

Compiler un programme C graphique sous Windows avec Code::Blocks et GLFW

Télécharger les pré-requis

Un petit projet vous permettant de tester votre système a été concocté pour vous, et vous pouvez le télécharger à l'adresse <http://perso.uclouvain.be/vincent.legat/teaching/iCampusOpen/documents/meca1120/myFem-BasicGraphic-mingw.zip>

1. Créez un répertoire tel que `C:\Users\Grungi\Documents\Projects`. Ne faites pas plus long, et évitez à tout prix les caractères spéciaux et les espaces dans votre chemin. Cela peut vous sembler une bonne idée de créer un répertoire du type

`C:\Users\Grungi\MesEtudes\Université catholique de Louvain\Master\Master 1\Mécanique\Eléments` mais croyez notre longue expérience : c'est à éviter. Vous aurez sûrement à retaper ce chemin à la main, ou à naviguer dans les répertoires. Bref, do not do that...

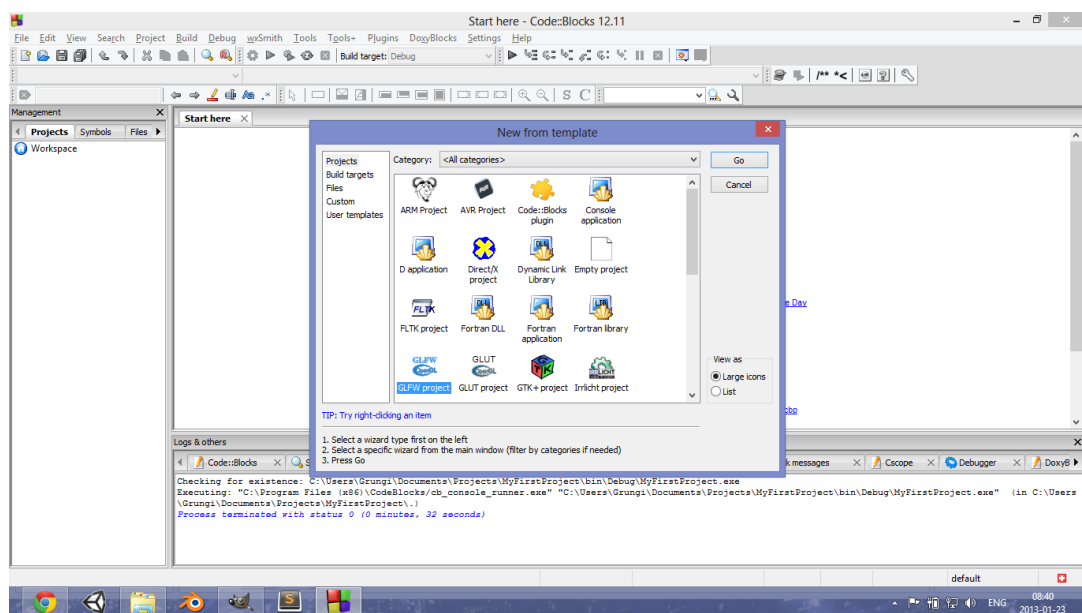
2. Placez le fichier `BasicGraphic.zip` dans votre nouveau dossier.
3. Décompressez le fichier. Pour ce faire, cliquez avec le bouton droit et choisissez "Extraire les fichiers". L'explorateur Windows peut se promener dans un fichier zip sans le décompresser, ne vous laissez pas méprendre.

Attention : Vous ne devez pas installer GLFW séparément, les fichiers vous sont fournis dans le fichier zip.

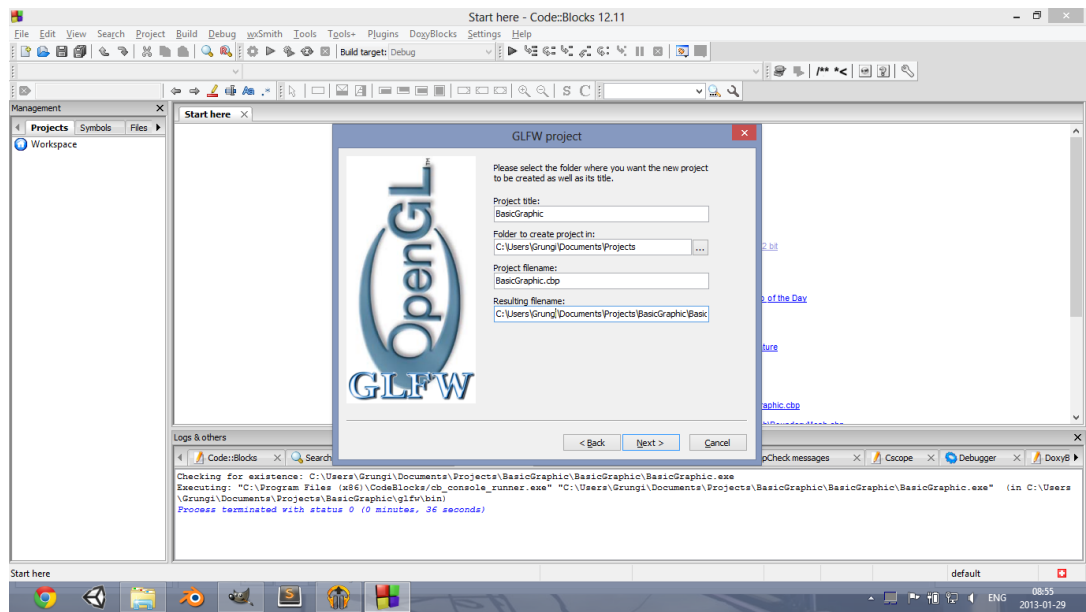
Créer un nouveau projet GLFW

Nous allons commencer par créer un nouveau projet dans Code::Blocks.

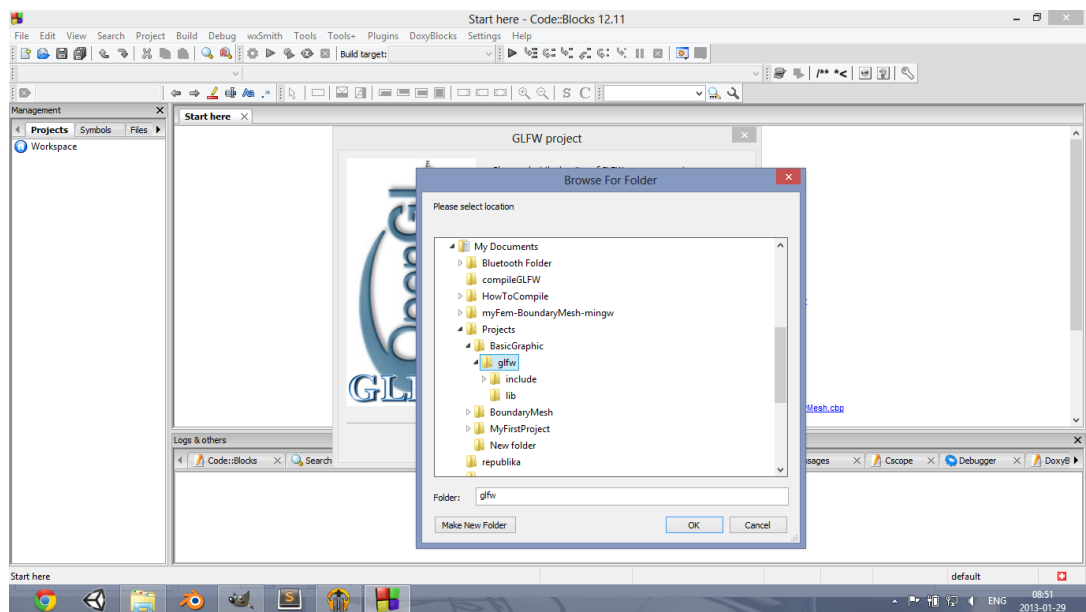
1. Cliquez sur `File` → `New` → `Project...`, et sélectionnez GLFW comme type de projet.



2. Donnez un nom au projet : pour simplifier, nommez le projet `BasicGraphic`, et choisissez le dossier où nous avons placé l'archive zip à la section précédente (donc, `C:\Users\Grungi\Documents\Projects` chez moi, à adapter pour vous).



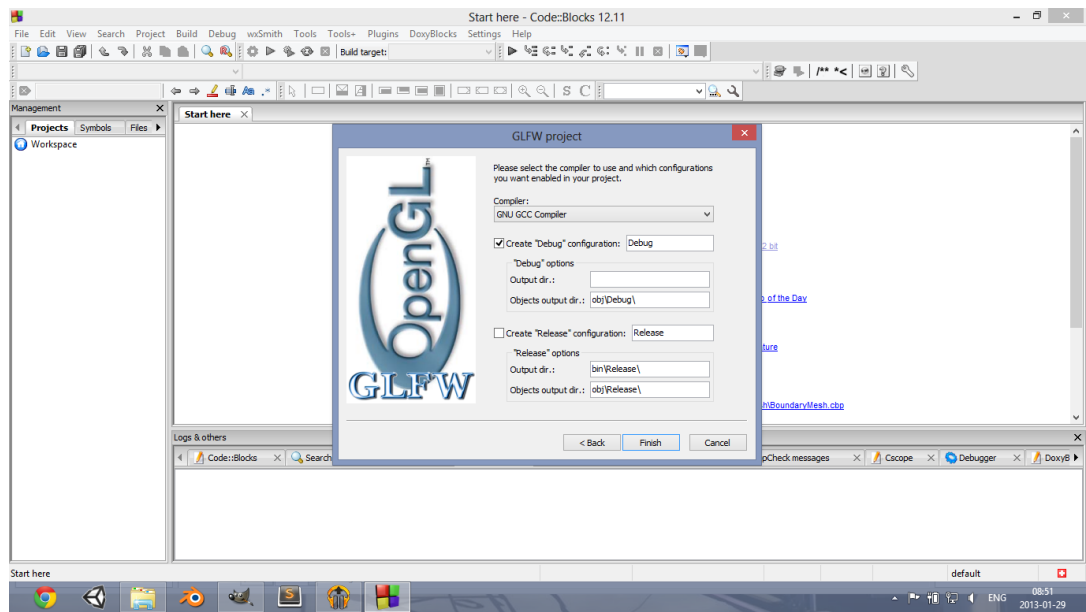
- Indiquez à Code::Blocks où il doit trouver la librairie GLFW. Indiquez simplement le chemin vers le sous-répertoire `glfw` à l'intérieur du projet. Si vous avez suivi les instructions il se trouve à `C:\Users\Grungi\Documents\Projects\BasicGraphics\glfw`.



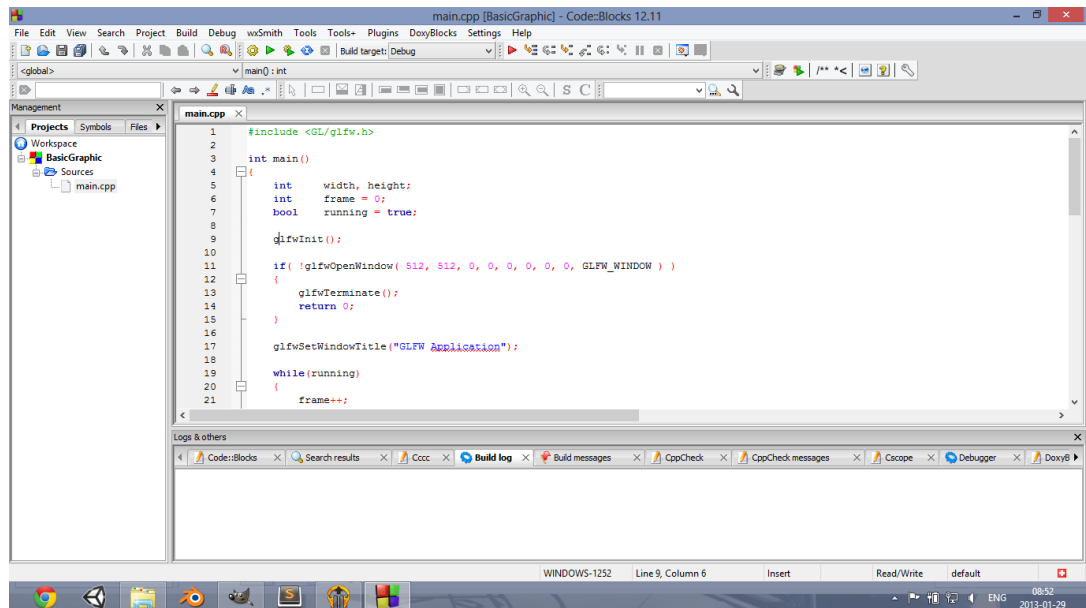
- Indiquez les autres options du projet comme sur la figure ci-dessous.

- Décochez la case "release"¹.
- Supprimez le contenu du champ "Output dir"

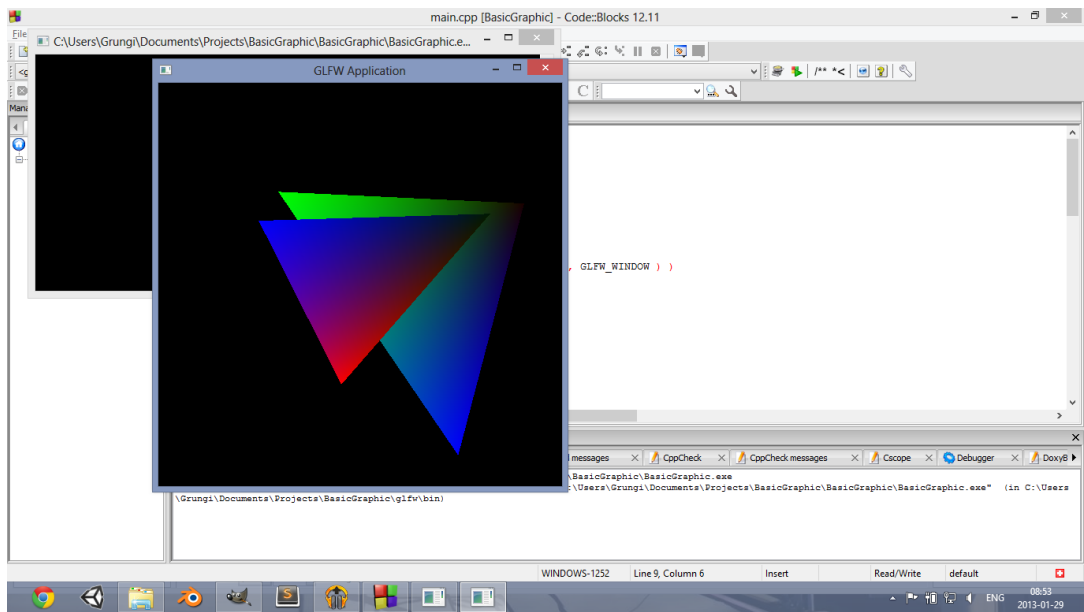
¹ D'habitude, on fait la différence entre "debug", qui représente une version non-optimisée, et donc plus facile à décortiquer par les outils d'analyse du code, et "release", qui représente une version plus optimisée, et donc moins aisée à déboguer.



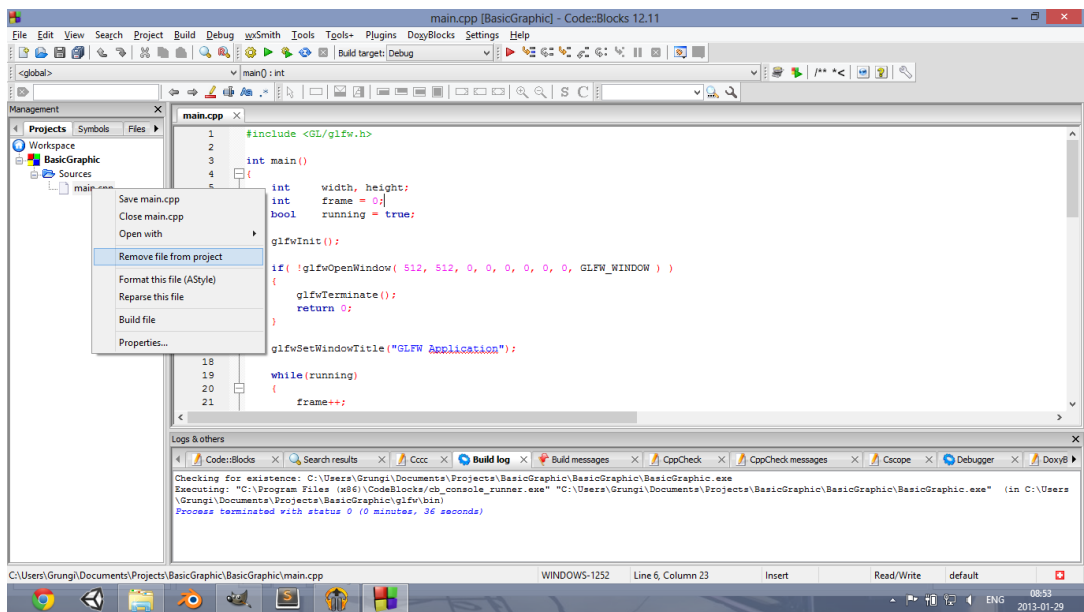
5. Compilez le projet de test (main.cpp) de Code::Blocks avec Build and Run (F9).



Si tout va bien, vous devriez voir des triangles tournant sur votre écran.



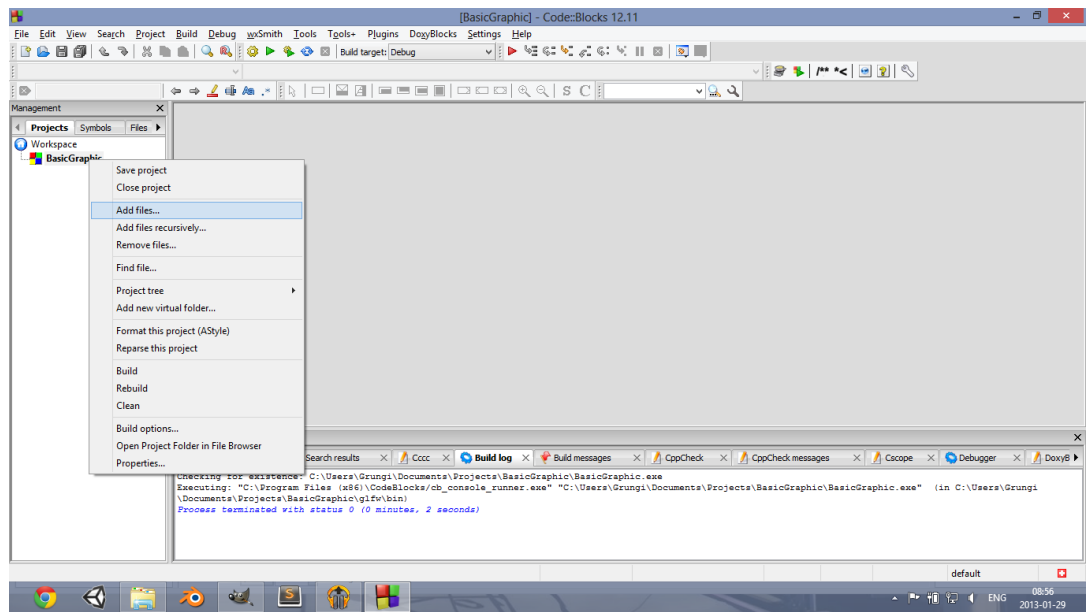
6. Fermez la fenêtre avec les triangles, et appuyez sur une touche de votre clavier pour fermer la fenêtre de terminal encore ouverte.
7. Supprimez `main.cpp` en cliquant avec le bouton droit de votre souris sur son nom dans la colonne de gauche, et en sélectionnant `Remove file from project`.



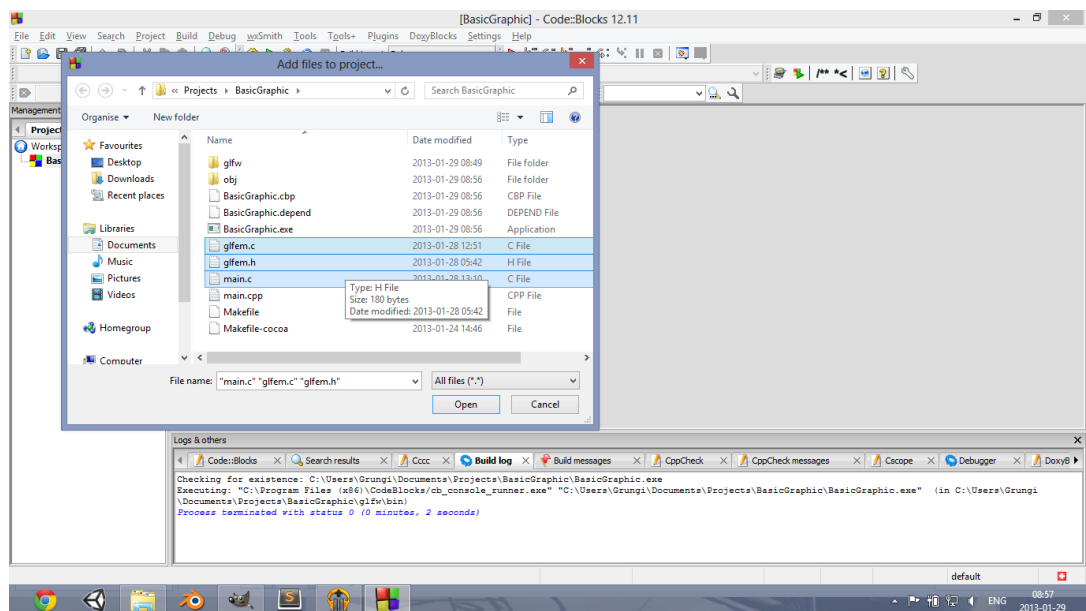
Voilà, il va enfin être temps de faire un peu de travail sérieux (ou en tous cas qui a un rapport avec le cours !)

Compiler le programme de test du cours

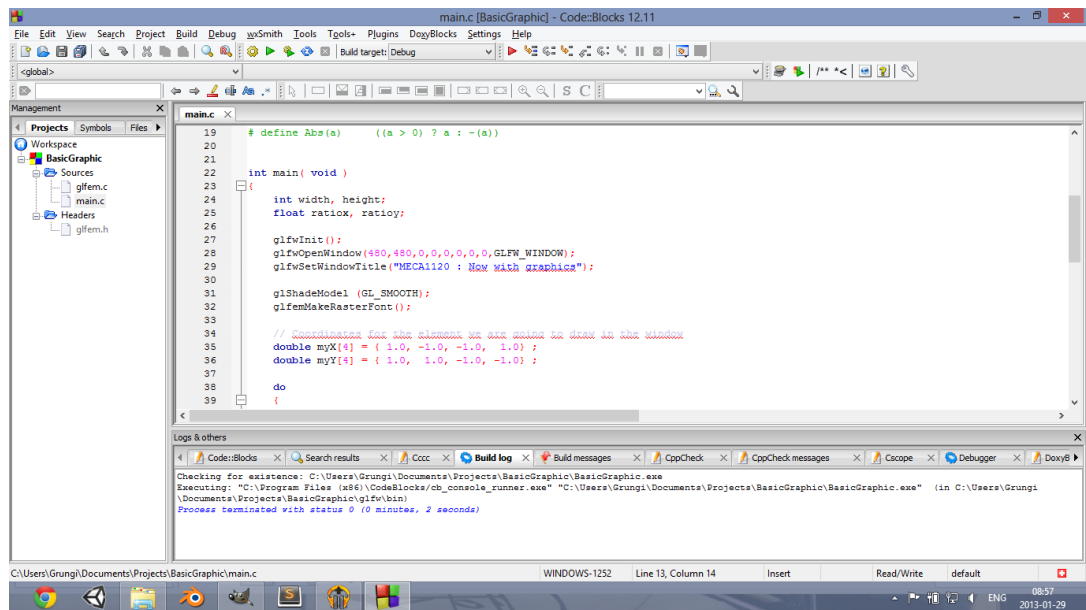
1. Cliquez avec le bouton droit sur le nom du projet à gauche de l'écran, et sélectionnez `Add files...`



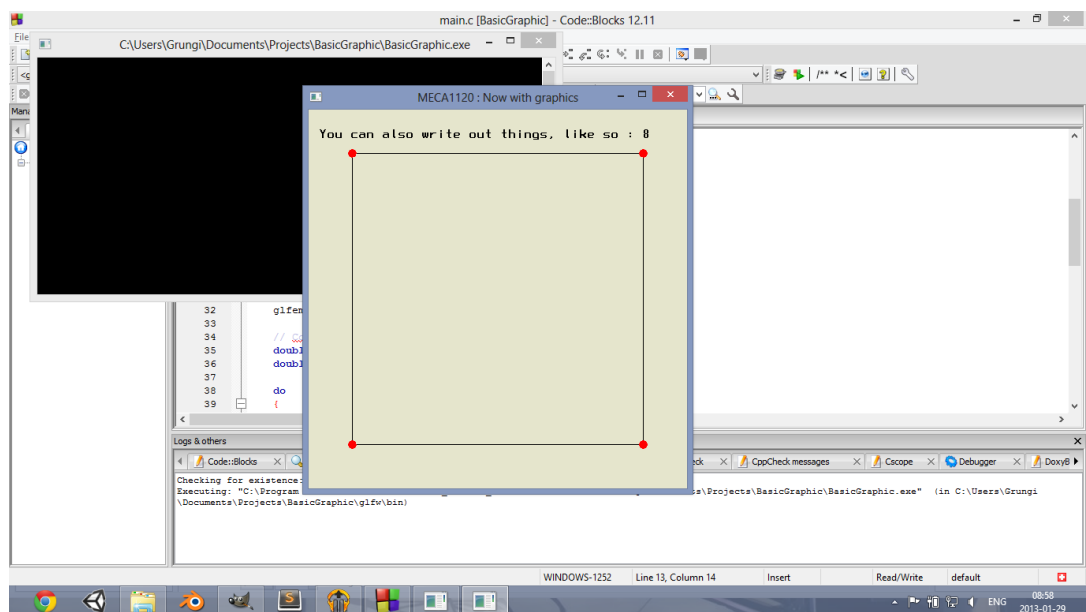
2. Sélectionnez trois fichiers : `glfem.c`, `glfem.h` et `main.c` (attention, pas `main.cpp` !)



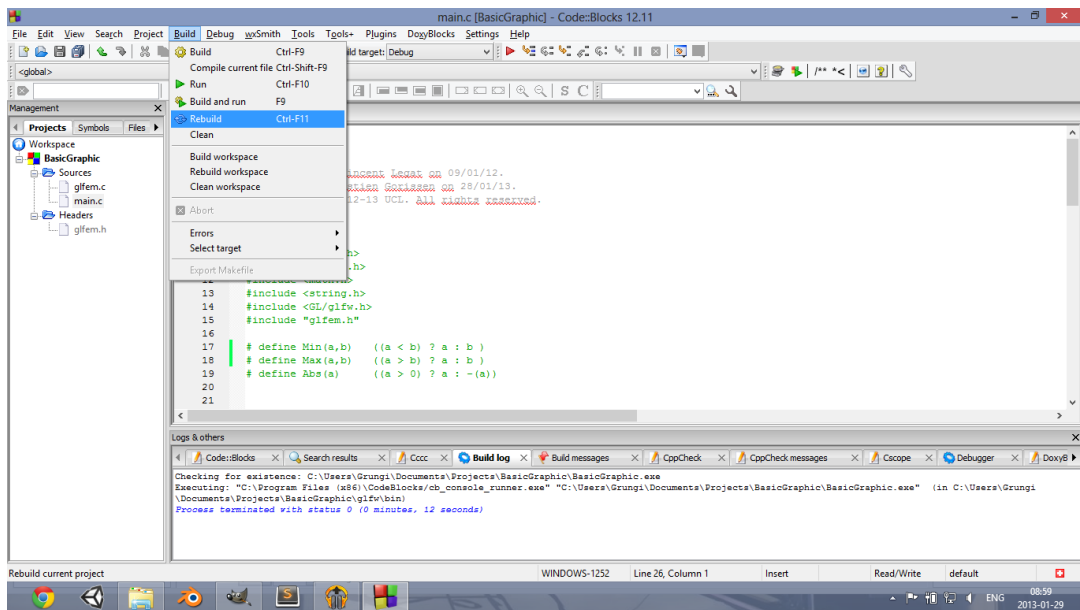
3. Vérifiez que les fichiers ont bien été ajoutés dans l'arborescence du projet.



4. Recompilez le projet et lancez-le via **F9**, pour voir apparaître un somptueux maillage composé d'un unique carré. C'est peu, mais c'est un début !



5. Si vous retombez sur vos anciens amis les triangles qui tournent, il se peut que Code::Blocks n'ait pas daigné recompiler la version actuelle de vos sources. Faites simplement **Build** → **Rebuild** (ou bien **Ctrl+F11**) et ensuite **F9** pour régler ce soucis.

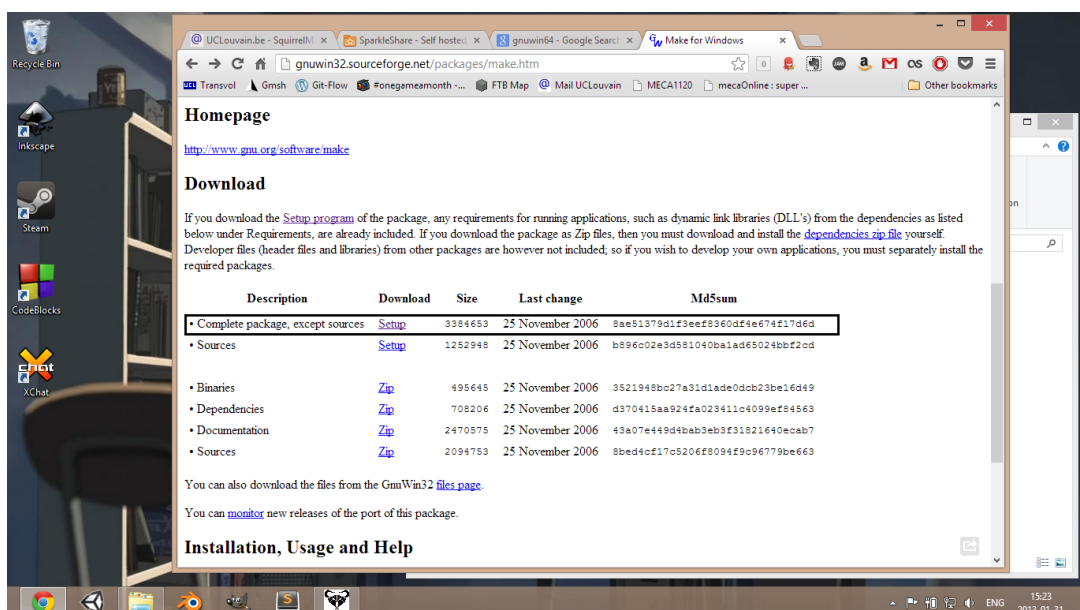


Voilà, vous avez maintenant un environnement de développement fonctionnel. Il ne vous reste plus qu'à l'utiliser pour mener vos projets à bon port. Prenez le temps de parcourir le code du projet exemple pour avoir une idée des fonctions à votre disposition.

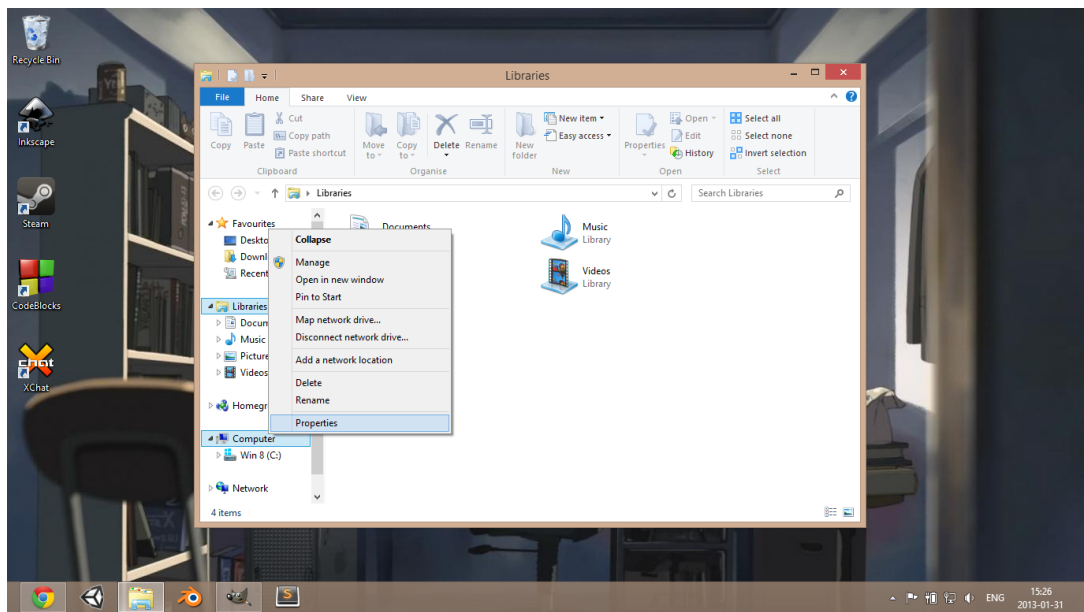
Pour ceux qui préfèrent compiler le projet avec un Makefile sous Windows

Si vous préférez utiliser un Makefile à la place de Code::Blocks pour compiler vos programmes, c'est également possible.

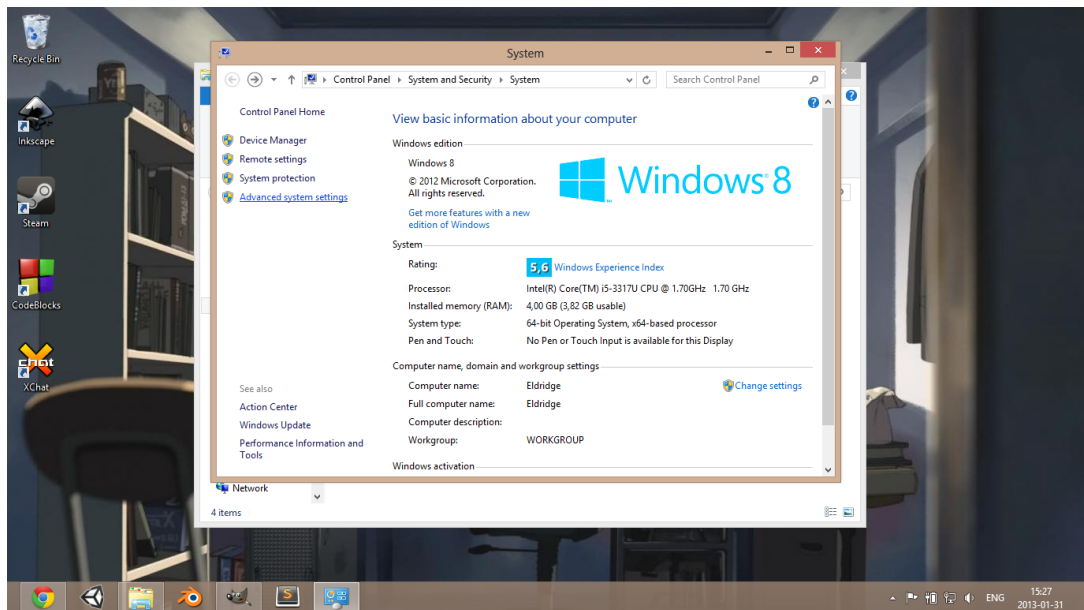
1. Téléchargez make pour windows sur <http://gnuwin32.sourceforge.net/packages/make.htm>, prenez le "Complete package, except sources".



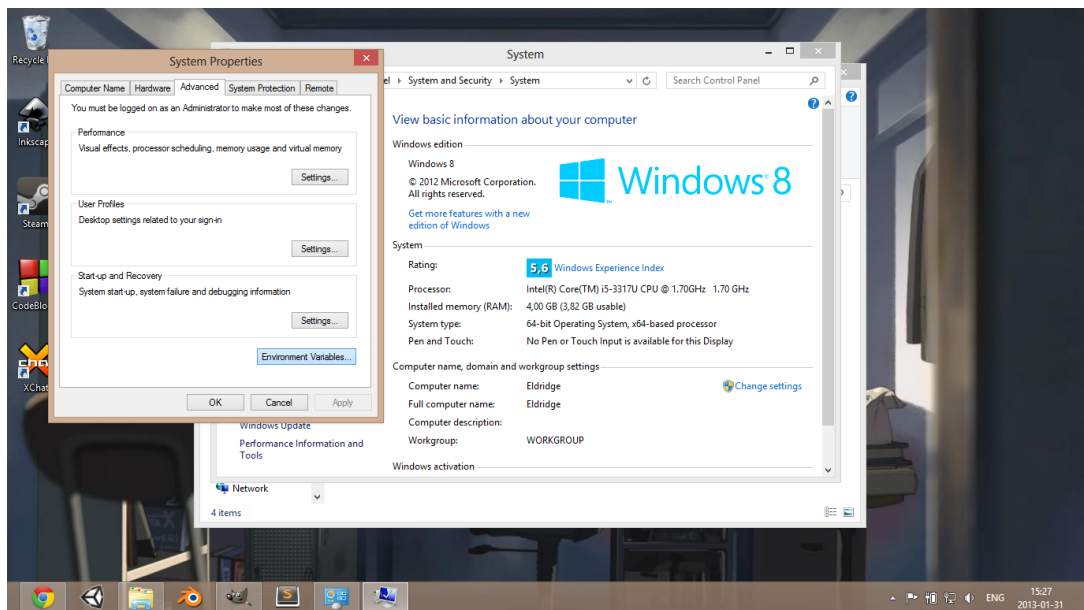
2. Installez make : Next, Agree, Next, Next, Next... Vous connaissez la musique.
3. Ouvrez l'explorateur Windows, et cliquez avec le bouton droit sur Computer. Sélectionnez "Propriétés" (Properties).



4. Sélectionnez "Paramètres avancés" (Advanced system settings) dans la nouvelle fenêtre.

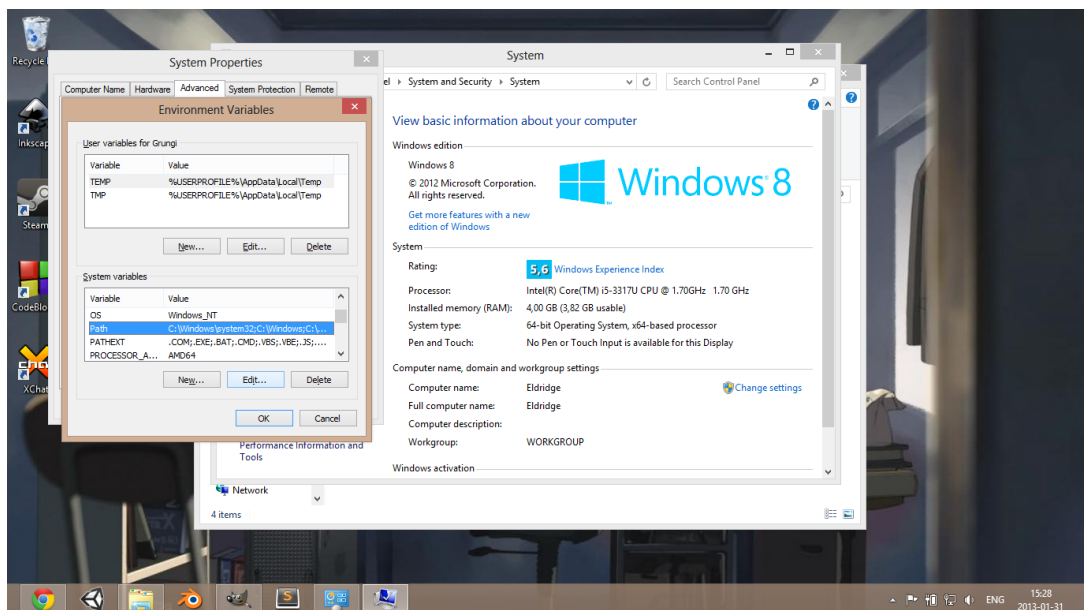


5. Cliquez sur "Variables d'environnement" (Environment variables).

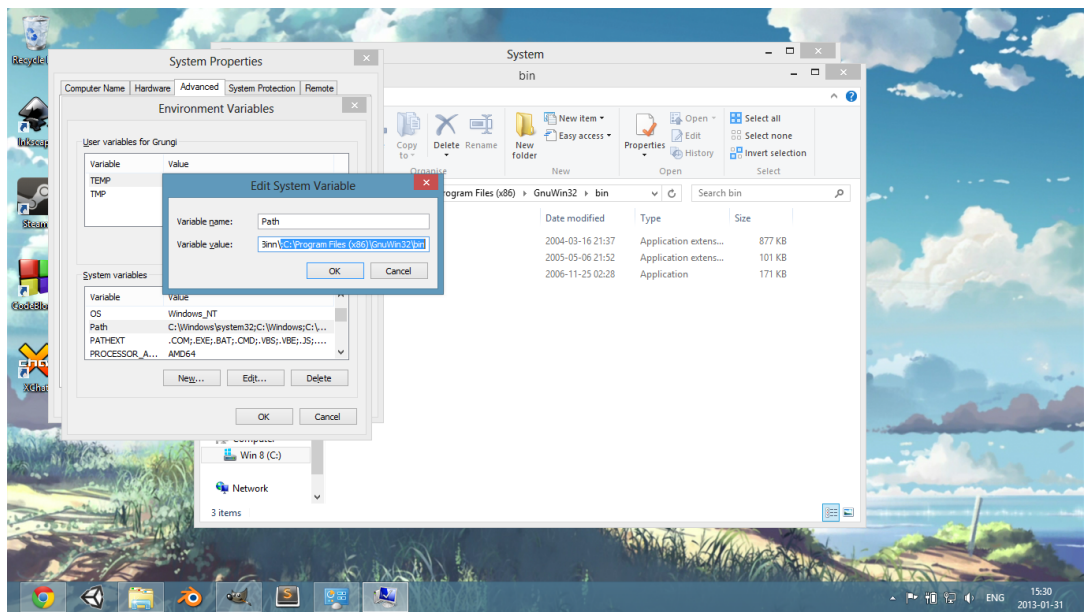


6. Trouvez la variable "Path" dans la liste et cliquez sur éditer.

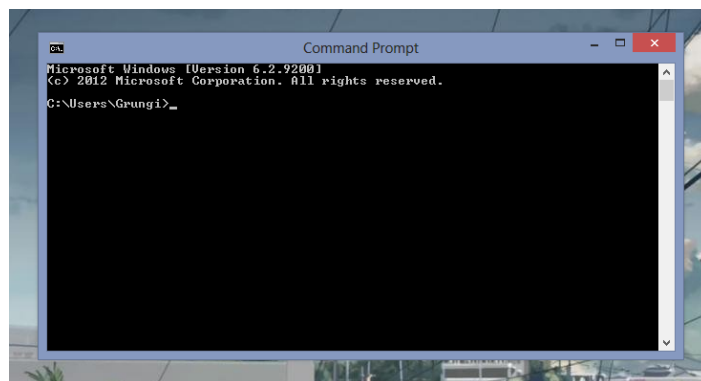
ATTENTION ! Rajouter une valeur dans cette variable ne pose aucun soucis, mais surtout ne supprimez rien de ce qui s'y trouve déjà, sous peine de rendre votre système inutilisable.



7. Rajoutez, à la fin de la liste de valeurs de la fonction : ;C:\Program Files (x86)\GnuWin32\bin (n'oubliez pas le ";" au début !)

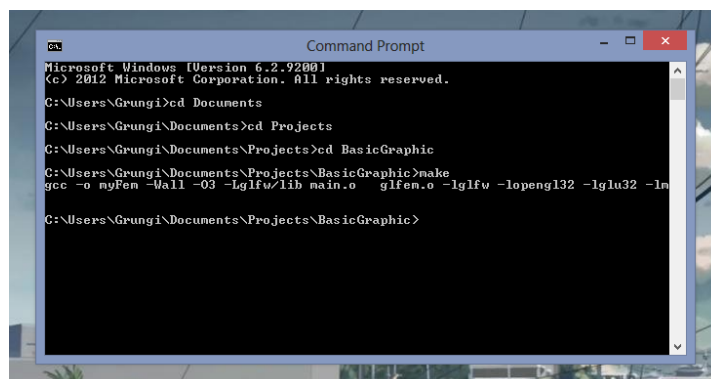


8. De la même façon, rajoutez encore ;C:\Program Files (x86)\CodeBlocks\MinGW\bin. Modifiez ces chemins si vous n'avez pas installé Code::Blocks et/ou make à l'endroit par défaut.
9. Cliquez sur Ok quelques fois pour quitter toutes ces fenêtres.
10. Ouvrez une invite de commande (Command prompt).

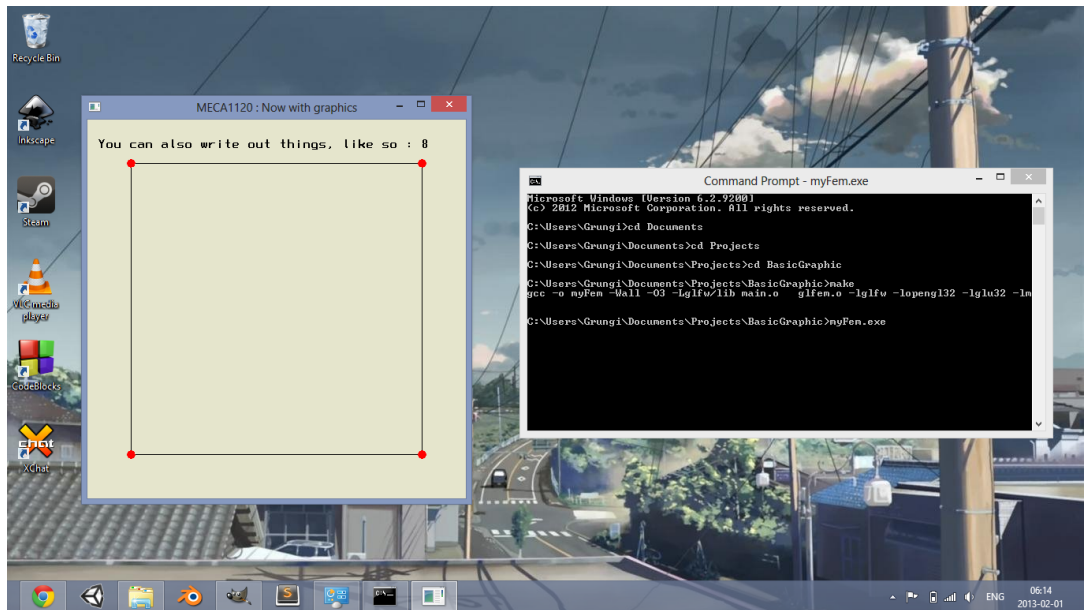


11. Entrez les commandes suivantes pour retrouver le dossier du programme et le compilez (adaptez si besoin) :

```
cd Documents
cd Projects
cd BasicGraphic
make
```



12. Si tout s'est bien déroulé, vous pouvez maintenant lancer le programme en tapant `myFem.exe`.



Et voilà :)