OPENWOR v1.9

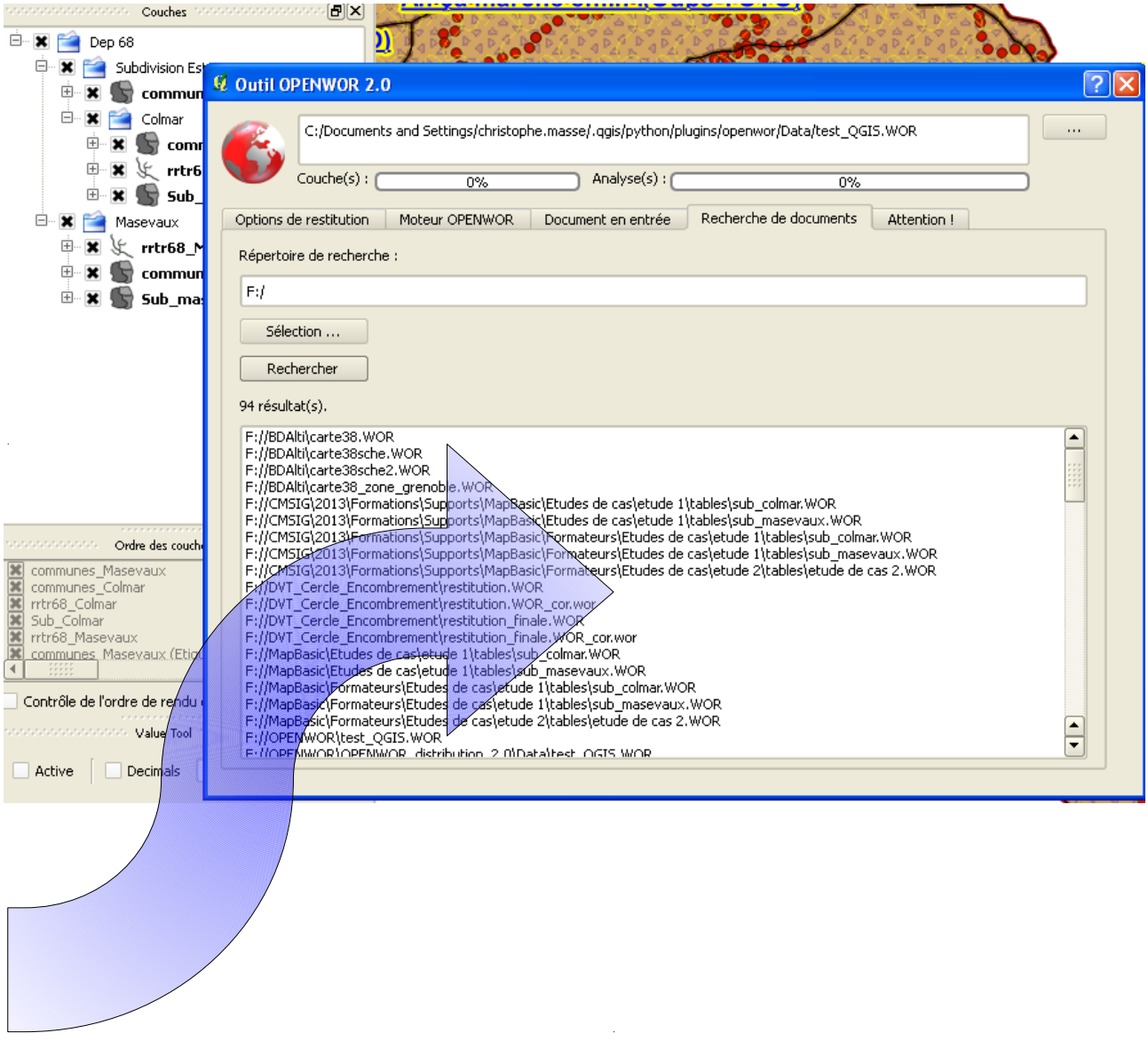

Couches
Dep 68
Colmar
Masevaux
Outil OPENWOR 2.0
C:/Documents and Settings/christophe.masse/.qgis/python/plugins/openwor/Data/test_QGIS.WOR
Couche(s) :
0%
Analyse(s) :
0%
Options de restitution
Moteur OPENWOR
Document en entrée
Recherche de documents
Attention !
Répertoire de recherche :
F:/
Sélection ...
Rechercher
94 résultat(s).
F://BDAlti\carte38.WOR
F://BDAlti\carte38sche.WOR
F://BDAlti\carte38sche2.WOR
F://BDAlti\carte38_zone_grenoble.WOR
F://CMSIG\2013\Formations\Supports\MapBasic\Etudes de cas\etude 1\tables\sub_colmar.WOR
F://CMSIG\2013\Formations\Supports\MapBasic\Etudes de cas\etude 1\tables\sub_masevaux.WOR
F://CMSIG\2013\Formations\Supports\MapBasic\Formateurs\Etudes de cas\etude 1\tables\sub_colmar.WOR
F://CMSIG\2013\Formations\Supports\MapBasic\Formateurs\Etudes de cas\etude 1\tables\sub_masevaux.WOR
F://CMSIG\2013\Formations\Supports\MapBasic\Formateurs\Etudes de cas\etude 2\tables\etude de cas 2.WOR
F://DVT_Cercle_Encombrement\restitution.WOR
F://DVT_Cercle_Encombrement\restitution.WOR_cor.wor
F://DVT_Cercle_Encombrement\restitution_finale.WOR
F://DVT_Cercle_Encombrement\restitution_finale.WOR_cor.wor
F://MapBasic\Etudes de cas\etude 1\tables\sub_colmar.WOR
F://MapBasic\Etudes de cas\etude 1\tables\sub_masevaux.WOR
F://MapBasic\Formateurs\Etudes de cas\etude 1\tables\sub_colmar.WOR
F://MapBasic\Formateurs\Etudes de cas\etude 1\tables\sub_masevaux.WOR
F://MapBasic\Formateurs\Etudes de cas\etude 2\tables\etude de cas 2.WOR
F://OPENWOR\test_QGIS.WOR
communes_Masevaux
communes_Colmar
rrtr68_Colmar
Sub_Colmar
rrtr68_Masevaux
Contrôle de l'ordre de rendu
Value Tool
Active
Decimals

Table des matières

Version date	Auteur	Remarques
15/05/2012	Christophe MASSE	Rédaction du document
12/06/2012	Christophe MASSE	Ajout des fonctions inhérentes aux groupes MapInfo
05/07/2012	Christophe MASSE	-
15/01/13	Christophe MASSE	Mise à jour des documents de l'extension

Documentation OPENWOR

1 Les classes :

1.1 UI_Dialog_OW (object)

Classe maîtresse de l'extension.

Interface d'OpenWor.

Localisation : fopenwor.py

- ***def __init__(****self, iface****):***
- ***def setupUi(****self, Dialog****):***
- ***def CallQgsProjectionSelector(****self****): → SRSDialog()***
- ***def LoadWOR(****self****):***
- ***def FindComposer(****self, iMap****):***
- ***def MakeListTables(****self****):***
- ***def FixeInfosViewHTML(****self****):***
- ***def ReturnInfosLayerMap(****self****):***
- ***def MakeMap(****self****):***
- ***def ChangeSymbology(****self****):***
- ***def InterActiveMode(****self****):***
- ***def FixenProj4(****self, theProj****):***
- ***def FixeUniProjection(****self****):***
- ***def FixeTxtLabel(****self, Dialog****):***
- ***def MakeWarnings(****self****):***

1.2 SRSDialog (QDialog)

Classe d'appel boîte de dialogue choix de projection QGIS.

Interface d'OpenWor.

Localisation : fopenwor.py

- ***def __init__(****self, iface****):***

- ***def epsg(****self****):***

1.3 MeasureType(Type)

Classe de mesure et de conversion d'unité. Appelé depuis les fonctions de représentation des couches, notamment pour fixer les seuils de zoom :

- ***def GetValueZoom(****nValue, nUnits,nSizeMap, nSizeMapUnits****)***

Localisation : measured_units.py

Auteur : Georges SAKKIS

2 Les principales fonctions sur le document :

2.1 AnalyseWOR(self, nFile)

Fonction d'analyse du document sélectionné.

Localisation : fopenwor.py

Cette fonction permet l'alimentation des tableaux utilisés pour stocker et organiser les informations contenues dans le document chargé.

Liste des principaux tableaux :

- **tLayer** : contient les types des fichiers ressources ;
- **uLayer** : contient l'URL des fichiers ressources ;
- **tComposer** : contient les informations des mises en page ;
- **tBrowse** :contient les fenêtres données à restituer ;
- **tMap** : contient les informations des cartes à réaliser ;
- **tLayerMap** : contient les informations des couches des cartes à réaliser ;
- **tLayerMapAna** : contient les couches qui portent des analyses thématiques. Les informations des analyses thématiques sont stockées dans les entités **tMap** ;
- **tSelect** : contient les sélections du document ;
- **tJoin** : contient les jointures du document.

2.2 Localisation des fichiers ressources

Diverses fonctions pour la localisation des fichiers ressources sont présentes dans l'extension.

- ***def GetAdress(****nTable, nDir****):***
- ***def GetRasterFileMIG(****nTable****):***
- ***def GetRessourceFile(****nTable****):***
- ***def GetRasterFile(****nTable, IsSHP, IsASCII****):***
-

Localisation : fopenwor.py

3 Les fonctions de restitution dans l'interface :

3.1 InitAllTableau

Cette fonction permet la remise à blanc du contenu de tous les tableaux de stockage des informations du document chargé.

Localisation : fopenwor.py

3.2 Liste des sous-fonctions tableaux :

Les deux tableaux **tLayer** et **uLayer** sont directement alimentés depuis **AnalyseWOR**. Pour les autres tableaux, des fonctions sont dédiées à chaque type de contenu.

Localisation : fopenwor.py

- ***def MaketComposer(****self, zlistWOR, Rac, wpos****):***
- ***def MaketBrowse(****self, zlistWOR, Rac, wpos****):***
- ***def MaketMap(****self, zlistWOR, Rac, wpos****):***
- ***def MaketLayerMap(****self, zlistWOR, wpos, indexMap****):***
- ***def MaketSelect(****self, zlistWOR, Rac, wpos****):***
- ***def MaketJoin(****self, zlistWOR, Rac, wpos****):***

La fonction ci-dessous est nécessaire pour inverser l'ordre des couches thématiques (exploitation correcte des informations de visibilité).

- ***def ReOrgtLayerMapAna():***

Localisation : fopenwor.py

3.3 Liste des fonctions de restitution :

L'interface d'OpenWor comporte différents éléments pour la restitution d'information et la sélection d'entités (carte, mise en page).

Localisation : fopenwor.py

- ***def MakeHTMLView(****self, tempLayer, subtempLayer****):***
- ***def MakeTreeView(****self****):***
  - **def MakeParentItem(**self, zStr**):**
  - **def MakeItem(**DicGen, parentItem**):**

La fonction **FixeInfosViewHTML** permet de localiser un élément de la vue structurée (Treeview) dans la présentation HTML.

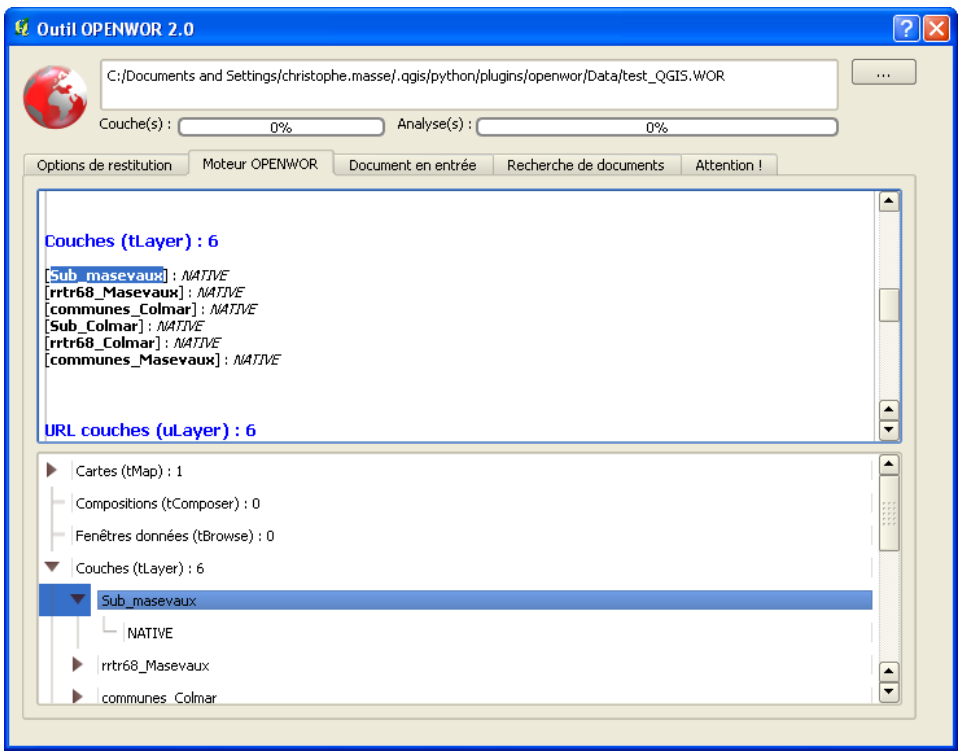
Cette vue structurée est mise en forme par les fonctions **MakeTreeView**, **MakeParentItem** et **MakeItem**. Ce découpage en fonctions élémentaires n'est dictée que par des objectifs de maintenance facilitée.

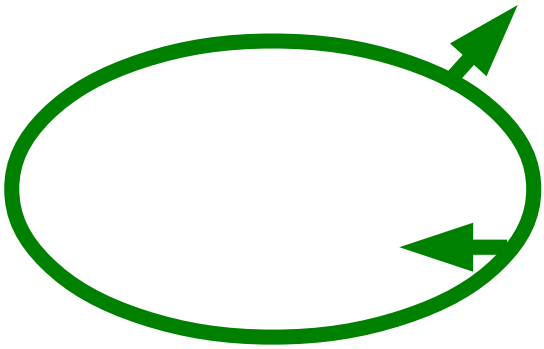
4 L'interface

La boîte de dialogue mise en place par la classe **UI_Dialog_OW** est composée de cinq onglets.

- **Options de restitution** : qui permet de sélectionner les entités à représenter et les options pour la représentation ;
- **Moteur OPENWOR** : qui contient les vues HTML et TreeView des entités OpenWor ;
- **Document en entrée** : qui permet de consulter le contenu natif du document MapInfo ;
- **Recherche de documents** : qui permet de lister les documents présents sur un disque, un répertoire (avec les sous-réperoires) ;
- **Attention !** : qui est à la fois une boîte “A propos ...” complémentaire de celle accessible depuis le menu “OpenWor 1.0” et une boîte d'avertissements sur certaines limites de la version utilisée.

Une interaction a minima a été mise en place sur le double clic d'un élément de la vue structurée pour rechercher les chaînes correspondantes dans la vue HTML.

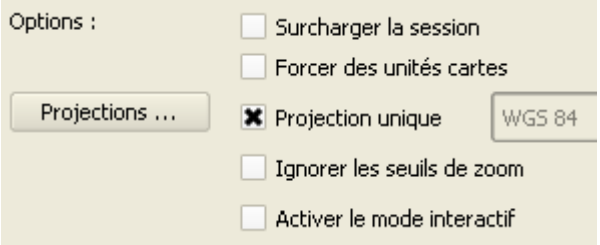




5 Les options et les contrôles

5.1 Les options

Les options sont présentées dans la partie basse du premier onglet :



Pour rappel, la liste des fonctions portées par la classe **UI_Dialog_OW** :

- ***def InterActiveMode(****self****):***
- ***def FixenProj4(****self, theProj****):***
- ***def FixeUniProjection(****self****):***

Des listes de choix permettent de sélectionner la carte et la mise en page à représenter.

Les fonctions dédiées à l'alimentation et la restitution d'informations sont :

- ***def MakeListTables(****self****):***
- ***def ReturnInfosLayerMap(****self****):***
- ***def FindComposer(****self, iMap****):***

Localisation : fopenwor.py

La dernière fonction “**FindComposer**” s'assure que la mise en page comporte au moins une référence à la carte sélectionnée.

5.2 Les contrôles

L'extension “OpenWor” dans sa version 1.9 ne supporte toujours pas ou ne restitue toujours pas un certain nombre d'informations du document originel.

Des contrôles sont aussi présents pour vérifier la disponibilité des ressources et leur intégrité (une couche peut être vide). Un document peut être partiellement restituable, soit parce certains éléments ne sont pas supportés par l'extension, soit parce que certaines ressources ne sont pas accessibles. Dans ce second cas, le nombre de fichiers ressources (**tLayer** et **uLayer**) sera différent du nombre de fichier ressources du document originel (“*Open Table ...*”).

Si aucun document n'est sélectionné, si l'intégralité d'un document n'est pas accessible en ressources, si aucune carte n'est contenue par ce document, le bouton “Charger la carte et la mise en page” n'est pas disponible.

La fonction qui contient ce contrôle est la suivante :

- ***def MajCtrlButtonMap(****self****):***

Localisation : fopenwor.py

6 La réalisation d'une carte, d'une mise en page et l'affichage de fenêtre de données

Les fonctions suivantes gérent ces différents aspects :

- ***def MakeMap(****self****):***
  - **def MakeMapCanvas(**self, iMap**):**
    - **def MakeLayer(**self, cLayer, cLayerMap, cInGroup, iMap, nSizeMap, nSizeMapUnits**):**
    - **def MakeGroupLayer(**self, cGroup**):**
    - **def MakeGroupExpanded(**self, nCond**):**
    - **def CountLayersMap(**zMap**):**

Localisation : fopenwor.py

  - **def MakeComposer(**self, iComposer, iInfosComposer**):**

Localisation : symbology.py

La fonction **MakeMapCanvas** fait appel à la fonction **MakeLayer** pour la représentation d'une couche dans la carte à réaliser.

La fonction **MakeLayer** contient les appels aux fonctions de symbologie suivant la nature des couches chargées et des actions à réaliser (Symbologie simple, analyse(s) thématique(s), étiquettes ou annotations).

La fonction **MakeLayer** contient aussi la restitution des fenêtres données.

La fonction **MakeGroupLayer** est une fonction pour regouper les couches par groupes. Notez que lors de la surcharge de projet, des groupes “partagés” par deux projets MapInfo ne seront pas recréés. Si le groupe est déjà présent, la couche est placé dans le groupe de destination.

La fonction **MakeGroupExpanded** permet de fermer ou d'ouvrir l'ensemble des noeuds du bloc légende.

7 La symbologie

- ***Def MakeLayer(****self, cLayer, cLayerMap, cInGroup, iMap, nSizeMap, nSizeMapUnits****):***
  - ***def DefautLayerStyle(****self, uLayer****):***
  - ***def GetWMSInfos(****nTableRaster****):***
  - ***def AddLayerASCII(****self, AsciiFile, cLayer, NbRows****):***
    - **def MakeLineASCII(**self, rb, feat, TempoPoint, iX, iY, LigneData, zDelimiter, zQuoteCaracter, NbFields**):**
  - ***def HasHeaderASCII(****AsciiFILE, Ligne, zDelimiter, zQuoteCaracter****):***

Localisation : fopenwor.py

- ***def NbRowAsciiFILE(****AsciiFILE****):***

Localisation : ow_utils.py

Avant la symbolisation, la couche doit être au préalable chargée.

Si la couche est fournie avec un fichier “**QML**” le style porté par ce fichier est restitué pour la couche. Sinon en l'absence de symbologie expressement définie dans MapInfo, l'habillage à défaut est de type “N & B”.

Toute analyse thématique portée par une couche implique un rechargement de la ressource.

Les fonctions suivantes sont utilisées dans le pré-traitement des analyses thématiques :

- ***def CountAna(****DicGen, Target****):***
- ***def CountAnaiMapSup(****iMap, index****):***
- ***def PosANA(****DicGen, Target, iMap****):***
- ***def CountAnaMapi(****iMap****):***
- ***def CountAnaInfMapi(****iMap****):***

Localisation : fopenwor.py

Pour les raster WMS, des pré-traitements sont aussi nécessaires (récupération des informations de session WMS du fichier XML associé au fichier TAB MapInfo) :

- ***def GetWMSInfos(****nTableRaster****):***
- ***def ExtractValWMS(****nStr, nRub****):***
- ***def ExtractLayers(****nStr****):***

Localisation : symbology.py

Une fois la ressource chargée, par appel aux fonctions natives QGIS idoines (*addVectorLayer*, *addRasterLayer*) ou par une fonction OpenWor (*AddLayerASCII*), l'appel à la fonction de représentation est réalisé.

- ***def Symbo2Vector(****self, zLayer, sSymboLayer, nSizeMap, nSizeMapUnits, zType, nForceUnitsMap, nFixeUniProj, zForceCRS, zForceZoom****):***
- ***def SymboRaster(****self, zLayer, sSymboLayer, nSizeMap, nSizeMapUnits, zForceZoom****):***
- ***def SymboVectorAna(****self, zLayer, sSymboLayer, nSizeMap, nSizeMapUnits, iSymboLayer, sInfosMap,nSymbologyV2, zType, sSymboLayerFather, HasLabeller, zReproject, nForceUnitsMap, zLayerVisibility, zForceCRS, zForceZoom****):***
- ***def SymboRasterMIG(****self, zLayer, zURL, sSymboLayer, nAnaMIG, nLabelFieldName, nSizeMap, nSizeMapUnits, sInfosMap, nSymbologyV2, vType, nForceUnitsMap****):***

Localisation : symbology.py

La plupart des fonctions ci-dessus comportent des appels aux fonctions suivantes :

- ***def FixeZOOM(****zInfos, zLayer, zSizeMap, zSizeMapUnits, indexUnitsZoom, indexValueMinZoom, indexValueMaxZoom, IsZoomLabels****):***
- ***def FixeALPHA(****zInfos, zLayer****):***
- ***def MakeVisibility(****self, sLayer, sLayerVisibility****):***

- ***def MakeLabels(****self, zLayer, sSymboLayer, nSizeMap, nSizeMapUnits, zType, nForceUnitsMap, zReproject, zForceCRS, zForceZoom****):***

Localisation : symbology.py

La fonction **MakeLabels** traite des étiquettes classiques et personnalisées (transposées en annotations).

Différentes sous-fonctions sont implémentées pour les styles appliqués par nature d'objets. Par ailleurs, l'ancienne et la nouvelle symbologie sont présentes dans l'extension.

7.1 Symbologie V2

Depuis sa version 1.9, l'ancienne symbologie de QGIS n'est plus utilisée par l'extension.

Des fonctions de la nouvelle symbologie sont utilisées par la symbolisation classique et par les analyses thématiques. Le paramétre **zDisplayGraphic** est à False pour les analyses thématiques.

- ***def MakeMARKERV2(****zDisplayGraphic, zLayer, zSymbol, zInfosSymbos****):***
  - **def MakeSimpleMARKERV2(**vInfos**):**
  - **def MakeFontMARKERV2(**vInfos**):**
  - **def MakeSvgMARKERV2(**vInfos**):**

- ***def MakeBRUSHV2(****zDisplayGraphic, zLayer, zSymbol, zInfosSymbos, zIsAna****):***
  - **def MakeSimpleBRUSHV2(**vInfos**):**
  - **def MakeSvgBRUSHV2(**vInfos**):**
  - **def MakeCentroidBRUSHV2(**vInfos**):**

La fonction MakeCentroidBRUSHV2 bien que présente n'est pas appelée par l'extension.

- ***def MakeLINEV2(****zDisplayGraphic, zLayer, zSymbol, zInfosSymbos****):***
  - **def MakeSimpleLINEV2(**zSymbol, zColorLine, zOffset, zQtLine, zSizeLine, zNet**):**

Localisation : symbology.py

7.2 Les analyses thématiques

Les informations des couches thématiques sont la combinaison des informations contenues dans **tLayerMapAna** (visibilité, ...) et **tMap** (type d'analyse, symbologie,...).

L'extension OpenWor comprend des appels aux fonctions d'analyses existantes présentes dans QGIS mais aussi des fonctions propres. Les appels aux fonctions de l'API QGIS concernent :

Symbologie v2 :

- **QgsCategorizedSymbolRendererV2()**
- **QgsGraduatedSymbolRendererV2()**

Les fonctions codées sont des sauts conditionnels de la fonction **SymboVectorAna** :

- ***GRADUATED***
- ***DENSITY***
  - **def randomizePoints(**self, inLayer, minimum, design, value, nSymbologyV2, nMax, nSize, nRatio, nFillColor, nType**):**
    - **def vectorRandom(**self, n, layer, xmin, xmax, ymin, ymax**):**
- ***PIE / HALF***
- ***BAR / STACKED***
  - **QgsUniqueValueRenderer()**
  - **QgsRendererCategoryV2()**

Localisation : symbology.py

8 Les mises en page

Dans la version 1.9 de l'extension, la partie “Compositions” (mises en page) n'a connu aucune amélioration.

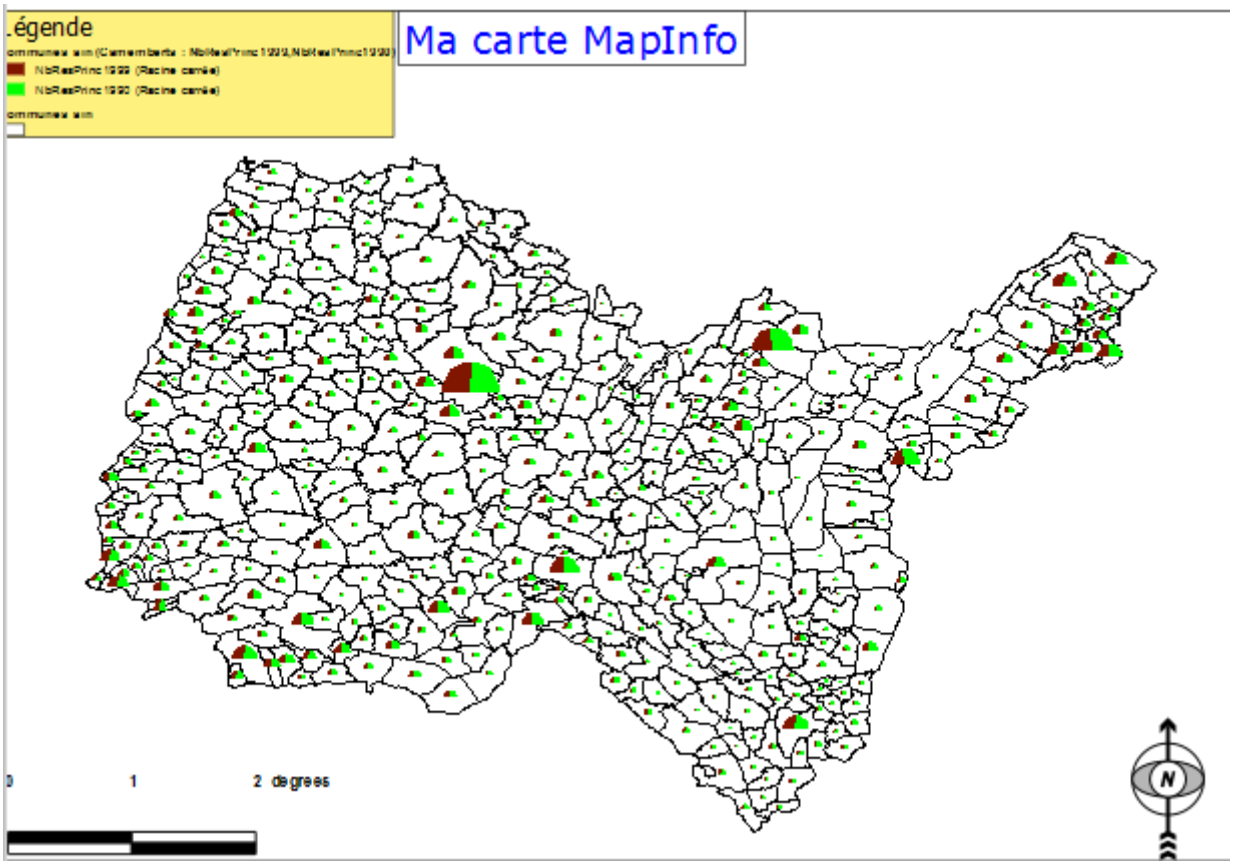
- ***def MakeComposer(****self, iComposer, iInfosComposer****):***
  - **def MakeTEXT(**self, c, composerView, posx, posy, zText, zColor, zpos, dpmm, isSimpleText**):**
  - **def MakeLEGEND(**self, c, composerView, posx, posy, zFillColor**):**
  - **def MakeTABLE(**self, c, composerView, posx, posy, w, h, zLayerName, zFillColor, zLineColor**):**
  - **def MakeOBJECT(**self, c, composerView, zlistCompo, Rac, wpos, dpmm**):**
  - **def MakeFRAME(**self, c, composerView, zlistCompo, Rac, wpos, dpmm**):**
  - **def MakeSCALEBAR(**self, c, composerView, composerMap**):**
  - **def MakeARROW(**self, c, composerView, zlistCompo, Rac, wpos, dpmm**):**
  - **def MakeNORTHARROW(**self, c, composerView, composerMap**):**

Les fonctions de mise en page utilisent les sous-fonctions ci-après :

- **def MakeFONT(**zPolice, zSize, zBold, zItalic, zUnderLine, zOverLine, zStrech**):**
- **def FixePaperSize(**zPaper**):**

Localisation : symbology.py

Dans la version 1.0 de l'extension, les mises en page du document originel ne sont quasiment pas exploitées. OpenWor propose aussi les options “*Aucune mise en page*” et “*Mise en page OpenWor*” qui retourne une mise en page de ce type :


égende
communes xm (Camembert : NbResPrinc1999,NbResPrinc1990)
NbResPrinc1999 (Racine carrée)
NbResPrinc1990 (Racine carrée)
communes xm
Ma carte MapInfo
0
1
2 degrees
N

9 Liste des principales nouveautés de la version 1.9

- Modifications de la fonction FixeALPHA ;
- Intégration des QgsExpression dans la fonction MakeLabel ;
- nouvelle fonction, SrtRGB(zColor), récupération valeurs élémentaires RGB chaîne de caractère de la couleur ;
- Suppression du code de tous les appels à l'ancienne symbologie (maintien des tableaux pour les objets des compositions) ;
- Suppression du répertoire "Brush" (Abandon de l'ancienne symbologie) ;
- Ajout du cas XLS dans AnalyseWOR et modification de la fonction GetRasterFile pour traiter le cas XLS ;
- Ajout du chargement des XLS dans MakeMapCanvas ;
- Ajout de la fonction GetRangeTableXLS(nTabFile) dans le module fopenwor.py pour récupération du bon onglet (plus tard, ajouter la définition de plage de cellules quand disponible sous QGIS) ;
- Introduction de règles de comportement (présence de fichiers type XLS, CSV et/ou de groupes) ;
- Prise en compte des couches logiques (SEAMLESS TAB) ;
- Injection des fonctions ChangeSETTINGS, DefineLayerProj et RestoreSETTINGS ;
- Ajout du nouvel onglet "Recherche de documents" (IHM, controles divers) ;
- Refonte onglet "attention" de l'interface ;
- Nouvelle ébauche de récupération des informations emprise / Zoom du document ;
- Corrections diverses sur le code ...