

SOPHYA, PI, PIAPP

EVOLUTION EN COURS ET PRÉVUES

R. ANSARI, AVRIL 2022

- **SkyMap** : Ajout de classes de projection sphériques ,
Molleweide, Sinus, Mercator ...
- **NTools** : classes de résolution numérique
SecantNumSolver et résolution équation-diff ODE_RK4_Solver
(en cours)
- **SysTools** : nouvelles classes socket (UDP) - Amélioration
de la classe d'interpréteur de base Commander - prise en
charge de lua dans piapp
- **ShapeLib** : classes adaptateur pour l'utilisation de la
bibliothèque shapelib (lecture des fichiers de données
géographiques - classes SLI_SHP_Interface ,
SLI_DBF_Interface (Merci Guy !)

Autres - pas fait du tout




- **SkyT** : Jouvence du module, y inclure les classes SED, Filtres, Schechter ... que j'ai développées par ailleurs
- **HexagonalMapHisto** : Une sorte de carte / Histo2D à pixels hexagonal, à développer
- **TArray** : Optimisation des opérations sur tableaux en profitant des techniques mises en place par Hadrien pour le corrélateur ...
- **TArray** : Interface avec Eigen - un petit programme test existe

PI / piapp

- Couverture complète 95%-99% de PI en version PIWxWidget (et PIXtMotif)
- Nouvelles classes : PIConsole2 , PIRoundGauge, PITableView
- Export Window/Widget en image (service de wxWidget) PNG, JPEG
- Nouvelles couleurs, une dizaine de tables de couleurs extraites de python/Matplotlib (Merci Olivier !)
- Problème de tracé dans les événements autre que PaintEvent (version wxWidget) a été contourné

PI / piapp : futur incertain



- Séparation de l'environnement piapp (NamedObjMgr, Interpreteur, commandes ...) de PI : permettrait à Guy de construire une appli piapp avec ses outils graphiques. Nécessite la définition de quels classes interface 
- Amélioration de piapp [tracés graphiques, commandes, amélioration interface manipulation des fichiers FITS et HDF5 ...]
- Utilisation de Scintilla Text Editor ?

LA SUITE (COURT TERME)

Slide d'Avril 2020

- Installer et tester PI et piapp version wxWidget sur Mac OS (Cocoa) et X11 (GTK)
- identifier les problèmes - corriger ou en faire part à l'équipe de développement wxWidget
- Partir sur la base du makefile ad-hoc ds PI/PIWxW
- Restructurer le code PI en **PIBase**, **PIX11Motif**, **PIWxW** et **PI**
- **PIBase** regrouperait classes support de base, ex *PIGraphicAtt*, les classes définissant l'interface xxxGen, ex *PIWdgGen*)
- **PIX11Motif** (ou PIX11) l'implémentation des classes xxxGen avec X11/Motif
- **PIWxW** l'implémentation des classes xxxGen avec wxWidget
- **PI** regrouperait alors uniquement les classes qui sont génériques, comme *PIScDrawWdg*
- les répertoires **Plext**, **PIGcont**, **ProgPI** resteraient inchangés
- Il faudra adapter les Makefiles et script ...



LA SUITE (MOYEN TERME)

Slide d'Avril 2020

- Voir comment intégrer le vrai 3D (PIOpenGL ou ??) ?
- Compléter les fonctionnalités non encore implémentés ds PIWxW (Copier/coller, drag & Drop (Mac OS X uniquement) ... ✓
- Construire une application complète, installable sur MacOS (Guy) et Linux
- Explorer l'utilisation d'objets wxWidget sans les implémenter en version X11 : Affichage HTML (wxHtmlWindow)?, affichage de Tables (wxGrid) ✓
- Améliorer / optimiser l'affichage d'images (code de l'interface des tableaux P2DArrayAdapter , PImage) , décodage coordonnées astro (fait en partie - intégration WCSTools) ... ✎
- Améliorer / compléter l'affichage des cartes sphériques (projections ...)
- Intégration de l'interpreteur LUA ✎
 - (voir SophyaProgs/Eval/DivTst/luatst.cc) ✓