



## Prise en main de GNOGA



Sowebio SARL  
15, rue du Temple  
17310 – St Pierre d’Oléron – France

Capital 15 000 EUR – SIRET 844 060 046 00019 – RCS La Rochelle – APE 6201Z – TVA FR00844060046

Prise en main de GNOGA

[www.soweb.io](http://www.soweb.io)  
[developpement@soweb.io](mailto:developpement@soweb.io)



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

ed. 186 of 2023-09-12  
page 1 of 119

<b>Ed.</b>	<b>Validation</b>	<b>Objet</b>	
1	20220711	Première version du document.	tg
7	20220712	Ajout de liens.	tg
13	20220713	Documentation de la modification des sources et changements mineurs.	tg
14	20220723	Relecture et corrections.	sr
20	20220725	Modification de l'architecture du document.	tg
27	20220725	Documentation des anomalies, astuces et modification des sources.	tg
30	20220726	Documentation de la recompilation de GNOGA.	tg
33	20220726	Ajout des modifications de jQuery-UI.	tg
42	20220726	Ajouts mineurs dans la section Astuces.	tg
48	20220727	Précisions sur les conséquences liées à l'utilisation de SQLite sur FS réseaux SMB ou NFS.	sr
49	20220728	Paquets de dépendances d'Ubuntu 18.04 LTS à 22.04 LTS pour les tests et TLS [https].	sr
51	20220728	Intégration TLS [https] avec GNOGA.	sr
58	20220729	Ajouts des fichiers en annexe, modifications des sources et code JavaScript.	tg
65	20220729	Modification des sources : templates.	tg
67	20220729	Ajouts mineurs dans Contexte.	tg
70	20220729	Changements mineurs.	tg
72	20220731	Validation build et intégration TLS.	sr
77	20220801	Modification des sections détaillant la déclaration statique/dynamique d'éléments.	tg
81	20220801	Ajout keycode.	tg
86	20220801	Modification du code en annexe.	tg
88	20220801	Ajout du chapitre Anomalies avec Handlers.	tg
89	20220803	Corrections hello3post et hello3post_File avec issues <a href="https://github.com/Blady-Com">https://github.com/Blady-Com</a> .	sr
94	20220803	Réorganisations « anomalies ». Intégration SSL dans l'intallation Gnoga.	sr
97	20220804	Synthèse patch Makefile, validation test SSL autres tickets de bugs, mise au propre.	sr
105	20220805	Relecture + Modifications mineures.	tg
110	20220809	Suppression du chapitre sur les erreurs sur la méthode Style pour le remplacer par un chapitre traitant de l'erreur plus générale du crash des boutons de l'application.	tg
112	20220810	Relecture et changements mineurs.	tg
116	20220810	Ajouts dans le chapitre Anomalies avec les Handlers.	tg
122	20220817	Ajouts des chapitres Exemple d'ajout de fichier JS et CSS, modification des chapitres Ajout de Folders.js, .css ...	tg
124	20220817	Ajout du chapitre Dictionnaire en annexe.	tg
128	20220817	Modifications mineures.	tg
133	20220818	Changements mineurs et Ajout de la section fuites mémoires.	tg
135	20220818	Ajouts dans Anomalie avec les Handlers.	tg
155	20220819	Modifications mineures et changement du logo Ada.	tg
157	20220822	Ajout de la section Résolution dans le chapitre Anomalie avec les Handlers.	tg
163	20220822	Ajout du logo GNOGA, modifications mineures sur la Résolution des problèmes des Handlers et Installation de GNOGA.	tg
164	20220823	Relecture et corrections.	sr

#### Prise en main de GNOGA



<i><b>Ed.</b></i>	<i><b>Validation</b></i>	<i><b>Objet</b></i>	
171	20220824	Ajout des chapitres liés au package Simple_Form	tg
173	20220831	Changements mineurs	tg
174	20220904	Changements mineurs	tg
176	20230728	Logos unifiés	sr
182	20230814	Clarifications ds dépendances, maj Gnoga 2.2a, déverminage des interfaces DB, maj des démos et tests MySQL et SQLite	sr
186			

Prise en main de GNOGA

[www.soweb.io](http://www.soweb.io)  
[developpement@soweb.io](mailto:developpement@soweb.io)



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

ed. 186 of 2023-09-12  
page 3 of 119

Auteur

Théodore Gigault – [t.gigault@gmail.com](mailto:t.gigault@gmail.com)

Relecteur

Stéphane Rivière – [sriviere@soweb.io](mailto:sriviere@soweb.io) [Maître de stage & CTO Sowebio]

Logos Ada et GNOGA [fin de document]

Senzo Rouhaud - Sowebio



# Table des matières

---

## Introduction

1	Contexte.....	11
2	Généralités.....	11
3	Environnement.....	12

## Installation de GNOGA

1	Avertissement.....	13
1.1	Mode d'installation.....	13
1.2	Modifications des sources.....	13
2	Dépendances.....	13
2.1	Tableau des dépendances.....	13
2.2	Détail des dépendances.....	14
3	Chemins.....	15
3.1	<GNOGA_PATH>.....	15
3.2	<ENV_DEV_PATH>.....	15
3.3	<GPR_PROJECT_PATH>.....	15
4	Installation.....	15
4.1	Build 2.2a au 06/08/23.....	15
4.2	Build 2.1a au 06/03/22.....	17
4.3	Exemples de programmes.....	19
4.4	Documentation installée.....	19
4.5	Documentation non installée.....	19
4.6	Outils installés.....	19
5	Environnement.....	20
5.1	Nautilus avec GNAT Studio.....	20
5.2	Midnight Commander avec GNAT Studio.....	20

## Découverte de GNOGA

1	Tutoriels.....	21
1.1	Tutoriel 01.....	21
1.2	Tutoriel 02.....	21
1.3	Tutoriel 03.....	21
1.4	Tutoriel 04.....	21
1.5	Tutoriel 05.....	21
1.6	Tutoriel 06.....	22
1.7	Tutoriel 07.....	22
1.8	Tutoriel 08.....	22
1.9	Tutoriel 09.....	22
1.10	Tutoriel 10.....	23
1.11	Tutoriel 11.....	24
2	Démos.....	25
2.1	AdaBlog.....	25
2.2	AdaEdit.....	27

Prise en main de GNOGA



2.3	Chattanooga.....	27
2.4	Connect Four.....	27
2.5	Leaves Main.....	28
2.6	Linxtris.....	29
2.7	Localize.....	29
2.8	Logo.....	30
2.9	Mine Detector.....	30
2.10	Movies.....	30
2.11	Othello.....	31
2.12	Password Gen.....	31
2.13	Random Int.....	31
2.14	Snake.....	31
2.15	Tic tac toe.....	32
3	Test.....	32
3.1	Align.....	32
3.2	Bootjs.....	33
3.3	boot_strap_demo.....	33
3.4	canvas_test.....	34
3.5	cli.....	34
3.6	db_active.....	34
3.7	db_mysql.....	35
3.8	db_sqlite.....	37
3.9	demo [aka gnoga-demo].....	38
3.10	ding.....	39
3.11	essai8.....	39
3.12	essai10b.....	40
3.13	essai11.....	40
3.14	files_ops.....	41
3.15	forms.....	42
3.16	jdemo.....	43
3.17	layouts.....	44
3.18	mac_test.....	44
3.19	media.....	44
3.20	menu.....	45
3.21	message_boxes.....	45
3.22	multiuser.....	46
3.23	pack.....	46
3.24	pixi_sprite_test.....	46
3.25	plot_test.....	47
3.26	popups.....	47
3.27	pragma_sqlite.....	48
3.28	singleton.....	48
3.29	sprite_test.....	50
3.30	storage.....	50
3.31	svg_demo.....	51
3.32	test_modal_dialog.....	51

3.33	tree.....	52
3.34	ttables.....	53
4	Test SSL.....	53
4.1	layouts_ssl.....	53
5	Documentation gnoga-wf.pdf.....	55
5.1	Création d'un projet GNOGA avec gnoga_make – version originale.....	55
5.2	Création d'un projet GNOGA avec gnoga_make – version modifiée.....	58
5.3	Utilisation basique.....	59
5.4	Les résultats des formulaires.....	60
5.5	Méthode Get.....	60
5.6	Méthode Post.....	61
5.7	Stockage de données.....	62
6	Fuites mémoires.....	63
6.1	Valgrind.....	63

## Utilisation de GNOGA

1	Environnement de développement.....	65
2	Base du codage avec GNOGA.....	65
3	Types.....	66
3.1	App_Data.....	66
3.2	Widget.....	66
4	Composants GNOGA.....	67
4.1	Dockers.....	67
4.2	Boutons.....	67
4.3	Boîtes de dialogue.....	68
4.4	Diverses méthodes sur les objets.....	69
4.5	Autres outils.....	69
4.6	Ajouter des images.....	69
4.7	Breadcrumb.....	69
4.8	Simple Form.....	69
5	Composants JS.....	70
5.1	JQuery.....	70
5.2	jQuery-UI.....	70
5.3	MNMenu.....	71
5.4	Ace Editor.....	73
5.5	Bootstrap.....	74
5.6	JsTree.....	74
5.7	MacGap.....	74
5.8	PixiJS.....	74
5.9	Message Boxes.....	75
6	Composants jQuery-UI.....	75
6.1	Accordion.....	75
6.2	Folders.....	76
6.3	Tabs.....	76
6.4	Cards.....	77
7	Fonctionnalité TLS [https].....	77
7.1	Proxy TLS.....	77

8	7.2 Support direct.....	77
9	CSS.....	78
9	Modification des sources.....	80
9.1	Généralités.....	80
9.2	Attribut Background.....	80
9.3	Sections des Accordions.....	81
9.4	jQuery-UI.....	82
10	Astuces.....	82
10.1	Récupérer le contenu d'un objet.....	82
10.2	Styliser les éléments avec CSS.....	83
10.3	Plantage de l'application.....	83
10.4	Inspecteur de pages Web.....	83
10.5	Création dynamique ou statique d'éléments.....	84
10.6	Architecture de développement.....	85
10.7	Exemple d'ajout de fichier CSS.....	85
10.8	Exemple d'ajout de fichier JS.....	86

## Annexes

1	Ressources GNOGA.....	87
1.1	Gnoga.....	87
1.2	Ada for automation.....	87
1.3	Document « Premiers pas avec GNOGA ».....	87
1.4	Vidéo de démonstration de GNOGA.....	87
2	Ressources Sowebio.....	88
2.1	AIDE.....	88
3	Dictionnaire.....	88
3.1	Dictionnaire général/Ada.....	88
3.2	Dictionnaire GNOGA.....	88
4	Ressources jQuery.....	89
4.1	Keycodes.....	89
5	Ressources jQuery-UI.....	91
5.1	Généralités.....	91
5.2	Extension.....	91
5.3	ThemeRoller.....	91
6	Ressources DOM.....	92
7	To do list.....	92
7.1	Intégration SQLite.....	92
7.2	Documentation makefile.....	93
8	Anomalies corrigées.....	93
8.1	hello3post.adb.....	93
8.2	hello3post_file.adb.....	93
8.3	Anomalies avec les Handlers.....	94
9	Anomalies signalées.....	101
9.1	Issues Github.....	101
9.2	Tickets Sourceforge.....	101
10	Anomalies à corriger.....	101
10.1	SQLite & les FS réseaux.....	101

10.2	Démos.....	102
10.3	Anomalies dans gnoga-wf.pdf.....	103
10.4	Accents dans les titres des sections de certains composants jQuery-UI.....	105
11	Modifications des sources.....	105
11.1	Modification du template singleton de gnoga_make.....	105
11.2	Modification du template multi_connect de gnoga_make.....	106
11.3	Ajout de Breadcrumb.ads.....	107
11.4	Ajout de Breadcrumb.adb.....	107
11.5	Ajout de Simple_Form.ads.....	108
11.6	Ajout de Simple_Form.adb.....	109
11.7	Ajout de Folders.ads.....	111
11.8	Ajout de Folders.adb.....	112
11.9	Ajout de Folders.js.....	113
11.10	Ajout de Folders.css.....	117
11.11	Modifications à apporter à boot.html pour inclure le package Folders.....	118



<https://this-page-intentionally-left-blank.org>

Prise en main de GNOGA



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

[www.soweb.io](http://www.soweb.io)  
[developpement@soweb.io](mailto:developpement@soweb.io)

ed. 186 of 2023-09-12  
page 10 of 119

# Introduction

---

## 1 Contexte

Ce document a été rédigé par Théodore Gigault, dans le cadre d'un stage en entreprise, afin de conclure sa première année à l'ENSEIRB-MATMECA.

L'objet de ce stage a été de découvrir les méthodes de programmation modulaire et structurée, orientée objet, ainsi qu'une méthode d'analyse top-down, puis de mettre en pratique les apprentissages avec la prise en main d'une infrastructure logicielle graphique en Ada et, enfin, d'effectuer l'analyse de l'application de gestion des contrats clients en prélèvements SEPA de l'entreprise. Afin d'implémenter cette application, GNOGA a été choisi comme framework.

## 2 Généralités

Quand on parle de développement Web, Ada n'est pas le premier langage auquel on pense. Le JavaScript est actuellement le plus populaire mais présente aussi des inconvénients : typage faible, langage interprété donc lent, syntaxe peu expressive, « enfer des dépendances », etc. Typescript, un dialecte « transpilé » en Javascript, et fortement typé, ne permet pas de débugger facilement le programme à cause de la « compilation » intermédiaire.

Ada ne présente pas tous les désavantages précédemment énoncés et autorise des développements productifs et maintenables. Ce langage a donc été choisi pour la création des développements internes de l'entreprise. À cet effet, le framework graphique GNOGA a été sélectionné.

GNOGA est un framework graphique en Ada utilisant des outils modernes pour le développement d'applications Web. De plus, les applications créées sont « responsives » et multi-plateformes.

Une partie de la bibliothèque est dédiée à certains plugins comme jQuery, BootStrap ou en PixiJS. GNOGA propose également les fichiers JavaScript, CSS et HTML qui sont par défaut nécessaires. Enfin, le programme peut aussi bien tourner sur le même ordinateur que le navigateur ou sur un serveur.

Son arrivée dans l'environnement Ada a été vue comme une bonne chose par de nombreux membres de la communauté. En effet, comme le dit Pascal Pignard, le mainteneur principal de GNOGA :

– Malgré le positionnement du langage Ada principalement dans le monde professionnel peuplé d'ingénieurs où le « fun » [...] n'est pas le critère principal, je pense au contraire que l'apport « natif » d'une bibliothèque graphique apporterait un peu de « fun » qui serait profitable à Ada pour son aura après des jeunes ingénieurs [ou moins jeunes], quitte à pencher vers plus d'instantanéité contraire au méthode de développement logiciel. [...] À l'inverse, il me semble que son absence est un frein dans l'environnement actuel où tout est interface graphique : la moindre petite démo est mieux valorisée dans une interface graphique habituelle. Quelques surprises pourraient venir

d'applications graphiques orientées Web indépendantes de la plateforme par nature comme le propose la toute nouvelle bibliothèque GNOGA.<sup>1</sup>

Un dictionnaire regroupant quelques termes techniques est proposé [voir Dictionnaire] pour permettre au lecteur une meilleure compréhension du document.

## 3 Environnement

L'environnement des stations est GNU/Linux Ubuntu 18.04 LTS et 22.04 LTS avec un serveur intranet GNU/Linux Debian 11 virtualisé sous Xen et supportant SAMBA.

L'environnement de développement est AIDE [Ada Instant Environment Development], fondé sur GNAT CE 2021. <https://github.com/sowebio/aide-bin>

<sup>1</sup> Extrait de Premiers pas avec GNOGA, Pascal Pignard [<https://blady.pagesperso-orange.fr/telechargements/gnoga/gnoga.pdf>]



# Installation de GNOGA

## 1 Avertissement

### 1.1 Mode d'installation

◊ Il est recommandé d'installer l'environnement de développement et GNOGA en local car les programmes de démonstration SQLite et différents outils GNAT utilisant SQLite [tels GnatDoc] sont incompatibles avec les systèmes de fichiers réseaux<sup>2</sup>, tels SMB, NFS, etc.

Voir SQLite & les FS réseaux pour plus d'information.

Pour partager une installation GNOGA dans un groupe de travail, sans modification de l'environnement de développement, il faudra utiliser un « drive » tels que Seafire, dropbox, etc.

Build « make all » :

- En local sur disque NVMe : 1m46
- En réseau 1 Gbps sur RAID5 16To : 5m24

### 1.2 Modifications des sources

Avant l'installation de GNOGA, il est conseillé d'étudier les changements opérés à la section Modification des sources.

## 2 Dépendances

### 2.1 Tableau des dépendances

		<b>Paquet</b>	<b>Commentaire</b>
MySQL	libmariadb3	MariaDB database client library	
	libmariadb-dev	MariaDB database development files	
	libmariadb-dev-compat <sup>3</sup>	MariaDB Connector/C, compatibility symlink avec MySQL	
	mariadb-client	Optionnel, pour le client CLI MySQL	
	mariadb-server	Optionnel, pour avoir le serveur en local sur sa station	
SQLite	libssqlite3-0	SQLite 3 shared library	
	libssqlite3-dev	SQLite 3 development files	
TLS/https	libgnutls30	GNU TLS library - main runtime library	
Gtk	pkg-config	Manage compile and link flags for libraries	
	libwebkit2gtk-4.0-37	Web content engine library for GTK	
	libwebkit2gtk-4.0-dev	Web content engine library for GTK - development files	
	libgtk-3-0	GTK graphical user interface library	
	libgtk-3-dev	Development files for the GTK library	
	libjavascriptcoregtk-4.0-dev	JavaScript engine library from WebKitGTK - development files	

<sup>2</sup> SQLite n'est pas conçu pour accéder à des fichiers de bases de données placés sur des systèmes de fichiers réseaux.

<sup>3</sup> Paquet inutile si on remplace --pragma Linker\_Options ["-lmysqlclient"]; par pragma Linker\_Options ["-lma-riadb"]; -- package libmariadb3

## 2.2 Détail des dépendances

### ❑ SQLite via la distribution pour les tutoriels

Sous Ubuntu 18.04 LTS à 22.04 LTS :

---

```
user@system: sudo apt install libssqlite3-0 libssqlite3-dev
```

---

### ❑ Paquets pour TLS

Pour l'utilisation sécurisée de GNOGA [<https://>]. Sous Ubuntu 18.04 LTS à 22.04 LTS :

---

```
user@system: sudo apt install libgnutls30
```

---

### ❑ Paquets pour MySQL

Pour le test MySQL [mariaDB], il sera nécessaire de créer une base de données et un utilisateur avec les droits sur cette base. Sous Ubuntu 18.04 LTS à 22.04 LTS, installer les paquets suivants :

---

```
user@system: sudo apt install mariadb-server mariadb-client libmariadbclient-dev-compat
```

---

### ❑ Paquets pour SQLite

---

```
user@system: sudo apt install mariadb-server mariadb-client libmariadbclient-dev-compat
```

---

### ❑ Paquets pour L'intégration Gtk

---

```
user@system: sudo apt install pkg-config libwebkit2gtk-4.0-37 libwebkit2gtk-4.0-dev libgtk-3-0 libgtk-3-dev libjavascriptcoregtk-4.0-dev
```

---

### ❑ Éditeur SQL Universel DBeaver

Pour éditer tout types de bases de données, dont MySQL, MariaDB et SQLite, il est conseillé d'installer DBeaver :

<https://dbeaver.io/download>

<https://github.com/dbeaver/dbeaver/releases>

### ❑ Éditeur SQLiteBrowser seulement

Pour éditer seulement les bases de données SQLite, il est conseillé d'installer :

---

```
user@system: sudo apt install sqlitebrowser
```

---

## 3 Chemins

### 3.1 <GNOGA\_PATH>

Le chemin <GNOGA\_PATH> désigne la racine de Gnoga.

Dans ce document, l'archive de Gnoga a été décompressée dans \$HOME/opt/gnoga.

Adaptez ce chemin à votre configuration préférée.

### 3.2 <ENV\_DEV\_PATH>

Le chemin <ENV\_DEV\_PATH> désigne la racine de l'environnement Ada utilisé. En procédant ainsi, il n'y a aucune variable d'environnement à mettre à jour.

Pour la définition précise de ce chemin, voir les sections suivantes.

### 3.3 <GPR\_PROJECT\_PATH>

Puisque GPRbuild est installé dans le sous-répertoire <ENV\_DEV\_PATH>/bin de l'environnement de développement Ada, alors le répertoire par défaut des fichiers de projets [.gpr] est <ENV\_DEV\_PATH>/share/gpr.

Dans ces conditions, il n'est pas non plus nécessaire de paramétriser la variable d'environnement GPR\_PROJECT\_PATH.

## 4 Installation

### 4.1 Build 2.2a au 06/08/23

#### □ Installation du fichier compressé dans <GNOGA\_PATH>

Dans ce document, l'archive de Gnoga est décompressée dans \$HOME/opt/gnoga :

---

```
user@system: cd $HOME
user@system: mkdir opt ; cd opt
user@system: wget https://sourceforge.net/projects/gnoga/files/gnoga-2.1a.zip
user@system: unzip gnoga-2.1a
user@system: mv gnoga-2.1a gnoga
```

---

#### □ Build principal

Pour le build de la démo SSL [https], des tests et de la documentation, pour la version 2.1a, il est nécessaire d'incorporer dans le Makefile les modifications ci-dessous.

Elles ont été créées par la commande : diff -u Makefile-org Makefile.

Détail des modifications :

- Pour le build des tests, ajout de native\_gtk et gnoga\_secure à la règle all et remplacement de webkit2gtk-3.0 par webkit2gtk-4.0 ;
- Pour le build de la documentation, remplacement de git: par https: ;
- Pour l'intégration de gnoga\_secure dans l'installation et sa désinstallation, ajout de deux commandes gprinstall.

```

--- Makefile.org          2022-03-06 11:01:02.000000000 +0100
+++ Makefile 2022-08-04 14:43:52.806908453 +0200
@@ -133,7 +133,7 @@
     ZB_OPTIONS=-XBUILD=${subst Release,Production,\${BUILD_MODE}} -XOS=unix
 endif

-all: deps ${BUILD_SQLITE3} basic_components gnoga gnoga_tools demo tutorials
+all: deps ${BUILD_SQLITE3} basic_components gnoga gnoga_secure gnoga_tools demo tutorials native_gtk

 basic_components:
     ${MAKE} -C components
@@ -161,7 +161,7 @@
     ${BUILDER} -P deps/uxstrings/lib_uxstrings.gpr

 native_gtk: src/gnoga_gtk_window.c
-    cd obj && gcc -c ..src/gnoga_gtk_window.c `pkg-config --cflags gtk+-3.0,webkit2gtk-3.0`
+    cd obj && gcc -c ..src/gnoga_gtk_window.c `pkg-config --cflags gtk+-3.0,webkit2gtk-4.0`

 native_osx:
     - cd deps && git clone https://github.com/MacGapProject/MacGap2.git
@@ -185,7 +185,7 @@
 .IGNORE: bin/multimarkdown
 bin/multimarkdown:
     - cd deps && git clone git://github.com/fletcher/MultiMarkdown-4.git
+    cd deps && git clone https://github.com/fletcher/MultiMarkdown-4.git
     cd deps/MultiMarkdown-4 && git submodule init
     cd deps/MultiMarkdown-4 && git submodule update
     cd deps/MultiMarkdown-4 && "${MAKE}"
@@ -215,6 +215,7 @@
 # Install Gnoga without deps
 install_gnoga: basic_components gnoga gnoga_tools
     ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga src/gnoga.gpr ${GN_OPTIONS}
+    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga ssl/gnoga_secure.gpr ${GN_OPTIONS}
     ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --mode=usage tools/tools_agg.gpr ${GN_OPTIONS}
     ${MAKE} -C components INSTALL_DIR="${PREFIX}/share/gnoga
@@ -232,6 +233,7 @@
     ${INSTALLER} --prefix="${PREFIX}" --install-name=pragmarc --uninstall pragmarc.gpr
     ${INSTALLER} --prefix="${PREFIX}" --install-name=uxstrings --uninstall lib_uxstrings.gpr
     ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --uninstall gnoga.gpr
+    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --uninstall gnoga_secure.gpr
     ${MAKE} -C deps/zanyblue/src INSTALL_DIR="${PREFIX}" uninstall
     ${RM} -fr "${PREFIX}/share/gnoga

```

Pour modifier le Makefile, créer un fichier Makefile.patch contenant le listing ci-dessus et, dans le répertoire de <GNOGA\_PATH>, lancer la commande :

---

user@system: patch Makefile Makefile.patch

---

Depuis <GNOGA\_PATH>, lancer la commande suivante pour inclure la bonne version des paquets :

---

sed -i 's/gtk-3.0/gtk-4.0/g' ./src/gnoga-application-gtk\_window.adb

---

Toujours depuis <GNOGA\_PATH>, lancer dans cet ordre :

---

user@system: make all  
user@system: make tests  
user@system: make html-docs rm-docs

---

## □ Installation de GNOGA dans <ENV\_DEV\_PATH>

L'environnement de développement utilisé pour le build 2.2a est un GNAT GCC 11.2.4 installé dans \$HOME/opt/gnat-1124



⇒ Ce chemin est désigné <ENV\_DEV\_PATH>

Remplacez-le par le chemin de votre configuration actuelle et lancer l'installation :

```
user@system: make PREFIX=<ENV_DEV_PATH> install
user@system: cp <GNOGA_PATH>/lib/gnoga-server-connection-secure.ali <ENV_DEV_PATH>/lib/gnoga/gnoga ; cp <GNOGA_PATH>/lib/libgnoga_secure.a <ENV_DEV_PATH>/lib/gnoga/gnoga
```

## 4.2 Build 2.1a au 06/03/22

□ Installation du fichier compressé dans <GNOGA\_PATH>

Dans ce document, l'archive de Gnoga est décompressée dans \$HOME/opt/gnoga :

```
user@system: cd $HOME
user@system: mkdir opt ; cd opt
user@system: wget https://sourceforge.net/projects/gnoga/files/gnoga-2.1a.zip
user@system: unzip gnoga-2.1a
user@system: mv gnoga-2.1a gnoga
```

□ Build principal

Pour le build de la démo SSL [https], des tests et de la documentation, pour la version 2.1a, il est nécessaire d'incorporer dans le Makefile les modifications ci-dessous.

Elles ont été créées par la commande : diff -u Makefile-org Makefile.

Détail des modifications :

- Pour le build des tests, ajout de native\_gtk et gnoga\_secure à la règle all et remplacement de webkit2gtk-3.0 par webkit2gtk-4.0 ;
- Pour le build de la documentation, remplacement de git: par https: ;
- Pour l'intégration de gnoga\_secure dans l'installation et sa désinstallation, ajout de deux commandes gprinstall.

```
--- Makefile-org      2022-03-06 11:01:02.000000000 +0100
+++ Makefile 2022-08-04 14:43:52.806908453 +0200
@@ -133,7 +133,7 @@
         ZB_OPTIONS=-XBUILD=${subst Release,Production,\${BUILD_MODE}} -XOS=unix
 endif

-all: deps ${BUILD_SQLITE3} basic_components gnoga gnoga_tools demo tutorials
+all: deps ${BUILD_SQLITE3} basic_components gnoga gnoga_secure gnoga_tools demo tutorials native_gtk

 basic_components:
     ${MAKE} -C components
@@ -161,7 +161,7 @@
     ${BUILDER} -P deps/uxstrings/lib_uxstrings.gpr

 native_gtk: src/gnoga_gtk_window.c
-        cd obj && gcc -c .. /src/gnoga_gtk_window.c `pkg-config --cflags gtk+-3.0,webkit2gtk-3.0`
+        cd obj && gcc -c .. /src/gnoga_gtk_window.c `pkg-config --cflags gtk+-3.0,webkit2gtk-4.0`

 native_osx:
     - cd deps && git clone https://github.com/MacGapProject/MacGap2.git
@@ -185,7 +185,7 @@
 .IGNORE: bin/multimarkdown
 bin/multimarkdown:
     - cd deps && git clone git://github.com/fletcher/MultiMarkdown-4.git
+    cd deps && git clone https://github.com/fletcher/MultiMarkdown-4.git
```

## Prise en main de GNOGA



---

```
cd deps/MultiMarkdown-4 && git submodule init
cd deps/MultiMarkdown-4 && git submodule update
cd deps/MultiMarkdown-4 && "${MAKE}"
@@ -215, 6 +215, 7 @@
# Install Gnoga without deps
install_gnoga: basic_components gnoga gnoga_tools
    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga src/gnoga.gpr ${GN_OPTIONS}
+    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga ssl/gnoga_secure.gpr ${GN_OPTIONS}
    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --mode=usage tools/tools_agg.gpr ${GN_OPTIONS}
    ${MAKE} -C components INSTALL_DIR="${PREFIX}/share/gnoga
@@ -232, 6 +233, 7 @@
    ${INSTALLER} --prefix="${PREFIX}" --install-name=pragmarc --uninstall pragmarc.gpr
    ${INSTALLER} --prefix="${PREFIX}" --install-name=uxstrings --uninstall lib_uxstrings.gpr
    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --uninstall gnoga.gpr
+    ${INSTALLER} --prefix="${PREFIX}" --install-name=gnoga --uninstall gnoga_secure.gpr
    ${MAKE} -C deps/zanyblue/src INSTALL_DIR="${PREFIX}" uninstall
${RM} -fr "${PREFIX}/share/gnoga
```

---

Pour modifier le Makefile, créer un fichier Makefile.patch contenant le listing ci-dessus et, dans le répertoire de <GNOGA\_PATH>, lancer la commande :

---

```
user@system: patch Makefile Makefile.patch
```

---

Depuis <GNOGA\_PATH>, lancer la commande suivante pour inclure la bonne version des paquets :

---

```
sed -i 's/gtk-3.0/gtk-4.0/g' ./src/gnoga-application-gtk_window.adb
```

---

Toujours depuis <GNOGA\_PATH>, lancer dans cet ordre :

---

```
user@system: make all
user@system: make tests
user@system: make html-docs rm-docs
```

---

## □ Installation de GNOGA dans <ENV\_DEV\_PATH>

L'environnement de développement utilisé pour la rédaction de ce document est un GNAT Studio 2021 CE issu de AIDE <https://github.com/sowebio/aide-bin> et installé dans \$HOME/opt/gnat-2021.

▷ Ce chemin est désigné <ENV\_DEV\_PATH>

Remplacez-le par le chemin de votre configuration actuelle et lancer l'installation :

---

```
user@system: make PREFIX=<ENV_DEV_PATH> install
user@system: cp <GNOGA_PATH>/lib/gnoga-server-connection-secure.ali <ENV_DEV_PATH>/lib/gnoga/gnoga ; cp <GNOGA_PATH>/lib/libgnoga_secure.a <ENV_DEV_PATH>/lib/gnoga/gnoga
```

---

## 4.3 Exemples de programmes

- Tutoriels

<GNOGA\_PATH>/tutorial

- Démos

<GNOGA\_PATH>/demo

- Tests

<GNOGA\_PATH>/test

## 4.4 Documentation installée

- Simple Components

<ENV\_DEV\_PATH>/share/doc/components/components.htm

- Zanyblue

<ENV\_DEV\_PATH>/share/doc/zanyblue/index.html

<ENV\_DEV\_PATH>/share/doc/zanyblue/ZanyBlue.pdf

- Gnoga documentation

<ENV\_DEV\_PATH>/share/gnoga/doc

- Gnoga Reference Manual

<ENV\_DEV\_PATH>/share/gnoga/gnoga\_rm

## 4.5 Documentation non installée

<GNOGA\_PATH>/FAQ

<GNOGA\_PATH>/HISTORY

<GNOGA\_PATH>/INSTALL

<GNOGA\_PATH>/README

<GNOGA\_PATH>/README-SSL

<GNOGA\_PATH>/TIPS.md

<GNOGA\_PATH>/TODO

## 4.6 Outils installés

- gnoga\_doc

Générateur de documentation à partir des spécifications de GNOGA.

- gnoga\_make

Générateur de squelette d'application.

## ❑ Zbinfo

Utilitaire d'information sur les données de la bibliothèque ZanyBlue.

## ❑ zbmcompile

Utilitaire de génération de sources Ada pour l'internationalisation.

# 5 Environnement

## 5.1 Nautilus avec GNAT Studio.

Nautilus démarre GNAT Studio pour fichier projet [.gpr] lancé.

▷ Si, après l'installation de GNOGA, des variables d'environnement ont été mises à jour, telles que PATH, GPR\_PROJECT\_PATH, etc. il faut *fermer toutes les instances* de Nautilus, puis le relancer, afin que le nouvel environnement soit pris en compte.

Parfois, il n'est pas possible de démarrer GNAT Studio à partir de Nautilus sans erreur à la compilation. La méthode ci-dessous est alors recommandée.

## 5.2 Midnight Commander avec GNAT Studio.

Une alternative à la console, plus efficace que Nautilus et plus universelle (puisque elle s'affranchit de l'environnement graphique), est Midnight Commander.

Cet utilitaire, avec ses deux fenêtres de dossiers, permet aussi une réorganisation rapide des répertoires.

Pour faciliter le lancement de GNAT Studio, ajouter dans le fichier `~/.config/mc/mc.ext` :

---

```
.../  
shell/.db  
    Open=nohup sqlitebrowser %f &  
  
shell/.gpr  
    Open=nohup gnatstudio -P%f &  
  
shell/.html  
    Open=nohup /opt/firefox/firefox %f &  
.../...
```

---

Ajuster le chemin de firefox en fonction de votre installation [ici une version originale, non snap, installée dans /opt].

Par ailleurs, une console est toujours à disposition via Ctrl+O.

# Découverte de GNOGA

---

Dans cette section, nous avons détaillé le contenu des tutoriels, des démos, etc.

Les tutoriels sont dans <GNONA\_PATH>/tutoriel et les exécutables des tutoriels dans <GNONA\_PATH>/bin.

## 1 Tutoriels

### 1.1 Tutoriel 01

Application singleton, affichage d'un message « Hello World » puis fermeture de l'application.

```
html/boot.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

### 1.2 Tutoriel 02

Boutons [Click Me!] [End App] et association d'une fonction pour chaque bouton. Fermeture de l'application à la demande du client.

```
html/boot.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

### 1.3 Tutoriel 03

Idem au précédent mais en multi-connexion utilisateur.

```
html/boot.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

### 1.4 Tutoriel 04

Implémentation des tâches avec GNOGA. Cela permet de réaliser plusieurs tâches en parallèle et non en série. Cela revient à incorporer dans le code des éléments de programmation événementielle. Les tâches modifient la couleur du texte « I love Gno-ga ! ».

```
html/boot.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

### 1.5 Tutoriel 05

Ajout d'un Canvas (zone de dessin).

```
html/boot.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```



## 1.6 Tutoriel 06

Fenêtres, Frames, Pop-up(s).

Lance deux fenêtres dont l'une affiche :

---

Since this popup is not a cross-site frame we have full control over it even though this popup was blank and has no websocket connection to our application. We are piggy backing off the connection in the main window.

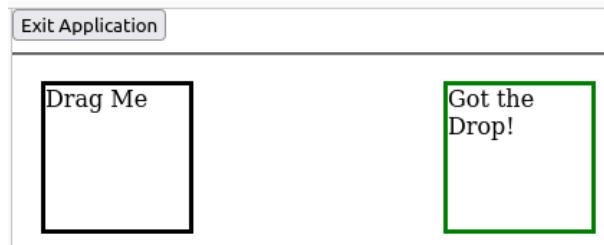
Click the close popups buttons in the main window.

---

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

## 1.7 Tutoriel 07

Drag and Drop.



html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

## 1.8 Tutoriel 08

Formulaire, interaction statique [une seule réponse possible au formulaire par connexion].

Présentation similaire au Tutoriel 09.

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

## 1.9 Tutoriel 09

Formulaire interactif [plusieurs réponses possibles au formulaire, visualisation des réponses].

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

[Exit Application](#)

Name	<input type="text" value="azeazeazeaze"/>
Message	<input type="text" value="1212212112"/>

[Submit](#)

[Submit]

Sur le navigateur :

Name : azeazeazeaze  
Message : 1212212112

En console :

Requested: Kind: FILE, File: result, Query: Name=azeazeazeaze&Message=1212212112

## 1.10 Tutoriel 10

Initialiser la base de données :

```
user@system: ./tutorial_10 setup

gnogaparams table created
2023-08-14 14:50:47.54 : Error End_Of_Recordset.
2023-08-14 14:50:47.54 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-server-database-sqlite.adb: 421
[./tutorial_10]
0x620c03 gnoga__server__database__sqlite__next at ???
0x682313 gnoga__server__model__find_where at ???
0x644342 gnoga__server__migration__migrate_to at ???
0x64cd58 gnoga__server__migration__setup at ???
0x64e4e0 gnoga__server__migration__migrations_handled_command_line at ???
0x40ec7c _ada_tutorial_10 at ???
0x41d9a5 main at ???
[/lib/x86_64-linux-gnu/libc.so.6]
0x7f2ed1429d8e
0x7f2ed1429e3e
[./tutorial_10]
0x40be83_start at ???
0xffffffffffffffff

Current migration level =0
Requested migration level up to4
Migrating to level1
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTOINCREMENT, lastname VARCHAR[80], firstname VARCHAR[80]]
Migrating to level2
Running : INSERT INTO users [`lastname`, `firstname`] VALUES ['Taft', 'Tucker']
Migrating to level3
Running : INSERT INTO users [`lastname`, `firstname`] VALUES ['Dewar', 'Robert']
Migrating to level4
Running : INSERT INTO users [`lastname`, `firstname`] VALUES ['Botton', 'David']
```

Prise en main de GNOGA



---

```
Done migration
Saving new migration_level = 4
```

---

Puis lancer :

---

```
user@system:./tutorial_10
```

---

Sortie Web :

---

```
Obtain list of tables from: test
Display list of tables: test
Table Name in database : gnogaparams
Table Name in database : sqlite_sequence
Table Name in database : users
Field Name in users : id
Field Name in users : lastname
Field Name in users : firstname
Column Name : id
Data Type : INTEGER
Field Type : INTEGER
Field Opts :
Field Size : 0
Decimals : 0
Can Be Null : FALSE
Default :
Column Name : lastname
Data Type : VARCHAR[80]
Field Type : VARCHAR
Field Opts : 80
Field Size : 80
Decimals : 0
Can Be Null : FALSE
Default :
Column Name : firstname
Data Type : VARCHAR[80]
Field Type : VARCHAR
Field Opts : 80
Field Size : 80
Decimals : 0
Can Be Null : FALSE
Default :
This will escape things like a '' or " or \ or whatever may be needed like a ` based
on the particular database engine.
id => 1
lastname => Taft
firstname => Tucker
id => 2
lastname => Dewar
firstname => Robert
id => 3
lastname => Botton
firstname => David
```

---

## 1.11 Tutoriel 11

---

```
user@system:./tutorial_11 setup

gnogaparams table created
2023-08-14 14:56:40.76 : Error End_of_Recordset.
2023-08-14 14:56:40.76 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-ser-
ver-database-sqlite.adb:421
[./tutorial_11]
```

---

Prise en main de GNOGA



---

```
0x6248db gnoga__server__database__sqlite__next at ???
0x685feb gnoga__server__model__find_where at ???
0x64801a gnoga__server__migration__migrate_to at ???
0x650a30 gnoga__server__migration__setup at ???
0x6521b8 gnoga__server__migration__migrations_handled_command_line at ???
0x40fd76 _ada_tutorial_11 at ???
0x42167d main at ???
[/lib/x86_64-linux-gnu/libc.so.6]
0x7f6cf029d8e
0x7f6cf029e3e
[./tutorial_11]
0x40be83 _start at ???
0xffffffffffffffe

Current migration level =0
Requested migration level up to5
Migrating to level1
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTOINCREMENT, lastname VARCHAR[80], firstname VARCHAR[80]]
Migrating to level2
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Taft', 'Tucker')
Migrating to level3
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Dewar', 'Robert')
Migrating to level4
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Botton', 'David')
Migrating to level5
Running : CREATE TABLE `foods` [id INTEGER PRIMARY KEY AUTOINCREMENT, user_id Integer, food VARCHAR[80]]
Done migration
Saving new migration_level = 5
```

---

Puis lancer :

---

```
user@system: ./tutorial_11
```

---

Sortie Web :

---

```
Using Gnoga.Server.Model.Table
Records in table : 3
Record : 1
First Name : Tucker
Last Name : Taft
Record : 2
First Name : Robert
Last Name : Dewar
He Likes : Pears
The first thing he liked was Pears
Record : 3
First Name : David
Last Name : Botton
He Likes : Apples
He Likes : Oranges
The first thing he liked was Apples
```

---

## 2 Démos

### 2.1 AdaBlog

Blog avec persistance sur base de données SQLite.

Prise en main de GNOGA

[www.soweb.io](http://www.soweb.io)  
[developpement@soweb.io](mailto:developpement@soweb.io)



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

ed. 186 of 2023-09-12  
page 25 of 119

```

user@system: ./adablog setup

gnogaparams table created
2023-08-14 14:59:01.08 : Error End_Of_Recordset.
2023-08-14 14:59:01.09 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-server-database-sqlite.adb: 421
[./adablog]
0x6956f3 gnoga__server__database__sqlite__next at ???
0x6f6e03 gnoga__server__model__find_where at ???
0x6b8e32 gnoga__server__migration__migrate_to at ???
0x6c1848 gnoga__server__migration__setup at ???
0x6c2fd0 gnoga__server__migration__migrations_handled_command_line at ???
0x40c0d9 _ada_adablog_main at ???
0x40de35 main at ???
[/lib/x86_64-linux-gnu/libc.so.6]
0x7f3685c29d8e
0x7f3685c29e3e
[./adablog]
0x40be83 _start at ???
0xfffffffffffffe

Current migration level =0
Requested migration level up to2
Migrating to level1
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTOINCREMENT, username VARCHAR[80], pass VARCHAR[80], last_session VARCHAR[80])
Migrating to level2
Running : CREATE TABLE `blog_entries` [id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER, entry_date DATE, entry_text TEXT]
Done migration
Saving new migration_level = 2

```

---

Puis lancer :

---

```

user@system: ./adablog

```

---

Sortie Web :

### Prise en main de GNOGA



## 2.2 AdaEdit

Éditeur de code en Ada. Il permet d'accéder et de modifier temporairement les sources puisque l'application ne permet pas de sauvegarder les modifications.  
Utilise le Plugin Ace\_Editor.

```
html/boot.html [text/html]
js/boot.js [text/javascript]
js/jquery.min.js [text/javascript]
html/favicon.ico [image/x-icon]
js/ace-builds/src-min-noconflict/ace.js [text/javascript]
js/ace-builds/src-min-noconflict/theme-monokai.js [text/javascript]
js/ace-builds/src-min-noconflict/mode-ada.js [text/javascript]
```

## 2.3 Chattanooga

Erreur à la connexion : On\_Connect\_Submit: raised PROGRAM\_ERROR : a-cohama.adb:472 access before elaboration

```
0x44657f Chattanooga.Db.User_Maps.Find at a-cohama.adb: 472
0x445199 Chattanooga.Db.User_Maps.Contains at a-cohama.adb: 255
0x44fb79 Chattanooga.Db.Control.ExistsN at chattanooga-db.adb: 128
0x44fc51 Chattanooga.Db.Control.ExistsSP at chattanooga-db.adb: 122
0x44edb3 Chattanooga.Db.Exists at chattanooga-db.adb: 54
0x41f9f9 Chattanooga.Ui.On_Connect_Submit at chattanooga-ui.adb: 230
0x4aedef0 Gnoga.Gui.Base.Fire_On_Submit at gnoga-gui-base.adb: 1082
0x4c8289 Gnoga.Gui.Base.On_Message at gnoga-gui-base.adb: 1988
0x6bd368 Gnoga.Server.Connection.Dispatch_Task_TypeT at gnoga-server-
connection.adb: 1767
```

```
html/boot.html [text/html]
js/boot.js [text/javascript]
js/jquery.min.js [text/javascript]
html/favicon.ico [image/x-icon]
```

## 2.4 Connect Four

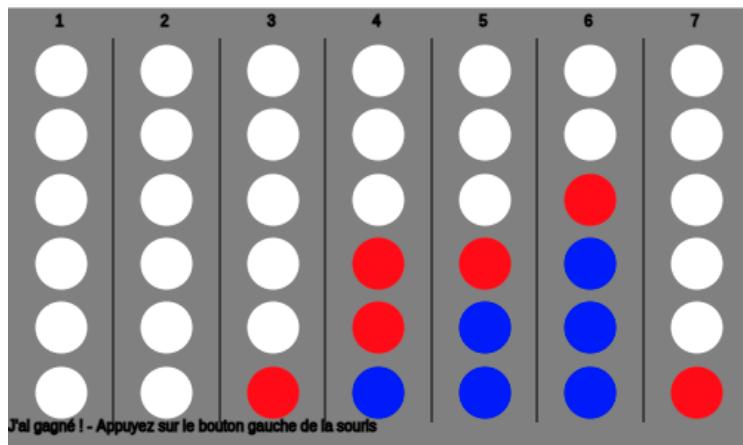
Jeu Puissance Quatre, codé avec une IA forte.  
Utilise ZanyBlue [i18n] [bibliothèque d'éléments pour framework].

Développé par Barry Fagin et Martin Carlisle, US Air Force Academy.

### Prise en main de GNOGA



Ce jeu de Puissance 4 [TM] a été codé initialement en Ada 2005 et compilé avec le compilateur JVM-GNAT. Adapté pour GNOGA par Pascal Pignard [20150613].



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

## 2.5 Leaves Main

Candy Crush Like, avec des feuilles. Utilise le Plugin PixiJS pour le rendu graphique. Manipulation de sprites.

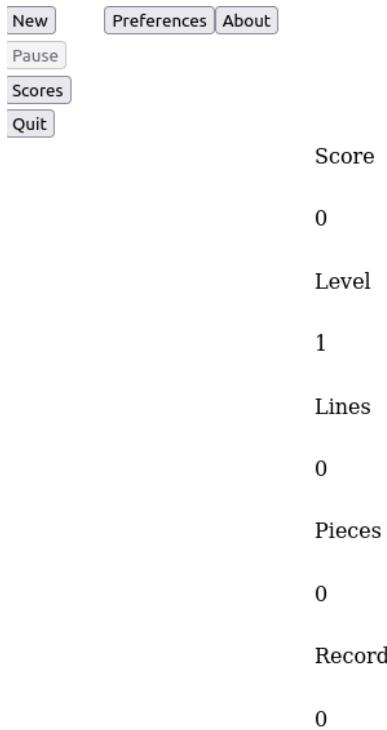


html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]  
html/favicon.ico [image/x-icon]  
js/pixi.min.js [text/javascript]  
img/board\_bkg\_1.jpg [image/jpeg]  
img/d\_0540.png [image/png]  
img/d\_0537.png [image/png]  
img/d\_0542.png [image/png]  
img/d\_0541.png [image/png]  
img/d\_0536.png [image/png]  
img/d\_0544.png [image/png]

Prise en main de GNOGA

## 2.6 Linxtris

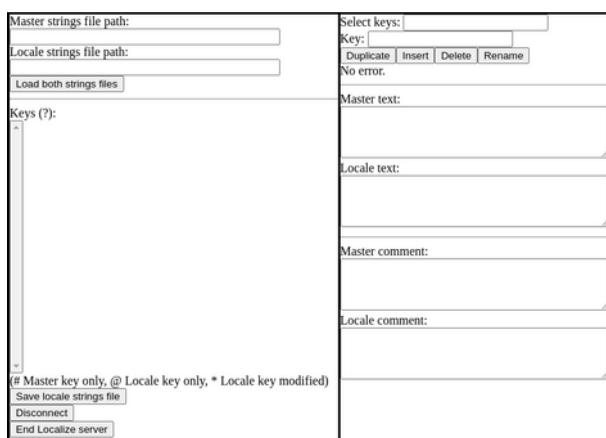
Jeu Tetris. Se lance mais non fonctionnel.



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

## 2.7 Localize

Gestionnaire de localisation. Procédure d'utilisation inconnue. Lire le readme.md dans ~/demo/localize



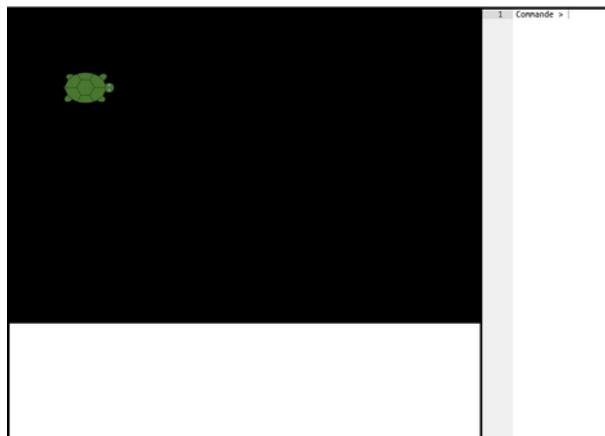
html/boot.html [text/html]  
js/boot.js [text/javascript]

Prise en main de GNOGA

js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

## 2.8 Logo

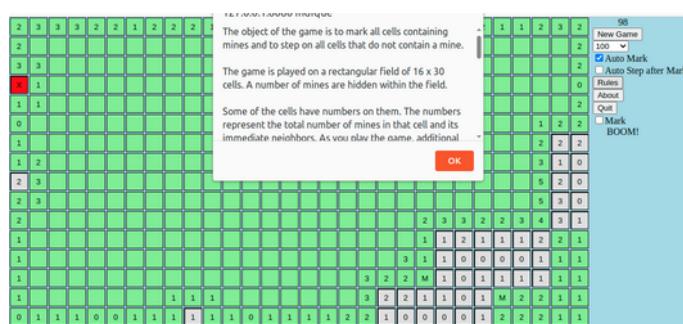
Déplacer une tortue via des lignes de commande.



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]  
js/pixi.min.js [text/javascript]  
js/ace-builds/src-min-noconflict/ace.js [text/javascript]  
img/hector.png [image/png]

## 2.9 Mine Detector

Jeu démineur.



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

## 2.10 Movies

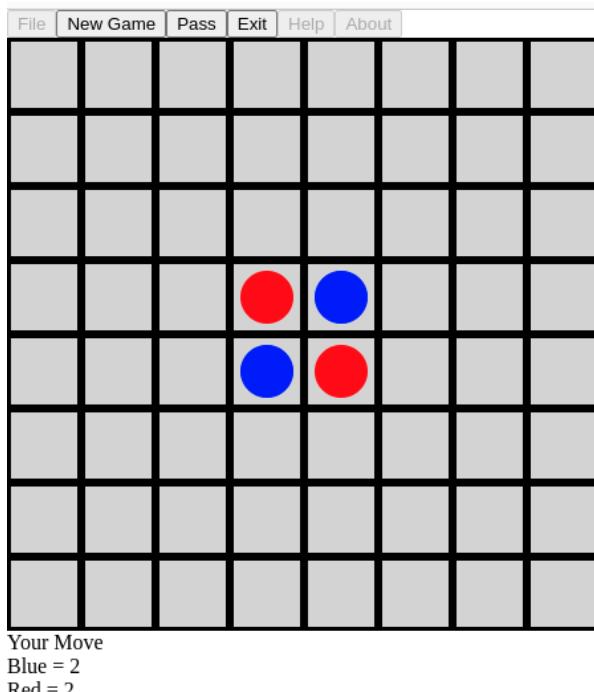
Zones de saisie qui permettent de stocker dans une liste des films puis de les chercher.

Source et binaire non présents.

Prise en main de GNOGA

## 2.11 Othello

Jeu Othello. Codé avec une IA très forte.



```
html/boot.html [text/html]
js/boot.js [text/javascript]
js/jquery.min.js [text/javascript]
html/favicon.ico [image/x-icon]
```

## 2.12 Password Gen

Générateur de mots de passe avec critère de longueur, de symbole, etc.

Source présent, binaire non construit.

## 2.13 Random Int

Générateur de nombre aléatoire entre deux bornes.

Minimum value	<input type="text" value="1"/>
Maximum value	<input type="text" value="10"/>
<input type="text" value="8"/>	<input type="button" value="Generate"/>
<input type="button" value="Quit"/>	

```
html/boot.html [text/html]
js/boot.js [text/javascript]
js/jquery.min.js [text/javascript]
html/favicon.ico [image/x-icon]
```

## 2.14 Snake

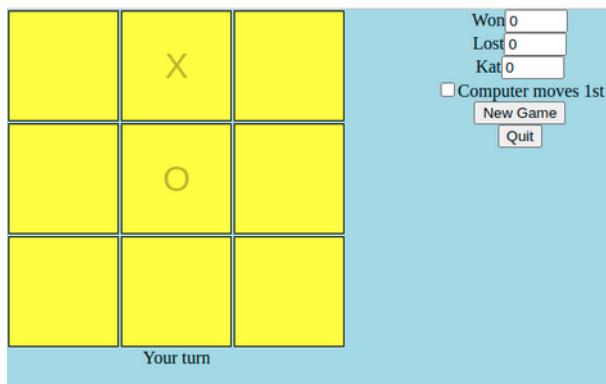
Jeu du Snake.

Prise en main de GNOGA

html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

## 2.15 Tic tac toe

## Jeu du morpion avec une IA forte.



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

3 Test

### 3.1 Align

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

## 3.2 Bootjs

<http://127.0.0.1:8080>

[go to bootjs\\_demo.html](#)

---

Test of French - Bienvenue.

---

HTML Injection from bootjs\_demo.html:

---

## This page has pre-existing html

---

We turn it in to a Gnoga ready page simply by adding a script tag that points to /js/boot.js

Once boot.js has been loaded, it is then possible from Gnoga to attach as needed to existing IDs or use Gnoga.Client.Bind\_Page to attach to all IDs on the page.

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

[http://127.0.0.1:8080/bootjs\\_demo.html](http://127.0.0.1:8080/bootjs_demo.html)

This page has pre-existing html

---

We turn it in to a Gnoga ready page simply by adding a script tag that points to /js/boot.js

Once boot.js has been loaded, it is then possible from Gnoga to attach as needed to existing IDs or use Gnoga.Client.Bind\_Page to attach to all IDs on the page.

html/boot.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

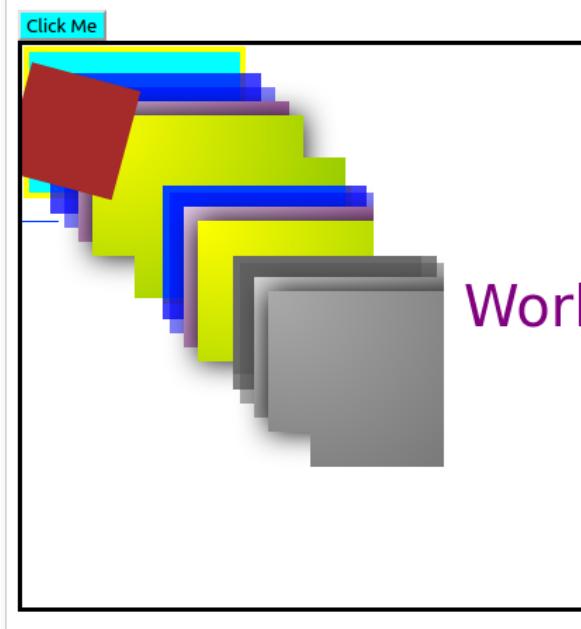
### 3.3 boot\_strap\_demo

# Gnoga The GNU Omnificent GUI for Ada

The Ada Open-Source Mission-Critical Cloud, Desktop and Mobile Application Development Framework

```
html/boot_bootstrap3.html [text/html]
js/jquery.min.js [text/javascript]
css/bootstrap.min.css [text/css]
css/bootstrap-theme.min.css [text/css]
js/bootstrap.min.js [text/javascript]
js/auto.js [text/javascript]
```

### 3.4 canvas\_test



html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.5 cli

```
Enter: azeazeaezaze
Line :@azeazeaezaze@
Enter exit to end session,
quit to end app: ls
Line (20 max):@ls@
Enter: exit
Line :@exit@
Enter exit to end session,
quit to end app: quit
Line (20 max):@quit@
End.
```

html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]  
js/ace-builds/src-min-noconflict/ace.js [text/javascript]

### 3.6 db\_active

Initialiser la base de données :

---

```
user@system: ./db_active setup OK

2023-08-14 10:39:26.03 : Openning database file: ./active_test.db
gnogaparams table created
2023-08-14 10:39:26.05 : Error End_Of_Recordset.
2023-08-14 10:39:26.06 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-server-database-sqlite.adb: 421
[./db_active]
0x624857 gnoga__server__database__sqlite__next at ???
0x685f67 gnoga__server__model__find_where at ???
```

---

Prise en main de GNOGA

```
0x647f96 gnoga__server__migration__migrate_to at ???
0x6509ac gnoga__server__migration__setup at ???
0x652134 gnoga__server__migration__migrations_handled_command_line at ???
0x40ff38 _ada_db_active at ???
0x422639 main at ???
[/lib/x86_64-linux-gnu/libc.so.6]
0x7fbe73a29d8e
0x7fbe73a29e3e
[./db_active]
0x40be83_start at ???
0xffffffffffffffe

Current migration level =0
Requested migration level up to5
Migrating to level1
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTOINCREMENT, lastname VARCHAR[80], firstname VARCHAR[80]]
Migrating to level2
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Taft', 'Tucker')
Migrating to level3
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Dewar', 'Robert')
Migrating to level4
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Botton', 'David')
Migrating to level5
Running : CREATE TABLE `foods` [id INTEGER PRIMARY KEY AUTOINCREMENT, user_id Integer, food VARCHAR[80]]
Done migration
Saving new migration_level = 5
```

---

Puis lancer :

---

```
user@system: ./db_active
```

---

Sortie Web :

---

```
Using Gnoga.Server.Model.Table
Records in table : 3
Record : 1
First Name : Tucker
Last Name : Taft
Record : 2
First Name : Robert
Last Name : Dewar
He Likes : Pears
He Likes : Pears
The first thing he liked was Pears
Record : 3
First Name : David
Last Name : Botton
He Likes : Apples
He Likes : Oranges
He Likes : Apples
He Likes : Oranges
The first thing he liked was Apples
```

---

## 3.7 db\_mysql

Pour tester :

- Créer une DB et un user avec mot de passe ayant les droits sur cette DB

– Renseigner ces identifiants en ligne 54 de db\_mysql.adb > Connection.Connect [Database => "xxx", Host => "localhost", User => "xxx", Password => "xxx"];

Initialiser la base de données :

---

```
user@system: ./db_mysql setup
2023-08-14 10:03:50.83 : Opening connection to MySQL database.

gnogaparams table created
2023-08-14 10:03:51.08 : Error End_Of_Recordset.
2023-08-14 10:03:51.08 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-server-database-mysql.adb:445
[./db_mysql]
0x61f677 gnoga__server__database__mysql__next at ???
0x67c20b gnoga__server__model__find_where at ???
0x63e23a gnoga__server__migration__migrate_to at ???
0x646c50 gnoga__server__migration__setup at ???
0x6483d8 gnoga__server__migration__migrations_handled_command_line at ???
0x40f0bf _ada_db_mysql at ???
0x41dfb9 main at ???
[/lib/x86_64-linux-gnu/libc.so.6]
0x7f22a5c29d8e
0x7f22a5c29e3e
[./db_mysql]
0x40bfa3 _start at ???
0xfffffffffffffe

Current migration level =0
Requested migration level up to4
Migrating to level1
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTO_INCREMENT, lastname VARCHAR[80], firstname VARCHAR[80]]
Migrating to level2
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Taft', 'Tucker')
Migrating to level3
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Dewar', 'Robert')
Migrating to level4
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Botton', 'David')
Done migration
Saving new migration_level = 4
```

---

Puis lancer :

---

```
user@system: ./db_mysql
```

---

Sortie Web :

---

```
Obtain list of tables from: test
Table Name in database : gnogaparams
Table Name in database : point
Table Name in database : users
Field Name in users : id
Field Name in users : lastname
Field Name in users : firstname
Column Name : id
Data Type : int[11]
Field Type : int
Field Opts : 11
Field Size : 11
Decimals : 0
Can Be Null : FALSE
```

---

Prise en main de GNOGA

---

```
Default :  
Column Name : lastname  
Data Type : varchar[80]  
Field Type : varchar  
Field Opts : 80  
Field Size : 80  
Decimals : 0  
Can Be Null : TRUE  
Default :  
Column Name : firstname  
Data Type : varchar[80]  
Field Type : varchar  
Field Opts : 80  
Field Size : 80  
Decimals : 0  
Can Be Null : TRUE  
Default :  
I've been thinking.. \"escaped\" \\ is it?  
id => 1  
lastname => Taft  
firstname => Tucker  
id => 2  
lastname => Dewar  
firstname => Robert  
id => 3  
lastname => Botton  
firstname => David
```

---

## 3.8 db\_sqlite

Initialiser la base de données :

---

```
user@system: ./db_sqlite setup  
2023-08-14 15:18:35.54 : Openning database file: ./test.db  
gnogaparams table created  
2023-08-14 15:18:35.56 : Error End_Of_Recordset.  
2023-08-14 15:18:35.57 : raised GNOGA.SERVER.DATABASE.END_OF_RECORDSET : gnoga-server-database-sqlite.adb:421  
[./db_sqlite]  
0x62022f gnoga__server__database__sqlite__next at ???  
0x68193f gnoga__server__model__find_where at ???  
0x64396e gnoga__server__migration__migrate_to at ???  
0x64c384 gnoga__server__migration__setup at ???  
0x64db0c gnoga__server__migration__migrations_handled_command_line at ???  
0x40ee33 _ada_db_sqlite at ???  
0x41e011 main at ???  
[/lib/x86_64-linux-gnu/libc.so.6]  
0x7f0f05629d8e  
0x7f0f05629e3e  
[./db_sqlite]  
0x40be83 _start at ???  
0xffffffffffffffff  
  
Current migration level =0  
Requested migration level up to4  
Migrating to level1  
Running : CREATE TABLE `users` [id INTEGER PRIMARY KEY AUTOINCREMENT, lastname VARCHAR[80], firstname VARCHAR[80]]  
Migrating to level2  
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Taft', 'Tucker')  
Migrating to level3  
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Dewar', 'Robert')  
Migrating to level4  
Running : INSERT INTO users (`lastname`, `firstname`) VALUES ('Botton', 'David')  
Done migration
```

---

Prise en main de GNOGA

---

```
Saving new migration_level = 4
```

---

Puis lancer :

---

```
user@system: ./tutorial_10
```

---

Sortie Web :

---

```
Obtain list of tables from: test
Display list of tables: test
Table Name in database : gnogaparams
Table Name in database : sqlite_sequence
Table Name in database : users
Field Name in users : id
Field Name in users : lastname
Field Name in users : firstname
Column Name : id
Data Type : INTEGER
Field Type : INTEGER
Field Opts :
Field Size : 0
Decimals : 0
Can Be Null : FALSE
Default :
Column Name : lastname
Data Type : VARCHAR[80]
Field Type : VARCHAR
Field Opts : 80
Field Size : 80
Decimals : 0
Can Be Null : FALSE
Default :
Column Name : firstname
Data Type : VARCHAR[80]
Field Type : VARCHAR
Field Opts : 80
Field Size : 80
Decimals : 0
Can Be Null : FALSE
Default :
I've been thinking.. "escaped" \ is it?
id => 1
lastname => Taft
firstname => Tucker
id => 2
lastname => Dewar
firstname => Robert
id => 3
lastname => Botton
firstname => David
```

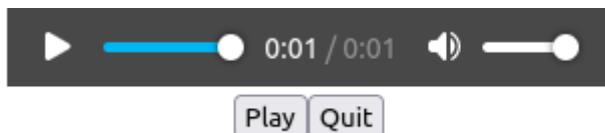
---

### 3.9 demo [aka gnoga-demo]



html/debug.html [text/html]  
 js/jquery.min.js [text/javascript]  
 js/boot.js [text/javascript]

### 3.10 ding



html/boot.html [text/html]  
 js/jquery.min.js [text/javascript]  
 js/boot.js [text/javascript]

### 3.11 essai8

~/test/tickets/007

Prise en main de GNOGA

0  
1  
2  
3  
4  
5

Mouse coordinates X, Y:

210,10  
218,24  
225,35  
242,63  
267,96  
293,126  
309,143

### 3.12 essai10b

~/test/tickets/019

Keyboard characters:

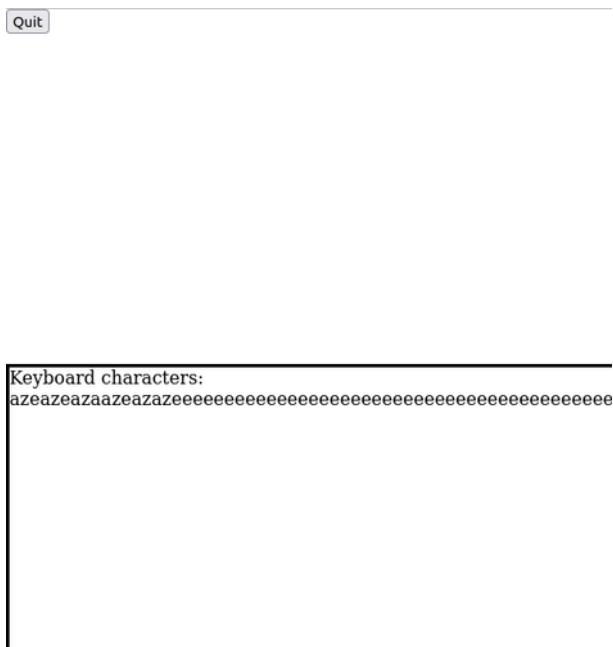
```
1 Keyboard characters 2:qsdqsdqdqsdqsd
2 qsdqsdqdsqsdqsdqsdqsdqsdqsd
3 qsdqsdqsdqsdqsdqsdqsdqsdqsd
4 qsdqsdqsdqsdqsdqsdqsdqsd
5 qsdqsdqsdqsdqsdqsdqsd
6 qsdqsdqsdqsd
7 qsdqsdqsd
8 |
```

### 3.13 essai11

~/test/tickets/011

Prise en main de GNOGA

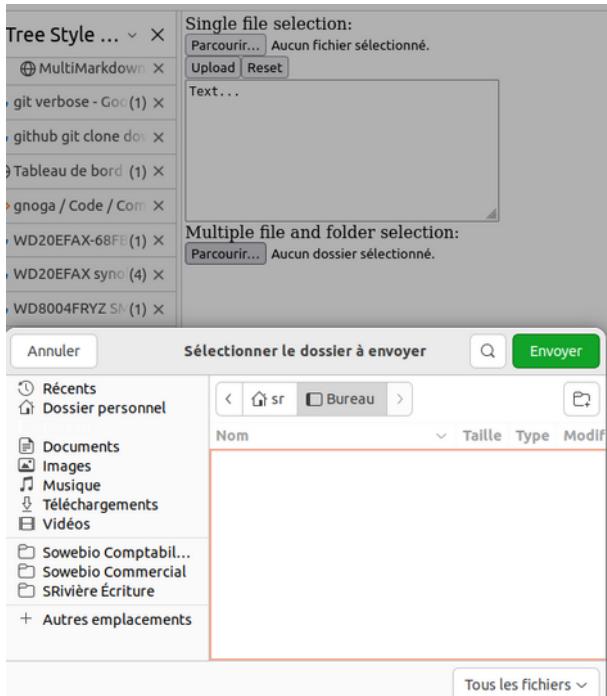




## Sortie console :

```
2023-08-06 15:22:50.15 : KEY_DOWN, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.15 : KEY_PRESS, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.18 : KEY_DOWN, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.18 : KEY_PRESS, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.21 : KEY_DOWN, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.21 : KEY_PRESS, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.24 : KEY_DOWN, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.24 : KEY_PRESS, 32, , FALSE, FALSE, FALSE, FALSE
2023-08-06 15:22:50.26 : KEY_UP, 32, , FALSE, FALSE, FALSE, FALSE
```

## 3.14 files\_ops



html/boot.html [text/html]

## Prise en main de GNOGA

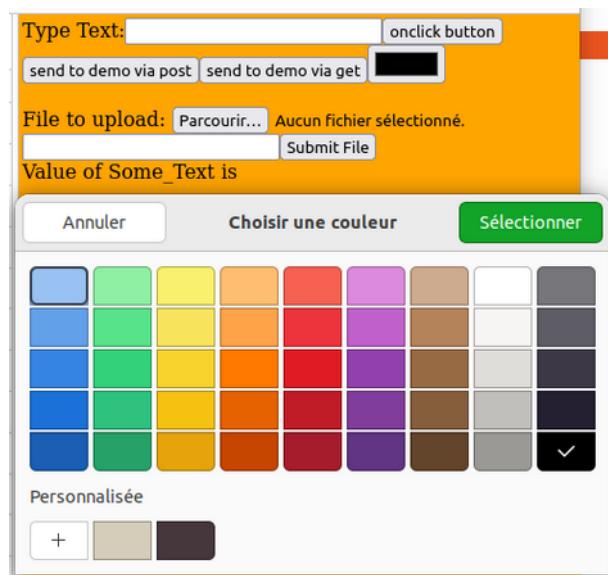


CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

ed. 186 of 2023-09-12  
page 41 of 119

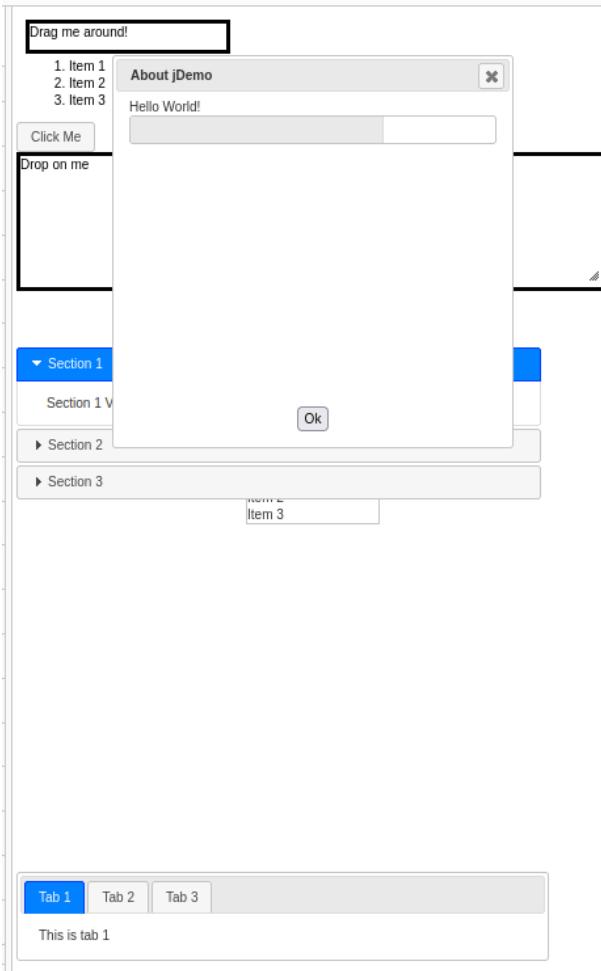
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.15 forms



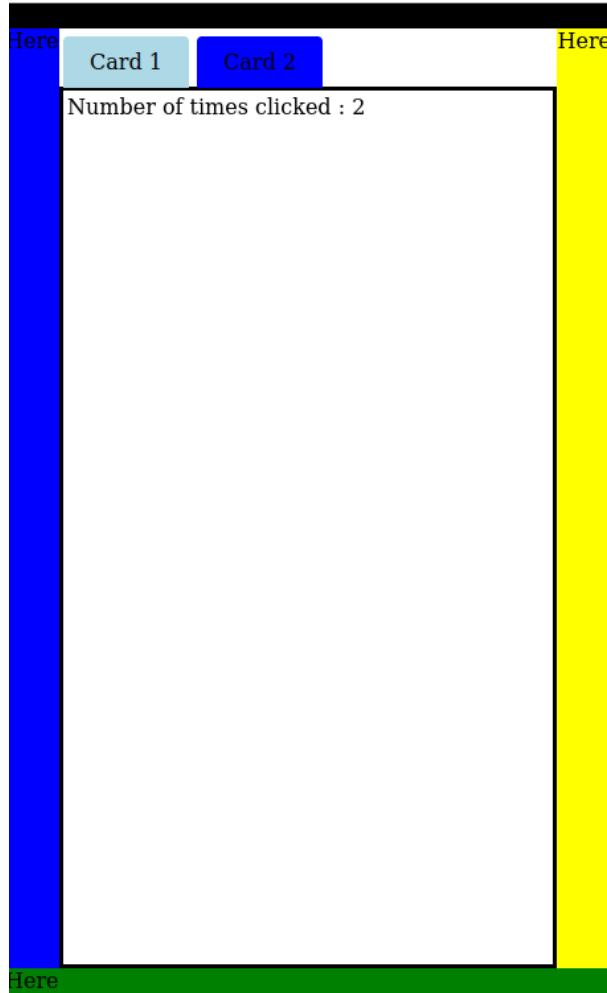
html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.16 jdemo



```
html/boot_jqueryui.html [text/html]
js/jquery.min.js [text/javascript]
css/jquery-ui.min.css [text/css]
js/jquery-ui.min.js [text/javascript]
js/auto.js [text/javascript]
css/images/ui-icons_444444_256x240.png [image/png]
css/images/ui-icons_ffffff_256x240.png [image/png]
css/images/ui-icons_777777_256x240.png [image/png]
css/images/ui-icons_555555_256x240.png [image/png]
```

### 3.17 layouts

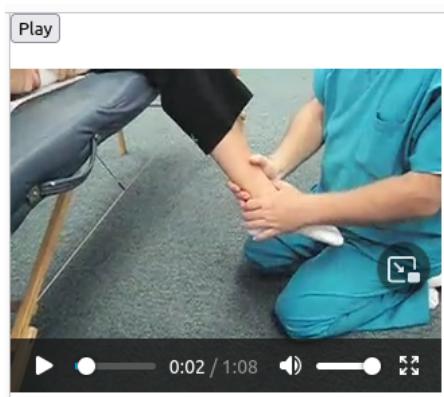


html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.18 mac\_test

Non pertinent dans notre contexte.

### 3.19 media



html/boot.html [text/html]

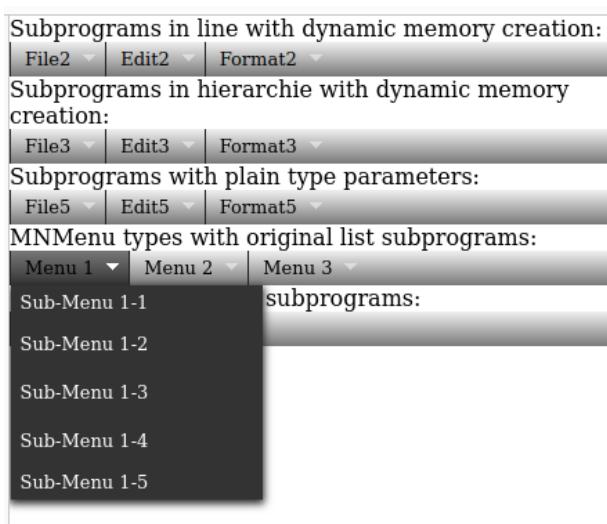
Prise en main de GNOGA



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.20 menu



html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]  
css/mnmenu.css [text/css]  
js/jquery.mnmenu.js [text/java  
css/mnmenu.css [text/css]  
js/jquery.mnmenu.js [text/java  
img/menuLight.png [image/png]  
img/menuArrowsFFFFFF.png [imag  
img/menuDark.png [image/png]

### 3.21 message\_boxes



html/boot\_jqueryui.html [text/html]  
js/jquery.min.js [text/javascript]  
css/jquery-ui.min.css [text/css]  
js/jquery-ui.min.js [text/javascript]  
js/auto.js [text/javascript]

css/images/ui-icons\_777777\_256x240.png [image/png]  
css/images/ui-icons\_555555\_256x240.png [image/png]

### 3.22 multiuser

## Multi\_Connect App Demo

19

57

---

Click to close the connection  
Click to close down server  
[Gnoga Home Page](#)

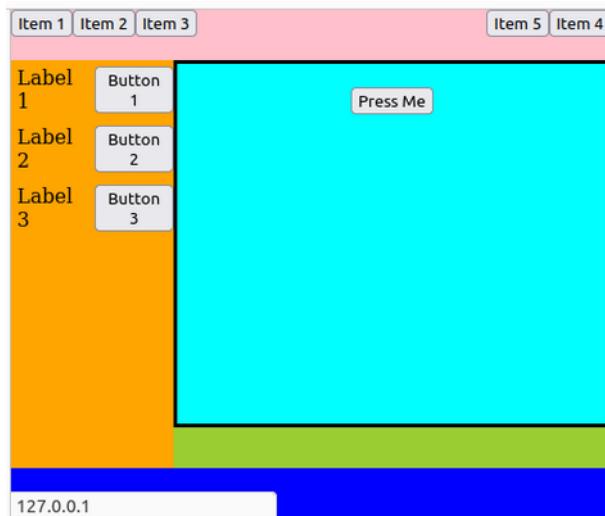
---



html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

New connection ID 1  
New connection ID 2  
.../...

### 3.23 pack



html/ajax.html [text/html]  
js/jquery.min.js [text/javascript]  
js/ajax.js [text/javascript]

### 3.24 pixi\_sprite\_test

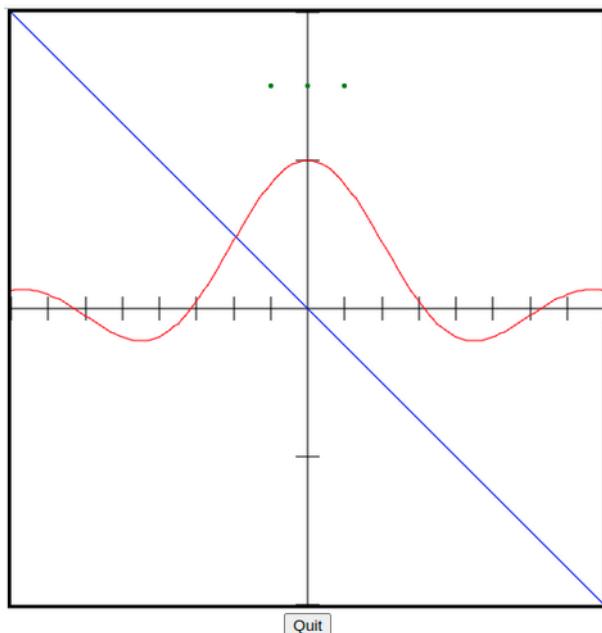
Démo non fonctionnelle avec Firefox. Fonctionnelle avec Chrome.

Prise en main de GNOGA



html/debug.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]  
js/pixi.min.js [text/javascript]  
img/E4a.png [image/png]

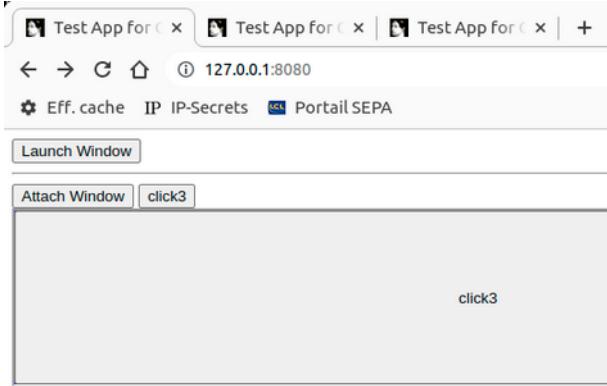
### 3.25 plot\_test



html/boot.html [text/html]  
js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

### 3.26 popups

Ouvre un nouvel onglet.



```
html/debug.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

### 3.27 pragma\_sqlite

Sortie console :

---

```
At creation Short_Column_Names: TRUE
After swap Short_Column_Names: FALSE
At creation Full_Column_Names: FALSE
After swap Full_Column_Names: TRUE
At creation Encoding: UTF-8
At creation UTF8_SString: TRUE
Insert and read Blansec Adèle
users.id => 1
users.lastname => Blansec
users.firstname => Adèle
With UXStrings the result is ok because the text is stored in database following UTF8_String attribut, then is read in the same way. Here UTF-8 text is expected by database regarding Encoding attribut.
After swap UTF8_SString: FALSE
Insert and read Lupin Arsène
users.id => 1
users.lastname => Blansec
users.firstname => Adèle
users.id => 2
users.lastname => Lupin
users.firstname => Arsène
The result 1 is incorrect because the value stored in database as UTF-8 is read as Latin-1.
The result 2 is ok because the value is stored in database as Latin 1 then is read as latin-1.
```

---

### 3.28 singleton

Application Gtk

Sortie console :

---

```
2023-08-06 13:07:37.54 : Sending to single route for Mozilla/5.0 (X11; Linux x86_64; rv:98.0) Gecko/20100101 Firefox/98.0 on Linux x86_64 from 127.0.0.1:34040
2023-08-06 13:07:37.54 : Connection established.
2023-08-06 13:07:37.54 : You can add to the path of app, e.g. http://url:8080/abc
2023-08-06 13:07:37.54 : if a file exists in server /html/ will be served if not
2023-08-06 13:07:37.54 : the app boot file will be served.
2023-08-06 13:07:37.58 : path on url : /
2023-08-06 13:07:37.58 : page_id can be set using http://url:8080?page_id=xxx
2023-08-06 13:07:37.63 : page_id = undefined
2023-08-06 13:07:37.68 : User Agent = Mozilla/5.0 (X11; Linux x86_64; rv:98.0) Gecko/20100101 Firefox/98.0
```

---

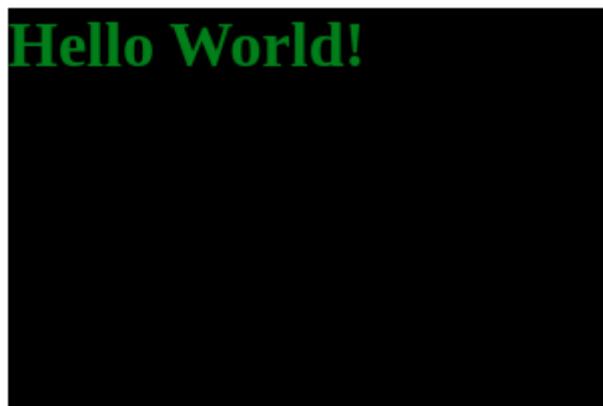
```
2023-08-06 13:07:55.32 : Deleting connection ID 1
2023-08-06 13:07:55.32 : Message loop released.
2023-08-06 13:07:55.32 : Done.
2023-08-06 13:07:55.32 : Application terminating.
2023-08-06 13:07:55.34 : Normal exit of task: main_task_0000000001D0C340
```

---

Fenêtre Gtk :



Sortie navigateur sur localhost:8080 :

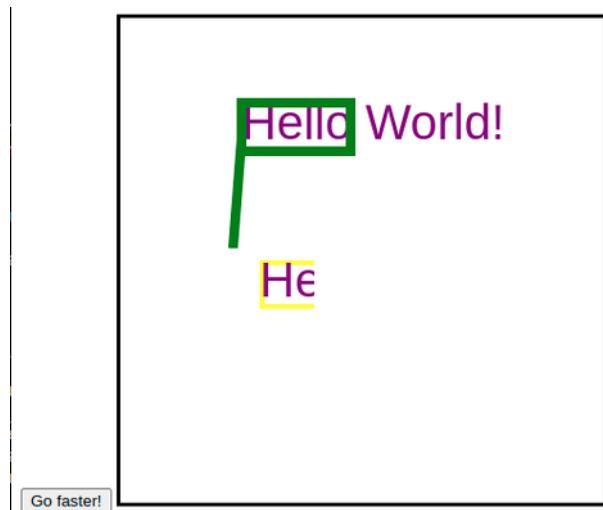


---

**Click when Done**

Prise en main de GNOGA

### 3.29 sprite\_test



js/boot.js [text/javascript]  
js/jquery.min.js [text/javascript]  
html/favicon.ico [image/x-icon]

### 3.30 storage

Mémorisation locale de session sur un premier navigateur :

Click me and I will hide.  
Last access was at null  
Last access was at 1659012599  
Session id is 202272812493500038530000

Lecture de l'ID de session après le clic :

Last access was at 1659012599  
Session id is 202272812493500038530000

Sur un autre navigateur :

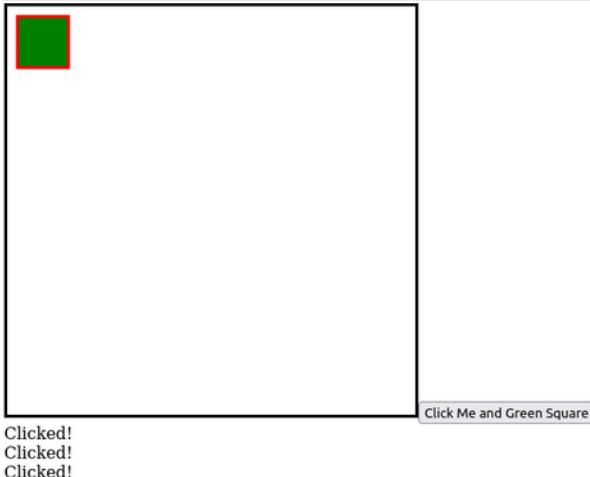
Click me and I will hide.  
Last access was at null  
New session assigned.  
Session id is 202272812523400358918000

En console :

```
2022-07-28 14:49:59.69 : Session 1 = ID : 202272812493500038530000
2022-07-28 14:49:59.69 : Local 1 = Last_View : 1659012599
2022-07-28 14:49:59.69 : Number of active connections = 1

2022-07-28 14:52:34.36 : Session 1 = gid : 202272812523400358918000
2022-07-28 14:52:34.36 : Local 1 = Last_View : 1659012754
2022-07-28 14:52:34.36 : Number of active connections = 2
```

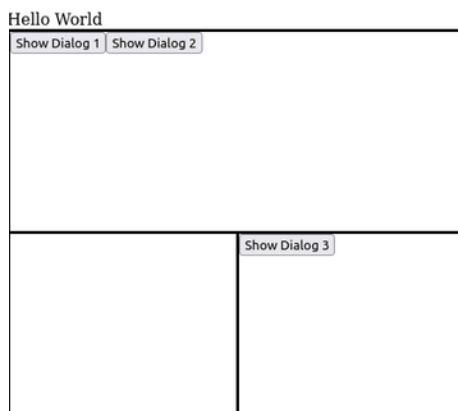
### 3.31 svg\_demo



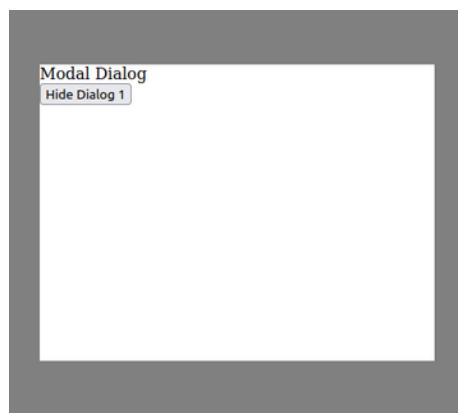
html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

### 3.32 test\_modal\_dialog

Écran à l'init :



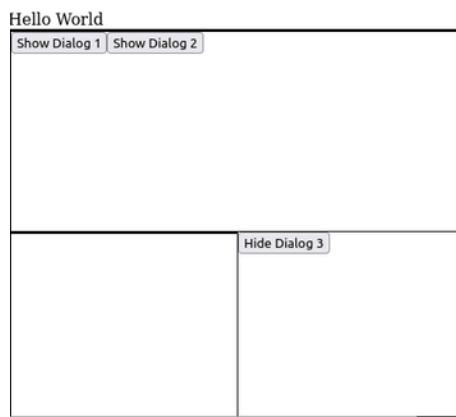
[Show Dialog 1]



[Hide Dialog] > [Show Dialog 2]

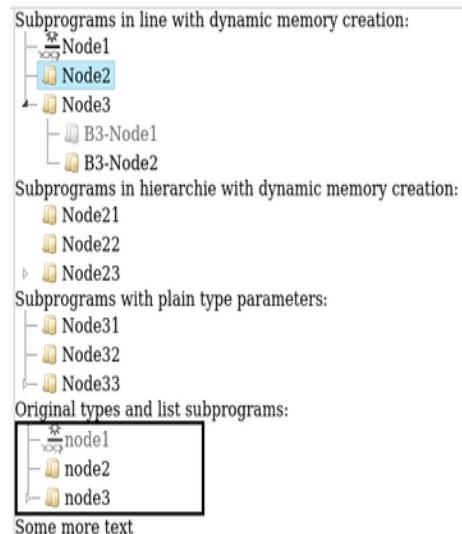


[Hide Dialog] > [Show Dialog 3]



html/debug.html [text/html]  
js/jquery.min.js [text/javasc  
js/boot.js [text/javascript]

### 3.33 tree



html/debug.html [text/html]

```
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
css/jstree_themes/default/style.min.css [text/css]
js/jstree.min.js [text/javascript]
img/gnoga_tiny.png [image/png]
css/jstree_themes/default/32px.png [image/png]
```

### 3.34 ttables

#### My First Table

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

```
html/debug.html [text/html]
js/jquery.min.js [text/javascript]
js/boot.js [text/javascript]
```

## 4 Test SSL

### 4.1 layouts\_ssl

#### Création du certificat SSL

Supprimer server.crt et server.key.

Avec le nom de station « ro6.genesix.org », créer le certificat via Let's Encrypt avec votre générateur préféré<sup>4</sup> et installez-le, par exemple selon cette arborescence :

---

```
└── certs
    ├── ca.cer
    ├── fullchain.cer
    ├── ro6.genesix.org.cer
    ├── ro6.genesix.org.conf
    ├── ro6.genesix.org.csr
    ├── ro6.genesix.org.csr.conf
    └── ro6.genesix.org.key
└── html
    ├── debug.html
    ├── favicon.ico
    └── robots.txt
└── js
    ├── boot.js
    ├── jquery.min.js
    ├── layouts_ssl
    ├── layouts_ssl.adb
    └── test_ssl.gpr
```

---

<sup>4</sup> <https://letsencrypt.org/docs/client-options>

## Modifications de layouts\_ssl.adb

Afin d'avoir une expérience « complète » en https, il peut être préférable de choisir le port 443 au lieu de 8443. Les ports inférieurs à 1024 nécessitant les droits root via sudo, cette option est à envisager en fonction de vos droits.

Commenter dans layouts\_ssl.adb, aux alentours de la ligne 143 :

---

```
Gnoga.Server.Connection.Secure.Register_Secure_Server
  [Certificate_File => Gnoga.Server.Application_Directory & "/test_ssl/server.-
crt",
   Key_File => Gnoga.Server.Application_Directory & "/test_ssl/server.key",
   Port      => 8_443, Disable_Insecure => False];
```

---

et remplacer par :

---

```
Gnoga.Server.Connection.Secure.Register_Secure_Server
  [Certificate_File => "certs/ro6.genesix.org.cer",
   Key_File => "certs/ro6.genesix.org.key",
   Port => 443, Disable_Insecure => False];
```

---

Ou Port => 8443 si vous n'avez pas les droits sudo ou root.

## Créer test\_ssl\_ins.gpr

Puisque gnoga.gpr est déjà installé dans l'environnement de développement :

## Commenter

---

```
with "../src/gnoga.gpr"
```

---

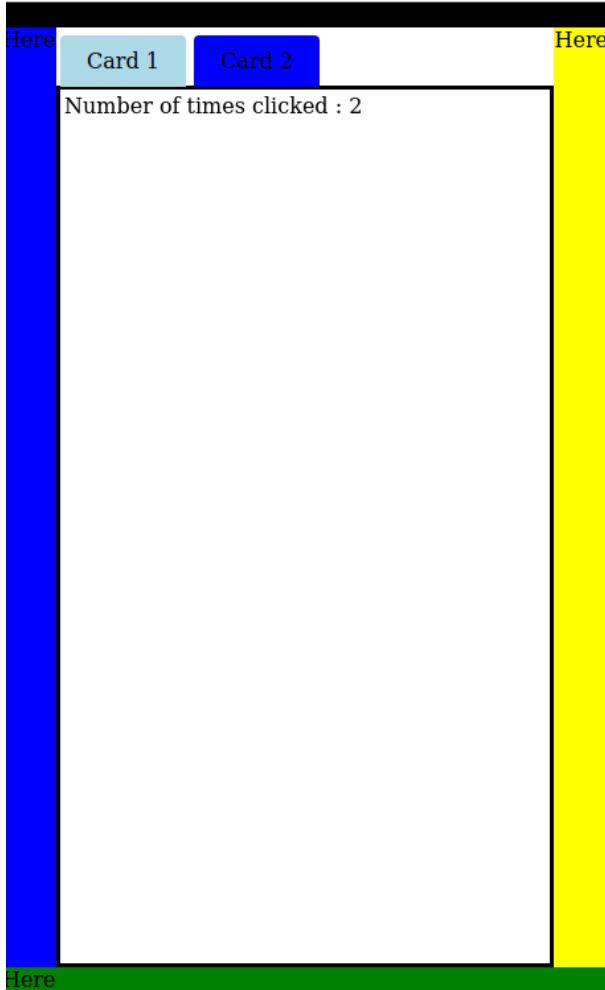
## Remplacer

---

```
-       with "../src/gnoga.gpr"
+       with "gnoga.gpr"
```

---

On retrouve alors l'affichage de l'application layouts :



html/debug.html [text/html]  
js/jquery.min.js [text/javascript]  
js/boot.js [text/javascript]

## 5 Documentation gnoga-wf.pdf<sup>5</sup>

Cette section est dédiée au document gnoga-wf.pdf.

### 5.1 Création d'un projet GNOGA avec gnoga\_make – version originale

Gnoma\_make permet de créer le squelette d'un projet GNOGA :

---

```
gnoga_make new new_app multi_connect
[Gnoga_Make Version 0.0 - pre-alpha]
.../...
```

---

Crée une nouvelle application nommée : new\_app avec le template de base multi\_connect (pour application multi-connexions).

<sup>5</sup> Premiers pas avec GNOGA, Pascal Pignard [<https://blady.pagesperso-orange.fr/telechargements/gnoga/gnoga.pdf>]

Templates :

- multi\_connect [par défaut]
- hello\_world
- singleton

Template Directory : .../gnoga/templates/gnoga\_make/

Cela permet de créer l'environnement de travail du projet : bin/, css/, html/, img/, js/, obj/, src/, les fichiers new\_app.gpr et .adb qui se situent dans le répertoire src/, et enfin un Makefile qui se situe dans le répertoire principal.

En fonction du templates choisi, d'autres fichiers seront potentiellement créés. Par défaut, les fichiers créés sont :

new\_app.ads, new\_app.gpr, new\_app-main.adb, new\_app-controller.adb, new\_app-controller.ads, new\_app-view.adb, new\_app-view.ads et le Makefile.

Il y a des erreurs à la compilation immédiatement après avoir créé le template. Une correction a été réalisée sur le template Multi\_Connect puisque c'est celui qui nous intéresse le plus. Les autres templates n'ont pas été corrigés. Cependant, la correction serait similaire.

Une fois la commande gnoga\_make utilisée, le projet est créé. Il contient alors déjà du code mais aussi quelques erreurs empêchant la compilation.

▷ Les modifications suivantes visent à faire compiler le code.

#### □ new\_app-main.adb

Modifier :

---

```
-      Gnoga.Log [Ada.Exceptions.Exception_Name (E) & " - "
-      & Ada.Exceptions.Exception_Message (E)];
+      Gnoga.Log (E);
```

---

#### □ new\_app-view.ads

Ajouter :

---

```
+      with UXStrings; use UXStrings;
```

---

#### □ new\_app-view.ads et new\_app-view.adb

Supprimer :

---

```
-      overriding
```

---

## □ new\_app-view.ads

Modifier :

---

```
procedure Create
  [View   : in out Default_View_Type;
  Parent : in out Gnoga.Gui.Base.Base_Type'Class;
-  ID     : in      String  := "";
+  ID     : in      UXString := ""];
```

---

## □ new\_app-view.adb

Modifier :

---

```
procedure Create
  [View   : in out Default_View_Type;
  Parent : in out Gnoga.Gui.Base.Base_Type'Class;
-  ID     : in      String  := "";
+  ID     : in      UXString := ""];
```

---

## □ new\_app-controller.adb

Modifier :

---

```
-      View.Create [Main_Window];
+      View.Click_Button.Create [Main_Window];
```

---

▷ Ça compile !

Le .gpr se situe dans les sources, il est possible de le mettre dans le répertoire courant mais il faut alors effectuer les changements suivants dans le fichier .gpr :

---

```
-      for Source_Dirs use [". "];
-      for Object_Dir use "../obj";
-      for Exec_Dir use "../bin";
+
+      for Source_Dirs use ["src"];
+      for Object_Dir use "obj";
+      for Exec_Dir use "bin";
```

---

Le code initial de multi\_connect semble assez différent du code des démos. Le découpage même des fichiers et répertoires semblent différer. Il est alors difficile de partir des démos pour développer une application à partir de multi\_connect.

Vu les nombreuses erreurs générées par les templates et l'organisation des fichiers qui peut paraître compliquée, la base la plus saine pour commencer un projet serait les tutoriels.

Le tutoriel 03 pour les applications multi-connexions, le tutoriel 01 pour les applications singltons. Contrairement au code généré par gnoga\_make, les tutoriels compilent parfaitement sans avoir à modifier les sources. L'organisation des fichiers semble également plus simple [il n'y a pas vraiment d'organisation puisque le peu de

code est contenu dans un voire deux fichiers maximum, au lieu de 5... ] et contiennent tous les objets et fonctions nécessaires au développement de l'application [On\_Connect, Main\_View, Main\_Window, etc.].

◊ Suite à cette expérience, nous avons créé une nouvelle version de gnoga\_make pour le rendre plus simple d'utilisation, corriger les erreurs et intégrer les fonctions de base comme dans les tutoriels, afin d'avoir déjà une base minimale pour compiler et afficher une page Web.

## 5.2 Création d'un projet GNOGA avec gnoga\_make – version modifiée

En partant des tutoriels 02 et 03, respectivement pour les applications singletons et multi-connexions, pour cette nouvelle version de gnoga\_make. Les templates ne sont plus les mêmes :

- multi\_connect [défaut]
- singleton
- multi\_plugin

hello\_world a été supprimé car il n'apportait rien de plus que le template singleton.

Le code des fichiers <template>.adb.tpl est documenté à Modifications des sources.

Le template multi\_plugin ajoute à multi\_connect la totalité des fichiers JavaScript, CSS, et HTML nécessaire pour faire fonctionner l'ensemble des plugins.

La commande peut être utilisée de la même façon et permet de choisir un nom de projet, et les différents fichiers s'adapteront à ce nom. Les répertoires sont organisés d'une manière similaire : on retrouve bin/, obj/, js/, css/, html/, src/ mais cette fois-ci le .gpr est dans le répertoire courant du projet [./]. Le seul fichier de code est le .adb dans les sources [src/]. L'utilisateur est donc libre de rajouter ou non des fichiers.

Le code généré du .gpr, comprends les commandes classiques qui permettent la compilation. Elles peuvent être modifiées à la volée. Néanmoins, quelques modifications ont été apportées. En effet, on avait par exemple :

---

```
for Exec_Dir use Settings.Exec_Dir;
```

---

...pour détailler le répertoire de build, ce qui n'est pas très clair pour l'utilisateur [il peut ne pas comprendre qu'il s'agit du répertoire "bin/" à moins de regarder le fichier de settings]. Pour moins d'ambiguïté, "Settings.Exec\_Dir" a été remplacé par "bin/".

Dans le code généré du .adb, les types App\_Data et App\_Access ont été ajoutés, ainsi que la fonction On\_Connect et les objets View et Window qui sont indispensables à la création d'une application.

Un bouton pour quitter l'application, présent dans les tutoriels, en guise de premier élément de l'application, a également été ajouté. Il est facilement supprimable et peut faire office de rappel dans un contexte de formation.

Les anciens templates ont été conservés dans :

---

```
.../gnoga/templates/gnoga_make/.ancien_gnoga_make
```

---

Une sauvegarde des nouveaux templates a été créée dans :

---

```
.../gnoga/templates/gnoga_make/.nouveau_gnoga_make
```

---

## 5.3 Utilisation basique

On apprend ensuite à insérer des boutons et des zones de saisies sans les positionner [voir section Diverses méthodes sur les objets]. Il en est de même pour une case à cocher et un curseur.

### □ Intégrer un bouton

---

```
-- Imports
with Gnoga.Gui.Base;
with Gnoga.Gui.Element.Common;

<...>

-- Déclaration
Quit_Button : Gnoga.Gui.Element.Common.Button_Type;

<...>

-- Procédure associé à l'utilisation du bouton
procedure On_Quit [Object : in out Gnoga.Gui.Base.Base_Type' Class]
is
    pragma Unreferenced [Object];
begin
    Gnoga.Application.Singleton.End_Application;
end On_Quit;

<...>

-- Main
Quit_Button.Create [Main_View, "Quitter"];
Quit_Button.On_Click_Handler [On_Quit'Unrestricted_Access];
```

---

### □ Ajout d'une zone de saisie

---

```
-- Imports
with Gnoga.Gui.Element.Form;

<...>

-- Déclaration
Fact_Form : Gnoga.Gui.Element.Form.Form_Type;
Input_Text : Gnoga.Gui.Element.Form.Text_Type;
Question_Label : Gnoga.Gui.Element.Form.Label_Type;

<...>

-- Main
Fact_Form.Create [Main_View];
Input_Text.Create [Fact_Form, 40];
Question_Label.Create [Fact_Form, Input_Text, "Saisir un nombre entier : "];
```

---

## □ Ajout d'une case à cocher et d'un curseur

---

```
-- Déclaration
Dec_Check_Box : Gnoga.Gui.Element.Form.Check_Box_Type;
Dec_Label : Gnoga.Gui.Element.Form.Label_Type;
Dec_Range : Gnoga.Gui.Element.Form.Range_Type;
Dec_Range_Label : Gnoga.Gui.Element.Form.Label_Type;
Dec_Value_Label : Gnoga.Gui.Element.Form.Label_Type;

<...>

-- Main
Dec_Check_Box.Create [Fact_Form];
Dec_Label.Create [Fact_Form, Dec_Check_Box, "Case à cocher", False];
Dec_Range.Create [Fact_Form];
Dec_Range.Minimum [0];
Dec_Range.Maximum [10];
Dec_Range.Value [0];
Dec_Value_Label.Create [Fact_Form, Dec_Range, "0", False];
Dec_Range_Label.Create [Fact_Form, Dec_Range, "Value", False];
Dec_Range.On_Change_Handler [Dec_Range_Change' Unrestricted_Access];
```

---

## 5.4

### Les résultats des formulaires

Suite au paragraphe « création de formulaires », nous allons voir comment récupérer les informations envoyées par l'utilisateur.

Il y a trois méthodes pour récupérer les informations d'un formulaire :

- La méthode GET : les données sont ajoutées à l'URI spécifiée dans le formulaire avec un encodage particulier, à utiliser principalement pour la consultation, la taille des données est limitée ;
- La méthode POST : les données sont envoyées par une transaction spécifique, à utiliser pour la modification, la taille n'est pas limitée ;
- La méthode des événements : les données sont récupérées soit au changement d'un élément, au clic d'un bouton ou à la soumission du formulaire.

La mise en place du formulaire est la même pour les deux premières méthodes, mais la récupération des données est différente.

## 5.5

### Méthode Get

La procédure suivante permet de récupérer l'ensemble des données :

```
procedure Resultats
  [Main_Window : in out Gnoga.Gui.Window.Window_Type' Class;
  Connection :
    access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
  is

    pragma Unreferenced [Connection];
    Main_View : Gnoga.Gui.View.Console.Console_View_Type;

  begin
    Main_View.Create [Main_Window];
    if Main_Window.Location.Search /= "" then

      Main_View.Put_Line [Main_Window.Document.Input_Encoding];
      Main_View.Put_Line [Main_Window.Location.Search];
```

---

---

```

        for C in Gnoga.Gui.Location.Parse [Main_Window.Location.Search, Main_Win-
dow.Document.Input_Encoding].Iterate
            loop
                begin
                    Main_View.Put_Line
                        [ Gnoga.Types.Data_Maps.Key [C] & Gnoga.Types.Data_Maps.Element
[C]];
                end;
            end loop;

        end if;
    end Resultats;

```

---

On récupère un à un tous les champs du formulaire et on remplit une autre page avec le nom du champ ainsi que la réponse.

## 5.6 Méthode Post

On indique les champs que l'on veut récupérer de la façon suivante :

---

```

procedure On_Post_Request
    [URI           : in      UXString;
     Accepted_Parameters : out UXString]
is
    pragma Unreferenced [URI];
begin
    Accepted_Parameters := From_Latin_1
        ["Texte multi-ligne, Champ Caché, Case à Cocher, Boutons radio, Image," &
         "Texte, Mel, MDP, URL, Recherche, Couleur, Date, Heure, Mois, Semaine," &
         "Date heure locale, Nombre, Glisseur, Sélection, Image.x, Image.y, Fichier, Télé-
phone"];
end On_Post_Request;

procedure On_Post
    [URI           : in      UXString;
     Parameters : in out Gnoga.Types.Data_Map_Type]
is
    pragma Unreferenced [URI];
begin
    Last_Parameters := Parameters;
end On_Post;

procedure Resultats
    [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
     Connection  : access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
is
    pragma Unreferenced [Connection];
    Main_View : Gnoga.Gui.View.Console.Console_View_Type;
begin
    Main_View.Create [Main_Window];
    for C in Last_Parameters.Iterate loop
        begin
            Main_View.Put_Line
                ["POST parameter: " & Gnoga.Types.Data_Maps.Key [C] & " = " & Gnoga.-
Types.Data_Maps.Element [C]];
        end;
    end loop;
    Last_Parameters.Clear;
end Resultats;

<...>

Gnoga.Application.Multi_Connect.On_Connect_Handler [Resultats' Unrestricted_Access, "/
resultats"];
-- Erreurs aux deux lignes suivantes
Gnoga.Server.Connection.On_Post_Handler [On_Post' Unrestricted_Access];

```

---

```
Gnoga.Server.Connection.On_Post_Request_Handler[On_Post_Request'Unrestricted_Access];
```

---

Pour compiler et s'exécuter sans erreurs, le code original doit être modifié [voir Anomalies corrigées].

## 5.7 Stockage de données

### □ Stockage de session

---

```
with Gnoga.Client.Storage;

type App_Data_Type is new Gnoga.Types.Connection_Data_Type with record
    Main_View : Gnoga.Gui.View.Console.Console_View_Type;
    Mon_Bouton : Gnoga.Gui.Element.Common.Button_Type;
    Session : Gnoga.Client.Storage.Session_Storage_Type;
end record;

-- On Action / Input
App_Data_Access [Object.Connection_Data].Session.Set ["TEXT", Client_Text];

-- On Connect
App_Data.Session := Gnoga.Client.Storage.Session_Storage [Main_Window];
App_Data.Session.Set ["TEXT", " -> "];
```

---

### □ Stockage local

---

```
with Gnoga.Client.Storage;

type App_Data_Type is new Gnoga.Types.Connection_Data_Type with record
    Main_View : Gnoga.Gui.View.Console.Console_View_Type;
    Mon_Bouton : Gnoga.Gui.Element.Common.Button_Type;
    Mon_Formulaire : Gnoga.Gui.Element.Form.Form_Type;
    Mon_Texte_Multi : Gnoga.Gui.Element.Form.Text_Area_Type;
    Local : Gnoga.Client.Storage.Local_Storage_Type;
end record;

procedure On_Envie [Object : in out Gnoga.Gui.Base.Base_Type'Class] is
begin
    App_Data_Access [Object.Connection_Data].Main_View.Put_Line ["Texte envoyé : "
& App_Data_Access [Object.Connection_Data].Mon_Texte_Multi.Value];
    App_Data_Access [Object.Connection_Data].Mon_Texte_Multi.Value [""];
    App_Data_Access [Object.Connection_Data].Local.Set ["TEXT", ""];
end On_Envie;

procedure On_Modif_Texte [Object : in out Gnoga.Gui.Base.Base_Type'Class] is
begin
    App_Data_Access [Object.Connection_Data].Local.Set ["TEXT", App_Data_Access
[Object.Connection_Data].Mon_Texte_Multi.Value];
end On_Modif_Texte;

procedure On_Modif_Stockage
    [Object
        : in out Gnoga.Gui.Base.Base_Type'Class;
    Storage_Event : in Gnoga.Gui.Window.Storage_Event_Record]
is
begin
    if Storage_Event.Name = "TEXT" then
        App_Data_Access [Object.Connection_Data].Mon_Texte_Multi.Value
[Storage_Event.New_Value];
    end if;
end On_Modif_Stockage;
```

---

```
-- On Connect
Main_Window.On_Storage_Handler [On_Modif_Stockage' Unrestricted_Access];
App_Data.Local := Gnoga.Client.Storage.Local_Storage [Main_Window];
```

Les quelques anomalies rencontrées sont référencées au chapitre Anomalies dans gno-ga-wf.pdf.

## 6 Fuites mémoires

### 6.1 Valgrind

Nous avons lancé des commandes valgrind sur différents binaires dans GNOGA pour vérifier les fuites mémoires. Certaines applications en ont plus que d'autre. Nous avons pris l'exemple de la démo « mine\_detector » et avons arrêté le programme après une partie. De plus, la commande faisait visiblement ralentir l'exécution du programme.

```
user@system: valgrind --leak-check=full --show-reachable=yes ./mine_detector
```

```
==4308== LEAK SUMMARY:
==4308==   definitely lost: 4,978,928 bytes in 622,149 blocks
==4308==   indirectly lost: 0 bytes in 0 blocks
==4308==   possibly lost: 53,736 bytes in 33 blocks
==4308==   still reachable: 701,284 bytes in 2,104 blocks
==4308==           suppressed: 0 bytes in 0 blocks
==4308==
==4308== For counts of detected and suppressed errors, rerun with: -v
==4308== ERROR SUMMARY: 5444 errors from 5444 contexts (suppressed: 0 from 0)
```

<https://this-page-intentionally-left-blank.org>

Prise en main de GNOGA



CC-by-nc-sa: Attribution + Noncommercial + ShareAlike

[www.soweb.io](http://www.soweb.io)  
[developpement@soweb.io](mailto:developpement@soweb.io)

ed. 186 of 2023-09-12  
page 64 of 119

# Utilisation de GNOGA

---

## 1 Environnement de développement

Pour créer une application avec GNOGA, pas uniquement des fichiers Ada sont nécessaires. En effet, comme pour n'importe quelle page Internet, on a besoin d'un fichier HTML, voir de fichiers CSS et JavaScript.

Ainsi, pour chaque bibliothèque utilisée [jQuery, MNMenu...] des fichiers CSS et JavaScript correspondant sont nécessaires pour un affichage correct des éléments graphiques.

Certains de ces fichiers n'étaient pas ajoutés nativement par la commande gnoga\_make : seul le fichier HTML, indispensable, était ajouté. Ainsi, nous avons apporté des modifications sur la commande gnoga\_make afin de pouvoir ajouter à l'environnement de développement de l'application l'ensemble des fichiers nécessaires [voir Crédit pour la création d'un projet GNOGA avec gnoga\_make – version modifiée].

Dans le cas de l'utilisation de la commande gnoga\_make originale, ces fichiers doivent être copiés depuis ".../gnoga/js", ".../gnoga/css" etc. Une fois copiés dans le répertoire de travail, il est tout à fait possible de modifier ces fichiers afin de pouvoir changer la forme des widgets, où même des animations.

Une fois cela fait, il est temps de développer l'application.

## 2 Base du codage avec GNOGA

Il est nécessaire de comprendre la structure du code avec GNOGA. Dans le « main », le code sera organisé de la façon suivante :

```
-- Importation des bibliothèques nécessaires au programme.
with Gnoga;
with Gnoga.Gui.Window;
with Gnoga.Gui.View;
with Gnoga.Gui.View.Docker;
with UXStrings; use UXStrings;

procedure project_example is

-- déclaration des types des variables (pour App_Data, cf : PLUS LOIN)
type App_Data is new Gnoga.Types.Connection_Data_Type with record
    My_Window : Gnoga.Gui.Window.Pointer_To_Window_Class;
    My_Docker : Gnoga.Gui.View.Docker.Docker_View_Type;
    My_View   : aliased Gnoga.Gui.View.View_Type;
end record;

-- déclaration / implémentation des procédures
procedure On_Exit [Object : in out Gnoga.Gui.Base.Base_Type'Class];

-- La procédure On_Connect définit ce qui va se passer au moment de la connexion.
-- C'est probablement la procédure la plus importante de toutes vos futures applications avec Gnoga : c'est donc dans cette procédure qu'il est nécessaire de créer est d'ajouter le contenu de la page.

procedure On_Connect
    [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
     Connection : access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
is
    pragma Unreferenced [Connection];
```

---

Prise en main de GNOGA



---

```

App      : App_Access := new App_Data;
begin
    null;
    -- code
end On_Connect;

-- procédure principale, initialise les connexions, etc.
begin
    Gnoga.Application.Title ["Example"];
    Gnoga.Application.HTML_On_Close ["Application ended."];
    Gnoga.Application.Multi_Connect.Initialize;
    Gnoga.Application.Multi_Connect.On_Connect_Handler [Event => On_Connect' Unrestricted_Access, Path => "default"];
    Gnoga.Application.Multi_Connect.Message_Loop;
end project_example;

```

---

## 3 Types

Dans de nombreux exemples, on peut voir que les implémentations du code ajoutent de nouveaux types. Certains nous ont marqué par leur praticité.

### 3.1 App\_Data

Le type « App\_Data » peut contenir une partie des éléments de l'application. Cela répertorie tous ou partie des éléments statiques de la page.

En instanciant une variable de type App\_Data, on instancie tous les éléments qu'elle contient. Par exemple :

---

```

type App_Data is new Gnoga.Types.Connection_Data_Type with record
    Window : Gnoga.Gui.Window.Pointer_To_Window_Class;
    Docker : Gnoga.Gui.View.Docker.View_Type;
    View   : aliased Gnoga.Gui.View.View_Type;
end record;

```

---

Instancier un type App\_Data instanciera également les éléments Window, Docker et View.

Cependant, il ne faut pas mettre tous les éléments de l'application dans ce type App\_Data, en particulier les éléments qui vont être créés dynamiquement [Voir Création dynamique ou statique d'éléments].

### 3.2 Widget

Le type Widget est notamment utilisé avec les Cards. Il permet de référencer tout le contenu d'un widget. Exemple :

---

```

type My_Widget_Type is new Gnoga.Gui.View.View_Type with record
    Widget_Form : Gnoga.Gui.Element.Form.Form_Type;
    Name_Input  : Gnoga.Gui.Element.Form.Text_Type;
    Message     : Gnoga.Gui.Element.Form.Text_Area_Type;
    My_Submit   : Gnoga.Gui.Element.Form.Submit_Button_Type;
end record;

```

---

## 4 Composants GNOGA

### 4.1 Dockers

D'après le fichier de spécification des Dockers : « Un Docker permet de disposer des vues enfants en les ancrant sur les côtés de Docker\_View\_Type. Docker\_View\_Types peut être imbriqué comme tous les types de vue et tous les côtés n'ont pas besoin d'être ancrés, ce qui permet facilement des mises en page complexes. »

Dans l'optique de développer une application classique [avec des menus sur les côtés, et une fenêtre principale où s'affichera le réel contenu de la page, comme nous avons pu essayer de développer lors de cette prise en main de GNOGA], il est nécessaire de comprendre et de travailler avec les Dockers.

Les dockers permettent de placer les vues [View, Frames] sur les bords de la fenêtre ou de remplir l'espace restant dans la fenêtre. Il distribue l'espace à ses vues enfants. Cela permet de structurer rapidement des pages qui peuvent être complexes. Un docker peut être l'enfant de la fenêtre :

---

```
Main_Window.Connection_Data [App];
App.My_Window := Main_Window'Unchecked_Access;
App.My_Docker.Create [Main_Window];
```

---

Pour placer une vue tout en haut du docker :

---

```
App.My_View.Create [App.My_Docker];
App.My_Docker.Top_Dock [App.My_View'Unchecked_Access];
```

---

### 4.2 Boutons

Déclaration et instantiation d'un bouton [dans Ada] :

---

```
Button : constant Gnoga.Gui.Element.Pointer_To_Element_Class := new Gnoga.Gui.Element.Common.Button_Type;
```

---

Affichage de ce bouton [dans le DOM] :

---

```
Gnoga.Gui.Element.Common.Button_Access[View.New_Element["Exit", Button]] .Create
[View, "Exit Application"];
```

---

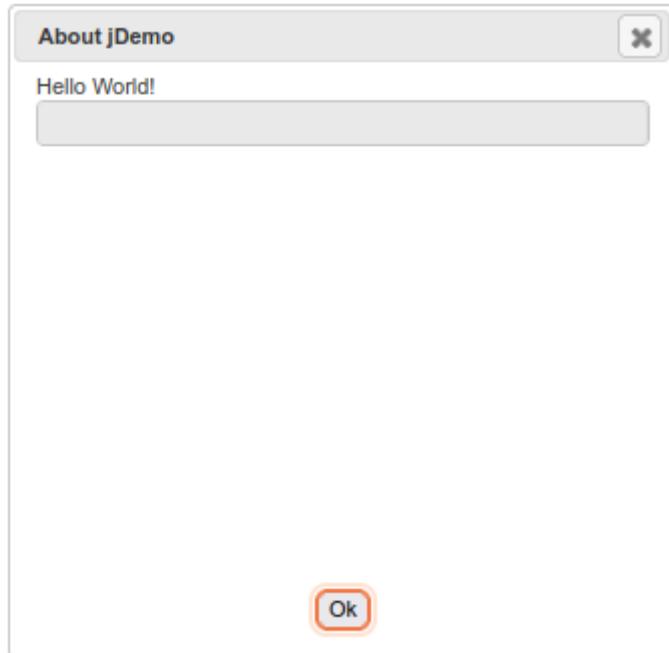
On associe le bouton à la fonction On\_Exit :

---

```
View.Element ["Exit"].On_Click_Handler [On_Exit'Unrestricted_Access];
```

---

## 4.3 Boîtes de dialogue



Exemple d'une boîte de dialogue, issu de la démo jDemo.

Déclaration :

---

```
Dialog      : aliased jQueryUI.Widget.Dialog_Type;
```

---

Les boîtes de dialogue créées peuvent être très complètes, en témoignent les nombreuses options que comporte leur création.

Il est ainsi possible de les dimensionner et les positionner telles que l'on veut, de les rendre redimensionnables, etc. Le programme jDemo est un bon exemple de ce que l'on peut faire facilement avec cet outil.

---

```
procedure Create
  [Dialog      : in out Dialog_Type;
   Parent       : in out GnoGA.Gui.Base.Base_Type' Class;
   Title        : in String;
   Content      : in String := "";
   Height       : in Natural := 0;
   Width        : in Natural := 0;
   Position_My  : in String := "center";
   Position_At  : in String := "center";
   Resizable    : in Boolean := False;
   Minimum_Height: in Natural := 150;
   Minimum_Width : in Natural := 150;
   Maximum_Height: in Natural := 0;
   Maximum_Width : in Natural := 0;
   Modal        : in Boolean := True;
   Close_On_Escape: in Boolean := True;
   Draggable    : in Boolean := True;
   ID           : in String := ""]
```

---

## 4.4 Diverses méthodes sur les objets

Différentes méthodes existent sur les objets en GNOGA afin d'éviter l'écriture directe dans les fichiers CSS. Les fonctions ou procédures en GNOGA ont alors le même nom et le même but que ces mêmes attributs en CSS. On a alors les méthodes suivantes :

---

```
Top, Left, Position [relative, absolue, fixe, statique, etc.]  
Background, Background_Color, Color  
Width, Height  
etc.
```

---

## 4.5 Autres outils

D'autres outils de développement sont disponibles sur GNOGA mais ne sont pas détaillés dans cette documentation. Il y a notamment les items déplaçables (exemple dans jDemo), les zones de saisie (tutoriels), les barres de progression (jDemo et tutoriels), les infobulles (exemple dans jDemo), etc.

## 4.6 Ajouter des images

Des images peuvent facilement être ajoutées, que ce soit des images de fond (penser à la méthode `Background_Image` en GNOGA plutôt que d'aller l'écrire dans le CSS). On peut aussi ajouter une image de la façon suivante, en déclarant au préalable l'objet.

Déclaration :

---

```
Image : Gnoga.Gui.Element.Common.IMG_Type;
```

---

Code :

---

```
-- Crédation  
App.Image.Create [Parent => App.My_View, URL_Source => "img/image.png"];
```

---

## 4.7 Breadcrumb

Les breadcrumb trail, ou fil d'Ariane en français, sont des widgets permettant de faciliter la navigation en gardant en mémoire et en affichant le chemin parcouru à l'intérieur d'un programme, d'une application.

Afin de faciliter le développement d'applications, nous en avons fait un package qui permet de créer dynamiquement une ligne de boutons successifs, formant ce fil d'Ariane. L'implémentation repose sur le principe d'étages, ou profondeurs, que nous avons nommé *depth* dans le package. Bien que rendant l'utilisation un peu plus complexe, ce mécanisme facilite l'implémentation puisqu'il permet de déterminer le nombre de boutons affichés dans le fil d'Ariane à tout instant.

Le package est visible au chapitre Modifications des sources.

## 4.8 Simple Form

`Simple_Form` est un package de manipulation de chaînes de caractères `UXStrings` et de création d'un formulaire contenant de multiples zones de saisie de texte.

Dans ce package, sont définies plusieurs fonctions :

- `Field_By_Index` : découpe la chaîne de caractères en plusieurs champs séparés par un séparateur, puis retourne le champ dont le numéro est passé en paramètres. Le séparateur doit être un caractère en UXString.
- `All_Fields` : découpe la chaîne de caractères en plusieurs champs séparés par un séparateur, puis retourne l'ensemble des champs dans une liste d'UXStrings. Le séparateur doit être un caractère en UXString.
- `Remove_First_Occurrence` : retire la première occurrence du caractère passé en paramètre. Cela doit être un caractère en UXString.
- `Remove_All_Occurrences` : retire toutes les occurrences du caractère passé en paramètre. Cela doit être un caractère en UXString.
- `Create` : créé un formulaire contenant de multiples zones de saisie de texte. Son paramètre `Strings` doit être une liste de chaînes UXString. Autant de zones de saisie de texte seront créées que de chaînes UXString dans la liste. Ces chaînes de caractères seront à gauche de la zone de saisie de texte, afin d'indiquer ce que l'on remplit.

Le package est visible au chapitre Modifications des sources.

## 5 Composants JS

Plusieurs composants Javascript ont été incorporés dans GNOGA. Plus ou moins spécialisés dans un type de widget [JsTree] ou un thème graphique [rendu 2D], ils ajoutent de nouvelles fonctionnalités aux applications GNOGA.

### 5.1 JQuery

JQuery est une bibliothèque Javascript pour le Web.

#### □ Liens

Version GNOGA : 3.5.0 [2020-04]

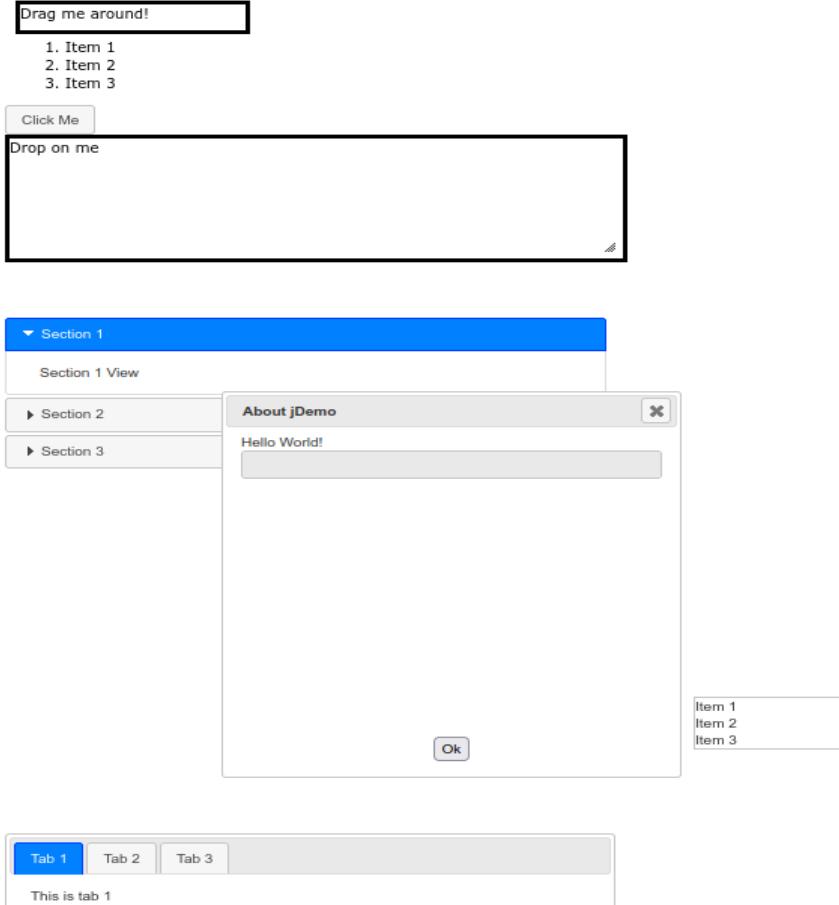
Version stable : 3.6.0 [2021-03]

### 5.2 jQuery-UI

jQuery-UI est une bibliothèque Javascript d'interface IHM, avec thèmes, construite au-dessus de jQuery.

GNOGA dispose d'une interface pour jQuery-UI.

Un excellent exemple d'application simple avec cette bibliothèque est jDemo. Cette démonstration reprend la quasi-totalité des widgets de jQuery. Elle permet de comprendre comment ceux-ci marchent sans pour autant nous donner des combinaisons de widgets complexes.



Application jDemo, reprenant les widgets de jQuery

Le fichier CSS de jQuery étant à part, il est tout à fait possible de modifier son code afin de modifier la forme des différents widgets.

Étant une bibliothèque de fonctions à part entière et très complète, la prochaine section est dédiée à quelques-uns de ses widgets utiles.

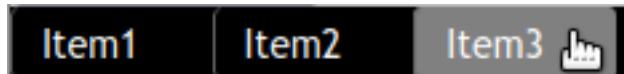
## □ Liens

<https://jquery.com>  
<https://api.jquery.com>

Version GNOGA : 3.5.0 [2020-04]  
 Version stable : 3.6.0 [2021-03]

## 5.3 MNMenu

MNMenu est composé de code Ada, JS et CSS permettant d'afficher un menu horizontal potentiellement déroulant.



Deux exemples de menus créés à partir de MNMenu.  
Le premier exemple vient de la démo de MNMenu.

Utilisation du composant :

```
with Gnoga.Gui.Plugin.MNMenu;
```

Déclaration :

```
Menu_Bar
Sub_Menu      : Gnoga.Gui.Plugin.MNMenu.MNMenu_Access;
Menu_Item_1,
Menu_Item_2,
Menu_Item_3   : Gnoga.Gui.Plugin.MNMenu.MNMenu_Item_Access;
```

Code :

```
-- Chargement de la bibliothèque
Gnoga.Gui.Plugin.MNMenu.Load_MNMenu [Main_Window];

-- Création de la barre de menu
Gnoga.Gui.Plugin.MNMenu.New_Menu_Bar [App.My_View' Unchecked_Access,      "MenuBar",
Menu_Bar, App.My_View' Unchecked_Access];

-- Ajout d'un item
Gnoga.Gui.Plugin.MNMenu.New_Item [Menu_Bar, "Item1", Menu_Item_1];

-- Ajout d'un sous-menu et de 2 items à ce sous-menu
Gnoga.Gui.Plugin.MNMenu.New_Sub_Menu_Item [Menu_Bar, Menu_Item_Edit, Sub_Menu];
Gnoga.Gui.Plugin.MNMenu.New_Item [Sub_Menu, "Item2", Menu_Item_2];
Gnoga.Gui.Plugin.MNMenu.New_Item [Sub_Menu, "Item3", Menu_Item_3];
Gnoga.Gui.Plugin.MNMenu.Display_Menu [Menu_Bar];
```

Deux implémentations existent pour créer des sous-menus :

- Crée directement un sous-menu avec la fonction New\_Sub\_Menu\_Item ;
- Crée d'abord les sous-menus comme étant des items puis on les convertit en sous-menus avec la fonction New\_Sub\_Menu.

Le fichier CSS permet de donner un effet ombré à la barre de menu. Il est obtenu en répétant sur toute la longueur du menu une ligne verticale de pixel allant du blanc au

noir, pour le dégradé. Le CSS permet aussi de griser l'item lorsque le curseur de la souris passe dessus, etc.

▷ Le composant est obsolète.

## □ Liens

<https://github.com/manusa/mnmenu>

<https://www.marcnuri.com/fr/content/mnmenu-dropdown-jquery-menu>

Version Gnoga : 0.0.19 [2014-12]

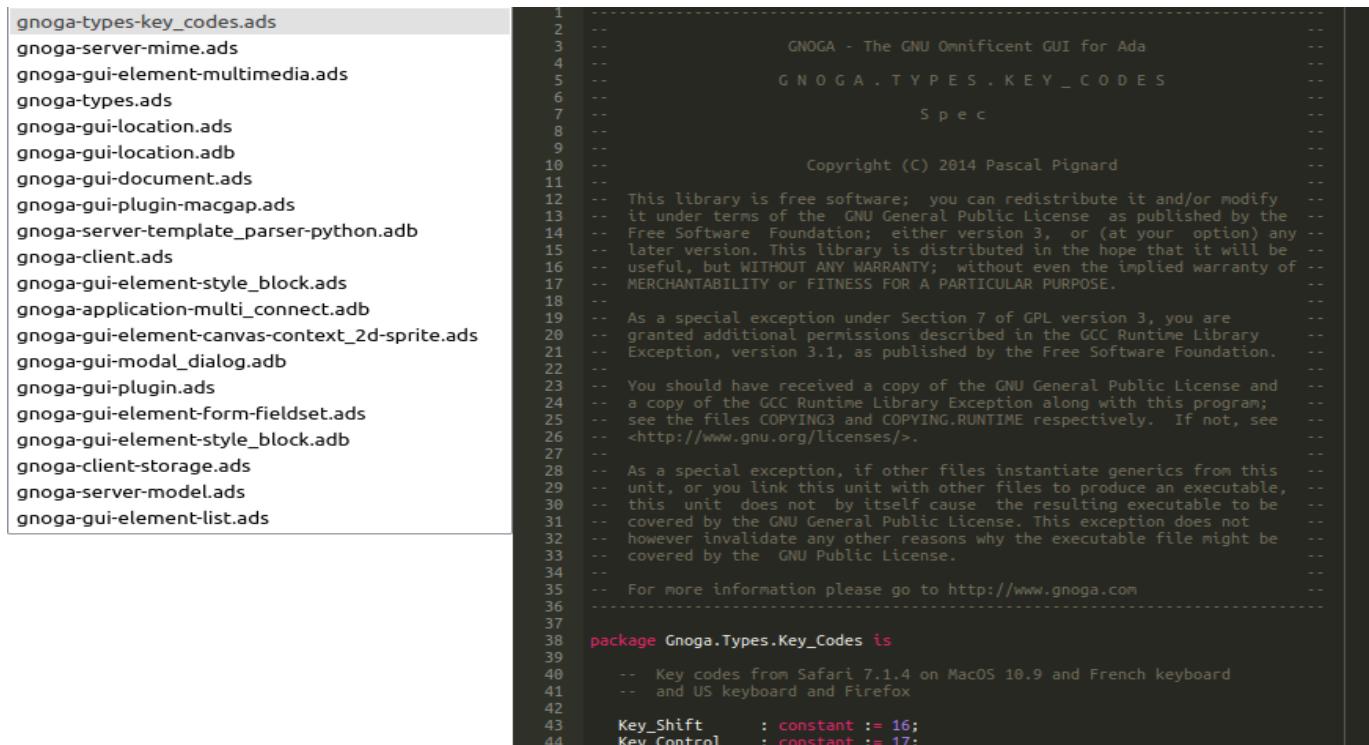
Version stable : 0.0.19 [2014-12]

## 5.4

### Ace Editor

Ace Editor est un plugin permettant de créer un éditeur de texte. Son exemple d'utilisation est la démo AdaEdit. Il n'est pas possible, en l'état, de sauvegarder les données modifiées dans la démo, mais cela doit être possible.

Ace Editor nous permet de modifier le thème de l'éditeur, de choisir le mode de coloration du texte et propose un ensemble de fonctionnalités et raccourcis simples qu'on retrouve sur la plupart des éditeurs. Cependant, certains outils plus complexes, comme les debuggers, ne sont pas présents.



```
gnoga-types-key_codes.ads
gnoga-server-mime.ads
gnoga-gui-element-multimedia.ads
gnoga-types.ads
gnoga-gui-location.ads
gnoga-gui-location.adb
gnoga-gui-document.ads
gnoga-gui-plugin-macgap.ads
gnoga-server-template_parser-python.adb
gnoga-client.ads
gnoga-gui-element-style_block.ads
gnoga-application-multi_connect.adb
gnoga-gui-element-canvas-context_2d-sprite.ads
gnoga-gui-modal_dialog.adb
gnoga-gui-plugin.ads
gnoga-gui-element-form-fieldset.ads
gnoga-gui-element-style_block.adb
gnoga-client-storage.ads
gnoga-server-model.ads
gnoga-gui-element-list.ads

1  --
2  --          GNOGA - The GNU Omnipotent GUI for Ada
3  --
4  --          G N O G A . T Y P E S . K E Y _ C O D E S
5  --
6  --          S p e c
7  --
8  --
9  --
10 --          Copyright (C) 2014 Pascal Pignard
11 --
12 -- This library is free software; you can redistribute it and/or modify
13 -- it under terms of the GNU General Public License as published by the
14 -- Free Software Foundation; either version 3, or (at your option) any
15 -- later version. This library is distributed in the hope that it will be
16 -- useful, but WITHOUT ANY WARRANTY; without even the implied warranty of
17 -- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
18 --
19 -- As a special exception under Section 7 of GPL version 3, you are
20 -- granted additional permissions described in the GCC Runtime Library
21 -- Exception, version 3.1, as published by the Free Software Foundation.
22 --
23 -- You should have received a copy of the GNU General Public License and
24 -- a copy of the GCC Runtime Library Exception along with this program;
25 -- see the files COPYING3 and COPYING.RUNTIME respectively. If not, see
26 -- <http://www.gnu.org/licenses/>.
27 --
28 -- As a special exception, if other files instantiate generics from this
29 -- unit, or you link this unit with other files to produce an executable,
30 -- this unit does not by itself cause the resulting executable to be
31 -- covered by the GNU General Public License. This exception does not
32 -- however invalidate any other reasons why the executable file might be
33 -- covered by the GNU Public License.
34 --
35 -- For more information please go to http://www.gnoga.com
36 --
37 --
38 package Gnoga.Types.Key_Codes is
39   --
40   -- Key codes from Safari 7.1.4 on Mac OS 10.9 and French keyboard
41   -- and US keyboard and Firefox
42   --
43   Key_Shift    : constant := 16;
44   Key_Control : constant := 17;
```

Application AdaEdit, utilisant le plugin AceEditor

## □ Liens

<https://ace.c9.io>

<https://github.com/ajaxorg/ace-builds/blob/master/ChangeLog.txt>

Version Gnoga : 1.4.12 [2020-07] date provenant de \$HOME/opt/gnoga/HISTORY

Version stable : 1.4.14 [2022-01]



## 5.5

### Bootstrap

Composant JS et CSS un peu similaire à jQuery-UI. Il contient cependant beaucoup moins d'éléments. Cette bibliothèque est plus focalisée sur le CSS que le Javascript, contrairement à jQuery-UI. Elle dispose de moins de fonctionnalités mais de visuels plus travaillés. Bootstrap permet de créer des tableaux, des containers, et des formulaires.

▷ La version utilisée par GNOGA est obsolète.

#### □ Liens

<https://getbootstrap.com>

<https://github.com/twbs/bootstrap/releases>

Version Gnoga : 3.3.5 [2015-06]

Version stable : 5.2.0 [2022-07]

## 5.6

### JsTree

Composant JS permettant la création d'arborescences, similaires à celles des gestionnaires de fichiers.

▷ La version utilisée par GNOGA est obsolète.

#### □ Liens

<https://getbootstrap.com>

<https://github.com/twbs/bootstrap/releases>

Version Gnoga : 3.1.1 [2015-06]

Version stable : 3.3.12 [2021-09]

## 5.7

### MacGap

Composant permettant de coder des applications adaptées à MacOS.

▷ Sans intérêt pour notre contexte

## 5.8

### PixiJS

Bibliothèque de rendu 2D WebGL très complète avec de nombreux exemples interactifs et le code correspondant.

L'implémentation GNOGA semble elle aussi assez complète. Cela permet de faire des animations autrement qu'en codant en JavaScript. Cependant, il y a très peu de fonctions qui pourrait servir pour un site internet simple. C'est une option à envisager pour une page d'accueil d'application sortant de l'ordinaire, à l'étape de la saisie des identifiants.

#### □ Liens

<https://pixijs.com>

<https://lib.haxe.org/p/pixijs/versions>

Version Gnoga : 3.3.0 [2020-06]

## 5.9 Message Boxes

Simples boîtes de dialogue paramétrables avec plusieurs boutons différents. N'apporte rien de plus que ce qu'on a déjà dans jQuery-UI dans la mesure où ces versions modifiées des boîtes sont ré-implementables.

La démo ne fonctionne pas correctement, après un premier débug le projet compile mais les boîtes de dialogue ne s'ouvrent pas.

▷ Sans intérêt pour notre contexte.

# 6 Composants jQuery-UI

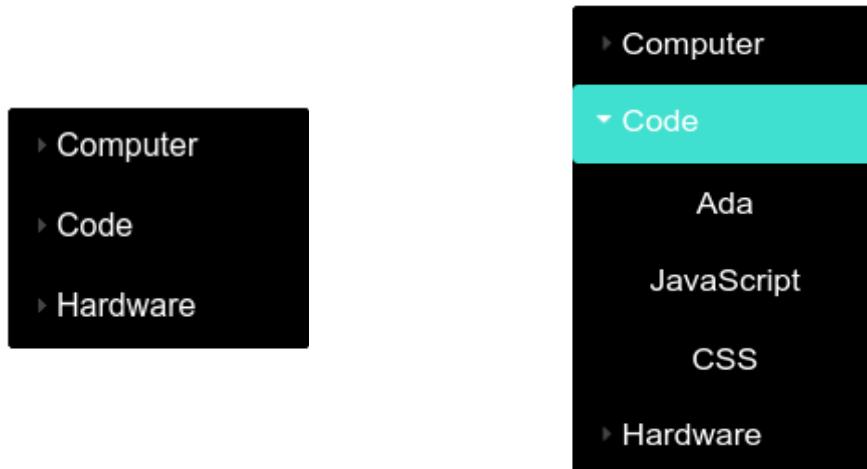
## 6.1 Accordion

Un accordéon est un menu affichant un contenu vertical non directement clicable.

Chaque section de l'Accordion permet d'accueillir un composant GNOGA et un seul.

Le bon usage est d'abord d'insérer un Docker puis, dans ce conteneur, d'insérer ensuite autant de Boutons que nécessaire.

Par défaut, une section est toujours ouverte. Passer Allow\_Collapse à True dans la procédure Render\_Accordion permet de refermer complètement l'Accordion.



Exemple d'accordéon.

À gauche, l'accordéon est plié, tandis qu'à droite il est déplié sur "Code".

Déclaration :

---

```
Accordion      : aliased Gnoga.Gui.Plugin.jQueryUI.Widget.Accordion_Type;
Folder        : aliased Folders.Folder_Type;
```

---

Code :

```

-- Création de l'élément
Folders.Create_Folder [App.Folder, App.Left_Panel];

-- Création de la section [bouton cliquable qui déroulera le contenu]
App.Folder.Create_Section [Folder];

-- Ajout du contenu qui sera découvert au clic, ici en l'occurrence, ajout d'un dossier
Gnoga.Gui.View.Docker.Pointer_To_Docker_View_Class
[Gnoga.Gui.View.New_Element [App.My_View' Unrestricted_Access, "Docker_Folder", App.Docker_Folder' Unrestricted_Access ]].Create [App.Folder];

-- Après ajout du contenu, on active le rendu du folder.
App.Folder.Render_Folder[True];

```

---

## 6.2 Folders

Folders est un composant que nous avons créé en nous inspirant de Accordion. Folders se déplie horizontalement tandis que Accordion se déplie verticalement.

À cet effet, des fichiers .adb, .css et .js doivent être ajoutés dans les sources de votre application pour pouvoir être utilisés. Le détail du code à ajouter est visible au chapitre Modifications des sources

## 6.3 Tabs

Les Tabs sont des onglets qui permettront de naviguer entre les différentes sections d'une page. Ils permettent de changer de carte au clic sur l'onglet correspondant, de la même façon qu'un moteur de recherche par exemple.



This is tab 1

Exemple de d'onglets Tabs.

Déclaration :

---

```
My_Tabs : aliased Gnoga.Gui.View.Card.Tab_Type;
```

---

Code :

---

```

-- Procédure qui permettra de changer de carte
procedure On_Button [Object : in out Gnoga.Gui.Base.Base_Type' Class] is
  pragma Unreferenced [Object];
begin
  Gnoga.Gui.View.Card.Show_Card[View => App.My_Cards, Name => "Widget"];
end On_Button;

-- Création des cartes
App.My_Tabs.Create [Parent => App.My_Docker, Card_View => App.My_Cards];

-- Création de la carte associé à My_Widget
App.My_Widget.Create [App.My_Cards];

```

---

Prise en main de GNOGA

```
App. My_Tabs. Add_Tab[Card => "Widget", Label => "Widget_Tab", Selected => False];
```

---

## 6.4 Cards

Associées aux différents menus, les Cards nous permettent de créer des applications où les menus ne changeront pas, mais le contenu de la fenêtre dépendra de l'onglet dans lequel on se trouve.

Déclaration :

---

```
My_Cards : aliased Gnoga.Gui.View.Card.Card_View_Type;
```

---

Code :

---

```
-- Procédure qui permettra de changer de carte
procedure On_Button [Object : in out Gnoga.Gui.Base.Base_Type' Class] is
  pragma Unreferenced (Object);
begin
  Gnoga.Gui.View.Card.Show_Card[View => App.My_Cards, Name => "Widget"];
end On_Button;

-- Création des cartes
App.My_Cards.Create [App.My_Docker];
App.My_Docker.Fill_Dock [App.My_Cards' Unchecked_Access];

-- Création de la carte associé à My_Widget
App.My_Widget.Create [App.My_Cards];
App.My_Cards.Add_Card [Name => "Widget", Card => App.My_Widget' Access];
```

---

## 7 Fonctionnalité TLS [https]

Cette fonctionnalité peut être mise en œuvre de deux manières :

- Via un proxy TLS (Nginx par exemple) sortant sur le port http ;
- Directement avec GNOGA.

### 7.1 Proxy TLS

La solution du proxy TLS est intéressante si plusieurs serveurs GNOGA, ayant plusieurs noms de domaines ou de sous-domaines, doivent être accédés à partir d'une seule IP.

On aura alors, par exemple, le schéma suivant :

- <https://serveur1.domaine1.tld> > <http://localhost:8081> ;
- <https://serveur2.domaine1.tld> > <http://localhost:8082> ;
- <https://serveur3.domaine1.tld> > <http://localhost:8083> ;
- <https://serveur1.domaine2.tld> > <http://localhost:8091> ;
- <https://serveur2.domaine2.tld> > <http://localhost:8092> ;

### 7.2 Support direct

Dès qu'un seul serveur doit être mis en œuvre, le support direct est recommandé.

Les points suivants doivent être validés :

- Création d'un certificat par Let's Encrypt et installation ;
- Installation du paquet libgnutls30 sur le système Debian ou dérivés ;

- Utilisation de l'extension GNOGA gnoga-server-connection-secure.ad\*.

Avec GNOGA installé dans votre environnement de développement, il suffira alors d'intégrer dans le fichier projet :

---

```
with "gnoga";
with "gnoga_secure";

project test_ssl_ins is
.../...
```

---

Et de faire précéder :

---

```
Application.Multi_Connect.Initialize [Event => On_Connect' Unrestricted_Access, ...
```

---

Par :

---

```
Gnoga.Server.Connection.Secure.Register_Secure_Server
[Certificate_File => Gnoga.Server.Application_Directory & "./ro6.genesix.org.cer",
 Key_File => Gnoga.Server.Application_Directory & "./ro6.genesix.org.key",
 Port      => 443, Disable_Insecure => False];
```

---

ro6.genesix.org.cer et ro6.genesix.org.key étant générés par Let Encrypt et le serveur ayant les droits root [port 443 < 1024].

Sans ces conditions, vous pouvez aussi utiliser les certificats livrés avec GNOGA, au prix d'une expérience utilisateur dégradée :

---

```
Gnoga.Server.Connection.Secure.Register_Secure_Server
[Certificate_File => Gnoga.Server.Application_Directory & "./server.crt",
 Key_File => Gnoga.Server.Application_Directory & "./server.key",
 Port      => 8443, Disable_Insecure => False];
```

---

Dans tous les cas, le chemin des clés et certificats est à ajuster en fonction du contexte. Si un certificat Let Encrypt est utilisé, il sera à renouveler automatiquement pendant le troisième [et dernier] mois de validité.

Un exemple complet est proposé dans ce document [voir section Test SSL].

## 8

## CSS

Il est possible de modifier les fichiers CSS une fois ceux-ci copiés dans le répertoire de travail. On peut modifier plusieurs aspects mais il faut retenir que le CSS ne permet, entre autres, de ne modifier que les couleurs, les positions et les tailles des différents éléments. Si on veut modifier les animations, il va falloir modifier les fichiers JavaScript.

D'une manière générale, voici quelques propriétés en CSS qui sont facilement modifiables si l'on veut changer quelques aspects visuels simples :

- margin, padding, border : relatif aux marges.  
Unité : px, em, %.  
Exemple : 0 25px 0 10px : marge de 25 px sur la droite, et de 10px sur la gauche. Ainsi, l'ordre est : haut | droit | bas | gauche.
- background, background-color : couleur de fond.  
Code : # ou nom de la couleur en anglais.
- Color : couleur du texte.  
Code : # ou nom de la couleur en anglais.
- Border-color : couleur de la bordure.
- Border-bottom-left-radius : rayon de la bordure en bas à droite de l'élément.  
Unité : px, em.
- Position : façon dont laquelle sera définie la position de l'élément.  
Valeurs : absolute, fixed, relative, static.
- opacity : opacité.  
Type : flottant
- height, width : respectivement hauteur et largeur.  
Unité : px ou %.
- font-size : taille de la police.  
Unité : px ou em.
- background linear 0.4s : temps après lequel la couleur de fond sera celle de l'état souhaité.
- cursor : affichage du curseur de la souris quand celui-ci passera sur l'élément.  
Valeurs : pointer.
- top, left, right, bottom : position.  
Unités : px, em, %.  
Si top est utilisé, ne pas utiliser bottom. Idem avec left et right.
- background-image: url("img/image.png") : image de fond.
- Background-position : position du background, donc potentiellement de l'image de fond.  
Unité : 2 valeurs [pour top et left] en px ou em.
- Z-index : position de l'élément sur l'axe Z. L'axe Z est l'axe dans la profondeur de l'écran. Valeur de base : 0. Un élément qui aura comme valeur de z-index supérieure à 0 apparaîtra au premier plan. Au contraire, un élément qui aura comme valeur de z-index inférieure à 0 n'apparaîtra sûrement pas puisqu'elle sera cachée par les autres éléments qui seront eux au plan 0.
- overflow : affichage ou non des barres de défilement.  
Valeur par défaut : visible. Valeur pour les cacher : hidden.
- État : hover : quand le curseur de la souris passe sur l'élément.

- Valeur en pourcentage : 100% représente la valeur du parent pour cette même propriété.

## 9 Modification des sources

### 9.1 Généralités

La modification des sources doit s'effectuer dans les répertoires :

---

```
... /gnoga/src
ou
... /gnoga/components
ou
... /gnogadeps
```

---

▷ Modifier des fichiers dans ".../gnoga/include" n'aura aucun impact sur le reste du code puisque ce ne sont que des fichiers copiés une fois la compilation de GNOGA réalisée.

### 9.2 Attribut Background

Durant nos essais pratiques de GNOGA, nous avons souhaité reproduire certaines parties d'autres sites internet qui nous ont inspirés. Un outil très simple et pratique existe en CSS pour appliquer à à-peu-près n'importe quel élément des dégradés simples : la fonction linear-gradient. Il n'est possible de l'utiliser qu'avec la propriété background, car background-color n'est là que pour recevoir des couleurs précises, pas de dégradés.

Cependant, il n'est pas possible de définir/modifier, avec GNOGA, le background de certains éléments, mais uniquement le background-color. Les sources de GNOGA ont été modifiées de la façon suivante :

gnoga-gui-element.ads : [ligne 677]

---

```
procedure Background
  [Element : in out Element_Type;
   Value    :          String];
procedure Background
  [Element : in out Element_Type;
   RGBA    :          Gnoga.Types.RGBA_Type];
procedure Background
  [Element : in out Element_Type;
   Enum    :          Gnoga.Types.Colors.Color_Enumeration];
function Background
  [Element : Element_Type]
  return Gnoga.Types.RGBA_Type;
```

---

gnoga-gui-element.adb : [ligne 1407]

---

```
-----
-- Background --
-----

procedure Background
  [Element : in out Element_Type;
   Value    :          String]
```

---

---

```

is
begin
    Element.Style ["background", Value];
end Background;

procedure Background
    [Element : in out Element_Type;
     RGBA      :          Gnoga.Types.RGBA_Type]
is
begin
    Element.Style ["background", Gnoga.Types.To_String [RGBA]];
end Background;

procedure Background
    [Element : in out Element_Type;
     Enum     :          Gnoga.Types.Colors.Color_Enumeration]
is
begin
    Element.Style ["background", Gnoga.Types.Colors.To_String [Enum]];
end Background;

function Background
    [Element : Element_Type]
    return Gnoga.Types.RGBA_Type
is
begin
    return Gnoga.Types.To_RGBA [Element.Style ["background"]];
end Background;

```

---

Cependant, après de nouvelles recherches, une alternative à cette modification du code a été trouvée [voir Styliser les éléments avec CSS].

### 9.3 Sections des Accordions

Les sections de ces composants sont définies comme des titres HTML. Cela ne permet pas d'y accéder depuis le code Ada, via GNOGA, nous empêchant alors de les modifier. Ainsi, au lieu de créer des titres HTML, il suffit de créer des boutons :

gnoga-gui-plugin-jqueryui-widget.adb : [ligne 62]

---

```

- procedure Create_Section
-     [View   : in out Accordion_Type;
-      Heading :          String]
- is
- begin
-     View.Put_HTML ["<h3>" & Escape_Qoutes [Heading] & "</h3>"];
- end Create_Section;

+ procedure Create_Section
+     [View   : in out Accordion_Type;
+      Heading :          String      ]
+ is

+     Button : constant Gnoga.Gui.Element.Pointer_To_Element_Class := new Gnoga.-
GUi.Element.Common.Button_Type;

+ begin

+     Gnoga.Gui.Element.Common.Button_Access [View.New_Element [From_Latin_1[ Hea-
ding], Button]] .Create [View, From_Latin_1[ Heading]];
+ end Create_Section;

```

---



## 9.4 jQuery-UI

La version de jQuey-UI dans GNOGA est la 1.12.1.

Cependant, le fichier jquery-ui.min.js contenu dans GNOGA, même une fois « déminifié », est différent du fichier jquery-ui.js officiel<sup>6</sup> en de nombreux points. Les modifications suivantes sont presque les mêmes pour les deux versions .

Plusieurs modifications ont été apportées à jQuery-UI. Dans un premier temps, nous avons fait en sorte qu'aucune section de l'Accordion ne soit ouverte lors de l'ouverture de l'application. Les modifications ont eu lieu dans jquery-ui.js aux lignes 4581 [approximativement] :

---

```
- this.active = this._findActive( options.active );
- this._addClass( this.active, "ui-accordion-header-active", "ui-state-active" )
-     ._removeClass( this.active, "ui-accordion-header-collapsed" );
- this._addClass( this.active.next(), "ui-accordion-content-active" );
- this.active.next().show();

+ this.active = this._findActive( options );
+ this._addClass( this.active, "ui-accordion-header-active", "ui-state-active" )
+     ._removeClass( this.active, "ui-accordion-header-collapsed" );
+ this._addClass( this.active.next(), "ui-accordion-content-active" );
```

---

[pour la version GNOGA, se rendre ligne 1966, effectuer les mêmes modifications, seule la variable option a été remplacée par la variable i]

Le second changement concerne l'affectation d'une valeur à un attribut dans le HTML. En effet, la hauteur « height » était définie à la ligne 4425 de jquery-ui.js mais il nous était impossible de la modifier via le CSS ou même GNOGA.

---

```
- if ( this.options.heightStyle !== "content" ) {
-     contents.css( "height", "" );
- }
```

---

[pour la version GNOGA, retirer la ligne 1892]

## 10 Astuces

Dans cette section, nous allons présenter quelques astuces qui nous ont permis de coder avec GNOGA de façon plus aisée.

### 10.1 Récupérer le contenu d'un objet

On peut facilement récupérer le contenu d'un objet avec la ligne de code suivante :

---

```
<objet>.jQuery_Execute["text[]"];
```

---

<sup>6</sup>Que l'on peut trouver sur : <https://jqueryui.com/download/all/> [version 1.12.1]

## 10.2 Styliser les éléments avec CSS

Plutôt que d'utiliser les fonctions codées avec GNOGA permettant de modifier les attributs CSS d'un objet, il est tout aussi possible d'arriver aux mêmes fonctionnalités avec la procédure Style, prenant en argument deux chaînes de caractères : la première étant l'attribut et la seconde la valeur. Ainsi, on peut avoir :

---

```
Accordion.Style["position", "absolute !important"];
```

---

Il est aussi possible de changer le background sans avoir à modifier les sources [voir Modification des sources] :

---

```
<objet>.Style["background", "<valeur de la propriété>"];
```

---

Cela permet également de mettre des nombres flottants comme valeur pour les méthodes :

---

```
<objet>.Style["left", "12.8px"];
```

---

fonctionne correctement, tandis que...

---

```
<objet>.Left{12.8};
```

---

...ne fonctionne pas car la valeur passée en paramètre de Left doit être un entier.

## 10.3 Plantage de l'application

Lorsque l'application plante, GNOGA donne le code d'erreur sous la forme d'adresses hexadécimales. Pour tracer l'erreur :

---

```
addr2line -e <nom_du_binaire> <code_d_erreur_en_hexadécimale>
```

---

## 10.4 Inspecteur de pages Web

Lorsque l'on crée une application, il peut être pratique de modifier diverses caractéristiques des éléments et de voir le résultat en temps réel. Pour cela, on peut utiliser l'inspecteur de pages Web [F12 avec Firefox]. Il nous est alors possible de modifier le CSS, le HTML sans pour autant que les modifications apportées soient enregistrées. Il ne reste donc plus qu'à reporter les modifications dans les fichiers Ada ou CSS.

L'inspecteur permet également vérifier qu'il n'y ait pas d'erreurs avec les fichiers HTML et JavaScript en cas de modifications, et propose un débogueur.

## 10.5 Crédit dynamique ou statique d'éléments

Les éléments dynamiques, qui vont être créés suite à l'appel d'une fonction [ou procédure], doivent être instanciés à part du type App\_Data créé, c'est-à-dire dans ladite fonction. De plus, il est important de comprendre la nuance entre les deux types d'instanciations : statique ou dynamique.

Nous allons voir cela avec l'exemple du bouton. Pour le moment, ce n'est qu'avec ces objets que j'ai pu différencier les moyens d'instanciations. En effet, même si je ne sais pas s'il est possible de le faire avec les vues, ni même comment le faire, cela ne semblait pas poser problème dans les applications que nous avions créé.

Pour des éléments créés statiquement, c'est-à-dire qui seront créés lors du démarrage de la page, il est important d'adopter la construction suivante :

---

```
type App_Data is new Gnoga.Types.Connection_Data_Type with record
    Bouton_Static : aliased Gnoga.Gui.Element.Common.Button_Type;
end record;
```

---

Il faut ensuite instancier un objet de type App\_Data dans la fonction On\_Connect :

---

```
procedure On_Connect
    [Main_Window : in out Gnoga.Gui.Window.Window_Type' Class;
     Connexion : access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
is
    pragma Unreferenced [Connexion];
    App : constant App_Access := new App_Data;
```

---

Enfin dans cette même fonction, créer le bouton avec :

---

```
App.Bouton_Static.Create[View, "bouton"];
```

---

Pour un bouton dynamique, c'est-à-dire créé lors d'un événement différent du démarrage de la page, la méthode est tout autre.

En effet, l'instanciation d'un bouton dynamique se fait de la façon suivante [dans la fonction qui va créer ce bouton] :

---

```
Bouton_Dynamique : constant Gnoga.Gui.Element.Pointer_To_Element_Class := new
Gnoga.Gui.Element.Common.Button_Type;
```

---

puis dans cette même fonction :

---

```
Gnoga.Gui.Element.Common.Button_Access[Bouton_Dynamique].Create [View, "bouton"];
View.Add_Element["bouton", Bouton_Dynamique];
Bouton_Dynamique.Dynamic;
```

---

Nous avons éviter la construction suivante qui semble nous apporter des bugs [voir Anomalies avec les Handlers] :

---

```
Gnoga.Gui.Element.Common.Button_Access  
[Gnoga.Gui.View.New_Element  
[View' Unrestricted_Access, "bouton", Bouton_Dynamique]].Create [View, "bouton"];
```

---

## 10.6 Architecture de développement

Il est conseillé de ne pas apporter de modifications aux fichiers JavaScript et CSS, présent dans le répertoire GNOGA, par exemple pour apporter des modifications à jQuery-UI. En effet, dans la mesure où ajouter un fichier puis l'inclure dans le DOM est facilement réalisable, il est préférable de ne pas toucher aux fichiers générés.

Un exemple permet d'illustrer ces propos dans le sous-chapitre suivant.

## 10.7 Exemple d'ajout de fichier CSS

Problème : Nous avons un Folder, dont le titre de la section est « Fichier ». Nous voulons colorer en rouge la première lettre de ce mot.

Nous avons alors le code suivant :

---

```
Folder.Create_Folder[Parent => Parent_View];  
Folder.Create_Section[Content => "Fichier", Name => "Fichier"];  
View_Folder.Create[Parent => Folder];  
Button.Create [Parent => View_Folder, Content => "Bouton"];
```

---

Et le résultat :



Dans un premier temps, nous allons créer un fichier CSS qui contiendra une nouvelle classe, qui nous permettra de colorier la lettre de la bonne couleur.

application.css :

---

```
.rouge{  
color : red;  
}
```

---

Incluons le fichier CSS nouvellement créé dans le fichier HTML :

---

```
<link href="/css/application.css" rel="stylesheet">
```

---

Précisons maintenant que nous voulons que la lettre « F » soit rouge :

---

```
Folder.Create_Folder[Parent => Parent_View];
```

---

```
-     Folder.Create_Section[Content => "Fichier", Name => "Fichier"];
+
+     Folder.Create_Section[Content  => "<span  class='rouge'>F</span>ichier",
Name => "Fichier"];
+
+     View_Folder.Create[Parent => Folder];
+     Button.Create [Parent => View_Folder, Content => "Bouton"];
```

---

Et voici le résultat :



## 10.8 Exemple d'ajout de fichier JS

Se reporter à l'ajout du widget Folders dans jQuery-UI (à partir de Ajout de Folders.js).

# Annexes

---

## 1 Ressources GNOGA

### 1.1 Gnoga

<https://sourceforge.net/projects/gnoga/> page Sourceforge du projet GNOGA, dans lequel l'ensemble du package GNOGA est présent en plus de quelques démonstrations qu'il faut parfois débugger.

<https://sourceforge.net/p/gnoga/wiki/Gnoga-Gallery> page Sourceforge des démos de GNOGA.

<https://sourceforge.net/p/gnoga/mailman/gnoga-list> mailing list pour communiquer avec la communauté GNOGA.

<https://sourceforge.net/p/gnoga/tickets> page pour signaler les problèmes relatifs à GNOGA.

### 1.2 Ada for automation

<https://slo-ist.fr/ada4automation> « Ada for Automation » est un cadriciel pour développer des applications d'automatisme évoluées dans le langage Ada par Stéphane Los.

L'IHM web de Ada for automation est fondée sur GNOGA. Voir [par exemple] <https://gitlab.com/ada-for-automation/ada-for-automation/-/blob/master/src/web/a4a-web-server.adb>

□ Extension JustGage pour GNOGA

<https://gitlab.com/ada-for-automation/ada-for-automation/-/commit/1b62805a77cc486d748e25bf48b2eb02e815ccd4>

<https://toorshia.github.io/justgage>

### 1.3 Document « Premiers pas avec GNOGA »

<https://blady.pagesperso-orange.fr/telechargements/gnoga/gnoga.pdf> Premiers pas avec GNOGA, documentation en Français de GNOGA permettant de découvrir et apprendre à programmer avec ce framework par Pascal Pignard.

<https://github.com/Blady-Com/Exemples-PPAG> page github des exemples du document gnoga\_wf.pdf de Pascal Pignard

### 1.4 Vidéo de démonstration de GNOGA

Par Jeffrey R. Carter.

<https://www.youtube.com/watch?v=-lCFJvlH7-k>

## 2 Ressources Sowebio

### 2.1 AIDE

*Ada Instant Development Environment.* Un environnement de développement ADA prêt à l'emploi pour GNU/Linux Debian et distributions dérivées telles Ubuntu.

<https://github.com/sowebio/aide>  
<https://github.com/sowebio/aide-doc>  
<https://github.com/sowebio/aide-bin>

## 3 Dictionnaire

Vocabulaire Ada et GNOGA pour aider à la compréhension de ce document.

### 3.1 Dictionnaire général/Ada

- Runtime : environnement d'exécution, permet l'exécution des programmes informatiques écrits dans un langage de programmation donné [ici, Ada].
- Template : gabarit, patron représentant la structure d'un fichier qui servira de modèle.

### 3.2 Dictionnaire GNOGA

- DOM : *Document Object Model*, est une interface de programmation pour les documents HTML, XML et SVG.
- Window : fenêtre du navigateur.
- View, Frame : portion de la fenêtre, cadre d'une dimension quelconque dans lequel d'autres éléments peuvent être ajoutés.
- Button : bouton.
- Heading : titre.
- Form : formulaire, zone dans laquelle on peut saisir un texte.
- Handler : fonction prenant en paramètre une autre fonction, et qui déclenchera l'exécution de cette seconde fonction dans des conditions particulières. Par exemple :

---

```
Button.On_Click_Handler [Foo1' Unrestricted_Access];  
Button.On_Double_Click_Handler [Foo2' Unrestricted_Access];
```

---

exécutera la fonction « Foo1 » lorsque le bouton « Button » sera cliqué, et exécutera la fonction « Foo2 » lorsque le bouton « Button » sera double-cliqué.

- Tab : étiquette, comme l'on peut avoir au dessus de chaque onglet du navigateur.
- Cards : ensemble de Views. On ne peut en afficher qu'une à la fois, mais on peut changer laquelle via une fonction. Fonctionne comme les onglets du navigateur.
- Widget : élément du DOM interactif.

- Accordion : widget se dépliant verticalement. Ses parties toujours visibles et cliquables sont les « sections », tandis que le contenu son des éléments placés en temps qu'éléments enfants de l'Accordion.
- Folder : comme un Accordion mais se dépliant horizontalement.
- Breadcrumb [trail] : littéralement « miettes de pain », cela correspond au fil d'Ariane. Ligne horizontale de boutons retraçant le parcours dans un arbre de fichiers, ou de Views etc. depuis le nœud racine.

## 4 Ressources jQuery

### 4.1 Keycodes

<b>KeyCode</b>	<b>Valeur numérique</b>
Backspace	8
Tab	9
Enter	13
Shift	16
CTRL	17
ALT	18
Pause/ Break	19
Caps Lock	20
ESC	27
Page Up	33
Page Down	34
End	35
Home	36
Arrow Left	37
Arrow Up	38
Arrow Right	39
Arrow Down	40
Insert	45
Delete	46
0	48
1	49
2	50
3	51
4	52
5	53
6	54
7	55
8	56
9	57
::	59
=+	61
a	65
b	66
c	67

d	68
e	69
f	70
g	71
h	72
i	73
j	74
k	75
l	76
m	77
n	78
o	79
p	80
q	81
r	82
s	83
t	84
u	85
v	86
w	87
x	88
y	89
z	90
Windows	91
Right Click	93
0[Num Lock]	96
1[Num Lock]	97
2[Num Lock]	98
3[Num Lock]	99
4[Num Lock]	100
5[Num Lock]	101
6[Num Lock]	102
7[Num Lock]	103
8[Num Lock]	104
9[Num Lock]	105
*[Num Lock]	106
+[Num Lock]	107
-[Num Lock]	109
.[Num Lock]	110
/[Num Lock]	111
F1	112
F2	113
F3	114
F4	115
F5	116
F6	117
F7	118
F8	119
F9	120

F10	121
F11	122
F12	123
Num Lock	144
Scroll Lock	145
My Computer	182
My Calculator	183
,<	188
.>	190
/?	191
'~	192
[{	219
\	220
}]	221
‘”	222

D'après <https://www.educba.com/jquery-keycode>

## 5 Ressources jQuery-UI

### 5.1 Généralités

<https://darchevillepatrick.info/js/js27.php>

### 5.2 Extension

[https://www.tutorialspoint.com/jqueryui/jqueryui\\_widgetfactory.htm](https://www.tutorialspoint.com/jqueryui/jqueryui_widgetfactory.htm)

[https://disic.github.io/rgaa\\_bibliotheques-javascript/tutoriels/jquery-ui.html](https://disic.github.io/rgaa_bibliotheques-javascript/tutoriels/jquery-ui.html)

[https://github.com/DISIC/rgaa\\_jquery-ui](https://github.com/DISIC/rgaa_jquery-ui) voir usage et sources, en relation avec la page ci-dessus]

### 5.3 ThemeRoller

<https://jqueryui.com/themeroller>

<https://learn.jquery.com/jquery-ui/themeroller> [mode d'emploi]

- Exemple

Sélectionner le thème : [Gallery] > UI Darkness theme

Revenir sur l'onglet : [Roll your own]

Poser un fond noir : [x] Reverse page background color

Télécharger le thème : [Download theme]

Laisser la version par défaut : [o] 1.13.2

[Download]

On obtient un fichier compressé jquery-ui-1.13.2.custom.zip qui, une fois décompressé, développe l'arborescence suivante :

---

```

    └── AUTHORS.txt
    └── external
        └── jquery
            └── jquery.js
    └── images

```

---

---

```

    ├── ui-bg_glass_20_555555_1x400.png
    ├── ui-bg_glass_40_0078a3_1x400.png
    ├── ui-bg_glass_40_ffc73d_1x400.png
    ├── ui-bg_gloss-wave_25_333333_500x100.png
    ├── ui-bg_highlight-soft_80_eeeeee_1x100.png
    ├── ui-bg_inset-soft_25_000000_1x100.png
    ├── ui-bg_inset-soft_30_f58400_1x100.png
    ├── ui-icons_222222_256x240.png
    ├── ui-icons_4b8e0b_256x240.png
    ├── ui-icons_a83300_256x240.png
    ├── ui-icons_cccccc_256x240.png
    └── ui-icons_ffffff_256x240.png
    index.html
    jquery-ui.css
    jquery-ui.js
    jquery-ui.min.css
    jquery-ui.min.js
    jquery-ui.structure.css
    jquery-ui.structure.min.css
    jquery-ui.theme.css
    jquery-ui.theme.min.css
    LICENSE.txt
    package.json

```

---

- Test

Lancer index.html pour tester. Le choix « Reverse page background color » n'a pas été pris en compte, c'est certainement une fonction de base de jQuery ou jQueryUI à activer séparément.

Les fichiers indispensables se réduisent à :

- jquery.js
- jquery-ui.js
- jquery-ui.css
- répertoire images

Ou leurs versions minifiées :

- jquery.min.js
- jquery-ui.min.js
- jquery-ui.min.css

- En pratique

<https://api.jqueryui.com/theming>

## 6 Ressources DOM

[https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction#what\\_is\\_the\\_dom](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction#what_is_the_dom)

## 7 To do list

### 7.1 Intégration SQLite

Documenter la recompilation/réinstallation [voir recompilation ci-dessous] de GNOGA après un changement de paramètre à l'ouverture d'une DB SQLite [gnoga-server-database-sqlite.ad\*].

## 7.2 Documentation makefile

Analyser/tester le makefile avec toutes les options, pour ensuite documenter toutes les actions proposées et leurs effets.

## 8 Anomalies corrigées

### 8.1 hello3post.adb

Programme issu de la documentation gnoga-wf.pdf.

<https://github.com/Blady-Com/Exemples-PPAG/blob/master/src/hello3post.adb>

Pour compiler et s'exécuter sans erreurs, le code original doit être modifié :

---

```
diff hello3post-org.adb hello3post.adb

1,3d0
< with Ada.Exceptions;
< with Ada.Strings.Unbounded;
<
9a7
> with UXStrings; use UXStrings;
128,129c126,127
<     [URI           : in      String;
<     Accepted_Parameters : out Ada.Strings.Unbounded.Unbounded_String]
<-
>     [URI           : in      UXString;
>     Accepted_Parameters : out UXString]
133,134c131
<     Accepted_Parameters :=
<     Ada.Strings.Unbounded.To_Unbounded_String
<-
>     Accepted_Parameters := From_Latin_1
141c138
<     [URI           :          String;
<-
>     [URI           : in      UXString;
151c148
<     Connection   :          access Gnoga.Application.Multi_Connect.Connection_Hol-
der_Type]
<-
>     Connection   : access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
181,182c178,179
<     Gnoga.Log [Ada.Exceptions.Exception_Name (E) & " - " & Ada.Exceptions.Excep-
tion_Message (E)];
< end hello3post;
<-
>     Gnoga.Log (E);
> end hello3post;
```

---

### 8.2 hello3post\_file.adb

Programme issu de la documentation gnoga-wf.pdf.

[https://github.com/Blady-Com/Exemples-PPAG/blob/master/src/hello3post\\_file.adb](https://github.com/Blady-Com/Exemples-PPAG/blob/master/src/hello3post_file.adb)

Pour compiler et s'exécuter sans erreurs, le code original doit être modifié :

---

```
diff hello3post_file-org.adb hello3post_file.adb
```

---

```

diff hello3post_file-org.adb hello3post_file.adb
1, 3d0
< with Ada.Exceptions;
< with Ada.Strings.Unbounded;
<
9a7
> with UXStrings; use UXStrings;
130, 131c128, 129
<     [URI           : in      String;
<     Accepted_Parameters : out Ada.Strings.Unbounded.Unbounded_String]
---
>     [URI           : in      UXString;
>     Accepted_Parameters : out UXString]
135, 136c133
<     Accepted_Parameters := 
<         Ada.Strings.Unbounded.To_Unbounded_String
---
>     Accepted_Parameters := From_Latin_1
143c140
<     [URI           :          String;
---
>     [URI           :          UXString;
154, 156c151, 153
<     [URI           : in      String;
<     File_Name    : in      String;
<     Temp_Name    : in      String]
---
>     [URI           : in      UXString;
>     File_Name    : in      UXString;
>     Temp_Name    : in      UXString]
197c194
<         Gnoga.Log [Ada.Exceptions.Exception_Name [E] & " - " & Ada.Exceptions.Exception_Message [E]];
---
>         Gnoga.Log [E];

```

## 8.3 Anomalies avec les Handlers

Nous avons remarqué que, parfois, ajouter simplement un élément à une application GNOGA pouvait faire crasher l'application. En effet, la plupart du temps, une partie ou la totalité des boutons lève une erreur au clic.

Nous pensions dans un premier temps qu'il s'agissait des méthodes de style, comme `<Object>.Style` ou encore `<Object>.Border` : commenter/supprimer une ligne où était appelée une de ces fonctions réglait parfois le problème. Cependant, après un peu de recherche, il pourrait s'avérer que nous faisons face à un problème un peu plus global, puisque c'est aussi à l'ajout des boutons, et plus particulièrement à l'exécution d'une fonction passée aux Handlers, que des erreurs sont levées.

Peut-être que nous créons nos objets d'une mauvaise manière... Nous avons cependant veillé à distinguer création dynamique de création statique.

Voici quelques-unes de nos méthodes de création :

- boutons, implémentation statique [+ handler]

---

```

Button.Create[View, "button"];
Button.On_Click_Handler[On_Button' Unrestricted_Access];

```

---

- boutons, implémentation dynamique [+ handler]

---

```
Gnoga.Gui.Element.Common.Button_Access[Button].Create [View, "button"];
View.Add_Element["button", Button];
Button.Dynamic;
Button.On_Click_Handler[On_Button' Unrestricted_Access];
```

---

En effet, nous n'utilisons pas la fonction New\_Element, tel que suit, car cela nous apportait plus souvent des levées d'erreurs :

---

```
Gnoga.Gui.Element.Common.Button_Access
[Gnoga.Gui.View.New_Element
[View' Unrestricted_Access, "button", Button]].Create [View, "button"];
```

---

### - Création des Cards

---

```
Cards.Create [Parent];
View_1.Create [Cards];
View_2.Create [Cards];
[création des éléments dans la vue...]
Cards.Add_Card [Name => "Card_1",
                Card => View_1' Access];
Cards.Add_Card [Name => "Card_2",
                Card => View_2' Access];
```

---

Les Cards sont activées par des boutons, et non pas par des Tabs, comme c'est parfois le cas dans les démos. Cela ne devrait pas poser des problèmes puisque la fonction Show\_Cards permettant de permuter les Cards semble fonctionnelle. Ainsi, on a :

---

```
procedure On_Button [Object : in out Gnoga.Gui.Base.Base_Type' Class] is
    pragma Unreferenced [Object];
begin
    Cards.Show_Card["Card_1"];
end On_Button;
```

---

Dans des deuxièmes Cards, le parent est un Accordion. La fonction Render\_Accordion est appelée juste avant l'appel à la méthode Add\_Card.  
On utilise les Accordion pour créer des menus. Ainsi, pour chaque section de l'Accordion, on hérite un View\_Type. Ça nous permet de placer plusieurs boutons sous une section de l'Accordion, au lieu d'un seul bouton par section.

Et c'est à peu près tout...

On utilise dans nos applications de test ces éléments plusieurs fois chacuns, mais aucun autre widget.

Les erreurs levées sont :

---

```
--> raised CONSTRAINT_ERROR : a-chtgke.adb: 127: 12 invalid data
--> raised CONSTRAINT_ERROR : a-chtgke.adb: 60 divide by zero
--> raised STORAGE_ERROR : s-sintman.adb: 136 explicit raise
```

---

▷ L'ensemble de ces erreurs sont dues à l'appel de la méthode Show\_Card.

▷ L'ensemble des boutons ouvrant les Cards sont défaillants à ce moment là. Le reste des boutons sont fonctionnels.

```
--> raised PROGRAM_ERROR : GNOGA.SERVER.CONNECTION.GNOGA_HTTP_CLIENT is still in use
[...]
Execution of ./application1 terminated by unhandled exception
raised PROGRAM_ERROR
Call stack traceback
Execution of ./application1 terminated by unhandled exception
raised PROGRAM_ERR locatiOR : finalize/adjust raised exception
```

▷ Erreur due à Message\_Loop => Server\_Wait.Hold;

▷ Tous les boutons lèvent cette erreur et font cracher l'application.

D'une façon générale :

▷ Le type de l'erreur dépend du nombre de boutons qu'il y a dans l'application. Cependant, cette valeur n'est pas fixée : pour un nombre N de boutons : on aura une certaine erreur. Pour N+1 et N-1, ça sera des erreurs différentes.

Nous avons trois Cards dans notre application :

Avec notre code actuel, avec toutes les méthodes de style :

▷ 0/3 est fonctionnel.

Avec toutes les méthodes de style commentées/supprimées des Cards :

▷ 1/3 est fonctionnel, donc une seule ne créée aucune erreur lorsque l'on appelle la fonction Show\_Card sur cette Cards.

Avec toutes les méthodes de style commentées/supprimées de l'application :

▷ 0/3 est fonctionnel.

Après avoir fait quelques tests sur cette même application, nous avons remarqué que retirer l'ensemble des Cards ne changeait rien. Nous avions toujours ces erreurs. Nous avons fait la même chose pour les Breadcrumb trails, et nous avons obtenu le même résultat. Lors de ces tests, nous avons gardé les boutons qui étaient déjà placés et créés avant, nous les avons juste attachés à leurs « grand-parents » si nous avons enlevé leurs parents.

▷ Pour chaque élément créé dans le DOM, GNOGA semble faire une copie de l'élément dans une table de hachage, en utilisant des méthodes du Runtime Ada. À l'appel de la procédure utilisant cet élément en paramètre, cet élément semble manquer à la table de hachage.

Nous avons remarqué dans les démos un type un peu spécial :

```
type Button_Set is array [Field.Valid_Row, Field.Valid_Column] of Gnoga.Gui.Element.-
Common.Button_Type;
```

...qui permettait de créer un ensemble de boutons assez rapidement , encore plus s'ils sont tous de même contenu et de même Handler. Cela semble vraiment pratique, même pour créer un ensemble de boutons, au lieu de 1 par ligne. L'appel sera aussi facilité :

Prise en main de GNOGA



ed. 186 of 2023-09-12  
page 96 of 119

Buttons[12] au lieu de Bouton\_Douze. Cependant, cela peut-il avoir un impact sur la table de hachage ?

Les objets sont insérés dans la table de hachage avec leur Unique\_ID [subtype Unique\_ID is Integer] quand ils sont créés. La table de hachage est l'objet Object\_Map mais n'est accessible que dans la partie protégée Object\_Manager. Trois méthodes permettant de manipuler cette table sont implémentées d'office : Insert, Get\_Object, Delete.

Cependant, il semble impossible de vérifier si la table de hachage est correctement remplie sans modifier les sources. En effet, la méthode Get\_Object est manquante dans gnoga-server-connection.ads. de plus, le fait qu'on fait appel à un package générique pour définir Object\_Map ne facilite pas sa mise dans le fichier de spécification.

Ainsi, nous avons tenté de modifier les sources de GNOGA pour s'assurer, par vérification, que la table de hachage est correctement remplie. Voici nos changements :

Dans /src/gnoga-server-connection.ads :

```
+      protected body Object_Manager is
+          function Get_Object
+              [ID : Gnoga.Types.Unique_ID]
+              return Gnoga.Gui.Base.Pointer_To_Base_Class
+
+          is
+          begin
+              if Object_Map.Contains [ID] then
+                  return Object_Map.Element [ID];
+              else
+                  raise No_Object with "ID " & To_ASCII
+ (Image [ID]);
+              end if;
+          end Get_Object;
```

Dans /src/gnoga-server-connection.adb :

```
+      Object_Map : Object_Maps.Map;
-  protected Object_Manager is
-      function Get_Object
-          [ID : Gnoga.Types.Unique_ID]
-          return Gnoga.Gui.Base.Pointer_To_Base_Class;
-      procedure Insert
-          [ID      : in Gnoga.Types.Unique_ID;
-           Object   : in     Gnoga.Gui.
- Base.Pointer_To_Base_Class];
-      procedure Delete [ID : Gnoga.Types.Unique_ID];
-      private
-          Object_Map : Object_Maps.Map;
-  end Object_Manager;
```

À cause de la levée d'erreur, la fonction Get\_Object retourne quelque chose de différent du pointeur nul quand l'objet sélectionné n'existe pas. Ainsi, nous avons créé une seconde méthode, similaire à Get\_Object mais qui retourne bien le pointeur nul. Voici son implémentation :

Dans /src/gnoga-server-connection.ads :

---

```
+     function Get_Object_Or_Null
+         [ID : Gnoga.Types.Unique_ID]
+             return Gnoga.Gui.Base.Pointer_To_Base_Class;
```

---

Dans /src/gnoga-server-connection.adb :

---

```
+     function Get_Object_Or_Null
+         [ID : Gnoga.Types.Unique_ID]
+             return Gnoga.Gui.Base.Pointer_To_Base_Class
+         is
+         begin
+             if Object_Map.Contains [ID] then
+                 return Object_Map.Element [ID];
+             else
+                 return null;
+             end if;
+         end Get_Object_Or_Null;
```

---

Nous avons pu alors effectuer quelques tests, permettant de vérifier que les éléments du DOM sont bien associés à un ID dans la table de hachage. Pour cela, nous avons ajouté le code suivant à la fin de la procédure On\_Connect, dans notre application de test défaillante :

---

```
Ptr_Base : Gnoga.Gui.Base.Pointer_To_Base_Class;
[...]
for i in 1..300 loop
    Ptr_Base := Gnoga.Server.Connection.Object_Manager.Get_Object_Or_Null (i);
    if Ptr_Base /= null then
        Gnoga.Log(From_Latin_1[i'Image] & " : " & Ptr_Base.ID);
    else
        Gnoga.Log(From_Latin_1[i'Image] & " : null");
    end if;
end loop;
```

---

Ce qui nous permet d'afficher les 300 premiers ID [de la table de hachage] et GID [GNOGA-ID : ID dans le DOM]. Il est évident possible d'en afficher plus, mais notre dernier ID était le 243, donc inutile d'aller plus loin. On peut remarquer aussi que les ID ne se suivent pas forcément.

Après avoir vérifié manuellement que tous les éléments du DOM ont bien leur GID dans la table de hachage, nous avons remarqué que pas seulement la page n'avait sa table de hachage, les Views avaient aussi chacune la leur. Elle ne semble cependant pas être remplie dans tous les cas. En effet, seules les Cards et les Dockers possèdent une table de hachage. Cela permet aux Cards d'avoir la Card actuelle : « current », et aux Dockers de savoir quelle View est « dockée » à gauche, en haut, etc. Nous avons alors voulu regarder dans ces tables de hachage. Nous avons commencé par la table des Cards :

---

```
Ptr_Card : Gnoga.Gui.View.Pointer_To_View_Base_Class;
[...]
Ptr_Card := <Cards>.Card [<Card_Name>];
```

---

---

```
Gnoga.Log [From_Latin_1 [Ptr_Card .Unique_ID' Image] & " " & Ptr_Card. ID];
```

---

...qui nous affiche l'ID dans la table de hachage principale ainsi que le GID du DOM. Les ID correspondaient une nouvelle fois, et entre-temps, les anomalies de notre application ont disparu. Tous les boutons marchaient correctement sans provoquer d'erreur. Mais ça n'a pas résolu le problème de manière générale. Cependant, rajouter :

---

```
Ptr_Card := Main_Frame.Cards.Card[Name => "current"];  
Gnoga.Log [From_Latin_1 [Ptr_Card.Unique_ID' Image] & " " & Ptr_Card. ID];
```

---

recréait des bugs.

Ainsi, nous n'avons pas réussi à corriger cette anomalie.

Dans un second temps, nous avons essayé de débugger l'anomalie avec le débogueur de GNAT Studio. Voici que nous avons pu avoir :

- L'application se crée correctement, il n'y a aucune erreur à ce moment là.
- Les variables ont toutes des adresses « normales ».

Lorsque l'on appuie sur un bouton, le message suivant apparaît pour l'ensemble des variables :

- « No frame is currently executing in block application.on\_connect »

Et c'est à ce moment là qu'il y a levées d'erreurs, plus précisément quand on veut chercher la Card dans la table de hachage des Cards.

Il semblerait que les Cards soient déréférencées au moment où l'on appuie sur les boutons.

Enfin, nous avons essayé, à l'aide d'une tâche, d'afficher les tables de hachage pendant l'exécution de l'application, afin de vérifier leur contenu juste avant la levée des erreurs. Malheureusement, comme nous l'avons précisé quelques lignes plus tôt, une fois qu'on affiche les tables, tous les éléments se mettent à fonctionner correctement... Impossible donc d'avoir l'affichage de la table et levée des erreurs en même temps.

Entre-temps, une nouvelle erreur est apparue :

---

```
raised CONSTRAINT_ERROR : s-taprop.adb:493 access check failed
```

---

L'erreur renvoie à des fichiers concernant les tâches. Elle est levée au moment de l'exécution de la première ligne de la fonction passée au Click Handler : à ce moment-là, cette première ligne faisait entrer dans la tâche.

Ainsi, la levée d'erreur pourrait ne pas être due aux tables de hachage des Cards, comme les premières erreurs pouvaient le laisser penser, mais peut-être à l'exécution des fonctions passées au Handlers.

## □ Résolution

Suite à une discussion avec Pascal Pignard, nous avons compris que notre erreur venait d'une *mauvaise compréhension des connexions multiples*.

En effet, comme l'explique Pascal Pignard<sup>7</sup> : « En fait la durée de vie d'un élément est liée à sa déclaration au sein de la procédure On\_Connect. À la fin de celle-ci, l'élément est désactivé mais pas supprimé de la page Web. Nous sommes confrontés à la persistance des éléments une fois la page créée. ». Ainsi, il nous faut un outil pour récupérer l'élément désactivé.

Tout d'abord, pour conserver l'ensemble des éléments, il faut tous les mettre dans le type App\_Data vu précédemment. C'est-à-dire que, même si vous avez créé un type Widget\_Type, il faut déclarer dans le type App\_Data ce widget :

```
type My_Widget_Type is new Gnoga.Gui.View.View_Type with record
    Widget_Form : Gnoga.Gui.Element.Form.Form_Type;
    Name_Input   : Gnoga.Gui.Element.Form.Text_Type;
    Message      : Gnoga.Gui.Element.Form.Text_Area_Type;
    My_Submit    : Gnoga.Gui.Element.Form.Submit_Button_Type;
end record;
```

```
type App_Data is new Gnoga.Types.Connection_Data_Type with record
    Window : Gnoga.Gui.Window.Pointer_To_Window_Class;
    Docker : Gnoga.Gui.View.Docker.Docker_View_Type;
    View   : aliased Gnoga.Gui.View.View_Type;
    Widget : My_Widget_Type;
end record;
```

Ensuite, dans la procédure On\_Connect :

```
App : constant App_Access := new App_Data;
[...]
Main_Window.Connection_Data [App];
```

Permet justement cette conservation des éléments.

Enfin, dans les fonctions que l'on passe aux Handlers, par exemple, il faut écrire :

```
procedure On_Click_Button [Object : in out Gnoga.Gui.Base.Base_Type'Class] is
    App : constant App_Access := App_Access [Object.Connection_Data];
begin
    App.Widget.Put_Line ["Click"];
end On_Click_Button;
```

Et ne *pas* écrire :

<sup>7</sup> Extrait de Premiers pas avec GNOGA, Pascal Pignard (<https://blady.pagesperso-orange.fr/telechargements/gno-ga/gnoga.pdf>)

```
procedure On_Click_Button [Object : in out Gnoga.Gui.Base.Base_Type' Class] is
    pragma Unreferenced [Object];
begin
    App.Widget.Put_Line("Click");
end On_Click_Button;
```

---

## 9 Anomalies signalées

### 9.1 Issues Github

- hello3post.adb - In order for this source to compile and run correctly, a few updates are required #1

<https://github.com/Blady-Com/Exemples-PPAG/issues/1>

- hello3post\_File.adb - In order for this source to compile and run correctly, a few updates are required #2

<https://github.com/Blady-Com/Exemples-PPAG/issues/2>

### 9.2 Tickets Sourceforge

- Make bugs

<https://sourceforge.net/p/gnoga/tickets/52>

- Bug in test\_ssl.gpr

<https://sourceforge.net/p/gnoga/tickets/53>

- Rename everywhere [doc and sources] SSL [deeply obsolete] to TLS

<https://sourceforge.net/p/gnoga/tickets/54>

## 10 Anomalies à corriger

### 10.1 SQLite & les FS réseaux

- Problème

Les systèmes de fichiers réseaux [NFS, CIFS/SMB] ne font pas bon ménage avec SQLite, à cause [probablement] de la fonction WAL [Write Ahead Logging] de journalisation, source d'incompatibilité entre les locks de SQLite et les systèmes de fichiers réseaux.

Un moyen simple de supprimer la gestion des locks [en s'assurant qu'il n'y aura pas d'accès concurrents à la DB SQLite] est de passer le nom de la base à ouvrir en URI [Uniform Resource Identifier]. Concrètement : « [file:tutorial\\_10.db?nolock=1](file:tutorial_10.db?nolock=1) ». Mais l'accès par URI n'est pas compilé par défaut dans le paquet Debian.

## □ Conséquences

La version SQLite par défaut des distributions Debian et dérivées ainsi que la version compilée de GNATdoc ne permettent pas d'installer GNOGA sur un système de fichiers réseau NFS ou SMB sans générer des problèmes de verrouillages.

Les tutoriaux 10 & 11 utilisant SQLite ne pourront être testés et la génération de la documentation GNOGA via Gnatdoc se terminera prématurément en erreur [SQL.ERROR] Failed to execute COMMIT error=database is locked.

## □ Solution 1

Recompiler SQLite avec l'accès par URI et recompiler les programmes Ada correspondants. Les programmes résultats ne gèrent plus la concurrence d'accès [utilisation mono-utilisateur].

## □ Solution 2

Désactiver le WAL via le PRAGMA journal\_mode=DELETE et recompiler les programmes Ada correspondants. Les programmes résultats gèrent toujours la concurrence d'accès [solution à valider].

## □ Solution 3

Synchronisation non temps-réel « à la Git » entre plusieurs utilisateurs qui déporteraient en local une copie temporaire de la DS SQLite [solution complexe].

## □ Notes

Finalement, on revient aux fondamentaux : avoir le contrôle des sources, en réintégrant les sources de SQLite dans la chaîne de développement de GNOGA [en s'inspirant de ce qui a été réalisé avec v20.sql].

## 10.2 Démos

### □ Anomalie 1

Une erreur récurrente est liée au typage de certaines fonctions lorsque l'on passe celles-ci en paramètre à des Handlers. Par exemple dans la démo de JSTree, on a :

---

```
Tree.On_Open_Node_Handler    [Handler => On_Open' Unrestricted_Access];
Tree.On_Close_Node_Handler   [Handler => On_Open' Unrestricted_Access];
Tree.On_Select_Node_Handler  [Handler => On_Select' Unrestricted_Access];
Tree.On_Deselect_Node_Handler[Handler => On_Select' Unrestricted_Access];
```

---

Le paramètre de chacun de ces Handlers est un pointeur vers le type JSTree\_Event, défini dans Gnoga.Gui.Plugin.JSTree de la façon suivante :

---

```
type JSTree_Event is access procedure
  (Object : in out Gnoga.Gui.Base.Base_Type' Class;
   Node   : in      String);
```

---

Ce qui correspond parfaitement aux déclarations des procédures On\_Open et On\_Select :

---

```
procedure On_Open
  [Object : in out Gnoga.Gui.Base.Base_Type' Class;
   Node   : in      String];
procedure On_Select
  [Object   : in out Gnoga.Gui.Base.Base_Type' Class;
   Selected : in      String];
```

---

Il ne devrait alors pas y avoir de problème : les procédures ont les bons types de paramètres, et on passe bien un pointeur sur ces procédures. Pourtant, on a l'erreur :

---

```
error: expected type "JSTree_Event" defined at gnoga-gui-plugin-jstree.ads: 110
error: found type access to procedure "On_Open" defined at line 183
error: ==> in call to "On_Open_Node_Handler" at gnoga-gui-plugin-jstree.ads: 220
```

---

Et cela pour l'ensemble des fonctions passées aux Handlers.

Cependant, la procédure On\_Connect est créée et associée à un Handler de la même façon, et fonctionne correctement.

## 10.3 Anomalies dans gnoga-wf.pdf

### □ Anomalie 1

Section 3 du document.

Une fois la commande gnoga\_make exécutée, un fichier « .adb » est créé et contient à la ligne 24 [approx.] :

---

```
24      Gnoga.Log [Ada.Exceptions.Exception_Name [E] & " - "
                  & Ada.Exceptions.Exception_Message [E]];
```

---

...produit une erreur. Log est une fonction implémentée pouvant prendre en paramètre :

- [UXString, Ada.Exceptions.Exception\_Occurrence]
- ou [UXString]
- ou [Ada.Exceptions.Exception\_Occurrence].

Or ce qui est entré en paramètre est une chaîne classique i.e. [Standard.]String.

Une correction possible est alors :

---

```
24      Gnoga.Log [E];
```

---

Le type de E est bien Ada.Exceptions.Exception\_Occurrence.

De plus, dans certaines démos [tic-tac-toe notamment], on retrouve bien :

## □ Anomalie 2

Erreur ligne 59 [approx.].

```
59      if not Dec_Check_Box.Checked then
        Result_Label.Text ["Résultat " &
                           Input_Text.Value &
                           " ! = " &
                           Long_Long_Integer' Image
                           [Factorielle
                           [Natural' Value
                           [Input_Text.Value]]]);
      else
        Result_Label.Text ["Résultat " &
                           Input_Text.Value &
                           " ! = " &
                           Factorielle [Natural' Value [Input_Text.Value]]];
      end if;
```

De la même façon, l'erreur semble être due à UXStrings. Result\_Label.Text prend en paramètre un UXString or on lui donne en argument une concaténation de chaînes de caractères standards.

La solution au premier problème consiste à importer la bibliothèque UXStrings :

```
with UXStrings; Use UXStrings;
<...>
60      if not Dec_Check_Box.Checked then
        Result_Label.Text ["Résultat "
                           & Input_Text.Value
                           & " ! = "
                           & From_ASCII [Long_Long_Integer' Image [Factorielle
                           [Input_Text.Value]]]);
      else
        Result_Label.Text ["Résultat "
                           & Input_Text.Value
                           & " ! = "
                           & From_ASCII [Factorielle [Input_Text.Value]]];
      end if;
```

▷ Pour afficher du texte avec GNOGA, il est indispensable d'utiliser UXStrings.

Un autre souci aurait pu être que la méthode Value d'Input\_Text soit une méthode polymorphe. En effet, elle retourne un entier [Integer] ou une chaîne de caractères [String]. Comme Factorielle prend en entrée un entier naturel [Natural], il serait nécessaire de caster la valeur de retour de Input\_Text.Value en Natural. Or Natural [Input\_Text.Value] produit une erreur puisque la conversion est ambiguë. Il peut y avoir conversion de Integer vers Natural mais aussi String vers Natural.

De fait, *il n'est pas nécessaire de caster en Natural, ni de modifier Factorielle*. La conversion en Integer est automatique puisque Natural est un sous-type de Integer.

## 10.4 Accents dans les titres des sections de certains composants jQuery-UI

Une fois les changements appliqués sur les Accordions – qui consistait à définir les sections comme des boutons plutôt que par des titres HTML – nous avons rencontré un problème avec les accents sur les libellés de boutons.

Il est apparu nécessaire de la convertir en UXStrings. Nous avons tenté sans succès de la convertir à l'aide de la méthode From\_UTF\_8. La solution a été d'utiliser la méthode From\_Latin\_1.

---

```
Gnoga.Gui.Element.Common.Button_Access [View.New_Element [From_Latin_1[ Heading], Button]] .Create [View, From_Latin_1[ Heading]];
```

---

## 11 Modifications des sources

Dans cette section sont partagés les code des différentes fonctionnalités que nous avons rajoutées à GNOGA. Certains ajouts, comme les Folders, ne sont que des Accordions légèrement modifiés. Les Breadcrumb trails [fils d'Ariane] ont quant à eux été créés de toute pièce mais nécessiteront peut-être des améliorations.

### 11.1 Modification du template singleton de gnoga\_make

---

```
with Gnoga.Application.Singleton;
with Gnoga.Gui.Window;
with Gnoga.Gui.View;

with Gnoga.Gui.Base;
with Gnoga.Gui.Element.Common;

procedure @@data.App_Name@@ is
  My_Window : Gnoga.Gui.Window.Window_Type;
  My_View : Gnoga.Gui.View.View_Type;
  My_Exit : Gnoga.Gui.Element.Common.Button_Type;

  procedure On_Exit [Object : in out Gnoga.Gui.Base.Base_Type' Class];
  -- When this action is fired it will end the application

  procedure On_Exit [Object : in out Gnoga.Gui.Base.Base_Type' Class] is
    pragma Unreferenced [Object];
begin
  My_View.New_Line;
  My_View.Put_Line ["Closing application"];

  My_Exit.Disabled;

  Gnoga.Application.Singleton.End_Application;
end On_Exit;
begin
  Gnoga.Application.Title ["@@data.App_Name@@"];
  Gnoga.Application.HTML_On_Close ["Application ended. "];
  Gnoga.Application.Singleton.Initialize [Main_Window => My_Window];
  My_View.Create [My_Window];
  My_Exit.Create [My_View, "End App"];
  My_Exit.On_Click_Handler [On_Exit' Unrestricted_Access];
  My_View.Horizontal_Rule;
  Gnoga.Application.Singleton.Message_Loop;
```

---

```
end @@data.App_Name@@;
```

## 11.2 Modification du template multi\_connect de gnoga\_make

```
with Gnoga.Application.Multi_Connect;
with Gnoga.Gui.Base;
with Gnoga.Gui.Window;
with Gnoga.Gui.View;
with Gnoga.Gui.Element.Common;
with Gnoga.Types;

procedure @@data.App_Name@@ is
    use all type Gnoga.String;

type App_Data is new Gnoga.Types.Connection_Data_Type with record
    My_Window : Gnoga.Gui.Window.Pointer_To_Window_Class;
    My_View   : Gnoga.Gui.View.View_Type;
    My_Exit   : Gnoga.Gui.Element.Common.Button_Type;
end record;
type App_Access is access all App_Data;

procedure On_Exit [Object : in out Gnoga.Gui.Base.Base_Type'Class];

procedure On_Exit [Object : in out Gnoga.Gui.Base.Base_Type'Class] is
    App : constant App_Access := App_Access [Object.Connection_Data];
    View : Gnoga.Gui.View.View_Type;
begin
    App.My_View.Remove;
    View.Create [App.My_Window.all];
    View.Put_Line ["Application exited."];
    App.My_Window.Close_Connection;
exception
    when E : others =>
        Gnoga.Log [Message => "On_Exit: ", Occurrence => E];
end On_Exit;

procedure On_Connect
    [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
     Connection   :                                         access Gnoga.Application.Multi_Connect.Connection_Holder_Type];

procedure On_Connect
    [Main_Window : in out Gnoga.Gui.Window.Window_Type'Class;
     Connection   :                                         access Gnoga.Application.Multi_Connect.Connection_Holder_Type]
is
    pragma Unreferenced [Connection];
    App : constant App_Access := new App_Data;
begin
    App.My_Window := Main_Window'Unchecked_Access;
    Main_Window.Connection_Data [App];

    App.My_View.Create [Main_Window];
    App.My_Exit.Create [App.My_View, "Exit App"];
    App.My_Exit.On_Click_Handler [On_Exit'Unrestricted_Access];

    App.My_View.Horizontal_Rule;
end On_Connect;

begin
    Gnoga.Application.Title ["@@data.App_Name@@"];
    Gnoga.Application.HTML_On_Close ["Application ended."];
    Gnoga.Application.Multi_Connect.Initialize;
```

```

    Gnoga.Application.Multi_Connect.On_Connect_Handler [Event => On_Connect' Unrestricted_Access, Path => "default"];

    Gnoga.Application.Multi_Connect.Message_Loop;
end @@data.App_Name@@;

```

---

## 11.3 Ajout de Breadcrumb.ads

---

```

with Gnoga.Gui.Base;
with Gnoga.Gui.View;
with UXStrings; use UXStrings;

package Breadcrumb is

    function Add_To_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Handler   : in     Gnoga.Gui.Base.Action_Event;
         Content   : in     UXString := "";
         Actual_Depth : in     Integer;
         Depth     : in     Integer := 0]
    return Integer;

    function Remove_From_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Actual_Depth : in     Integer]
    return Integer;

    function Update_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Handler   : in     Gnoga.Gui.Base.Action_Event;
         Content   : in     UXString := "";
         Actual_Depth : in out Integer;
         Depth     : in     Integer := 0]
    return Integer;

end Breadcrumb;

```

---

## 11.4 Ajout de Breadcrumb.adb

---

```

with Gnoga.Gui.Element.Common;
with Gnoga.Gui.Element;

package body Breadcrumb is

    function Add_To_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Handler   : in     Gnoga.Gui.Base.Action_Event;
         Content   : in     UXString := "";
         Actual_Depth : in     Integer;
         Depth     : in     Integer := 0]
    return Integer
    is
        Left_String : UXString := "";
        Element_Name : UXString := "";
        New_Button : constant Gnoga.Gui.Element.Pointer_To_Element_Class := new Gnoga.Gui.Element.Common.Button_Type;
    begin
        Element_Name := "Button_" & From_UTF_8[Integer'Image[Depth]].Delete[1,1];
        Left_String := From_UTF_8[Integer'Image[120 + [Depth] * 120]];
        Gnoga.Gui.Element.Common.Button_Access[New_Button].Create [View, Content];
        View.Add_Element(Element_Name, New_Button);
        New_Button.Dynamic;
        New_Button.On_Click_Handler[Handler];
        New_Button.Style["position", "absolute"];

```

---

Prise en main de GNOGA

---

```

        New_Button.Style["top", "0px"];
        New_Button.Style["left", Left_String & "px"];
        return Actual_Depth;
    end Add_To_Breadcrumb;

    function Remove_From_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Actual_Depth : in     Integer]
        return Integer
    is
        Element_Name : UXString := "";
    begin
        Element_Name := "Button_" & From_UTF_8[Actual_Depth' Image].Delete[1, 1];
        View.Element[Element_Name].Remove;
        return Actual_Depth;
    end Remove_From_Breadcrumb;

    function Update_Breadcrumb
        [View      : in out Gnoga.Gui.View.View_Type;
         Handler   : in     Gnoga.Gui.Base.Action_Event;
         Content   : in     UXString := "";
         Actual_Depth : in out Integer;
         Depth     : in     Integer := 0]
        return Integer
    is
    begin
        if Actual_Depth < Depth then
            Actual_Depth := Add_To_Breadcrumb[View    => View,
                                              Handler => Handler,
                                              Content => Content,
                                              Actual_Depth => Actual_Depth,
                                              Depth   => Depth];
        elsif Actual_Depth = Depth then
            Actual_Depth := Remove_From_Breadcrumb[View  => View,
                                                    Actual_Depth => Actual_Depth];
            Actual_Depth := Add_To_Breadcrumb[View    => View,
                                              Handler => Handler,
                                              Content => Content,
                                              Actual_Depth => Actual_Depth,
                                              Depth   => Depth];
        elsif Actual_Depth > Depth then
            while Actual_Depth > Depth loop
                Actual_Depth := Remove_From_Breadcrumb[View  => View,
                                                        Actual_Depth => Actual_Depth];
                Actual_Depth := Actual_Depth - 1;
            end loop;
        end if;
        Actual_Depth := Depth;
        return Actual_Depth;
    end Update_Breadcrumb;
end Breadcrumb;

```

---

## 11.5 Ajout de Simple\_Form.ads

---

```

with Gnoga.Gui.Base;
with Gnoga.Gui.Element.Form;
with Gnoga.Gui.View;
with UXStrings; use UXStrings;

package Simple_Form is

    type String_Array is array [Positive range <>] of UXString;

```

---

---

```

type Row_Type is new Gnoga.Gui.View.View_Type with record
    Input : aliased Gnoga.Gui.Element.Form.Text_Type;
    Row_Name : aliased UXString;
end record;

type Row_Set is array [Positive range <>] of Row_Type;
type Row_Set_Ptr is access all Row_Set;

type Form_View_Type is new Gnoga.Gui.View.View_Type with record
    Form      : Gnoga.Gui.Element.Form.Form_Type;
    Message   : Gnoga.Gui.Element.Form.Text_Area_Type;
    Submit    : Gnoga.Gui.Element.Form.Submit_Button_Type;
    Row_Count : Positive;
end record;

function Field_By_Index
    [Source : in UXString;
     Index_Num : in Positive;
     Separator : UXString]
return UXString;

function All_Fields
    [Source : in UXString;
     Separator : UXString]
return String_Array;

function Remove_First_Occurrence
    [Source : in UXString;
     Letter : UXString]
return UXString;

function Remove_All_Occurrences
    [Source : in UXString;
     Letter : UXString]
return UXString;

procedure Create
    [View      : in out Form_View_Type;
     Parent    : in out Gnoga.Gui.Base.Base_Type'Class;
     ID       : in      Gnoga.String := "";
     Strings  : in String_Array];
end Simple_Form;

```

---

## 11.6 Ajout de Simple\_Form.adb

---

```

with Gnoga.Gui.Element.Table;

package body Simple_Form is

    function Field_By_Index
        [Source : in UXString;
         Index_Num : in Positive;
         Separator : UXString]
        return UXString
    is
        Source_Copy : UXString := Source;
        Field : UXString;
        Field_Num : Integer := 0;
        Cursor : Integer;
    begin
        while Field_Num < Index_Num loop
            Cursor := Integer'Min [Index [Source_Copy, Separator], Source_Copy.Length];
            Field := Head[Source_Copy, Cursor-1];
            Source_Copy := Tail[Source_Copy, Source_Copy.Length-Cursor];
            Field_Num := Field_Num +1;
    
```

---

---

```

    end loop;
    return Field;
end Field_By_Index;

function All_Fields
  (Source : in UXString;
   Separator : UXString)
  return String_Array
is
  Source_Copy : UXString := Source;
  Separator_Count : constant Positive := Count[Source, Separator];
  Fields : String_Array[1..Separator_Count+1];
  Field_Num : Positive := 1;
  Cursor : Positive;
begin
  while Field_Num <= Separator_Count loop
    Cursor := Index[Source_Copy, Separator];
    Fields[Field_Num] := Head[Source_Copy, Cursor-1];
    Source_Copy := Tail[Source_Copy, Source_Copy.Length-Cursor];
    Field_Num := Field_Num +1;
  end loop;
  Cursor := Source_Copy.Length+1;
  Fields[Field_Num] := Head[Source_Copy, Cursor-1];
  return Fields;
end All_Fields;

function Remove_First_Occurrence
  (Source : in UXString;
   Letter : UXString)
  return UXString
is
  Result : UXString := Source;
  Tail_Source : UXString;
  Cursor : Integer;
begin
  Cursor := Index[Result, Letter];
  if Cursor > 0 then
    Tail_Source := Tail[Result, Result.Length-Cursor];
    Result := Head[Result, Cursor-1] & Tail_Source;
  end if;
  return Result;
end Remove_First_Occurrence;

function Remove_All_Occurrences
  (Source : in UXString;
   Letter : UXString)
  return UXString
is
  Letter_Count : constant Positive := Count[Source, Letter];
  Result : UXString := Source;
  Tail_Source : UXString;
  Cursor : Integer;
  Field_Num : Positive := 1;
begin
  while Field_Num <= Letter_Count loop
    Cursor := Index[Result, Letter];
    if Cursor > 0 then
      Tail_Source := Tail[Result, Result.Length-Cursor];
      Result := Head[Result, Cursor-1] & Tail_Source;
    end if;
    Field_Num := Field_Num +1;
  end loop;
  return Result;
end Remove_All_Occurrences;

procedure Create
  (View : in out Form_View_Type;
   Parent : in out Gnoga.Gui.Base.Base_Type'Class;
   ID : in Gnoga.String := "");

```

---

Prise en main de GNOGA



---

```

        Strings : String_Array]
is
use Gnoga.Gui.Element.Table;
Layout_Table : constant Table_Access := new Table_Type;

begin
    Gnoga.Gui.View.View_Type [View].Create [Parent, ID];

    View.Form.Create [View, "result", Gnoga.Gui.Element.Form.Post];
    Layout_Table.Dynamic;
    Layout_Table.Create [View];
    View.Row_Count := Strings'Length;

    declare
        Rows : constant Row_Set_Ptr := new Row_Set[1..View.Row_Count];
    begin
        for i in 1..View.Row_Count loop
            declare
                row : constant Table_Row_Access := new Table_Row_Type;
                col1 : constant Table_Column_Access := new Table_Column_Type;
                col2 : constant Table_Column_Access := new Table_Column_Type;
            begin
                row.Dynamic;
                col1.Dynamic;
                col2.Dynamic;

                row.Create [Layout_Table.all];
                Rows[i].Row_Name := Strings[i];

                Rows[i].Input.Create [Form => View.Form, Size => 40, Name =>
Remove_First_Occurrence[Rows[i].Row_Name, " "]];
                col1.Create [row.all, Rows[i].Row_Name];
                col2.Create [row.all];
                Rows[i].Input.Place_Inside_Top_Of [col2.all];
            end;
        end loop;
    end;
    View.Submit.Create [Form => View.Form, Value => "Envoyer"];
    View.Submit.Place_After [Layout_Table.all];
end Create;

end Simple_Form;

```

---

## 11.7 Ajout de Folders.ads

---

```

with Gnoga.Gui.Base;
with Gnoga.Gui.View;

package Folders is

    type Folder_Type is new Gnoga.Gui.View.View_Type with private;
    type Folder_Access is access all Folder_Type;
    type Pointer_To_Folder_Class is access all Folder_Type'Class;

    procedure Create_Folder
    (View   : in out Folder_Type;
     Parent : in out Gnoga.Gui.Base.Base_Type'Class;
     ID     : in      String := "");

    procedure Create_Section
    (View   : in out Folder_Type;
     Content :          String ;
     Name   :          String ;
     IMG_URL :          String    := "");

```

---

---

```

procedure Render_Folder
  (View          : in out Folder_Type;
   Allow_Collapse : in      Boolean := False);

private
  type Folder_Type is new Gnoga.Gui.View.View_Type with null record;
end Folders;

```

---

## 11.8 Ajout de Folders.adb

```

with Gnoga.Gui.Element;
with Gnoga.Gui.Element.Common;
with UXStrings; use UXStrings;

package body Folders is

  procedure Create_Folder
    (View      : in out Folder_Type;
     Parent   : in out Gnoga.Gui.Base.Base_Type'Class;
     ID       : in      String := "");
  is
  begin
    Gnoga.Gui.View.View_Type (View).Create (Parent, From_UTF_8 (ID));
  end Create_Folder;

  procedure Create_Section
    (View      : in out Folder_Type;
     Content  :          String      ;
     Name     :          String      ;
     IMG_URL :          String      := "") 
  is
    Button : constant Gnoga.Gui.Element.Pointer_To_Element_Class := new Gnoga.-
      Gui.Element.Common.Button_Type;
  begin
    Gnoga.Gui.Element.Common.Button_Access (Button).Create
    (View, From_Latin_1 (Content));
    View.Add_Element (From_Latin_1 (Name), Button);
    Button.Dynamic;
    if IMG_URL /= "" then
      Button.Background_Image [Value => From_Latin_1 (IMG_URL)];
      Button.Background ["linear-gradient(green, blue)"];
      Button.Background_Position [Value => "10px 10px"];
      Button.Background_Repeat [Value => "no-repeat"];
      Button.Background_Size [Value => "30px 30px"];
    end if;
  end Create_Section;

  procedure Render_Folder
    (View          : in out Folder_Type;
     Allow_Collapse : in      Boolean := False)
  is
    function params return String;
    function params return String is
    begin
      if Allow_Collapse = False then
        return "";
      else
        return "{ collapsible: true }";
      end if;
    end params;
  begin
    View.jQuery_Execute (From_UTF_8 ("folder[" & params & "]"));
  end Render_Folder;

```

---

```
end Folders;
```

---

## 11.9 Ajout de Folders.js

---

```
[function ($) {
$.widget("ui.folder", {
    version: "1.12.1",
    options: {
        active: 0,
        animate: {},
        classes: { "ui-folder-header": "ui-corner-top", "ui-folder-header-collapsed": "ui-corner-all", "ui-folder-content": "ui-corner-bottom" },
        collapsible: !1,
        event: "click",
        header: "> li > :first-child, > :not[li]:even",
        widthStyle: "auto",
        icons: { activeHeader: "ui-icon-triangle-1-e", header: "ui-icon-triangle-1-s" },
        activate: null,
        beforeActivate: null,
    },
    hideProps: { borderTopWidth: "hide", borderBottomWidth: "hide", paddingTop: "hide", paddingBottom: "hide", width: "hide" },
    showProps: { borderTopWidth: "show", borderBottomWidth: "show", paddingTop: "show", paddingBottom: "show", width: "show" },
    _create: function () {
        var e = this.options;
        [this.prevShow = this.prevHide = $[]],
        this._addClass("ui-folder", "ui-widget ui-helper-reset"),
        this.element.attr("role", "tablist"),
        e.collapsible || [e.active !== !1 && null != e.active] || [e.active = 0],
        this._processPanels(),
        0 > e.active && [e.active += this.headers.length],
        this._refresh();
    },
    _getCreateEventData: function () {
        return { header: this.active, panel: this.active.length ? this.active.next() : [] };
    },
    _createIcons: function () {
        var e,
        i,
        s = this.options.icons;
        s && [
            [{e = $("<span>")}, this._addClass(e, "ui-folder-header-icon", "ui-icon " + s.header),
            e.prependTo(this.headers),
            [i = this.active.children(".ui-folder-header-icon")], this._removeClass(i, s.header)._addClass(i, null, s.activeHeader)._addClass(this.headers, "ui-folder-icons")],
        ],
        _destroyIcons: function () {
            this._removeClass(this.headers, "ui-folder-icons"), this.headers.children(".ui-folder-header-icon").remove();
        },
        _destroy: function () {
            var $;
            this.elementremoveAttr("role"),
            this.headers.removeAttr("role aria-expanded aria-selected aria-controls tabIndex").removeUniqueId(),
            this._destroyIcons(),
            [$ = this.headers.next().css("display", "")].removeAttr("role aria-hidden aria-labelledby");
        },
        _setOption: function ($, e) {
            return "active" === $
        }
    }
});
```

---



---

```

        ? [this._activate[e], void 0]
        : ["event" === $ && [this.options.event && this._off[this.headers, this.options.event], this._setupEvents[e]],
          this._super[$, e],
          "collapsible" !== $ || e || this.options.active !== !1 || this._activate[0],
          "icons" === $ && [this._destroyIcons[], e && this._createIcons()],
          void 0];
      },
      _setOptionDisabled: function ($) {
        this._super[$], this.element.attr("aria-disabled", $), this._toggleClass(null, "ui-state-disabled", !$), this._toggleClass(this.headers.add[this.headers.next()], null, "ui-state-disabled", !$);
      },
      _keydown: function (e) {
        if (!e.altKey && !e.ctrlKey) {
          var i = $.ui.keyCode,
              s = this.headers.length,
              n = this.headers.index(e.target),
              o = !1;
          switch (e.keyCode) {
            case i.RIGHT:
            case i.DOWN:
              o = this.headers[(n + 1) % s];
              break;
            case i.LEFT:
            case i.UP:
              o = this.headers[(n - 1 + s) % s];
              break;
            case i.SPACE:
            case i.ENTER:
              this._eventHandler[e];
              break;
            case i.HOME:
              o = this.headers[0];
              break;
            case i.END:
              o = this.headers[s - 1];
          }
          o && [$[e.target].attr("tabIndex", -1), $[o].attr("tabIndex", 0),
$[o].trigger("focus"), e.preventDefault()];
        }
      },
      _panelKeyDown: function (e) {
        e.keyCode === $.ui.keyCode.UP && e.ctrlKey &&
t[e.currentTarget].prev().trigger("focus");
      },
      refresh: function () {
        var e = this.options;
        this._processPanels(),
        [e.active === !1 && e.collapsible === !0] || !this.headers.length
        ? ([e.active = !1], [this.active = t()])
        : e.active === !1
        ? this._activate[0]
        : this.active.length && !$.contains(this.element[0], this.active[0])
        ? this.headers.length === this.headers.find(".ui-state-dis-
bled").width
        ? ([e.active = !1], [this.active = $()])
        : this._activate[Math.max(0, e.active - 1)]
        : [e.active = this.headers.index(this.active)],
        this._destroyIcons[],
        this._refresh();
      },
      _processPanels: function () {
        var $ = this.headers,
            e = this.panels;
        [this.headers = this.element.find(this.options.header)],
        this._addClass(this.headers, "ui-folder-header ui-folder-header-
collapsed", "ui-state-default"),

```

---

Prise en main de GNOGA



```

        [this.panels = this.headers.next().filter(":not(.ui-folder-content-active)").hide()],
        this._addClass(this.panels, "ui-folder-content", "ui-helper-reset ui-widget-content"),
        e && [this._off(t.not[this.headers]), this._off(e.not[this.panels])];
    },
    _refresh: function () {
        var e,
            i = this.options,
            s = i.widthStyle,
            n = this.element.parent();
        [this.active = this._findActive(i)],
        this._addClass(this.active, "ui-folder-header-active", "ui-state-active")._removeClass(this.active, "ui-folder-header-collapsed"),
        this._addClass(this.active.next(), "ui-folder-content-active"),
        this.headers
            .attr("role", "tab")
            .each(function () {
                var e = $(this),
                    i = e.uniqueId().attr("id"),
                    s = e.next(),
                    n = s.uniqueId().attr("id");
                e.attr("aria-controls", n), s.attr("aria-labelledby", i);
            })
            .next()
            .attr("role", "tabpanel"),
        this.headers.not[this.active].attr({ "aria-selected": "false", "false", tabIndex: -1 }).next().attr({ "aria-hidden": "true" }).hide(),
        "aria-expanded": "true" ].hide(),
        this.active.length ? this.active.attr({ "aria-selected": "true", "aria-expanded": "true", tabIndex: 0 }).next().attr({ "aria-hidden": "false" }) : this.headers.eq(0).attr("tabIndex", 0),
        this._createIcons(),
        this._setupEvents[i.event],
        "fill" === s
            ? [{e = n.width}],
            this.element.siblings(":visible").each(function () {
                var i = $(this),
                    s = i.css("position");
                "absolute" !== s && "fixed" !== s && {e -= i.outerWidth(!0)};
            }),
            this.headers.each(function () {
                e -= $(this).outerWidth(!0);
            }),
            this.headers
                .next()
                .each(function () {
                    $(this).width(Math.max(0, e - $(this).innerWidth() + $[this].width())));
                })
                .css("overflow", "auto")
            : "auto" === s &&
            [{e = 0},
            this.headers
                .next()
                .each(function () {
                    var i = $(this).is(":visible");
                    i || $(this).show(), {e = Math.max(e, $[this].css("width", "").width())}, i || $(this).hide();
                })
                .width(e)];
        },
        _activate: function (e) {
            var i = this._findActive(e)[0];
            i !== this.active[0] && [{i = i || this.active[0]}, this._eventHandler({ target: i, currentTarget: i, preventDefault: t.noop })];
        },
        _findActive: function (e) {
            return "number" == typeof e ? this.headers.eq(e) : $[];
        }
    }
}

```

Prise en main de GNOGA



```

        },
        _setupEvents: function [e] {
            var i = { keydown: "_keydown" };
            e &&
                $.each[e.split(" "], function [$, e] {
                    i[e] = "_eventHandler";
                }],
                this._off[this.headers.add[this.headers.next()]],
                this._on[this.headers, i],
                this._on[this.headers.next(), { keydown: "_panelKeyDown" }],
                this._hoverable[this.headers],
                this._focusable[this.headers];
        },
        _eventHandler: function [e] {
            var i,
                s,
                n = this.options,
                o = this.active,
                a = ${e.currentTarget},
                r = a[0] === o[0],
                h = r && n.collapsible,
                l = h ? $[] : a.next(),
                c = o.next(),
                u = { oldHeader: o, oldPanel: c, newHeader: h ? $[] : a, newPanel: l };
            e.preventDefault(),
            [r && !n.collapsible] ||
                this._trigger["beforeActivate", e, u] === !1 ||
                [{n.active = h ? !1 : this.headers.index[a]}, {this.active = r ? $[] : a}, this._toggle[u], this._removeClass[o, "ui-folder-header-active", "ui-state-active"], n.icons && [{i = o.children[".ui-folder-header-icon"]}], this._removeClass[i, null, n.icons.activeHeader]._addClass[i, null, n.icons.header]], r ||
                    [this._removeClass[a, "ui-folder-header-collapsed"]._addClass[a, "ui-folder-header-active", "ui-state-active"], n.icons && [{s = a.children[".ui-folder-header-icon"]}], this._removeClass[s, null, n.icons.header]._addClass[s, null, n.icons.activeHeader]], this._addClass[a.next(), "ui-folder-content-active"]];
            },
            _toggle: function [e] {
                var i = e.newPanel,
                    s = this.prevShow.length ? this.prevShow : e.oldPanel,
                    this.prevShow.add[this.prevHide].stop(!0, !0),
                    [this.prevShow = i],
                    [this.prevHide = s],
                    this.options.animate ? this._animate[i, s, e] : [s.hide[], i.show[], this._toggleComplete[e]],
                    s.attr[{ "aria-hidden": "true" }],
                    s.prev().attr[{ "aria-selected": "false", "aria-expanded": "false" }],
                    i.length && s.length
                        ? s.prev().attr[{ tabIndex: -1, "aria-expanded": "false" }]
                        : i.length &&
                            this.headers
                                .filter[function () {
                                    return 0 === parseInt[$(this)].attr["TabIndex"], 10];
                                }]
                                .attr["TabIndex", -1],
                            i.attr["aria-hidden", "false"].prev().attr[{ "aria-selected": "true", "aria-expanded": "true" }, tabIndex: 0];
                },
                _animate: function [$, e, i] {
                    var s,
                        n,
                        o,
                        a = this,
                        r = 0,

```

---

```

        h = $.css["box-sizing"],
        l = $.length && [!e.length || $.index[] < e.index[]],
        c = this.options.animate || {},
        u = [l && c.down] || c,
        d = function () {
            a._toggleComplete[i];
        };
        return [
            "number" == typeof u && [o = u],
            "string" == typeof u && [n = u],
            [n = n || u.easing || c.easing],
            [o = o || u.duration || c.duration],
            e.length
                ? $.length
                    ? [{s = $.show().outerWidth()}, e.animate[this.hideProps, {
                        duration: o,
                        easing: n,
                        step: function ($, e) {
                            e.now = Math.round($);
                        },
                    }],
                    $.hide().animate[this.showProps, {
                        duration: o,
                        easing: n,
                        complete: d,
                        step: function ($, i) {
                            [i.now = Math.round($)], "width" !== i.prop ?
                                "content-box" === h && [r += i.now] : "content" !== a.options.widthStyle && [{i.now =
                                Math.round(s - e.outerWidth() - r)], [r = 0]};
                            },
                        },
                    void 0]
                    : e.animate[this.hideProps, o, n, d]
                : $.animate[this.showProps, o, n, d]
            ];
        },
        _toggleComplete: function ($) {
            var e = $.oldPanel,
            i = e.prev();
            this._removeClass[e, "ui-folder-content-active"],
            this._removeClass[i, "ui-folder-header-active"]._addClass[i, "ui-
            folder-header-collapsed"],
            e.length && [e.parent()[0].className = e.parent()[0].className],
            this._trigger["activate", null, $];
        },
    ];
}][jQuery]];

```

---

## 11.10 Ajout de Folders.css

---

```

.ui-folder .ui-folder-header {
    font-size: 40px;
    color: white;
    display: flex;
    cursor: pointer;
    position: relative;
    margin: 5px 5px 5px 5px;
    padding: 8px 5px 0px 5px;
    font-size: 100%;
    width: 100px;
    height: 35px;
}

.ui-folder .ui-folder-content {
    border-top: 0;
    overflow: hidden;
    position : absolute;

```

---

Prise en main de GNOGA

```
    top : 0px;
    width: 100px;
    left : 130px;
    z-index : 1;
    border-top-left-radius: 6px;
    border-top-right-radius: 6px;
    border-bottom-right-radius: 6px;
    border-bottom-left-radius: 6px;
}
```

---

## 11.11 Modifications à apporter à boot.html pour inclure le package Folders

---

```
<link href="/css/folders.css" rel="stylesheet">
<script src="/js/folders.js"></script>
```

---



Ada, « it's stronger than you ».  
Tribute to Daniel Feneuille, a legendary french Ada teacher (and much more)<sup>8</sup>



<sup>8</sup> <http://d.feneuille.free.fr>