Fugui Roadmap



* Components :
* Button
* ~~Ajouter les couleurs d’action dans les styles~~
* ~~Passer les sizes en flag (auto ou full)~~
* ~~Ajouter un flag (actionType : Default, selection, states)~~
* Labels
* ~~SmartText semble mal se wrapper (padding)~~
* ~~Ajouter un flag (stateType : Default, selection, states) maybe same actionType~~
* Checkbox
* ~~Pas besoin de style~~
* ~~A tester mais les couleurs disabled doivent foirer (frameBg or CheckMark selon isChecked)~~
* RadioButton
* OK
* Radiobutton group
* Ajouter des helper
* ~~Pas besoin de style~~
* Slider
* ~~Pas besoin de style~~
* ~~Flag pour la position du drag (left, right, none => default is right)~~
* Drag
* ~~Pas besoin de style~~
* ~~Maybe check le padding quand on met du text devant les drags~~
* Flag pour mettre le text à droite ou à gauche
* Combobox
* OK
* Collapsable
* Flag pour choisir si indentation ou pas
* Pas besoin de Style
* Text Input
* Ajouter des flags (surcouche aux flags ImGui car on veut en supprimer)
* Password
* CharTypes
* Uppercase
* Pas besoin de style
* Image
* OK
* ColorPicker
* Pas besoin de Style
* Flag pour afficher la prev en caré ou rect
* Ajouter les palets
* padding
* ToolTips
* Ajouter les Styles de textes
* ButtonsGroup
* Ajouter les styles par bouton
* PathField
* Ajouter le SaveFile
* Changer les ‘…’ en icon, selon le type de path
* Modal
* Ajouter les styles de button et text
* Ajouter des methodes pour aider (string title + body, errorBox, infoBox, etc)
* Gestion des couleurs des boutons
* Icons et couleur pour les statesBox
* Corriger les erreurs de padding
* Utiliser l’alignement des Panel pour dessiner les boutons à droite (si implémenté)
* ProgressBar
* Finir d’implémenter
* Notify
* A faire
* Tabs
* A faire
* Grid Item (pour les auto col grids => comment afficher un item dans un context x)
* A faire

* Global :
* Passer les bool de comportement en flags (num)
* Gestion de plusieurs fonts d’icons (icons par defaut + icons custom)
* Gestion generique du dockspace
* Déclarer des dockespaces
* Editeur de layout + déclarer layout et window name comme les layers
* Sans layout ?
* Comment stocker les UIWIndow/Definitions
* BitMask déclaré dans l’éditor
* Contraindre la duplication d’UIWindow (dans la def => flag)
* Contrainte de docking generique (eg : A dans B mais pas dans C)
* Ajouter des 3DContainer (pour la VR par exemple)
* Possibilité d’override les Styles des themes par context (replace or add)
* Re-Anchor les Overlay (context menu)
* Collapse ou réaligner les Overlay qui se chevauchent
* Ajouter les inputs keyboard par windows (opti, par container / context)
* Ranger les flags (file.s / folder)
* UIElementAnimationData doit être plus générique (pas seulement des bool) et utilisé dans l’anim des modals
* Les variables d’annimation doivent être passées dans le theme
* ImGui ne doit pas être accessible (ASMDEF => pas évident, dépendances cycliques à régler, maybe archi à revoir ades endroits si besoin)
* Wrapper les methodes ImGui utiles (dummy, cursor, avail)
* Ajouter la notion d’alignement (cursor custom qui dessine de gauche à droite ou de droite à gauche) dans les Panels
* Possibilitée d’ajouter des components dans Layout et Grid depuis l’exterieur (partial)
* Passer UILayout en partial, couper les components dans des fichiers qui l’extends (pareil pour UIGrid)
* Unifier les enums de colors dans les themes
* ///<summary> toutes les methodes et commenter le code