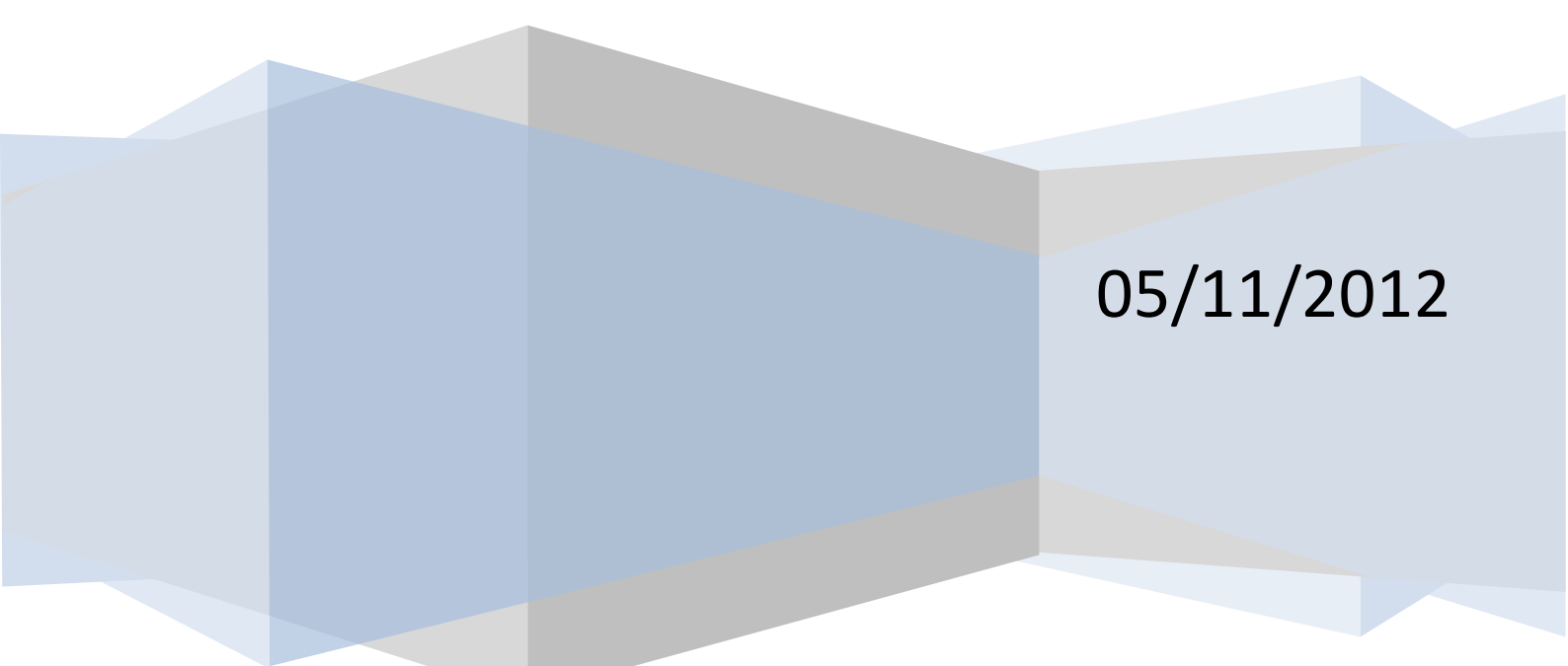


NeoAxis engine

HUDs et GUI

MONIER Vincent



05/11/2012

NeoAxis : Créer et éditer les HUDs

Exemple de HUD

Un HUD de présentation complet existe sur NeoAxis. Il est accessible dans le menu principal de la démo, en cliquant sur le bouton « Demo GUI ».



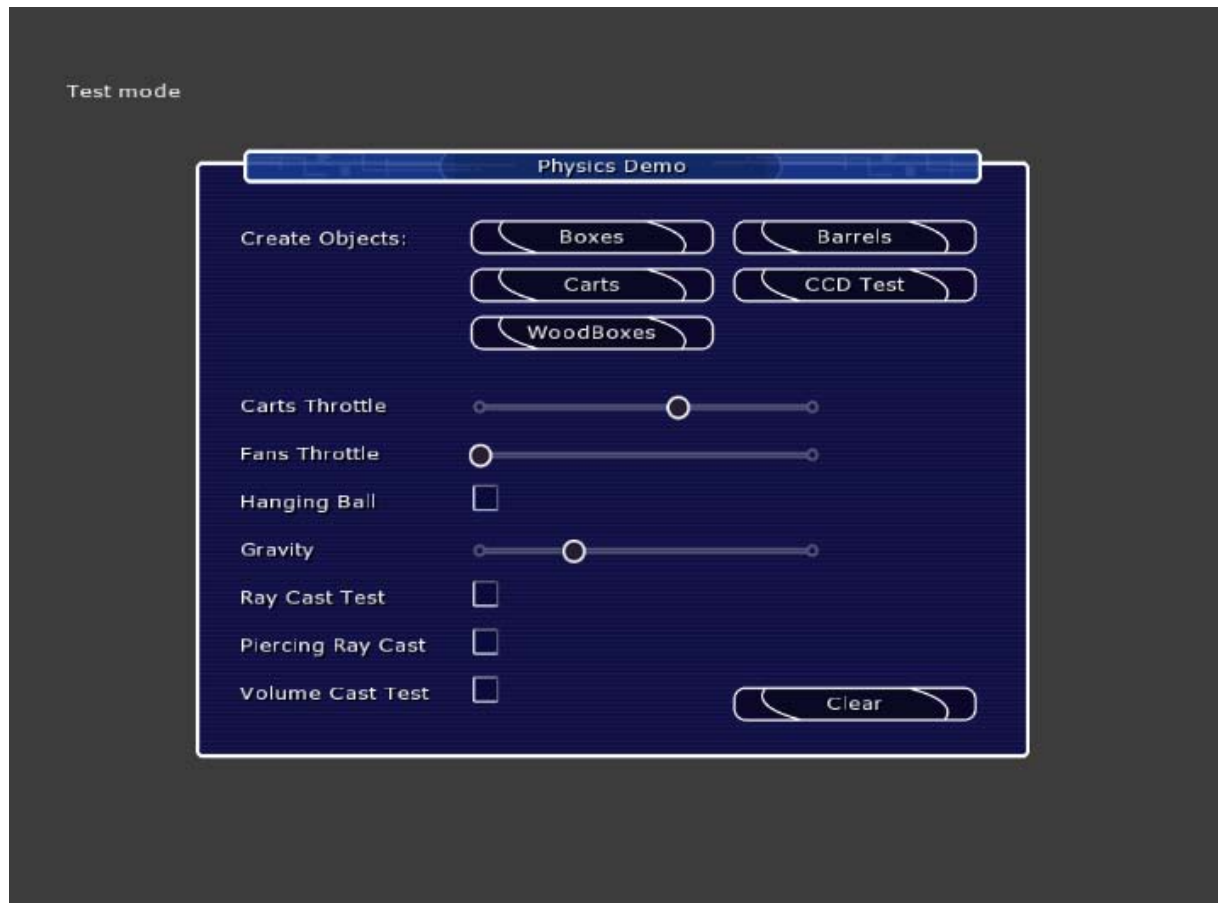
Dans la démo, le bouton « GUI Test » permet d'afficher un HUD contenant tous les éléments possibles des HUDs.



Le HUD d'exemple regroupe tous les types d'éléments pouvant être utilisés dans un HUD.

Le principe de templates

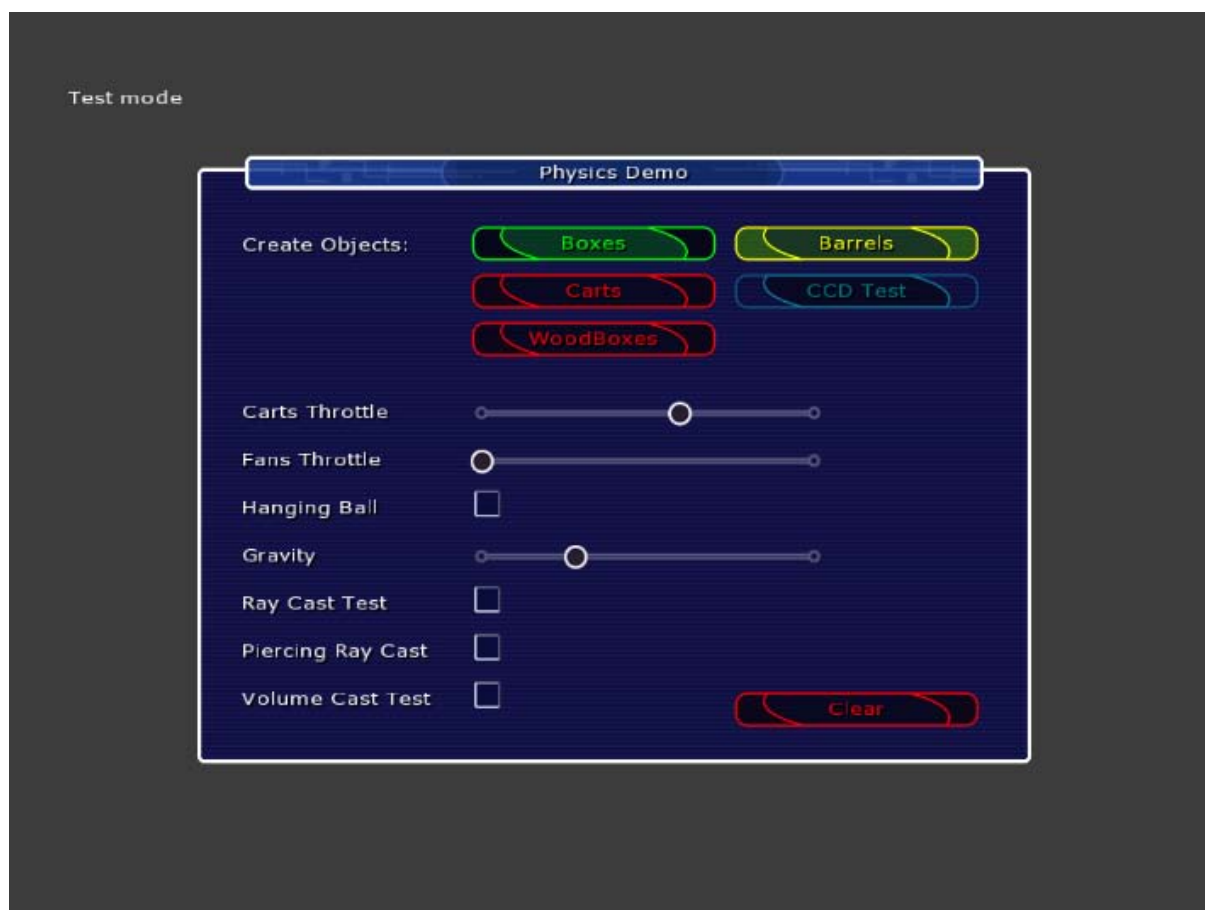
Des éléments par défaut existent, et peuvent être réutilisés (« GUI »). Ils permettent, par exemple, de créer un template (« patron ») pour certains éléments (boutons par exemple), avec un principe similaire aux feuilles de style (CSS ou styles Word) : si on édite le « GUI » inséré dans un HUD, alors toute inclusion de ce GUI sera modifiée.



Le menu de la zone de démonstration des effets physiques de NeoAxis.



Le template (GUI) du bouton utilisé dans le menu précédent.



Le menu précédent, après avoir modifié uniquement le template du bouton.

Il est donc conseillé de se créer un dossier de templates (similaire au dossier « Controls » déjà existant, qui contient les éléments template des menus d'origine), et d'utiliser ces templates dans les HUDs.

Elements utilisables dans les HUDs

Les HUDs peuvent utiliser les éléments suivants :

- Bouton (« Button », en bas à droite)
- Case à cocher (« Check Box », en haut à gauche, similaire aux pages web)
- Liste de sélection (« Combo Box », similaire aux pages webs)
- Objet (« Control », permettant l'affichage d'une image ou pour créer un élément masqué, « hidden »)
- Champ texte (« Edit Box », similaire aux champs textes des pages web)
- L'afficheur des stats du moteur (« Engine Profiler Window », réservé)
- Liste de sélection (« ListBox », dont les éléments sont bien sûr au choix du créateur)
- Barre de défilement (« Scroll Bar », permettant de créer des jauges par exemple)
- Onglets (« Tab Control », qui permet de regrouper des éléments dans un même « groupe » pour permettre un affichage par onglet par exemple)
- Textes (« Text Box », simples ou avec des effets spéciaux si on le souhaite)
- Videos (« Video Box », affichant une vidéo dans le HUD ; la vidéo peut se contrôler par les boutons du HUD)
- Un autre menu (« GUI », il s'agit d'un autre menu *.gui pouvant être un template)

Il est fortement conseillé de créer un template pour chaque élément, tout comme cela a été fait dans le dossier « Controls » donnant les templates par défaut.

Pour insérer un élément, il suffit d'éditer le HUD (double-clique sur le nom du HUD dans « Resources »), puis de cliquer droit dans le HUD, et de choisir « Add new control ». On peut alors choisir un élément de base (l'un de ceux-ci-dessus) ou un GUI (template).

Chaque élément possède des propriétés, modifiables dans le menu « Properties » (à droite, en dessous de « Control Hierachy »). Ces propriétés sont génériques ou spécifiques pour le type de cet élément.

Propriétés des éléments

Propriétés génériques

Background

Nom	Type	Description
BackColor	RGBA	Couleur d'arrière plan de l'élément
BackTexture	Path	Chemin vers une image à afficher en arrière-plan
BackTextureCoord	XYUV	Coordonnées de la texture à extraire pour l'arrière-plan (cf bas de page)
BackTextureTile	bool	La texture extraite peut être alignée sur la taille de l'objet

General

Nom	Type	Description
ColorMultiplier	RGB(A)	Multiplie les composantes de l'objet par cette valeur avant d'afficher l'objet ; permet de modifier la teinte et la luminosité
CopyTextFromParent	Bool	La valeur du text est héritée du parent (utile pour le template du bouton)
Enable	Bool	L'objet est activable
MouseCover	Bool	?
Name	Text	Le nom de l'objet, utile si l'objet doit être traité par un script dans l'environnement 3D
Text	Text	Le texte contenu dans l'objet
TopMost	Bool	?
Visible	Bool	L'objet est visible ou non

Layout

Nom	Type	Description
Anchor	List	?
HorizontalAlign	List	L'alignement horizontal de l'objet par rapport à son parent : <ul style="list-style-type: none"> - Left, les bords gauches de l'objet et du parent coïncident - Center, les centres de l'objet et du parent coïncident - Right, les bords droits de l'objet et du parent coïncident
LockEditorResizing	Bool	?
Position : Type	List	Le type de positionnement :

		<ul style="list-style-type: none"> - Parent, la valeur [0..1] est définie par rapport au parent - Screen, la valeur [0..1] est définie par rapport à l'écran - ScaleByResolution, la valeur est définie en pixels sur la base d'un écran 1024x768 - Pixel, la valeur est définie en pixel
Position : Value	Vec 2D	La valeur (0 0) réfère toujours au coin haut-gauche. La valeur du point bas droit dépend du type de positionnement (1 1 pour « Parent », 1 1 pour « Screen », 1024 768 pour « ScaleByResolution » et X Y pour Pixel, sous condition que la résolution d'écran soit X Y, ce qui n'est pas assuré car chaque client peut définir sa propre résolution en fonction de son écran).
Size : Type	List	Idem que « Position : Type »
Size : Value	Vec 2D	Idem que « Position : Value »
VerticalAlign	List	Similaire à « HorizontalAlign », mais pour la verticale : Top, Center ou Bottom.

Bouton

Le bouton réagit lorsque l'utilisateur passe dessus (souris) et clique. L'action de clic peut déclencher un événement et une action. L'action peut porter sur le menu lui-même (bouton fermant le menu par exemple) ou sur l'environnement (bouton commandant l'apparition d'un objet dans le décor par exemple).

Nom	Type	Description
Active	Bool	Le bouton est actif et cliquable
ClickMask	Path	Définit la zone active du bouton*
ClickMaskTextureCoord	XYUV	Coordonnées de la texture de Click
SoundClick	Path	Son à jouer (ogg de préférence) lorsque l'utilisateur clic sur le bouton
SoundMouseIntoArea	Path	Son à jouer quand l'utilisateur passe sa souris sur le bouton

*Zone active du bouton : Par défaut, un bouton occupe l'intégralité du rectangle de l'objet ; autrement dit, si le bouton possède les paramètres « Size : Type = Pixel » et « Size : Value = 100 30 », alors si la souris entre dans le rectangle de 100x30 pixels, le moteur considère qu'elle passe sur le bouton. ClickMask permet d'utiliser une image pour modifier la forme de cette zone.



Par défaut, toute la zone définie par la propriété « Size » du bouton est cliquable.



Le paramètre « ClickMask » permet de définir quelle est la zone effective du bouton. La zone blanche sera cliquable, la zone noire ne sera pas cliquable (la souris ne sera considérée comme étant « sur » le bouton que si elle est dans la zone blanche).



Alors, le bouton n'est plus cliquable que dans la zone définie par l'image ClickMask.

CheckBox

La checkbox fonctionne sur le même principe que les cases à cocher des formulaires web.

Nom	Type	Description
Checked	Bool	Indique si la case est cochée ou non. Le template permet de définir une forme personnalisée pour la case cochée et non cochée.

ComboBox

Une ComboBox est une liste déroulante similaire aux formulaires web.

Nom	Type	Description
TextIfNotSelected	Text	Texte à afficher si aucun élément n'est sélectionné.

Control

Les objets de type « Control » n'ont aucun paramètre particulier.

Ils permettent simplement l'affichage des éléments communs à tous les objets (issus de General et de Background, qui peut être une image : le control sert alors à afficher une simple image)

Edit Box

Une Edit Box est similaire aux champs textes sur le web. Les paramètres spécifiques sont :

Nom	Type	Description
MaxCharacterCount	Int	Le nombre maximum de caractères que l'on peut entrer dans ce champ texte.

Si le texte entré dans un champ texte est plus long que la zone d'affichage, le texte est tronqué (il ne « sort » pas de la zone graphique définie par l'objet) :



Si le texte est trop long pour la EditBox, il est tronqué à gauche : il ne déborde pas de la zone d'affichage.

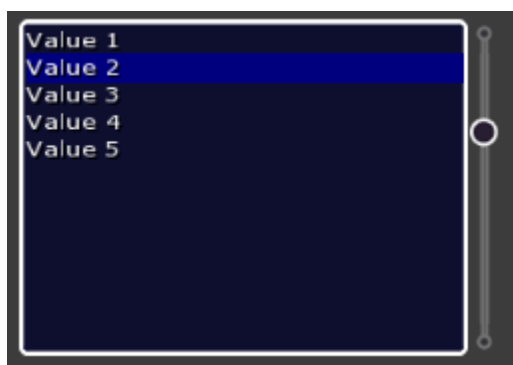
Scroll Bar

La scrollbar est similaire à un curseur « slide » (à ne pas confondre avec une barre d'ascenseur comme sur les pages webs). Un curseur slide est un curseur glissant le long d'un axe : sa position définit la valeur associée à ce curseur. Les paramètres spécifiques à une Scroll Bar sont les suivants :

Nom	Type	Description
Value	Double	La valeur courante du slide ; cette valeur est mise à jour quand l'utilisateur déplace le curseur
ValueRange	XY	La plage de valeurs du slide, la première valeur est le minimum, la seconde est le maximum ; le slide est linéaire (si le curseur est au milieu, la valeur est la moyenne de X et de Y)
Vertical	Bool	Indique si le slide se fait selon la verticale (true) ou l'horizontale (false)

List Box

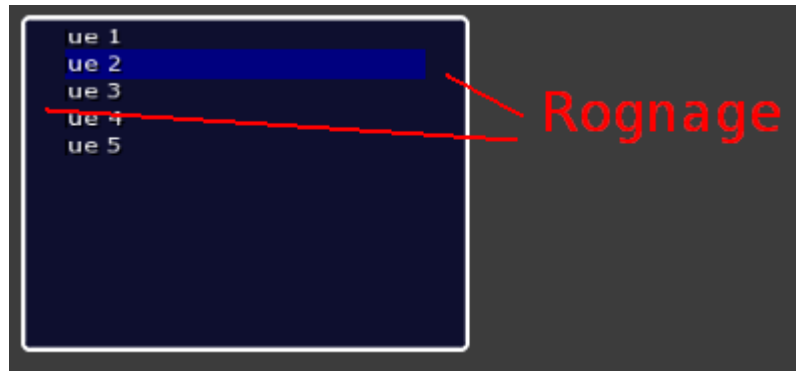
Une listbox est un ensemble de valeurs cliquables.



Un exemple de listbox

Les paramètres spécifiques sont les suivants :

Nom	Type	Description
ClipRectangleBorders : Type	List	Similaire à « Position : Type » dans « General »
ClipRectangleBorders : Value	XY	La valeur définit le rognage gauche/droit et le rognage haut/bas du cadre intérieur à la list box.
SelectedIndex	Int	Le numéro de la ligne que l'utilisateur a sélectionné dans la list box



Les valeurs dans « ClipRectangleBorders » définissent le rognage intérieur de la listbox.

Tab Control

Permet de construire une structure à onglets, jusqu'à 20 pages (de 0 à 19 inclus). On peut également utiliser les TabBox pour créer un ensemble de N pages (jusqu'à 20 donc), affichée une à une, avec un ensemble de boutons permettant de sélectionner la page à afficher (on peut donc ne pas présenter cela sous la forme d'onglets). Les paramètres spécifiques sont les suivants :

Nom	Type	Description
PageButtonsOffset : Type	List	Similaire à « Position : Type » dans « General »
PageButtonOffset : Value	XY	La valeur de l'offset des onglets ou boutons d'affichage des pages ; chaque bouton sera séparé du précédent par les valeurs horizontale et verticale indiquées
PageButtonsPosition : Type	List	Similaire à « Position : Value » dans « General »
PageButtonsPosition : Value	XY	Position du premier bouton d'affichage
SelectedIndex		Index de la page affichée (sélectionnée)

Text Box

Une text Box affiche un texte donné, avec une image de fond si besoin (définie par les paramètres généraux, dont « Background »). Les paramètres spécifiques sont :

Nom	Type	Description
Text Box		
AutoSize	Bool	La taille de l'objet est automatiquement calculée en fonction du texte contenu ; cela évite que le texte ne déborde de l'objet texte
Font	List	Le type de police à utiliser ; il est possible de créer une police personnalisée, à partir d'une image par exemple, ou d'un fichier de font (fichier de police de caractère). On peut également définir la hauteur de cette police.
SupportLocalization	Bool	Active la localisation (traduction) pour ce texte ; la localisation se

		fait via les fichiers « Bin/Data/Languages/*/*.language »
TextColor	RGB(A)	Couleur du texte (tout le texte est d'une seule et même couleur)
TextHorizontalAlign	List	Alignement horizontal du texte (Left, Center ou Right)
TextOffset : Type	List	Similaire à « Position : Type »
TextOffset : Value	XY	Valeur de positionnement du texte dans l'objet
TextVerticalAlign	List	Type d'alignement vertical (Top, Center ou Bottom)
Text Shadow		
Shadow	Bool	Active l'ombrage du texte
ShadowColor	RGB(A)	Couleur de l'ombrage du texte
ShadowOffset : Type	List	Similaire à « Position : Type » de « General »
ShadowOffset : Value	XY	Position de l'ombrage du texte
Word Wrap		
AlignByWidth	Bool	L'alignement des mots dans une ligne est de type « Justifié » : le premier mot de chaque ligne est tout à gauche, le dernier mot finit tout à droite
VerticalTextIndentation	Int	Hauteur des lignes du texte, définies par rapport au type de « Position » dans « General » (si « Position : Type » est « Parent », alors la hauteur de ligne sera par rapport à la taille du parent)
WordWrap	Bool	Active le retour à la ligne (true) pour éviter que le texte ne déborde de la zone d'affichage

Video Box

Permet d'insérer une vidéo (au format ogv) dans le HUD. Les paramètres spéciaux sont :

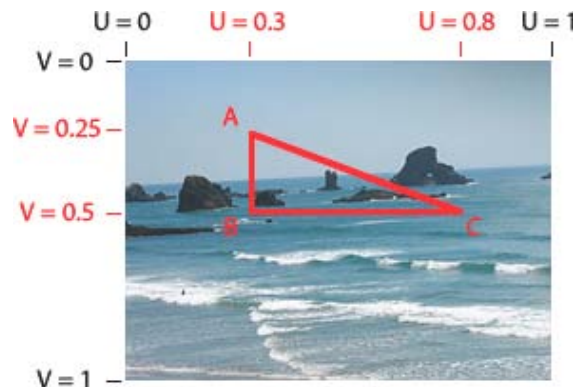
Nom	Type	Description
FileName	Path	Chemin de la vidéo à afficher
Loop	Bool	La vidéo est jouée en boucle (true)
Pause	Bool	La vidéo peut être mise en pause par un clic dessus
Volume	Double	Volume sonore pour la video

Types de données

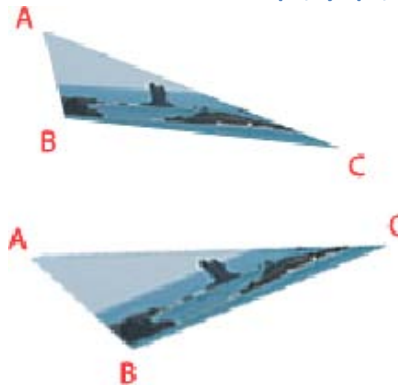
Les types de données utilisables sont les suivants :

Type	Description
Double / Single	Nombre décimal, signé
Int / integer	Nombre entier, signé
XY / vec 2D	Position/taille en 2D ou vecteur 2D
XYZ / vec 3D	Position/taille en 3D ou vecteur 3D
UV	Coordonnées 2D pour les textures, entre (0,0) et (1,1)
XYUV	Taille et position d'une zone de texture (XY représente el point de coordonnées minimales, et UV celui de coordonnées maximales)
RGB	Une couleur Rouge Vert Bleu, chaque composante est entre 0 et 255 inclus
RGBA	Idem que RGB, avec un chanel alpha entre 0 (transparent) et 255 (opaque)
RGB(A)	Idem que RGBA, sauf que la valeur alpha est facultative (255 par défaut)
Quad / vec 4D	Un quaternion pour les rotations 3D

Text	Un texte
Shader/HighMaterial	Un nom de shader (HighMaterial)
List	Une liste de valeur est acceptée, comme un « enum » / liste de sélection
Path	Un chemin vers un fichier
Bool	Un booléen, valant vrai (« true ») ou faux (« false »)



Les coordonnées UV d'une texture sont sur la base (0;0)..(1;1) partant d'en bas à gauche.



La texture est alors posée (avec une déformation si les coordonnées UV sont mal choisies) sur le triangle 2D ou 3D.

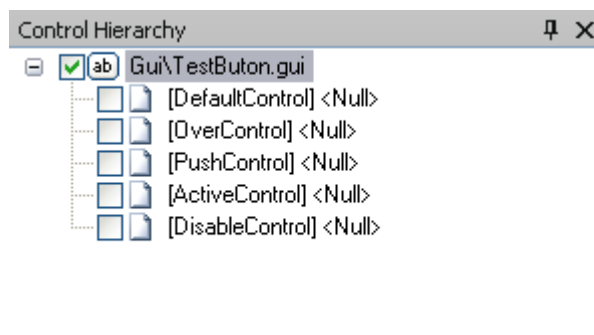
Créer les templates

Il est conseillé de créer un template pour chaque type d'objet. La procédure est la suivante :

- Créer un GUI pour ce template (cela revient à créer un nouveau menu dédié à ce template) :
 - Il est conseillé de rassembler tous les GUIs des templates dans un seul dossier
 - La création de ce GUI peut se faire par un clic droit sur le dossier dans la vue « Resource », puis « New Resource », « UI Model »
 - Renseignez le nom
 - Sélectionnez, comme base, le bon élément de « Base Element » (si vous créez un template de bouton, sélectionnez « Base Template / Button ») ; mais si vous choisissez, comme base, un GUI existant (exemple : « Controls/DefaultButton.gui »), alors cela revient à faire de votre nouveau GUI un « enfant » (au sens héritage objet) de DefaultButton.gui : si DefaultButton.gui est modifié, votre template le sera aussi
- Suivez les étapes ci-dessous, correspondantes au template que vous créez
- Sauvez
- Vous pouvez alors réutiliser ce template dans les autres menus

Bouton

Votre zone « ControlHierarchy » possède les éléments suivants (si vous avez sélectionné « Base Template / Button » lors de la création du GUI) :



Les éléments du « control hierarchy » d'un bouton, à partir de Base Template/Button.

L'élément racine est le GUI que vous avez créé (ici, « GUI\TestButton.gui »). Ce bouton contient des contrôles (éléments enfants dans la hiérarchie) :

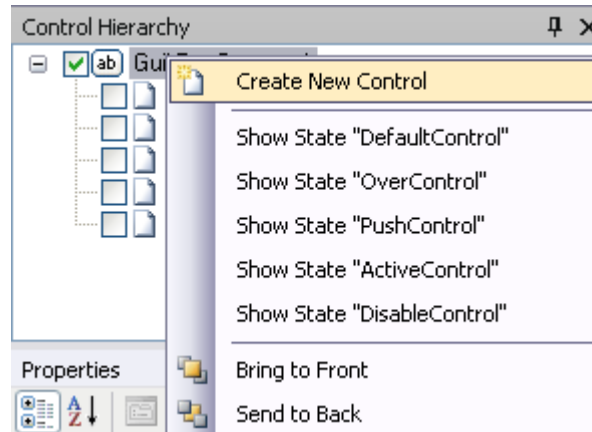
Nom du contrôle	Evènement associé
Default Control	Le bouton est cliquable
Over Control	La souris est au-dessus du bouton (ClickMask)
Push Control	Le bouton est cliqué (clic gauche enfoncé pendant que la souris est au-dessus du bouton)
Active Control	Le bouton est activé mais non-cliquable
Disable Control	Le bouton est désactivé et non-cliquable

Un bouton peut donc être cliquable (le cycle est alors DefaultControl → souris sur bouton → OverControl → Clic enfoncé → PushControl → clic relâché → DefaultControl, avec si besoin, l'exécution d'une commande liée au bouton et définie dans le script de la carte 3D), ou bien non cliquable. Dans ce cas, le bouton a deux états :

- non-cliquable car désactivé (DisableControl), c'est le cas des boutons « grisés » usuels
- non-cliquable car activé (ActiveControl), cela revient à dire que le bouton est forcément « enfoncé » : à l'instar d'un bouton non-cliquable car il est inactif, ce bouton est non-cliquable car il est « coincé » en position « active ». Cela peut servir, par exemple, dans le cas où un système est en route et ne peut pas être arrêté : le bouton « Marche/arrêt » est verrouillé sur la position « Marche ».

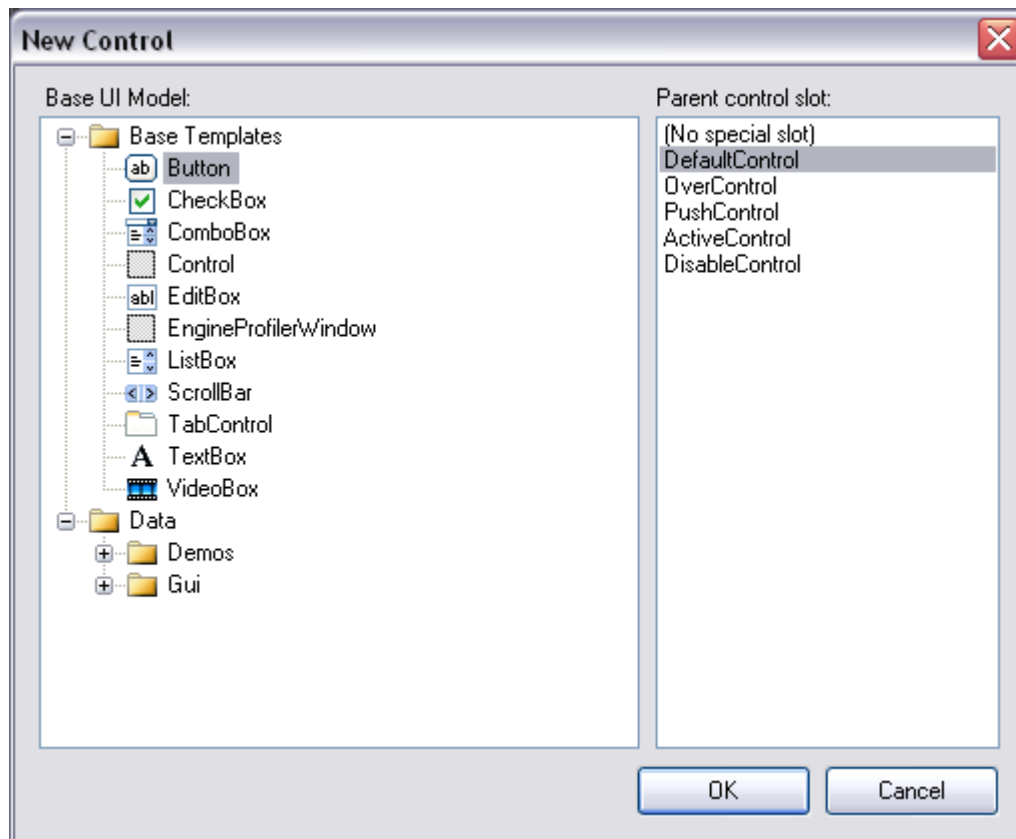
Par défaut, ces éléments sont null (vides). Il va falloir les définir un à un pour créer le template.

Pour ajouter un élément à un contrôle, sélectionnez le bouton (cochez la case dans le « Control Hierarchy » devant « GUI\TestButton.gui » ou double-cliquez sur le GUI dans la fenêtre d'affichage), puis faites un clic droit et cliquez sur « Create New Control » :



Cochez la case pour sélectionner un élément, et cliquez droit sur son nom pour ajouter un enfant dans la hiérarchie.

Dans la zone « Parent control slot », sélectionnez le contrôle que vous voulez définir (commencez, logiquement, par le DefaultControl). Dans « Base UI Model », sélectionnez le type d'élément à ajouter à ce contrôle (par exemple, un texte).



Sélectionnez le parent dans la zone de gauche, et sélectionnez la base (la « classe » de l'objet) dans le panneau de droite.

L'élément est rajouté en « enfant » du contrôle choisi. Si vous n'avez pas choisi de contrôle (« No special slot »), alors l'élément est ajouté en enfant de l'élément sélectionné précédemment (le bouton) et apparaîtra quelque soit le control active (que le bouton soit actif ou non, cliqué ou non etc). Vous pouvez définir des textures de fond sur les éléments des différents contrôles, pour que le bouton utilise ces textures suivant les évènements. Inspirez-vous de « DefaultButton.gui ».

Dans la suite, seuls les types de contrôles seront détaillés : le principe pour ajouter des éléments aux contrôles est inchangé.

Check Box

Les contrôles sont les suivants (ils doivent avoir un seul enfant, de type « Button ») :

Nom du contrôle	Evènement associé
Checked Button	La case est cochée, si l'utilisateur clique dessus pour la décocher, le contrôle « Push Control » du bouton associé à « Check Button » est appelé
Unchecked Button	La case est décochée, même principe que précédemment, mais dans le cas où l'utilisateur clic pour cocher

Combo Box

Les contrôles sont les suivants (de type Button ou Control) :

Nom du contrôle	Evènement associé
Item Button	Le bouton central, contenant la valeur choisie par l'utilisateur
Down Button	Bouton pour afficher le contenu de la liste déroulante
Down List Control	Zone affichée lorsque l'utilisateur déroule le contenu de la liste ; cette zone correspond à la liste des valeurs pouvant être choisies

Le contenu des choix de la liste n'est pas défini ici : il le sera dans le fichier script associé à l'environnement 3D.



Structure des contrôles d'un bouton.

Control

Cet élément n'a pas de contrôle, vous pouvez néanmoins ajouter des éléments directement à la racine de l'objet (« No Special Slot »). Si vous ouvrez « DefaultControl.gui », vous pouvez voir que, d'origine, le « DefaultControl » sert à créer un cadre sous la forme d'une matrice de 3x3 éléments, dont les bords sont le cadre, et le centre le contenu du contrôle.

Edit Box

Les contrôles sont les suivants (type Text) :

Nom du contrôle	Evènement associé
TextControl	Le texte associé à l'Edit Box

List Box

Les contrôles sont les suivants (note : l'ItemButton est de type Button, et ce Button possède lui-même des contrôles) :

Nom du contrôle	Evènement associé
ItemButton	Bouton associé aux éléments de la liste
ScrollBar	Barre (verticale généralement) affichée si la liste est trop longue pour tenir dans le cadre

Scroll Bar

Les contrôles sont les suivants :

Nom du contrôle	Evènement associé
BarControl	Le curseur associé à ce slide

Tab Control

Les contrôles sont les suivants (PageButton doit être de type Button, mais les pages peuvent être de types quelconque) :

Nom du contrôle	Evènement associé
PageButton	Le bouton de base pour la navigation entre onglets ; le même bouton est répété pour chaque onglet
PageN (N=0..19)	Les différentes pages ; il est recommandé d'utiliser un type conteneur (Contrôle ou autre GUI contenant la page complète)

Text Box

Aucun contrôle n'existe pour cet élément.

Video Box

Aucun contrôle n'existe pour cet élément.