

# Installation de Code::Blocks sous Ubuntu

Bonjour à tous.

Le but de ce tutoriel est de vous indiquer la marche à suivre pour installer Code::Blocks sur un ordinateur équipé d'une distribution linux Ubuntu.

Afin d'être le plus complet possible, nous allons démarrer avec une installation de Ubuntu entièrement vierge. Par « vierge », je ne veux pas dire « vide », mais je veux tout simplement parler d'une installation toute fraîche.

Je démarre donc avec la version 6.10 d'Ubuntu (32 bits) mise à jour avec les correctifs de sécurité proposés par les dépôts officiels. Si vous disposez d'une autre version (plus ou moins récente, ou même 64 bits), ne vous inquiétez pas, la procédure est la même.

Ce qu'il vous faudra :

- une connection internet active afin de pouvoir télécharger Code::Blocks et les paquets dont il aura besoin pour fonctionner
- un peu de patience, car je ne sais pas si vous l'avez remarqué, mais il est rare que tout marche du premier coup lorsque l'on débute sous linux (les actions avec la ligne de commande sont fréquentes, et les fautes de frappe sont courantes)

Comme je vous l'ai dit précédemment, je commence avec une installation totalement vierge d'Ubuntu. De votre côté, vous avez sûrement déjà essayé votre distribution, et installé certains logiciels. Il se peut donc que vous n'obteniez pas tous les messages d'erreurs listés dans ce tutoriel. Si c'est le cas, ne paniquez pas, et recherchez plus loin si le message d'erreur que vous obtenez est indiqué.

Si vous rencontrez de trop grosses difficultés lors de l'installation, n'hésitez pas à me contacter, je n'ai jamais refusé d'aider quelqu'un si je me sentais capable de la faire.

# Télécharger le paquet d'installation

Si vous avez un peu fouillé dans votre distribution, notamment dans le **Gestionnaire de Paquets Synaptic** qui est l'équivalent de **Ajout/Suppression de programmes** de Windows, vous vous êtes sûrement rendus compte que Code::Blocks n'est proposé nulle-part dans les paquets officiels.

Qu'à cela ne tienne : nous allons le télécharger, et de ce fait, nous aurons la toute dernière version (Code::Blocks étant continuellement en développement, une version "Nightly Build" est proposée quasiment tous les soirs).

Suivant votre version d'Ubuntu (32 ou 64 bits), le téléchargement ne s'effectue pas au même endroit.

- Si vous possédez une version 32 bits d'Ubuntu, le téléchargement se fait directement depuis le site officiel : [www.codeblocks.org](http://www.codeblocks.org). Rendez-vous sur le [forum](#), dans la section **Nightly Builds**. Vous y trouverez l'annonce correspondant à la dernière version disponible, ainsi que le lien vers le paquet ".deb" correspondant.
- Si vous possédez une version 64 bits d'Ubuntu (processeurs Amd uniquement), il n'existe, à l'heure ou j'écris ce tutoriel, aucun paquet officiel pour vous. N'ouvrez pas le gaz, ou ne vous faites pas sauter la cervelle pour autant, je vous ai concocté un paquet que vous pourrez trouver [ici](#).
- Si vous possédez une version Intel - 64 bits, soyez encore un peu patients. Je vous promet un petit tutoriel sur la marche à suivre pour compiler Code::Blocks directement depuis les sources.

Enregistrez le fichier téléchargé dans votre répertoire utilisateur (/home/votre\_nom\_d\_utilisateur)

**Heu, moi, il me l'a mis sur le bureau : comment je fais ?** 🤔

Pas de panique. Il est vrai que par défaut sous Ubuntu, le gestionnaire de téléchargements de Firefox est configuré de façon à enregistrer les fichiers automatiquement sur le bureau. Rassurez-vous, comme sous Windows, vous bénéficiez sous Linux d'un gestionnaire de fichier aussi performant que l'explorateur, et le Couper/Coller fait également partie de l'aventure. Pour le déplacer, faites un clic droit dessus, et sélectionnez "couper" dans le menu contextuel, puis ouvrez le gestionnaire de fichier directement dans votre répertoire utilisateur (menu "Raccourcis", "Dossier Personnel"), faites un clic droit dans la liste des fichiers à droite, et sélectionnez "Coller". Impressionnant, non ? 😎

Vous voyez bien que c'est du tout cuit Linux...

---

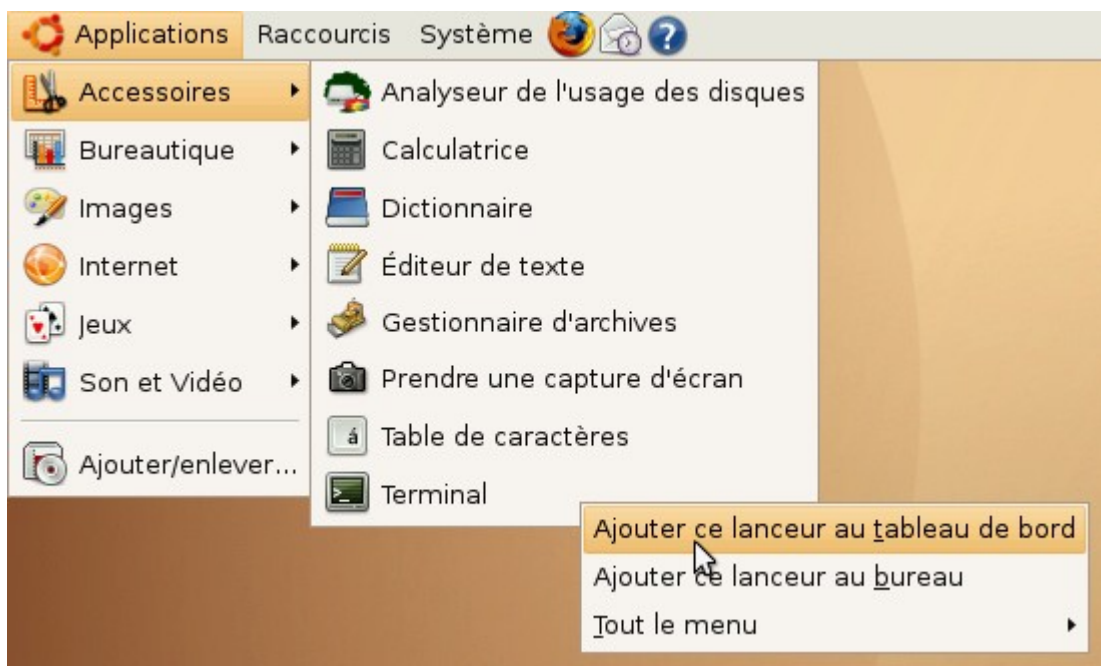
## On démarre l'installation

Nous allons maintenant passer à l'installation du paquet téléchargé précédemment. Et devinez comment cela va se faire : En ligne de commande !

**Quoi ? Et pourquoi pas ré-installer Ms-Dos, pendant qu'on y est ?** 🤔

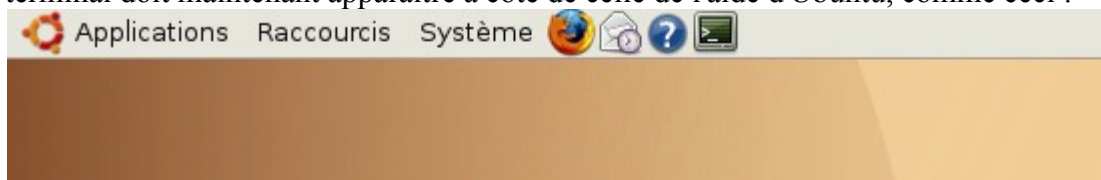
Ne paniquez pas! Vous verrez, la ligne de commande est un outil très puissant, et plus rapide que le mode graphique pour effectuer certaines tâches. Il suffit de connaître les commandes à entrer. 😊

Avant toute chose, comme il est fréquent sous linux d'utiliser la ligne de commande, je vous conseille d'en placer un raccourci sur le tableau de bord (la « barre de raccourcis » en haut de l'écran). Pour cela, allez dans le menu «**Applications**», «**Accessoires**», et repérez le raccourci «**Terminal**» (pour moi, il est au fond de ce menu). Faites un clic droit sur ce raccourci, et sélectionnez «**Ajouter ce lanceur au tableau de bord**».



*Capture d'écran N°1 : L'ajout du lanceur au tableau de bord*

L'icône du terminal doit maintenant apparaître à côté de celle de l'aide d'Ubuntu, comme ceci :



*Capture d'écran N°2 : Le lanceur en place sur le tableau de bord*

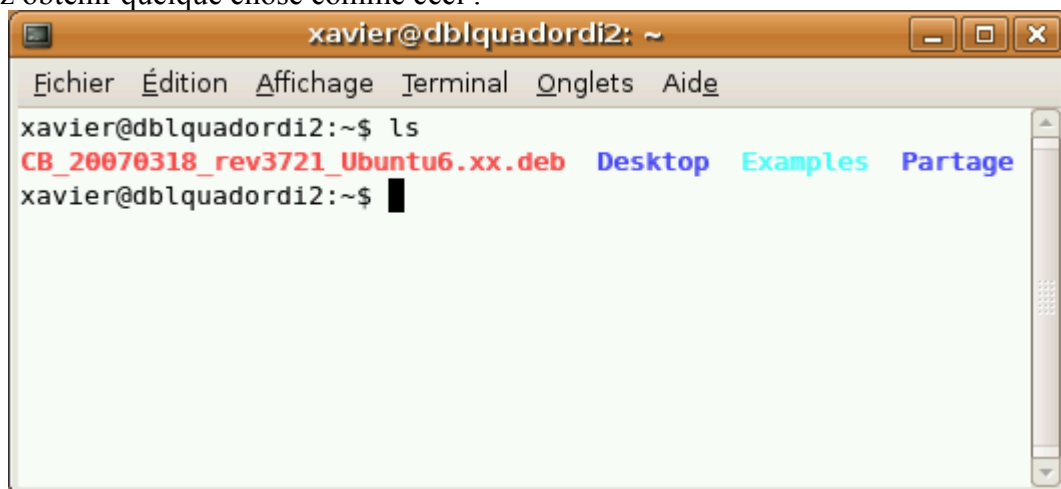
Maintenant, cliquez sur cet icône, pour lancer la fenêtre "Terminal", afin de pouvoir lancer l'installation de notre paquet.

Je ne vous ferais pas un cours sur l'utilisation du shell, car [Jarvis](#) l'a déjà fait. Jetez-y un coup d'oeil pour connaître les principales fonctions du shell, ainsi que l'organisation du système de fichiers.

Pour la suite de ce tuto, je pars du principe que vous connaissez le minimum vital expliqué dans le tutoriel cité ci-dessus.

Comme nous avons enregistré notre paquet, téléchargé précédemment, dans notre [répertoire utilisateur](#), et que la fenêtre terminal vous place, en démarrant, dans ce même répertoire, nous n'avons plus qu'à vérifier que notre paquet s'y trouve bien. Pour cela, un petit `ls` fera l'affaire.

Vous devriez obtenir quelque chose comme ceci :



*Capture d'écran N°3 : La fenêtre "Terminal"*

La commande pour installer un paquet manuellement s'appelle "**dpkg**". Et comme son utilisation nécessite les droits administrateurs, nous la ferons précéder de la commande "**sudo**".

Il faut ensuite lui donner le paramètre "-i" (pour Installer), et lui indiquer le nom du fichier à installer. Étant donné :

- La complexité du nom du fichier (CB\_20070318\_rev3721\_Ubuntu6.xx.deb) 🤔
- Le fait que vous n'avez sûrement pas le même fichier (le nom change en fonction du numéro de révision et de la date) 😞
- La flemme que j'ai à taper des trucs aussi compliqués 🙄

Je vais vous donner une petite astuce : la touche [TAB].

Entrez donc la commande : "**sudo dpkg -i CB\_**" et pressez [TAB]. Comme il n'existe qu'un seul fichier

commençant par "CB\_", le terminal l'a complété pour nous. Magique, non ? 🧙

**Ouais, c'est pas nouveau, ton truc, même la ligne de commande sous Windows est capable de faire ça !**

Peut-être, mais celle de linux est beaucoup plus puissante. Essayez, par exemple, de ne taper que "su", et de pressez [TAB]. Rien ne se passe. Re-pressez [TAB] une deuxième fois, et miracle, on vous liste toutes les commandes connues sur votre système commençant par "su". Pratique, pour retrouver une commande.

Bref, ne nous éloignons pas du sujet qui est, je vous le rappelle, l'installation de Code::Blocks.

Nous en étions donc à entrer la commande :

**Code : console**

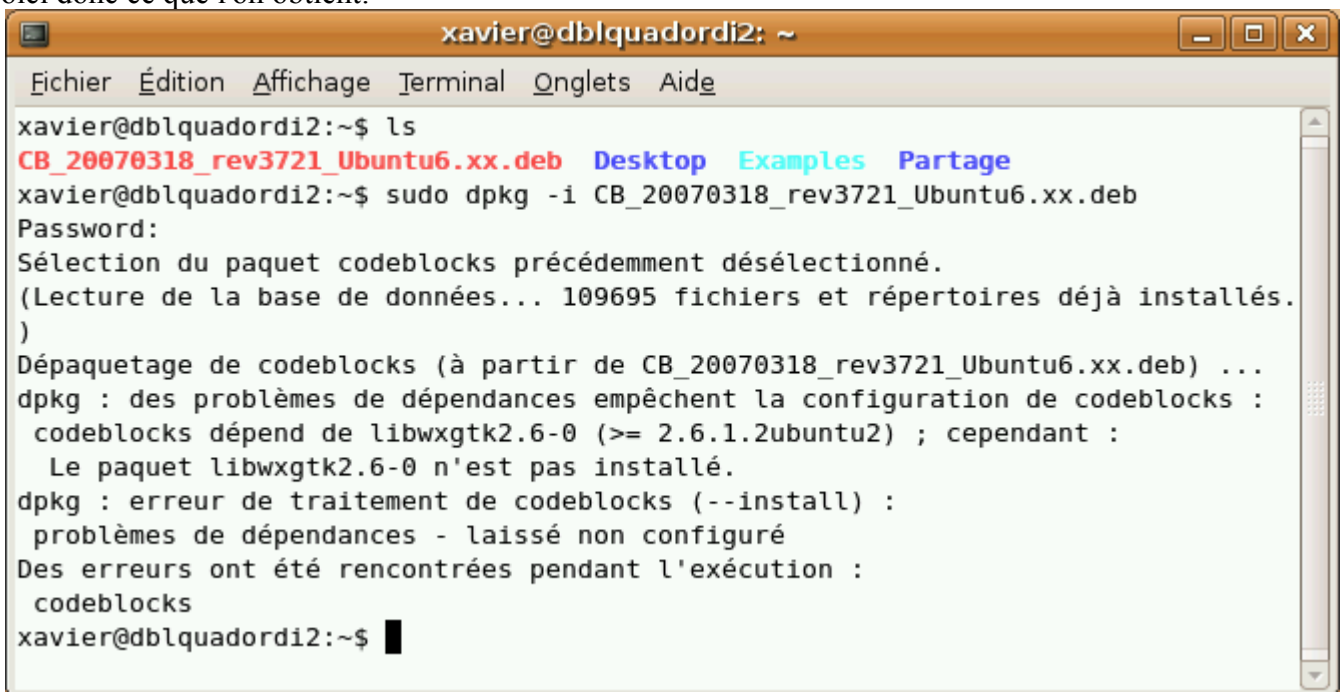
```
sudo dpkg -i le_nom_du_paquet_téléchargé"
```

Pressez [ENTREE], on vous demande votre mot de passe (à cause de la commande **sudo**) et l'installation commence.....

Attention, roulements de tambour, **Erreur !** 😞

Ça aurait été trop beau si ça avait marché du premier coup (et puis, il faut dire aussi que ce tuto aurait été un peu inutile).

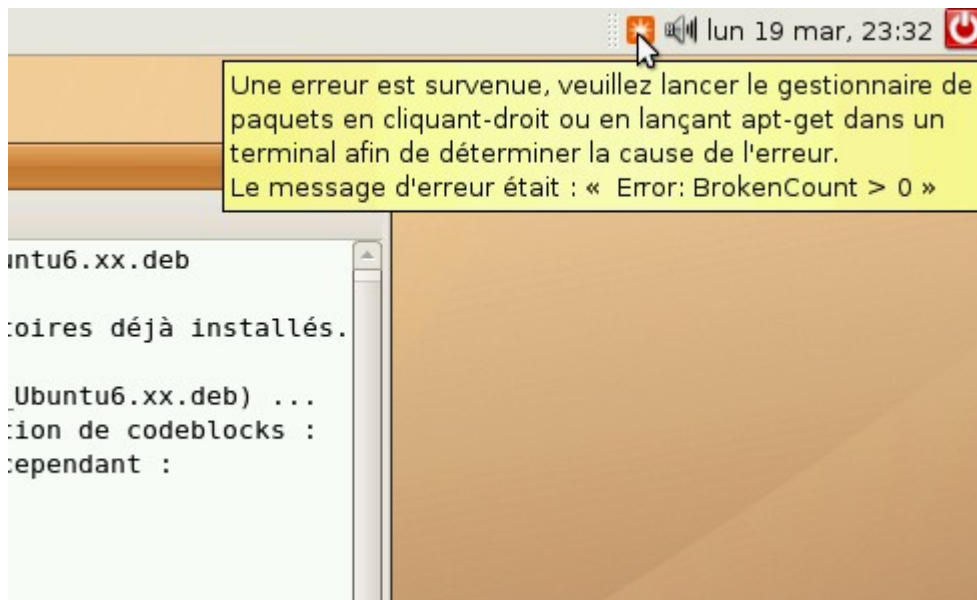
Voici donc ce que l'on obtient:



```
xavier@dblquadordi2: ~  
Fichier Édition Affichage Terminal Onglets Aide  
xavier@dblquadordi2:~$ ls  
CB_20070318_rev3721_Ubuntu6.xx.deb Desktop Examples Partage  
xavier@dblquadordi2:~$ sudo dpkg -i CB_20070318_rev3721_Ubuntu6.xx.deb  
Password:  
Sélection du paquet codeblocks précédemment désélectionné.  
(Lecture de la base de données... 109695 fichiers et répertoires déjà installés.)  
Dépaquetage de codeblocks (à partir de CB_20070318_rev3721_Ubuntu6.xx.deb) ...  
dpkg : des problèmes de dépendances empêchent la configuration de codeblocks :  
codeblocks dépend de libwxgtk2.6-0 (>= 2.6.1.2ubuntu2) ; cependant :  
Le paquet libwxgtk2.6-0 n'est pas installé.  
dpkg : erreur de traitement de codeblocks (--install) :  
problèmes de dépendances - laissé non configuré  
Des erreurs ont été rencontrées pendant l'exécution :  
codeblocks  
xavier@dblquadordi2:~$ █
```

*Capture d'écran N°4 : Notre première erreur*

De plus, au bout de quelques secondes, l'icône des mises à jour s'est allumé dans la zone système (à côté de l'horloge). Si vous laissez votre curseur dessus, vous obtiendrez un message vous indiquant qu'il ne s'agit pas de la disponibilité de nouvelles mises à jour, mais bel et bien d'une erreur.



*Capture d'écran N°5 : L'icône de notification d'erreur lors d'une installation*

C'est tellement bien expliqué, qu'on ne peut s'empêcher de suivre les recommandations qui nous sont données. Allez-y donc franchement: Clic droit sur l'icône, puis sélectionnez "Lancer le gestionnaire de paquets".

Il s'agit de l'interface graphique permettant d'installer ou de désinstaller des paquets.

**Quoi ? On pouvait le faire graphiquement ? Et tu nous fais passer par le Terminal ?** 🤔

Calmez-vous ! Même si des interfaces graphiques comme celle-ci existent sous linux, la plupart des aides que vous pourrez trouver sur les forums sont basées sur la ligne de commande. Il est presque indispensable de savoir s'en servir.

Donc, Synaptic (le gestionnaire de paquets) démarre, et nous indique qu'il y a un paquet "cassé" sur notre système, et qu'il faut utiliser le filtre "cassés" pour le trouver.

On valide, on clique sur le bouton "Filtres" (en bas à gauche), et on sélectionne "Cassés" dans la liste de gauche. Notre paquet apparaît dans la liste de droite, avec l'icône rouge, indiquant un problème. Cliquez sur l'icône, et choisissez "Sélectionnez pour suppression". Appliquez les modifications (je vous laisse deviner sur quel bouton il faut cliquer), confirmez, et laissez travailler votre PC. Lorsqu'il a fini de réparer vos bêtises (comment ça, c'est moi qui vous les ai fait faire ?), vous constaterez que l'icône de notification (capture ci-dessus) a disparu. Il ne reste plus qu'à fermer le gestionnaire de paquets, et à regarder ce qui a ~~fait~~ n'a pas marché lors de l'installation.

Si l'on analyse le message que l'on a obtenu dans la fenêtre "Terminal" (Capture N°4 ci-dessus), on s'aperçoit que l'erreur vient du fait qu'il nous manque quelque-chose (en l'occurrence, un paquet nommé **libwxgtk2.6-0**).

Comme Code::Blocks dépend de ce paquet (on dit dans ce cas que ce paquet est une dépendance de Code::Blocks), il va donc falloir que l'on l'installe. C'est le sujet du chapitre suivant.

# Installation des dépendances

Nous allons donc installer les paquets dont Code::Blocks a besoin pour fonctionner.

En l'occurrence, d'après l'erreur que l'on a obtenue lorsque l'on a lancé l'installation, il nous manque le paquet **libwxgtk2.6-0**.

Nous allons continuer l'installation avec notre fenêtre terminal, et pour cela, nous allons utiliser la commande **apt-get**.

Comme la commande **dpkg** que nous avons utilisé précédemment, il s'agit d'une commande nécessitant les droits administrateurs. Il faudra donc la faire précéder de **sudo**.

Nous voulons que cette commande installe le paquet que nous allons lui indiquer, il faudra la faire suivre de la commande **install**.

En dernier, il faudra lui indiquer quel est le paquet à installer.

Voici donc la commande à entrer dans la fenêtre Terminal :

**Code : console**

```
sudo apt-get install libwxgtk2.6-0
```

Vous avez sûrement remarqué que le nom du paquet à installer correspond exactement à ce qui était indiqué dans le message d'erreur que nous avons eu précédemment.

Lancez donc cette commande, entrez le mot de passe administrateur si on vous le demande, et qu'obtenez-vous ? Encore un message d'erreur !!!!! 😞

On nous dit que le paquet **libwxgtk2.6-0** n'est pas disponible.

En fait, si vous avez déjà fait quelques manipulations avec le gestionnaire de paquets, il se peut que vous ayez un message vous indiquant que le paquet n'est pas disponible, mais qu'il existe dans la base de données.

Cela est dû au fait que ce paquet ne fait pas partie de l'installation "standard" d'Ubuntu. Il faut, pour en bénéficier, activer un certain type de dépôts (et du coup, de paquets). Dans notre cas, il s'agit des dépôts

**Universe** dont nous avons besoin.

Il existe plusieurs méthodes pour activer ou désactiver des dépôts:

- La méthode graphique, par le gestionnaire de paquets
- La méthode basique, avec un éditeur de texte

Pour la méthode graphique, il suffit de lancer le gestionnaire de paquets **Synaptic** (menu **Système / Administration / Gestionnaire de paquets Synaptic**). Il faut ensuite aller dans le menu **Catégories / Dépôts**, dans le premier onglet (Ubuntu 6.10), et activer le ou les dépôts souhaités, puis faire une mise à jour de la base de données des paquets en cliquant sur **Recharger** (en haut à gauche).

Pour la méthode basique, il faut tout simplement éditer manuellement le fichier texte contenant la liste des dépôts, c'est à dire **/etc/apt/sources.list**, et comme il s'agit d'un fichier système, il faudra l'éditer avec les droits administrateurs (commande **sudo**). Entrez donc la commande suivante dans votre terminal:

**Code : console**

```
sudo gedit /etc/apt/sources.list
```

Il suffit ensuite de dé-commenter les deux lignes correspondantes aux dépôts **Universe**, c'est à dire qu'il faut enlever le signe "#" au début de ces lignes (pour information, il s'agit des lignes 17 et 18 de ce fichier texte). Enregistrez et fermez l'éditeur de texte. Il faut maintenant mettre à jour la base de données des paquets pour que les nouveaux dépôts soient pris en compte. Pour cela, nous allons ré-utiliser la commande **apt-get** avec l'option **update** (mise à jour, pour les anglophobes 🙄).

**Code : console**

```
sudo apt-get update
```

Il ne nous reste maintenant plus qu'à relancer la commande pour installer le paquet **libwxgtk2.6-0**:

**Code : console**

```
sudo apt-get install libwxgtk2.6-0
```

On nous demande confirmation, en disant qu'un autre paquet va être installé (**libwxbase2.6-0**), il suffit de confirmer par un appui sur la touche **O**, et l'installation se fait sans problèmes.

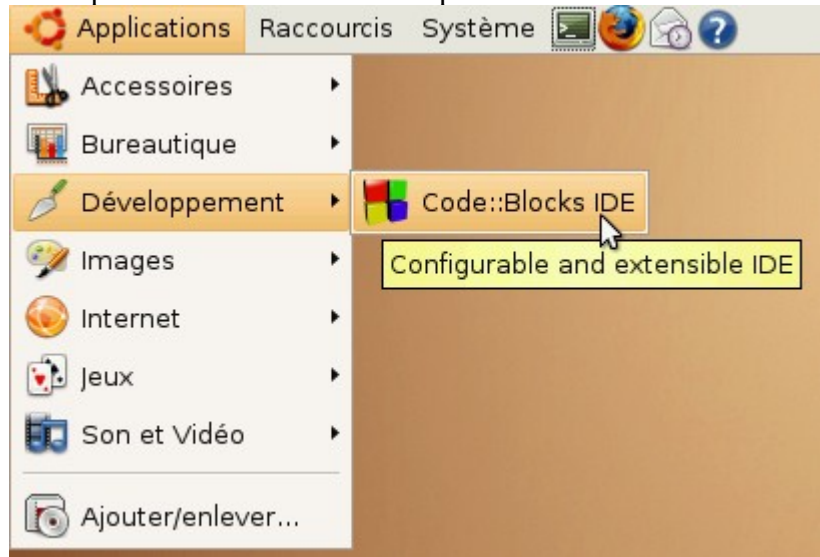
Nous allons maintenant pouvoir relancer l'installation du paquet de Code::Blocks avec la méthode vue au chapitre précédent:

**Code : console**

```
sudo dpkg -i CB_ [TAB]
```

Attention, roulements de tambours, et **ça marche !!!!!!!**

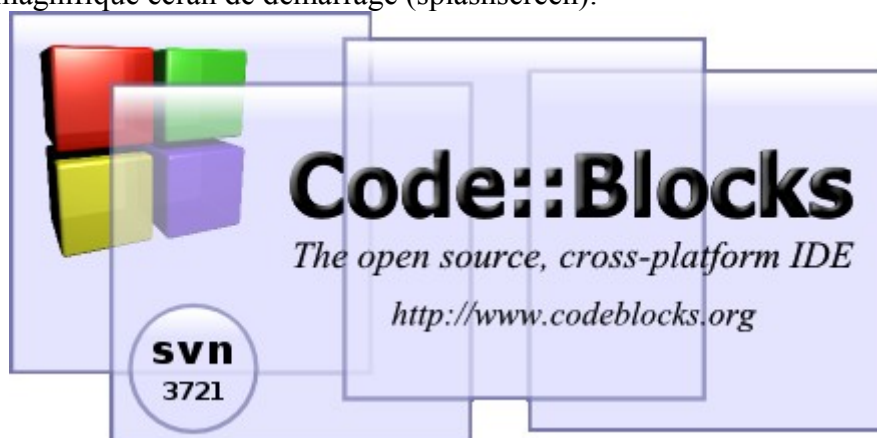
Pour vous en assurer, jetez un coup d'oeil dans le menu **Applications**, un sous-menu **Développement** a dû apparaître, dans lequel vous pourrez trouver un lanceur pour Code::Blocks.



*Capture d'écran N°6 : Le nouveau lanceur pour Code::Blocks*

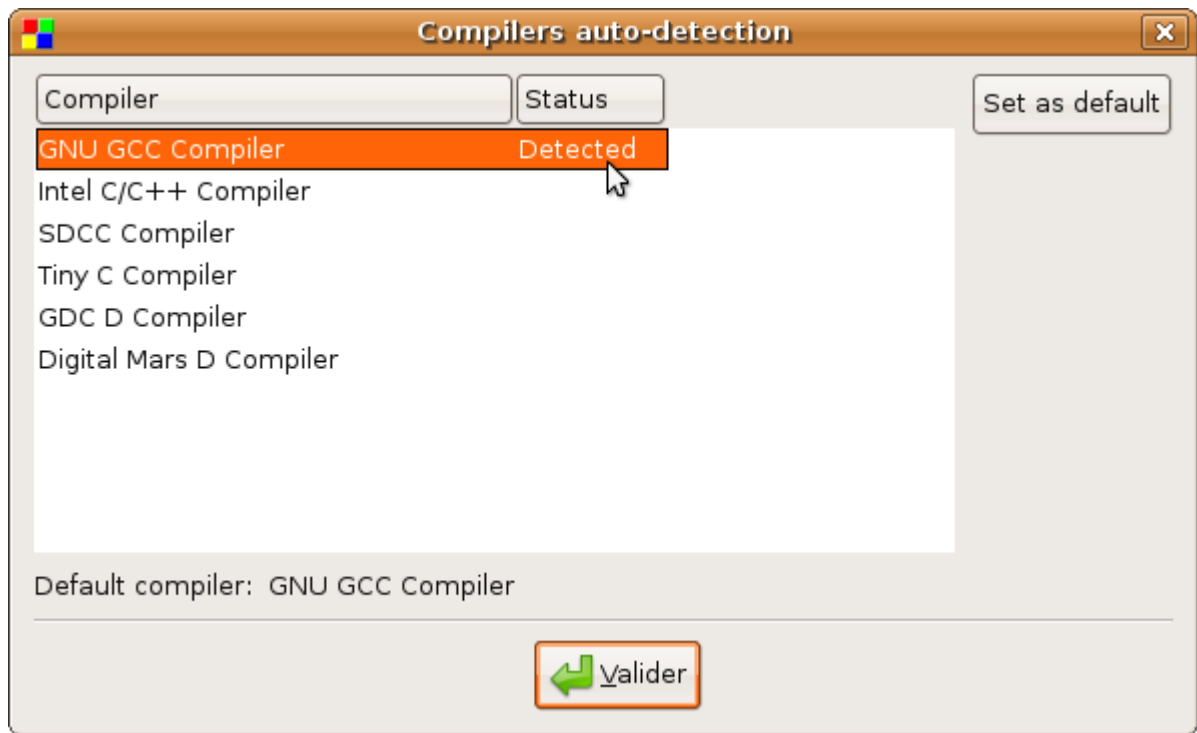
## Quelques essais

Ça y est, Code::Blocks est installé. Il ne nous reste plus qu'à l'essayer. Comme vous en mourrez d'envie, cliquez donc sur le fameux lanceur que a été créé par l'installation au chapitre précédent (voir capture n°6). Nous obtenons un magnifique écran de démarrage (splashscreen):



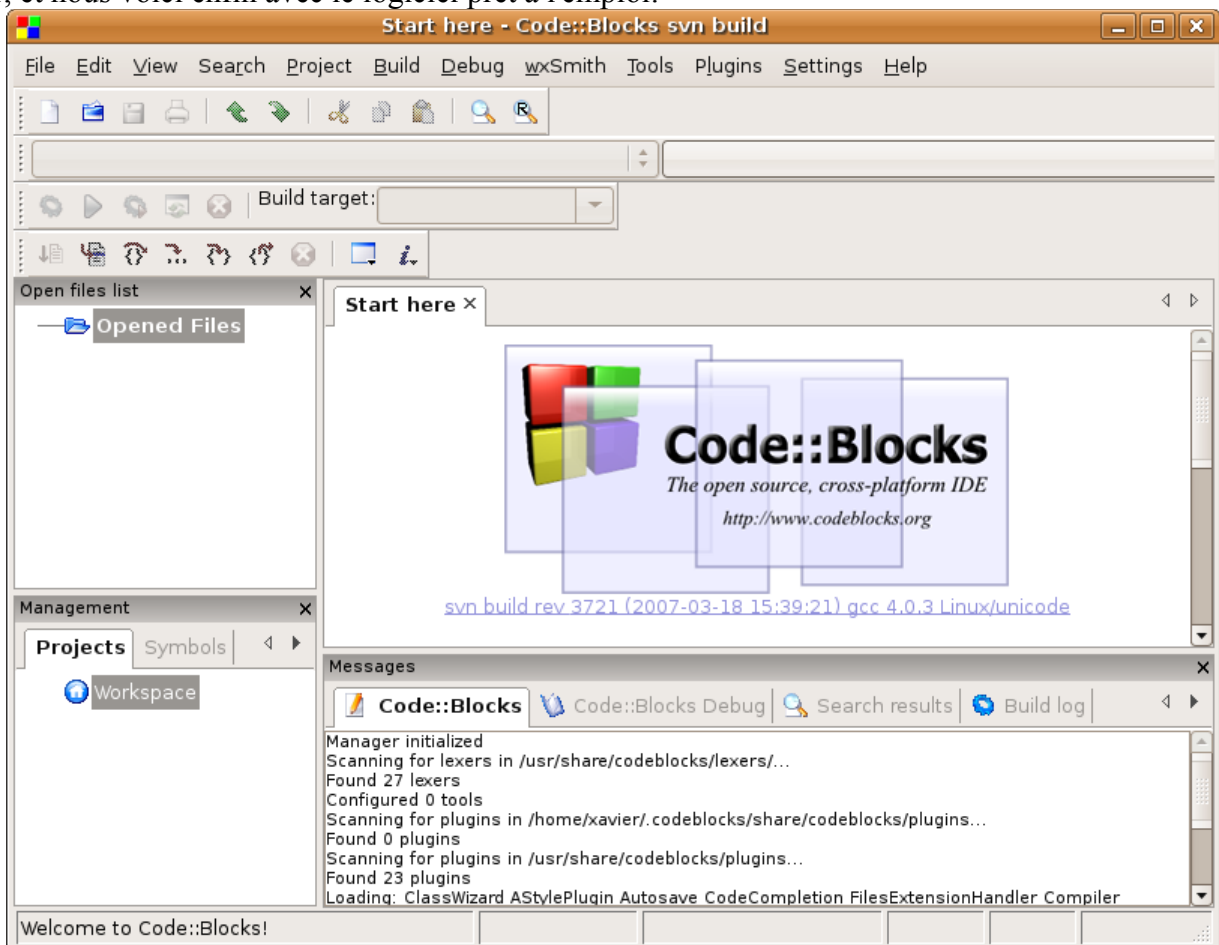
*Capture d'écran N°7 : L'écran de démarrage de Code::Blocks*

Puis, vient une boîte de dialogue permettant de choisir le compilateur par défaut, accompagnée d'une petite fenêtre d'informations qui apparaît en bas à droite de l'écran, et dont je n'ai pas réussi à obtenir de capture. Il suffit de cliquer dessus (ou d'attendre) pour qu'elle disparaisse.



*Capture d'écran N°8 : La boîte de dialogue pour le choix du compilateur*

Laissez le choix par défaut, et validez. Code::Blocks démarre, affiche une "Tips Box" que vous pouvez fermer, et nous voici enfin avec le logiciel prêt à l'emploi:



*Capture d'écran N°9 : L'interface de Code::Blocks*

Nous allons tout de suite créer un premier projet (le fameux "Hello, World !"). Mais avant tout, pour ne pas polluer notre répertoire utilisateur avec les futurs projets que nous allons créer, je vous suggère de créer un répertoire servant à tous vos projets de développement, que nous appellerons "devel".



Reprenez le Terminal (ou rouvrez-en un si vous l'avez fermé), et créez le répertoire "devel":

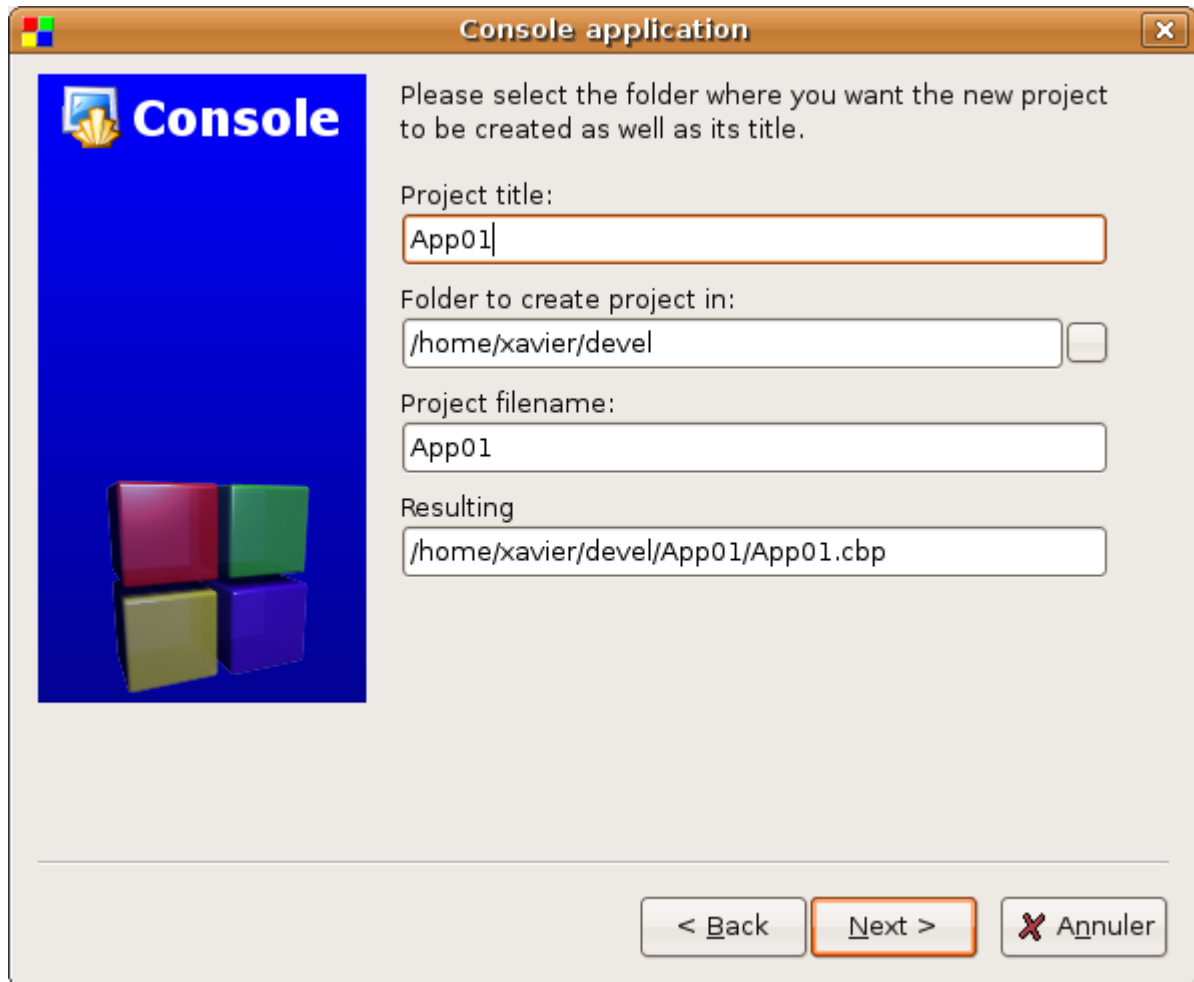
**Code : console**

```
mkdir devel
```

Dans Code::Blocks, sélectionnez "**File / New / Project**". L'assistant nouveau projet démarre.

Sélectionnez "**Console application**" (on va faire simple, pour commencer 😊) et cliquez su "**Go**". Passez la première étape de l'assistant (qui ne sert à rien, comme vous pouvez le constater)

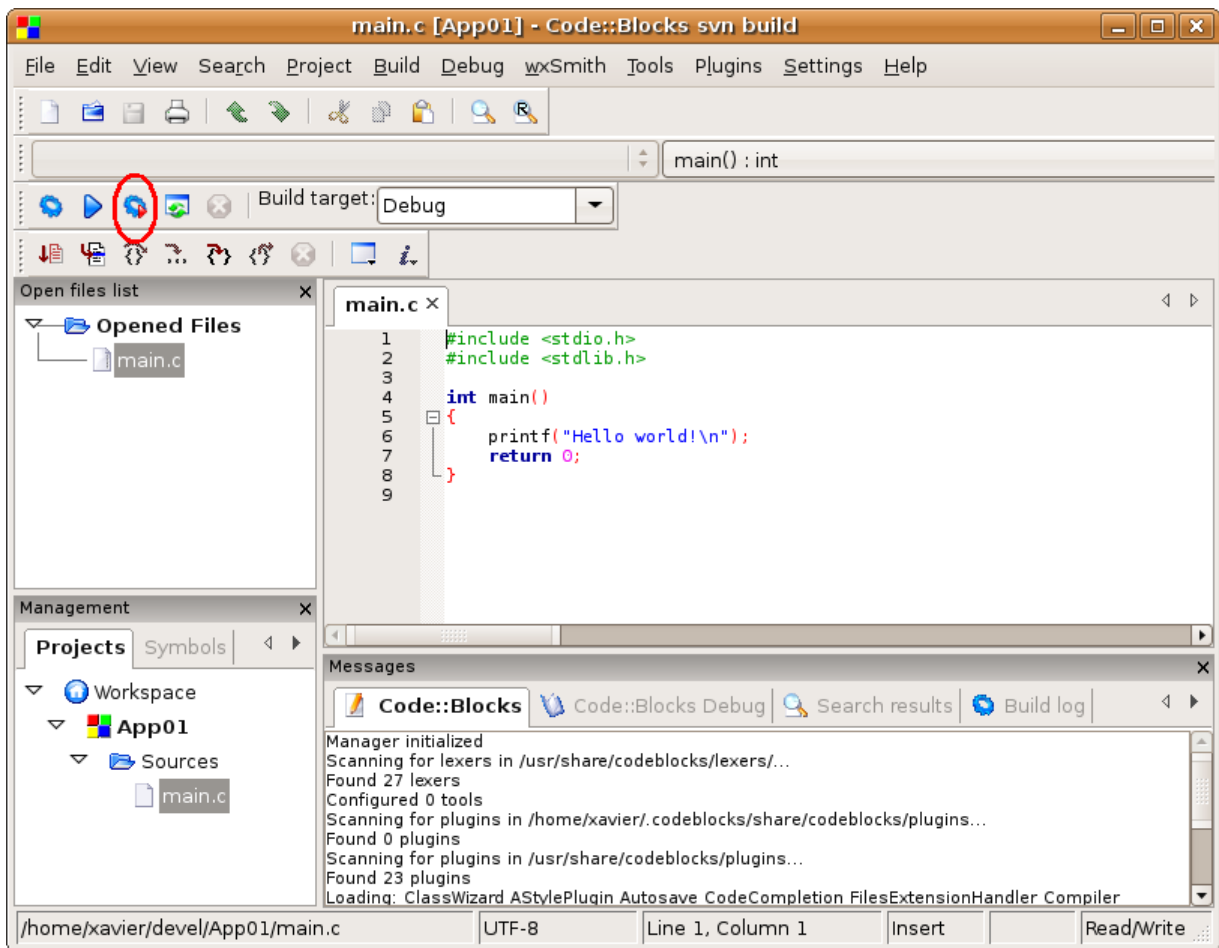
À l'étape suivante, donnez un titre à votre projet (je l'appelle **App01**, original, non ?) et renseignez le champ "**Folder to create project in**" de façon à ce qu'il pointe vers le répertoire "**devel**" que nous venons de créer. Vous remarquerez, dans le champ inférieur, que Code::Blocks va créer un sous-répertoire portant le nom du projet.



*Capture d'écran N°10 : Le nom et chemin du nouveau projet*

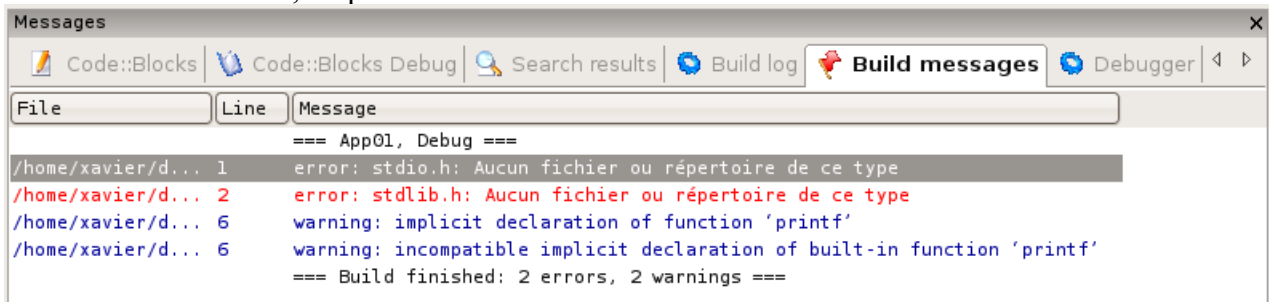
À l'étape suivante, laissez les options sélectionnées, et passez directement à la dernière étape vous demandant de choisir le langage de programmation à utiliser. Sélectionnez le "**C**" qui sera amplement suffisant pour l'instant, et cliquez sur le bouton "Terminer" (Finish). Notre projet est maintenant créé. Il ne reste plus qu'à le compiler et à l'exécuter. Pour regarder le code qui compose notre projet, il suffit d'ouvrir le fichier **main.c** qui a été créé par l'assistant : pour cela, il suffit de développer l'arborescence du projet (zone en bas à gauche) et de double-cliquer sur le nom du fichier. Comme vous pouvez le constater, le code est tout à fait simpliste et correspond au classique "Hello World !" dont tout le monde a entendu parler en débutant en programmation.

Pour compiler et exécuter, il suffit de cliquer sur le bouton prévu à cet effet (voir sur la capture n°10 ci-dessous).



*Capture d'écran N°11 : Le bouton permettant la compilation et l'exécution*

Cliquez donc sur ce bouton, et qu'obtenez-vous ? **Encore une erreur** 🤔



*Capture d'écran N°12 : Erreur à la compilation*

Et oui, il manque encore quelque-chose : les fichiers headers et librairies standard.

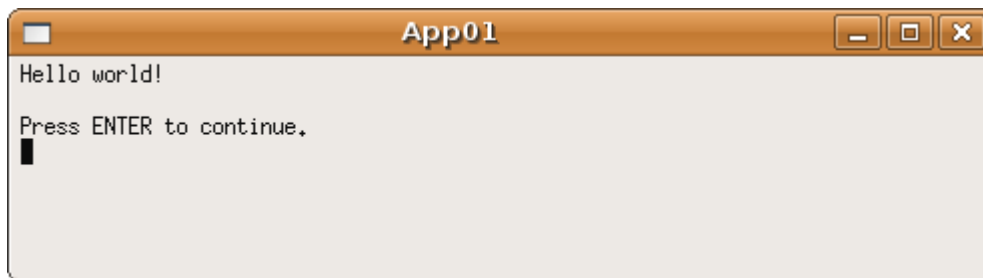
Comme je vois que vous commencez à en avoir marre, je vais vous donner la solution tout de suite : il faut que l'on installe le paquet **build-essential**. Vous vous souvenez de la commande ?

**Code : console**

```
sudo apt-get install build-essential
```

Maintenant, relancez la compilation et l'exécution : miracle, ça marche !!!! 🎉

On obtient une jolie petite fenêtre "console" dans laquelle s'affiche un superbe "Hello World!".



*Capture d'écran N°13 : Notre première application qui marche*

Nous allons maintenant créer un projet C++ en utilisant la librairie [wxWidgets](#), dont je suis un fervent utilisateur.

Vous vous souvenez, lors de l'installation, il nous a fallu installer un paquet appelé "**libwxgtk2.6-0**". Il s'agit des fichiers binaires de la version 2.6 de wxWidgets. Code::Blocks en a besoin, car c'est avec cette librairie qu'il a été programmé. Malheureusement pour nous, nous n'avons installé que la version "binaire", et non les fichiers headers et les libs de développement.

Nous allons remédier à cela avec la commande suivante:

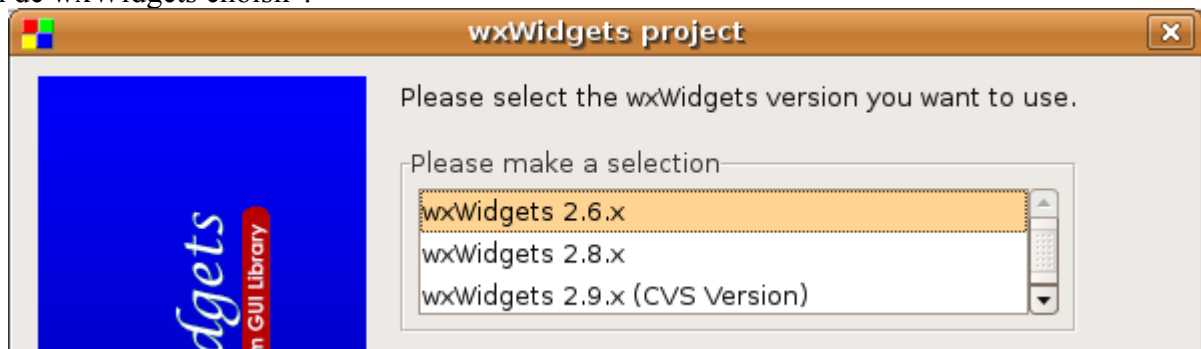
#### Code : console

```
sudo apt-get install libwxgtk2.6-dev
```

Confirmez l'installation, et laissez faire...

Reprenez maintenant Code::Blocks, et fermez le projet courant si vous êtes toujours sur l'application que nous avons créé précédemment.

Créez un nouveau projet (**File / New / Project**), sélectionnez le type "**wxWidgets project**" et cliquez sur "**Go**", passez la première étape qui ne sert toujours à rien, et nous arrivons au premier dilemme : quelle version de wxWidgets choisir ?



*Capture d'écran N°14 : Choix de la version de wxWidgets*

Je vais être franc avec vous : nous n'avons en fait pas le choix : nous ne disposons actuellement que de la version 2.6 dans les paquets d'Ubuntu.

Il existe bien une possibilité d'obtenir la version 2.8, mais il faut rajouter un dépôt à la liste actuelle. Vous pourrez toujours essayer de le faire vous-même : la procédure est indiquée sur la [page téléchargement de wxWidgets](#).

Quand à la version 2.9, il vous faudra carrément la compiler vous-même... 💡

Donc, nous utiliserons la version 2.6.

À l'étape suivante, nous retrouvons la boîte de dialogue nous permettant de saisir le nom et le chemin de notre nouveau projet. Je donne comme nom **wxApp02** au projet.

L'étape suivante vous demande votre nom, votre adresse e-mail, et l'adresse de votre site web. Mettez ce que vous voulez, sachant que ces informations seront insérées dans les commentaires au début de chaque fichier créé par l'assistant.

Ensuite, passez les étapes suivantes en laissant les options par défaut. Vous aurez bien le temps d'explorer tout ça plus tard.

Compilez et exécutez, et cette fois-ci ça marche du premier coup. Vous obtenez une jolie petite boîte de dialogue, avec 2 boutons.



*Capture d'écran N°15 : Notre application wxWidgets*

---

Et bien voilà, c'est terminé pour l'instant. Il ne vous reste plus qu'à nous pondre de jolies petites applications. J'espère que ce tutoriel vous sera utile, et qu'il est assez clair.

Je suis conscient qu'il est un peu rébarbatif par moments, mais il a au moins le mérite d'être complet (du moins, je le pense 😊)

Je suis bien sûr ouvert à toute piste d'amélioration, et disponible si vous rencontrez le moindre problème.