

## Table des matières

I – Définition	2
II – Les propriétés disponibles	2
III – Les méthodes disponibles	5
IV – Les événements disponibles	7

## I – Définition

**CameraControlFx** est un module conçu pour la gestion des caméras dans un jeu vidéo. Le module offre également aux développeurs des fonctionnalités afin de leur permettre de configurer facilement la caméra de leur joueur.

**NB :** Ce module est compatible à un jeu 3D et n'est pas sauvegardable.

## II – Les propriétés disponibles

- + **NodePath** **TargetCamera** : Contient l'instance d'un noeud de type *Camera*.
- + **NodePath** **Privot** : Contient l'instance d'un noeud de type *Spatial*.
- + **int** **Mode** = **0** : Contient le mode de la caméra que vous souhaitez utiliser. Les valeurs possibles sont :
  - > **CameraControlFx.Model.FPS** ou **0** : Mode FPS (First Person Shooter) activé.
  - > **CameraControlFx.Model.THIRD\_PERSON** ou **1** : Configuration de la caméra à la troisième personne.
- + **int** **KeepTransform** = **0** : Contrôle la transformation que doit écouter la caméra afin de pouvoir bien suivre sa cible.
  - > **MegaAssets.Transformation.LOCATION** ou **1** : La caméra suivra sa cible en écoutant sa position.
  - > **MegaAssets.Transformation.LOCROT** ou **4** : La caméra suivra sa cible en écoutant sa position ainsi que sa rotation.
- + **float** **Distance** = **5.0** : Quelle est la distance de la caméra par rapport à sa cible ?
- + **Vector3** **OffsetPosition** = **Vector3** (**0.0**, **0.0**, **0.0**) : Contrôle l'ajustement de la caméra en terme de position.
- + **Vector3** **Zooming** = **Vector3** (**0.0**, **100.0**, **0.0**) : Contrôle le niveau d'agrandissement de la caméra. La première valeur est le zoom minimum ; la deuxième, le zoom maximum et la troisième le taux d'agrandissement actuelle de la caméra.
- + **bool** **ZoomTurbulence** = **false** : La caméra va t-elle gérée automatiquement l'agrandissement en fonction de la vitesse de sa cible par rapport l'agrandissement fixé ?
- + **int** **Layout** = **2** : Contrôle l'angle de direction de la caméra. Les valeurs possibles sont :
  - > **MegaAssets.Disposal.TOP** ou **2** : Vue de haut.
  - > **MegaAssets.Disposal.BOTTOM** ou **4** : Vue de bas.
  - > **MegaAssets.Disposal.FORWARD** ou **6** : Vue de face.
  - > **MegaAssets.Disposal.BACKWARD** ou **7** : Vue de dos.

- > **MegaAssets.Disposal.LEFT** ou **5** : Vue de gauche.
- > **MegaAssets.Disposal.RIGHT** ou **3** : Vue de droite.
  
- + **Vector3** **OffsetRotation** = **Vector3** (**0.0**, **0.0**, **0.0**) : Contrôle l'ajustement de la caméra en terme de rotation.
  
- + **Vector2** **RotationLimit** = **Vector2** (**360.0**, **360.0**) : Contient l'angle limite de la rotation de la caméra.
  
- + **Vector2** **Velocity** = **Vector2** (**0.5**, **0.5**) : Contrôle la sensibilité et l'adoucissement de la rotation de la caméra. La première valeur est la sensibilité de la rotation et la deuxième, l'adoucissement.
  
- + **Vector2** **Cinematic** = **Vector2** (**1.0**, **0.1**) : Contrôle l'accélération et la décélération des mouvements de la caméra. La première valeur est l'accélération et la deuxième, la décélération.
  
- + **Vector3** **MaxSpeed** = **Vector3** (**1.0**, **1.0**, **1.0**) : Contrôle la vitesse des mouvements de la caméra.
  
- + **bool** **Collision** = **false** : Désirez-vous que la caméra percute tous les objets dotés d'un collisionneur ?
  
- + **bool** **Local** = **true** : Les mouvements de caméra vont-ils se faire dans un monde local ?
  
- + **bool** **RotatePivot** = **false** : Désirez-vous tourner l'objet pivot que chasse la caméra ?
  
- + **int** **AutoReset** = **0** : Voulez-vous réinitialiser les mouvements de la caméra de façon automatique ?  
Les valeurs possibles sont :
  - > **CameraControlFx.Reset.NONE** ou **0** : Aucune réinitialisation ne sera effectuée.
  - > **CameraControlFx.Reset.At\_ONCE** ou **1** : Réinitialisation immédiate de la caméra.
  - > **CameraControlFx.Reset.GRADUALLY** ou **2** : Réinitialisation progressive de la caméra. Ce type de réinitialisation suit le déplacement de sa cible.
  
- + **Vector2** **Gap** = **Vector2** (**0.0**, **0.0**) : A quel angle ou à quelle position de la cible devons nous réinitialiser la caméra ? Cette option peut s'avérer très pratique dans certains cas d'utilisation. La première valeur est l'angle à atteindre pour réinitialiser la caméra et la deuxième, la distance à atteindre.
  
- + **Vector2** **ShakeAxis** = **Vector2** (**25.0**, **25.0**) : Quel sont les axes qui seront affectés par la vibration.
  
- + **Vector2** **ShakeDensity** = **Vector2** (**25.0**, **1.0**) : Contient la densité de la vibration de la caméra. La première valeur contient le taux de roulage et la seconde, l'intensité de la vibration.
  
- + **bool** **TargetingSystem** = **false** : Désirez-vous activer le système de pistage de la caméra ?
  
- + **NodePath** **TargetingArea** : Contient l'instance d'un noeud de type *Area*.

+ **Array Targets** : Tableau de dictionnaires contenant toutes les différentes configurations sur chaque objet prise en charge par le développeur. Les dictionnaires issus de ce tableau supportent les clés suivantes :

- » **String id** : Quel est l'identifiant du noeud à prendre en charge ? L'utilisation de cette clé est obligatoire.
  - » **int search = 3** : Quel moyen utilisé pour chercher le noeud à prendre en charge ? Notez que l'identifiant donné est pisté à par un programme de recherche. Les valeurs possibles sont :
    - > **MegaAssets.NodeProperty.NAME ou 0** : Trouve un noeud en utilisant son nom.
    - > **MegaAssets.NodeProperty.GROUP ou 1** : Trouve un noeud en utilisant le nom de son groupe.
    - > **MegaAssets.NodeProperty.TYPE ou 2** : Trouve un noeud en utilisant le nom de sa classe.
    - > **MegaAssets.NodeProperty.ANY ou 3** : Trouve un noeud en utilisant l'un des trois moyens cités plus haut.
  - » **bool ignored = false** : Le pisteur de la caméra doit-il ignorer l'identifiant précisé ?
  - » **float transition = 1.0** : Combien de temps prend le passage d'une cible à une autre ?
  - » **int type = 0** : Quel type de transition adopté ? Les valeurs possibles de ce champ sont celles de Godot. Cette propriété est sollicitée au changement de cible.
  - » **int easing = 2** : Quel assouplissement adopté ? Les valeurs possibles de ce champ sont celles de Godot. Cette propriété est sollicitée au changement de cible.
  - » **Array | Dictionary entered** : Signal déclenché lorsque l'objet entre dans le champ de vision du pisteur. Cette clé exécute les différentes actions données à son déclenchement. Pour soumettre les actions à exécutées référez vous à la méthode utilisée au niveau de la clé *actions* de la propriété *EventsBindings* dans les bases du framework.
  - » **Array | Dictionary exited** : Signal déclenché lorsqu'un objet sort du champ de vision du pisteur. Cette clé exécute les différentes actions données à son déclenchement. Pour soumettre les actions à exécutées référez vous à la méthode utilisée au niveau de la clé *actions* de la propriété *EventsBindings* dans les bases du framework.
- + **int TargetingDirection = 2** : Sur quelle direction les pistages se feront ? Les valeurs possibles sont :
  - > **MegaAssets.Axis.X ou 1** : L'axe des abscisses.
  - > **MegaAssets.Axis.Y ou 2** : L'axe des ordonnées.
  - > **MegaAssets.Axis.Z ou 3** : L'axe des côtes.
  - > **MegaAssets.Axis.\_X ou 4** : L'opposé de l'axe des abscisses.
  - > **MegaAssets.Axis.\_Y ou 5** : L'opposé de l'axe des ordonnées.
  - > **MegaAssets.Axis.\_Z ou 6** : L'opposé de l'axe des côtes.

- + **int** **FroozenTargetingAxis** = **1** : Quel axe bloqué au cours des pistages ? Les valeurs possibles sont :
  - > **MegaAssets.Axis.NONE** ou **0** : Aucun blockage.
  - > **MegaAssets.Axis.X** ou **1** : Blockage de l'axe des abscisses.
  - > **MegaAssets.Axis.Y** ou **2** : Blockage de l'axe des ordonnées.
  - > **MegaAssets.Axis.Z** ou **3** : Blockage de l'axe des côtes.
  - > **MegaAssets.Axis.XY** ou **7** : Blockage des axes x et y.
  - > **MegaAssets.Axis.XZ** ou **8** : Blockage des axes x et z.
  - > **MegaAssets.Axis.YZ** ou **9** : Blockage des axes y et z.
  - > **MegaAssets.Axis.XYZ** ou **10** : Blockage des axes x, y et z.
  
- + **bool** **TargetingPriority** = **true** : Devront nous rendre prioritaire le pistage automatique devant les actions du joueur ? En d'autres termes, le joueur peut-il manipuler la caméra lorsqu'elle piste automatique un objet ?

### III – Les méthodes disponibles

- + **void** **reset\_camera** (**delay** = 0.0) : Réinitialise la caméra par rapport à sa cible.
  - » **float** **delay** : Quel est le temps mort avant la réinitialisation ?
  
- + **void** **change\_target** (**index** = -1, **delay** = 0.0) : Force la caméra à changé de cible parmit celles détectées. Par défaut, une cible est générée si l'index de la nouvelle cible n'a pas été donné.
  - » **int** **index** : Contient l'index de la nouvelle cible à pistée.
  - » **float** **delay** : Quel est le temps mort avant le changement ?
  
- + **void** **zoom** (**depth** = -1, **transition** = 1.0, **type** = 0, **easing** = 2, **delay** = 0.0) : Génère un effet d'agrandissement en avant.
  - » **float** **depth** : Contient le degré d'agrandissement. Par défaut, celui précisé dans les configurations de la caméra est utilisé pour faire le traitement.
  - » **float** **transition** : L'agrandissement se fera sur combien de temps ?
  - » **int** **type** : Quel type de transition adopté ? Les valeurs possibles de ce champ sont celles de Godot.
  - » **int** **easing** : Quel assouplissement adopté ? Les valeurs possibles de ce champ sont celles de Godot.
  - » **float** **delay** : Quel est le temps mort avant l'agrandissement ?
  
- + **void** **unzoom** (**depth** = -1, **transition** = 1.0, **type** = 0, **easing** = 2, **delay** = 0.0) : Génère un effet d'agrandissement en arrière.
  - » **float** **depth** : Contient le degré de désagrandissement. Par défaut, celui précisé dans les configurations de la caméra est utilisé pour faire le traitement.
  - » **float** **transition** : Le désagrandissement se fera sur combien de temps ?

- » **int type** : Quel type de transition adopté? Les valeurs possibles de ce champ sont celles de Godot.
  - » **int easing** : Quel assouplissement adopté? Les valeurs possibles de ce champ sont celles de Godot.
  - » **float delay** : Quel est le temps mort avant le désagrandissement?
- + **void zooming (velocity, depth = -1)** : Etablit un effet d'agrandissement en fonction de la valeur contenu dans le paramètre *velocity*.
- » **float depth** : Contient le degré de désagrandissement. Par défaut, celui précisé dans les configurations de la caméra est utilisé pour faire le traitement.
  - » **float velocity** : Contient la vitesse de l'agrandissement.
- + **void unzooming (velocity, depth = -1)** : Etablit un effet de désagrandissement en fonction de la valeur contenu dans le paramètre *velocity*.
- » **float depth** : Contient le degré de désagrandissement. Par défaut, celui précisé dans les configurations de la caméra est utilisé pour faire le traitement.
  - » **float velocity** : Contient la vitesse du désagrandissement.
- + **Dictionary get\_targets\_data (json = false)** : Renvoie toutes les données concernant les cibles de la caméra.
- » **bool json** : Voulez-vous renvoyer les données au format json?
- + **void rot (velocity)** : Pivote la caméra. Cette méthode s'adapte en fonction du mode choisi par le développeur.
- » **Vector2 | Vector3 depth** : Contient la vitesse du pivotement.
- + **void local\_translation (velocity)** : Déplace la caméra d'une position à une autre. Cette méthode agit sur la propriété *OffsetPosition* pour provoquer un mouvement de translation dans les axes.
- » **Vector2 | Vector3 depth** : Contient la vitesse du déplacement.
- + **void local\_rotation (velocity)** : Pivote la caméra d'un angle à un autre. Cette méthode agit sur la propriété *OffsetRotation* pour provoquer un mouvement de rotation dans les axes.
- » **Vector2 | Vector3 depth** : Contient la vitesse du pivotement.
- + **Spatial get\_current\_target ()** : Renvoie la référence de l'objet ou du noeud actuellement ciblé par le système de pistage automatique.
- + **Spatial get\_preview\_target ()** : Renvoie la référence de l'objet ou du noeud ayant été précédemment ciblé par le système de pistage automatique.

- + **Spatial** **get\_next\_target** () : Renvoie la référence du future objet ou noeud qui sera ciblé par le système de pistage automatique.
- + **bool** **is\_reset** () : A t-on déjà réinitialisé la rotation de la caméra ?
- + **void** **shaken** (**amount**, **delay** = 0.0) : Vibre la caméra.
  - » **float** **amount** : Contient la force de vibration de la caméra.
  - » **float** **delay** : Quel est le temps mort avant la vibration ?
- + **void** **generate\_shaken** (**amount**, **delay** = 0.0) : Génère aléatoirement une vibration de la caméra. Notez que cette méthode affecte les configurations effectuées sur la caméra sur ses vibrations. Il se peut donc qu'après une appelle de cette méthode, les valeurs des champs *ShakeAxis* et *ShakeDensity* est changées.
  - » **float** **amount** : Contient la force de vibration de la caméra.
  - » **float** **delay** : Quel est le temps mort avant la vibration ?

## IV – Les événements disponibles

- + **reset** (**node**) : Signal déclenché lorsque la caméra a été réinitialisée.
  - » **Node** **node** : Contient le noeud où ce signal a été émit.
- + **target\_changed** (**data**) : Signal déclenché lorsque le système de pistage automatique de la caméra a changé de cible. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node** **node** : Contient le noeud où ce signal a été émit.
  - » **Node** **target** : Contient la référence de la nouvelle cible de la caméra.
- + **target\_entered** (**data**) : Signal déclenché lorsqu'un objet entre dans le champ de vision du pisteur automatique de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node** **node** : Contient le noeud où ce signal a été émit.
  - » **Node** **target** : Contient la référence de l'objet détecté.
- + **target\_exited** (**data**) : Signal déclenché lorsqu'un objet sort du champ de vision du pisteur automatique de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node** **node** : Contient le noeud où ce signal a été émit.
  - » **Node** **target** : Contient la référence de l'objet détecté.
- + **target\_generated** (**data**) : Signal déclenché lorsqu'une cible a été générée par le pisteur automatique de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node** **node** : Contient le noeud où ce signal a été émit.
  - » **Node** **target** : Contient la référence de la future cible du pisteur automatique de la caméra.

- + **targeting\_enabled (node)** : Signal déclenché lorsque le système de pistage automatique de la caméra s'active.
  - » **Node node** : Contient le noeud où ce signal a été émit.
- + **targeting\_disabled (node)** : Signal déclenché lorsque le système de pistage automatique de la caméra se désactive.
  - » **Node node** : Contient le noeud où ce signal a été émit.
- + **min\_zoom (data)** : Signal déclenché lorsqu'on atteint l'agrandissement minimale de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le degré actuelle de l'agrandissement de la caméra.
- + **max\_zoom (data)** : Signal déclenché lorsqu'on atteint l'agrandissement maximale de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le degré actuelle de l'agrandissement de la caméra.
- + **zoom\_started (data)** : Signal déclenché lorsqu'on démarre l'opération d'agrandissement de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le degré actuelle de l'agrandissement de la caméra.
- + **zoom\_finished (data)** : Signal déclenché lorsque l'opération d'agrandissement de la caméra s'arrête. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le degré actuelle de l'agrandissement de la caméra.
- + **zoom (data)** : Signal déclenché au changement de la valeur d'agrandissement de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le degré actuelle de l'agrandissement de la caméra.
- + **generate\_zoom (data)** : Signal déclenché lorsqu'une valeur d'agrandissement a été générée suite à un calcul effectué en fonction de la vitesse de l'objet pivot de la caméra. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float depth** : Contient le future degré d'agrandissement de la caméra.



- + **max\_distance (node)** : Signal déclenché lorsque la distance de l'objet pivot de la caméra est supérieur ou égale à la deuxième valeur du champ *Gap*.
  - » **Node node** : Contient le noeud où ce signal a été émit.
  
- + **max\_angle (node)** : Signal déclenché lorsque la distance de l'objet pivot de la caméra est supérieur ou égale à la première valeur du champ *Gap*.
  - » **Node node** : Contient le noeud où ce signal a été émit.
  
- + **flat\_angle (data)** : Signal déclenché lorsque l'angle de rotation de la caméra est plat. Cet événement renvoie un dictionnaire contenant les clés suivantes :
  - » **Node node** : Contient le noeud où ce signal a été émit.
  - » **float angle** : Contient l'angle de rotation de la caméra. Notez que cet angle peut changé de signe en fonction du sens de rotation de la caméra.