

Premiers pas avec GNAT Programming Studio

GNAT Programming Studio (GPS) est l'environnement de programmation graphique fournit par AdaCore pour accompagner le compilateur Ada GNAT. Il est disponible en licence GPL.

Sommaire

1.	Installer GPS	2
2.	Première utilisation	2
3.	Mon environnement GPS personnalisé	2
4.	Création de notre premier projet	4
5.	Édition d'un code source	5
6.	Recherche dans un code source	7

1. Installer GPS

GPS est présent avec l'installation de GNAT Community présent sur www.adacore.com/community. L'installation pour macOS est décrite sur Blady blady.pagesperso-orange.fr/creations.html#gnaosxinstall. Un conteneur d'application macOS pour GPS est inclus avec la bibliothèque XNAdaLib présente sur Blady blady.pagesperso-orange.fr/creations.html#xadalib (dossier apps).

2. Première utilisation

Lancer GPS en ligne de commande ou avec l'application GPS (sous macOS par exemple).

```
$ gps &
```

Lors du premier lancement de GPS, une fenêtre apparaît nous proposant de sélectionner le thème graphique par défaut ou un mode sombre, cliquer sur le bouton "Next" après avoir fait son choix. L'affichage suivant propose de sélectionner le mode des raccourcis macOS ou Emacs, cliquer sur le bouton "Next" après avoir fait son choix, je choisis de conserver les raccourcis macOS. L'affichage qui vient propose des réglages généraux, cliquer sur le bouton "Next" après avoir fait son choix, je choisis juste de décocher "Save desktop on exit". Le dernier affichage propose un choix de plugins, cliquer sur le bouton "Apply" après avoir fait son choix, je choisis d'activer le composant "Copy Paste". La fenêtre principale de GPS apparaît avec au centre des liens sur la documentation.

Il nous reste quelques paramètres à positionner avant de nous lancer dans notre premier projet.

Astuce : pour repartir d'une configuration initiale de GPS si vous l'avez déjà utilisé, effacer alors ou renommer le dossier "\$HOME/.gps".

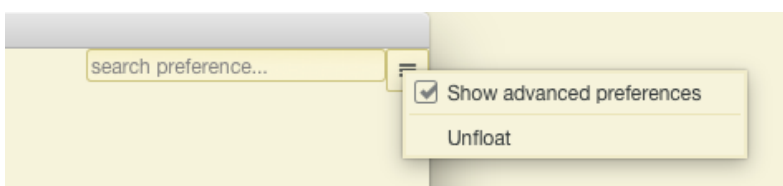
3. Mon environnement GPS personnalisé

GPS offre une profusion de configurations pour personnaliser son utilisation quitte parfois à ne pas savoir quel est exactement leurs effets.

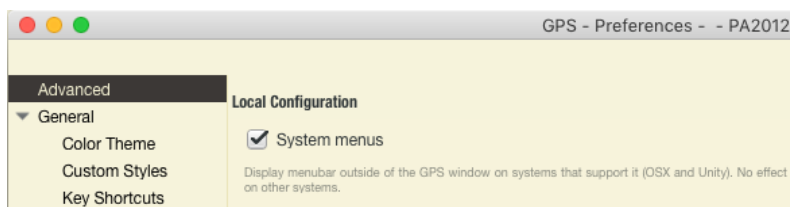
Voyons quelques paramètres, raccourcis clavier et plugins qui vont me servir pour tous mes projets de programmation avec GPS.

a) Les paramètres

Tout d'abord j'active l'affichage des préférences avancées avec le menu Edit -> Preferences puis le menu hamburger en haut à droite et en cochant "Show advanced preferences" :



Le menu Advanced apparait alors en haut à gauche, j'active tout de suite le placement de la barre de menu à la place habituelle pour macOS :



Dans le panneau Advanced, je positionne :

- "System menus" case cochée
- "Jump to first location" case décochée
- "Show system files" case cochée
- "Show files from project only" case cochée
- "Show hidden files" case cochée
- "Ellipsize long file names in views" case cochée
- "Select on match" case cochée

Dans le panneau General, je positionne :

- "Save desktop on exit" case décochée
- "Save editor in desktop" avec Never
- "Character set" avec "Unicode UTF-8"

Dans le panneau Editor, je positionne :

- "Right margin" avec 120

Récapitulatif des préférences enregistrées dans le fichier `$HOME/.gps/preferences.xml` :

```
<pref name="show-advanced-preferences" >TRUE</pref>
<pref name="system-menus" >TRUE</pref>
<pref name="locations-auto-jump-to-first" >FALSE</pref>
<pref name="browser_show_system_files" >TRUE</pref>
<pref name="explorers-file-show-project-only" >TRUE</pref>
<pref name="explorer-show-hidden-directories" >TRUE</pref>
<pref name="explorer-show-ellipsis" >TRUE</pref>
<pref name="Search-Select-On-Match" >TRUE</pref>
<pref name="General-Save-Desktop-On-Exit" >FALSE</pref>
<pref name="General-Editor-Desktop-Policy" >Never</pref>
<pref name="General-Charset" >UTF-8</pref>
<pref name="Src-Editor-Highlight-Column" > 120</pref>
```

b) Les raccourcis clavier

Dans le panneau General -> Key Shortcuts, je positionne les raccourcis :

- "Decrease text size" avec aucune touche
- "Comment lines" avec la touche `<cmd>-`
- "Uncomment lines" avec la touche `<maj><cmd>_`
- "Redo" avec la touche `<maj><cmd>Z`
- "Format selection" avec la touche `<ctrl><tab>`
- "Complete identifier" avec la touche `<tab>`

Récapitulatif des préférences enregistrées dans le fichier `$HOME/.gps/keys6.xml` :

```
<key action="Decrease text size" />
<key action="Comment lines" >primary-minus</key>
<key action="Uncomment lines" >shift-primary-underscore</key>
<key action="Redo" >shift-primary-Z</key>
<key action="Format selection" >control-Tab</key>
<key action="Complete identifier" >Tab</key>
```

c) Les plugins

J'ajoute deux composants de mon cru dans le dossier `$HOME/.gps/plug-ins` :

- "atos.py" : convertit les adresses fournies en lignes de code source lors d'une exception (pour macOS)
- "title.py" : créé un pragma Title en un signet

(Codes disponibles sur Blady blady.pagesperso-orange.fr/a_savoir.html#gnat)

Dans le panneau Plugins, je positionne les plugins :

- "ada2wsdl" case décochée
- "atos" case cochée
- "aws" case décochée
- "aws api" case décochée
- "copy file name" case cochée
- "copy paste" case cochée
- "create missingdirs" case cochée
- "gnatr" case cochée
- "pep8 integration" case décochée
- "qgen" case décochée
- "title" case cochée
- "unused entities" case cochée
- "wsdl2aws" case décochée

Récapitulatif des préférences enregistrées dans le fichier `$HOME/.gps/startup.xml` :

```
<startup file="copy_file_name.py" load="TRUE" />
<startup file="gnatr.py" load="TRUE" />
<startup file="pep8_integration.py" load="FALSE" />
<startup file="aws.xml" load="FALSE" />
<startup file="unused_entities.py" load="TRUE" />
<startup file="wsdl2aws.xml" load="FALSE" />
<startup file="aws_api.xml" load="FALSE" />
<startup file="create_missing_dirs.py" load="TRUE" />
<startup file="qgen" load="FALSE" />
<startup file="copy_paste.py" load="TRUE" />
<startup file="ada2wsdl.xml" load="FALSE" />
```

4. Création de notre premier projet

Au démarrage de GPS cliquer sur "Create new project" dans le panneau "Welcome to GPS" ou une fois ouvert sélectionner le menu File -> New Project.

Une fenêtre apparait avec "Simple Ada Project" sélectionner, cela nous va très bien, cliquer sur le bouton "Next". Indiquer le dossier de votre nouveau projet dans "Deploy project in", son nom

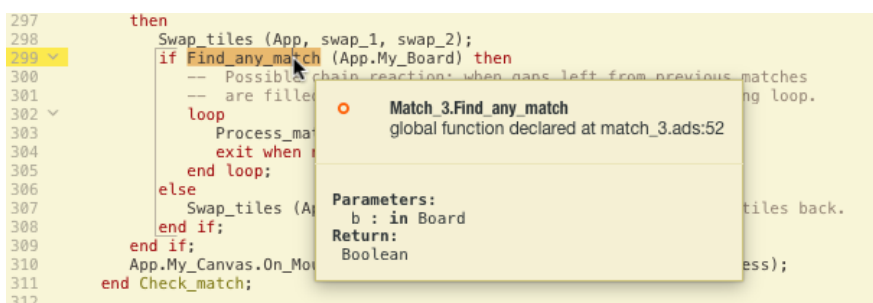
dans "Project name" ainsi que le nom du programme principal dans "Main Name". Cliquer enfin sur le bouton "Apply". Dans l'onglet Project, ouvrir le dossier src puis ouvrir le source du programme principal, tout est prêt pour insérer ses premières lignes en Ada.

Il n'y a plus qu'à cliquer sur le bouton "Build & Run" pour lancer la compilation et l'exécution du programme.

5. Édition d'un code source

Le code source est affiché avec la syntaxe mise en valeur : mot-clé, commentaires, chaînes de caractères, nombres, caractères... Ces mises en valeurs sont modifiables dans le menu préférences.

En parcourant le code source avec le curseur de la souris, il apparaît des fenêtres contextuelles avec des informations sur l'objet survolé.



```
297     then
298     Swap_tiles (App, swap_1, swap_2);
299     if Find_any_match (App.My_Board) then
300     -- Possible chain reaction: when gaps left from previous matches
301     -- are filled, a new loop may be needed.
302     loop
303     Process_matches;
304     exit when no_matches;
305     end loop;
306     else
307     Swap_tiles (App, swap_1, swap_2);
308     end if;
309 end if;
310 App.My_Canvas.On_Move (App, X, Y);
311 end Check_match;
312
```

Match 3.Find_any_match
global function declared at match_3.ads:52

Parameters:
b : in Board

Return:
Boolean

Ces informations seront d'autant plus complètes avec une compilation aboutie.

Un clic droit de la souris fait apparaître un menu contextuel avec de nombreuses possibilités correspondant à l'identificateur cliqué.



```
298 Swap_tiles (App, swap_1, swap_2);
299 if Find_any_match (App.My_Board) then
300 -- Possible chain reaction: when gaps left from previous matches
301 -- are filled, a new loop may be needed.
302 loop
303 Process_matches;
304 exit when no_matches;
305 end loop;
306 else
307 Swap_tiles (App, swap_1, swap_2);
308 end if;
309 end if;
310 App.My_Canvas.On_Move (App, X, Y);
311 end Check_match;
312
313 procedure On_Connect
314 (Main_Window : in out Window;
315 Connection : in Connection;
316 .Connection_Holder : Connection_Holder) is
317
318 procedure Mouse_Move
319 (Object : in out Object;
320 Mouse_Event : in Mouse_Event) is
321
322 procedure Mouse_Up
323 (Object : in out Object;
324 Mouse_Event : in Mouse_Event) is
325
326 procedure On_Connect
327 (Main_Window : in out Window;
328 Connection : in Connection;
329 .Connection_Holder : Connection_Holder) is
330
331 pragma Unreferenced (Main_Window);
332 App : constant App := App;
333 G : Plugin.Pixi.Game;
334 begin
335 Main_Window.Connect (App, G);
336 App.Main_Window := Main_Window;
337
```

Paste

Goto declaration of Find_any_match

Goto body of Find_any_match

Goto type declaration of Find_any_match

Goto file spec<->body

Find_any_match calls

Find_any_match is called by

Debug

References

Browsers

Locate in Project View: leaves_main.adb

Version Control

Refactoring

Coverage

Casing

Align

Expanded code

Check Coding standard of leaves_main.adb

Preprocessing

Run OS shell in leaves

SPARK

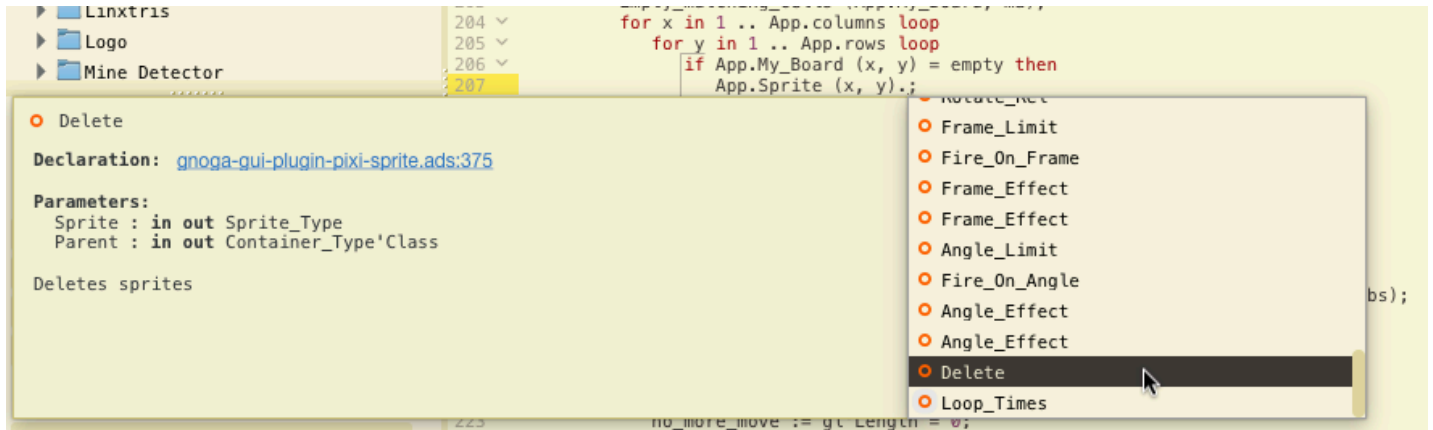
Properties...

Un <ctrl>clic sur un identificateur fait apparaître un lien qui emmène vers sa déclaration de la même façon que le menu contextuel "Goto declaration".

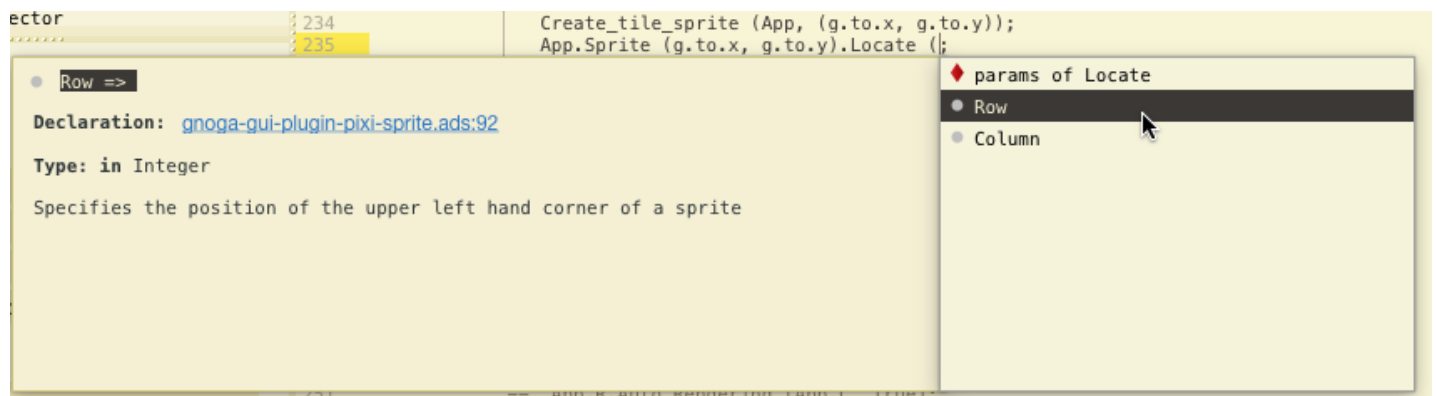
Pour sélectionner un bloc utiliser la souris en tirant sur le bloc à sélectionner ou utiliser la touche <maj> avec les touches fléchées. La sélection de bloc s'effectue de la même façon, le début et la fin de bloc correspondent respectivement au début et à la fin de sélection, les opérations de bloc sont dans le menu Edit -> Rectangle.

Pour indenter une ligne ou un bloc du code source faire <ctrl><tab>.

Pour compléter automatiquement un nom faire <tab>. Pour afficher les propositions faire <cmd><espace>. Une boîte de sélection apparaît, cliquer sur le choix désiré. Si aucun ne convient taper <ESC>. La fenêtre de choix apparaît également automatiquement après un point :



Pour compléter une liste de paramètres d'une procédure ou une fonction taper parenthèse, la liste des paramètres apparaît.



S'il y a plusieurs paramètres, taper virgule entre chaque paramètre.

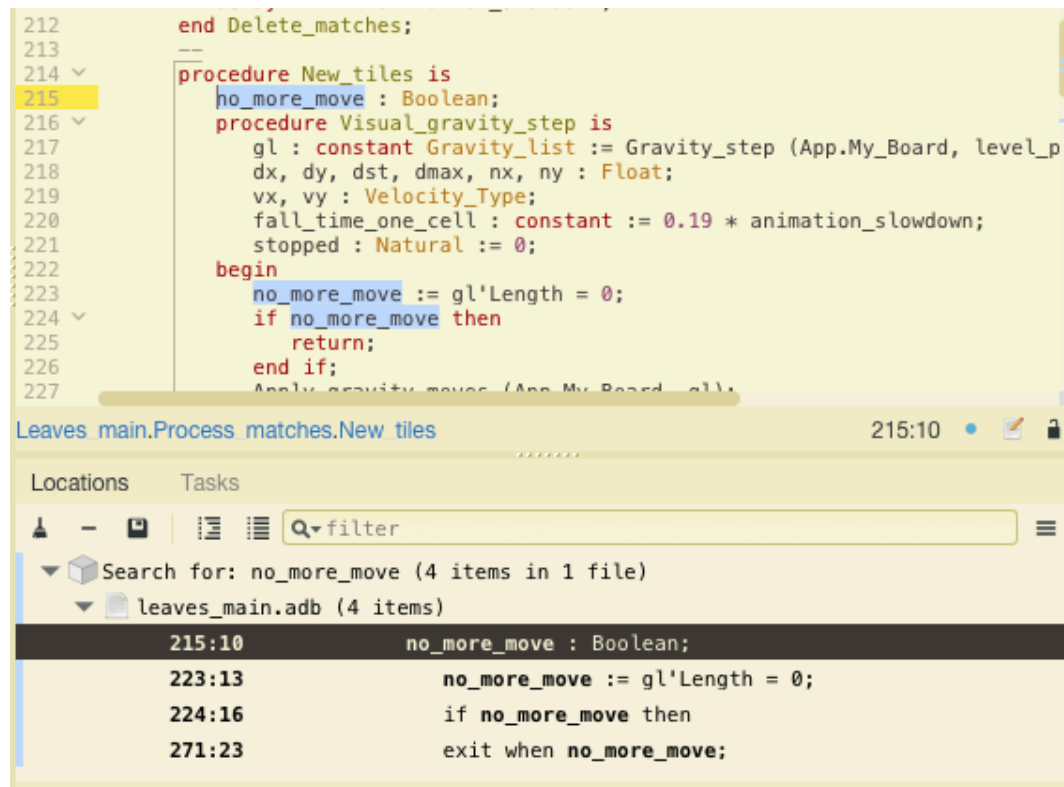
6. Recherche dans un code source

a) Un mot

Double-cliquer sur le mot à rechercher et cliquer sur le menu Navigate -> Find ou plus rapidement sur <cmd>F. Le panneau de recherche s'affiche. Cliquer sur Next ou <cmd>N, chaque mot trouvé est surligné. Cliquer sur Previous ou <cmd>P pour revenir sur le précédent mot trouvé.

Plusieurs options sont à découvrir comme la recherche dans plusieurs fichiers en sélectionnant Where ou In.

Cliquer sur Find All affiche en couleur surligné l'ensemble des mots trouvés avec leur position dans la fenêtre Locations. Un clic-droit sur le menu contextuel "Clear Locations" supprime les surlignages.



```
212     end Delete_matches;
213     --
214     procedure New_tiles is
215     no_more_move : Boolean;
216     procedure Visual_gravity_step is
217     gl : constant Gravity_list := Gravity_step (App.My_Board, level_p
218     dx, dy, dst, dmax, nx, ny : Float;
219     vx, vy : Velocity_Type;
220     fall_time_one_cell : constant := 0.19 * animation_slowdown;
221     stopped : Natural := 0;
222     begin
223     no_more_move := gl.Length = 0;
224     if no_more_move then
225     return;
226     end if;
227     Apply_gravity_moves (App.My_Board, all);
```

Locations	Tasks
Search for: no_more_move (4 items in 1 file)	
leaves_main.adb (4 items)	
215:10	no_more_move : Boolean;
223:13	no_more_move := gl.Length = 0;
224:16	if no_more_move then
271:23	exit when no_more_move;

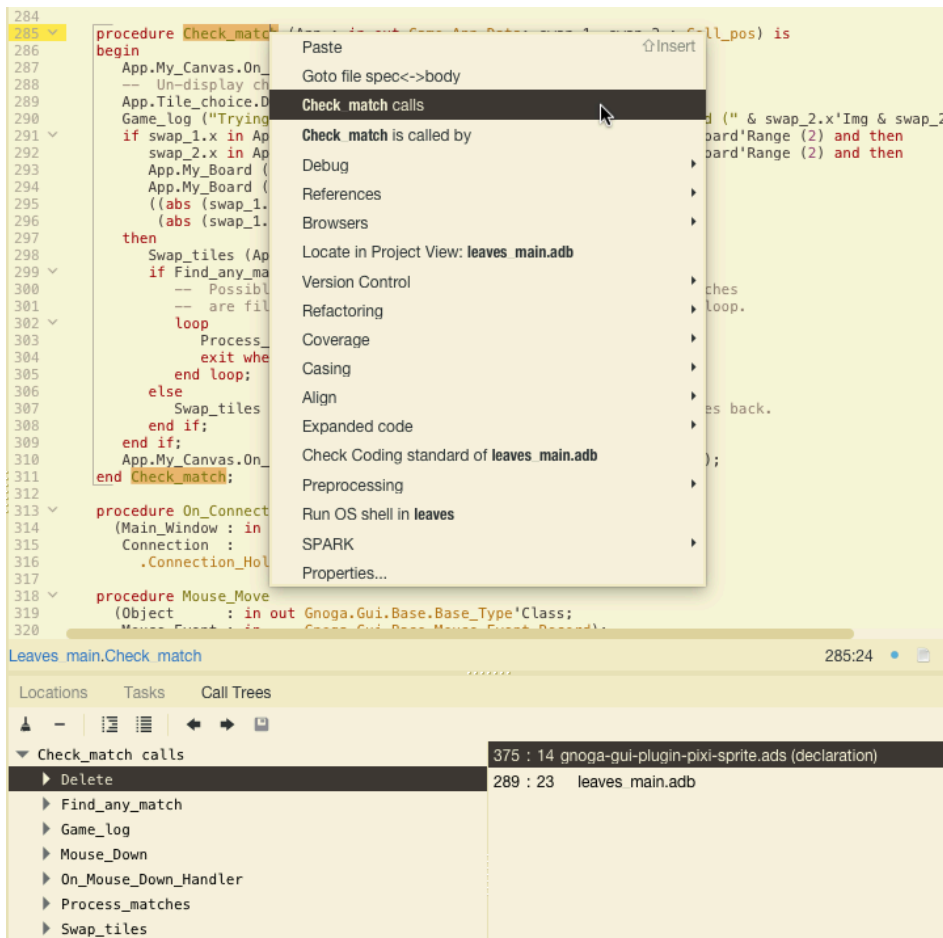
La combinaison de touche <maj><cmd>F fait apparaitre en plus le champ Replace avec les boutons d'actions correspondants.

b) Une déclaration, une définition

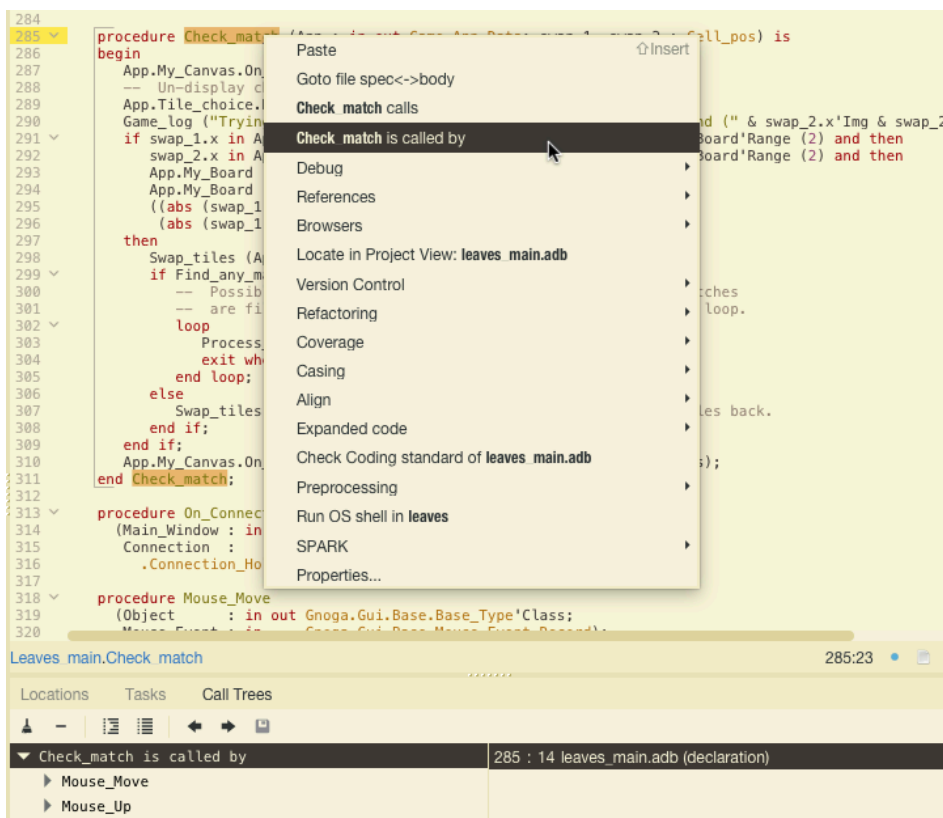
Positionner le curseur sur un identificateur du code source, le menu Navigate -> Goto Declaration affiche l'endroit où l'identificateur est déclaré et Navigate -> Goto Body affiche l'endroit où l'identificateur est défini (parfois il s'agit du même endroit). Le menu contextuel avec un clic-droit sur l'identificateur propose les mêmes fonctions.

c) Les appels

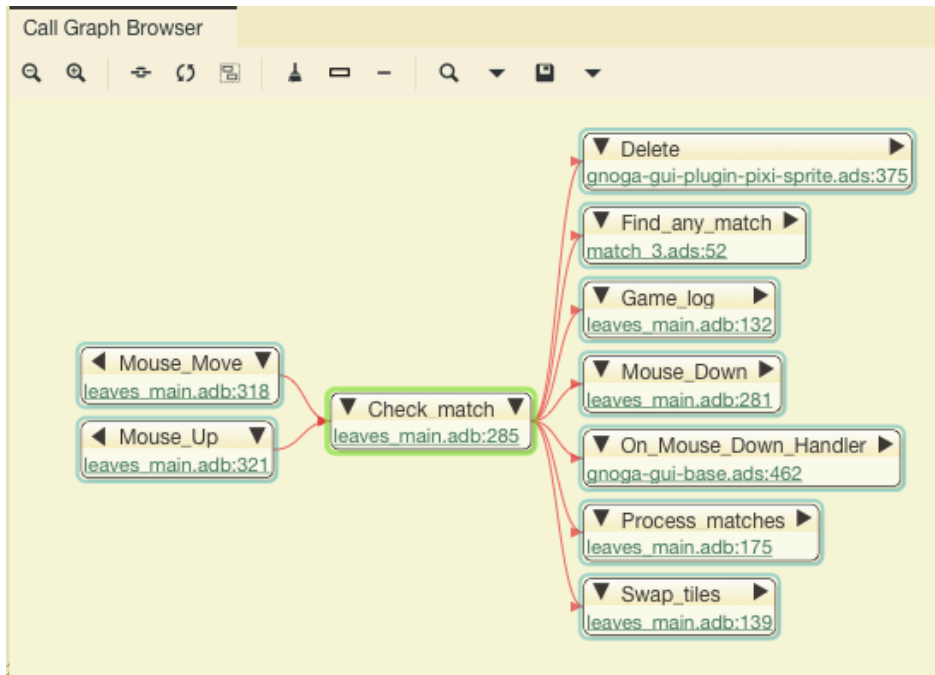
Positionner le curseur sur un identificateur du code source et afficher le menu contextuel avec clic-droit, le nom de l'identificateur s'affiche avec calls pour afficher la liste des procédures appelées :



et avec called by pour afficher la liste des procédures appelantes :



Ces deux informations sont également disponibles graphiquement en sélectionnant Browsers -> Calls ou Called by dans le menu contextuel :



d) Une ligne

Cliquer sur le menu Navigate -> Goto Line ou <cmd>G pour afficher ligne spécifique.

Le mois prochain nous verrons quelques astuces de correction d'erreur dans un source Ada avec le debugger intégré.

Pascal Pignard, novembre-décembre 2008, février 2009, juin 2019.