mort avant le passage.
- + **void h\_next (delay = 0.0)** : Passe à l'option suivante dans le menu vertical. N'utilisez cette méthode que le menu est configuré en deux dimension à savoir les valeurs des champs *ColumnCount* et *RowCount* sont supérieur à zéro. Notez que si l'une des valeurs de ces deux champs est supérieur à zéro et que l'autre est nulle, le menu est configuré en une dimension. Il est très important de comprendre cela.
  - » **float delay** : Quel est le temps mort avant le passage.

- + **void v\_next** (delay = 0.0) : Passe à l'option suivante dans le menu horizontal. N'utilisez cette méthode que le menu est configuré en deux dimension à savoir les valeurs des champs *ColumnCount* et *RowCount* sont supérieur à zéro. Notez que si l'une des valeurs de ces deux champs est supérieur à zéro et que l'autre est nulle, le menu est configuré en une dimension. Il est très important de comprendre cela.
  - » **float delay** : Quel est le temps mort avant le passage.
  
- + **void preview** (delay = 0.0) : Passe à l'option précédente dans le menu établi.
  - » **float delay** : Quel est le temps mort avant le passage.
  
- + **void h\_preview** (delay = 0.0) : Passe à l'option précédente dans le menu vertical. N'utilisez cette méthode que le menu est configuré en deux dimension à savoir les valeurs des champs *ColumnCount* et *RowCount* sont supérieur à zéro. Notez que si l'une des valeurs de ces deux champs est supérieur à zéro et que l'autre est nulle, le menu est configuré en une dimension. Il est très important de comprendre cela.
  - » **float delay** : Quel est le temps mort avant le passage.
  
- + **void v\_preview** (delay = 0.0) : Passe à l'option précédente dans le menu horizontal. N'utilisez cette méthode que le menu est configuré en deux dimension à savoir les valeurs des champs *ColumnCount* et *RowCount* sont supérieur à zéro. Notez que si l'une des valeurs de ces deux champs est supérieur à zéro et que l'autre est nulle, le menu est configuré en une dimension. Il est très important de comprendre cela.
  - » **float delay** : Quel est le temps mort avant le passage.
  
- + **void next\_to** (id, delay = 0.0) : Change d'option grâce à son identifiant.
  - » **String | int id** : Quel est l'identifiant de l'option à rendre active ?
  - » **float delay** : Quel est le temps mort avant le passage.
  
- + **int get\_active\_option\_index** () : Renvoie l'index de l'option actuellement sélectionnée dans le menu établi.
  
- + **int get\_preview\_option\_index** () : Renvoie l'index de l'option précédemment sélectionnée dans le menu établi.
  
- + **int get\_next\_option\_index** () : Renvoie l'index de la prochaine option à sélectionnée dans le menu établi.
  
- + **void run\_actions** (id = null, delay = 0.0) : Exécute toutes les actions que cache une option donnée. Par défaut, l'exécution se fera sur l'option sélectionnée.
  - » **String | int id** : Quel est l'identifiant de l'option à ciblée ?
  - » **float delay** : Quel est le temps mort avant les exécutions.

## IV – Les événements disponibles

- + **menu\_action (data)** : Signal déclenché lorsqu'on exécute le(s) action(s) d'une option donnée. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option ayant été frappée par la méthode *run\_actions* *O*.
- + **option\_changed (data)** : Signal déclenché lorsqu'on change d'option dans le menu établi. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int preview** : Contient l'index de position de l'option précédemment sélectionnée.
  - » **int current** : Contient l'index de position de l'option actuellement sélectionnée.
- + **option\_ignored (data)** : Signal déclenché lorsqu'on sélectionne une option qui doit être ignorée par le module. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option ignorée.
- + **option\_locked (data)** : Signal déclenché lorsqu'on déclenche les actions d'une option bloquée. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option bloquée.
- + **menu\_begin (node)** : Signal déclenché lorsqu'on appel des fonctions en faveur des options du menu.
  - » **Node node** : Contient le noeud où ce signal a été émit.
- + **menu\_end (node)** : Signal déclenché lorsque le temps de volatilisation du menu est écoulé.
  - » **Node node** : Contient le noeud où ce signal a été émit.
- + **mouse\_enter (data)** : Signal déclenché lorsque le curseur de la souris entre dans le survol d'une option du menu établi. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option en question.
- + **mouse\_exit (data)** : Signal déclenché lorsque le curseur de la souris sort du survol d'une option du menu établi. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option en question.

- + **mouse\_over (data)** : Signal déclenché lorsque le curseur de la souris survol une option du menu établit. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option en question.
  
- + **mouse\_clicked (data)** : Signal déclenché lorsque le curseur de la souris clique une option du menu établit. Ce événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **int index** : Contient l'index de position de l'option en question.
  - » **int button** : Contient l'index du bouton qui a été appuyée.