

Bien débiter en Langage C

par Franck Hecht ([retour aux articles](#))

Date de publication : 24/03/2007

Dernière mise à jour : 18/10/2010

Vous désirez apprendre le Langage C ? Vous ne savez pas par quoi commencer ? Ce guide va vous aider dans vos premiers pas, du choix de l'éditeur jusqu'à celui du livre ou des tutoriels en ligne ainsi que de nombreux liens pour vous permettre de vous exercer et d'aller plus loin !

I - Qu'est-ce que le C ?.....	3
II - Je débute :.....	4
II-A - Quels livres/tutoriels en ligne puis-je lire ?.....	4
II-B - Quels livres puis-je lire ?.....	4
II-C - Quels sont les outils dont j'ai besoin ?.....	4
III - Je veux approfondir mes connaissances :.....	5
III-A - Quels livres/tutoriels en ligne puis-je lire ?.....	5
III-B - Quels livres puis-je lire ?.....	5
III-C - Quels outils supplémentaires sont à ma disposition ?.....	5
IV - Où puis-je trouver des exercices à faire ?.....	7
V - Je veux aller plus loin :.....	8
V-A - Comment créer des applications graphiques ?.....	8
V-A-a - GTK+ (la GUI portable) :.....	8
V-A-b - L'API Win32 (uniquement pour Windows) :.....	8
V-B - Comment créer des jeux ?.....	8
V-B-a - La 2D :.....	8
V-B-b - La 3D :.....	9
VI - J'ai tout lu mais il y a des choses que je n'ai pas compris, que faire ?.....	10
VII - Conclusion.....	11
VIII - Remerciements.....	12

I - Qu'est-ce que le C ?

Créé au début des années 70 par **Dennis Ritchie** et **Brian Kernighan**, le Langage C reste encore aujourd'hui un des langages les plus utilisés au monde !




Ce langage de programmation est à la base des systèmes d'exploitations que nous connaissons aujourd'hui ou au moins du noyau de ces systèmes comme par exemple Unix/Linux. Le Langage C a justement été créé pour un seul et unique but au départ, développer un système d'exploitation (Unix) mais au fil du temps, grâce à sa puissance, il a été adopté par une large communauté de développeurs ce qui a permis au langage d'évoluer et surtout d'être standardisé.

Ce langage est multi plate-forme, c'est-à-dire qu'un programme que vous créez par exemple sous Linux pourrait tout à fait être re- compilé sous Windows, BeOS, Amiga, etc.... sans devoir changer grand chose (voir rien du tout) dans le code source si vous respectez les normes en vigueur ANSI/ISO.

II - Je débute :

Les débuts ne sont jamais très aisés, notamment en ce qui concerne le choix des livres, la recherche de tutoriels en ligne et dans le choix des éditeurs ou EDI (Environnement de Développement Intégré). Les catégories ci-dessous vous permettrons de vous acquitter de cette tâche souvent très longue et désagréable pour les débutants, tout ce que vous avez à faire, c'est de suivre ces différentes parties !

II-A - Quels livres/tutoriels en ligne puis-je lire ?


-  [Liste des cours complets sélectionnés par developpez.com](#)
-  [Initiation au Langage C - Emmanuel Delahaye](#)
-  [Codage en Langage C - Emmanuel Delahaye](#)

II-B - Quels livres puis-je lire ?


-  [Méthodologie de la programmation en C](#)

II-C - Quels sont les outils dont j'ai besoin ?

Il vous faut pour bien commencer, un EDI (*Environnement de Développement Intégré*). Plusieurs EDI sont à votre disposition mais, la plupart d'entre eux ne sont disponibles que pour un seul système d'exploitation.


C'est là que  **Code::Blocks** se distingue des autres EDI car il est portable, ce qui signifie qu'il est disponible sur au moins les systèmes les plus utilisés et plus connus soit Windows, Linux et Mac OS X.


Ci-dessous vous trouverez un lien direct de téléchargement de Code::Blocks pour chacun de ces systèmes:

 *Sous Windows, vous avez la possibilité d'utiliser la version dite "Stable" ce qui est souvent préférable: **Téléchargement de la version stable: 10.05***
Cette version est livrée avec la suite de compilation MingW !

 *Si votre système est un système Linux, vous avez alors le choix entre différents package selon votre distribution: **Téléchargement des versions 10.05 pour Linux***


 *Voici la version 8.02 si vous disposez de Mac OS X: **Téléchargement de la version Mac***

Vous pouvez également utiliser les versions en cours de développement, ça ne reste que des versions beta donc plus ou moins stables:  **Nightly Builds**








Voici un tutoriel pour l'installation de Code::Blocks sur Linux Ubuntu:  **Installation de Code::Blocks sous Ubuntu**
La procédure d'installation sur d'autres distributions devrait être relativement identique à celle-ci !

Voici également une présentation et test de cet EDI :  **Test de l'EDI C/C++ Code::Blocks**






III - Je veux approfondir mes connaissances :

Vous avez acquis de bonnes bases en Langage C, vous désirez approfondir vos connaissances soit dans le C ANSI/ISO ou voir les normes  **POSIX** ou encore vous plonger dans la programmation spécifique à Linux ? Les livres et tutoriels ci-dessous vous sont proposés par developpez.com !

III-A - Quels livres/tutoriels en ligne puis-je lire ?


-  [Liste des tutoriels hébergés sur developpez.com](#)
-  [Notes sur le Langage C - Emmanuel Delahaye](#)
-  [Concevoir un composant logiciel en C - Emmanuel Delahaye](#)
-  [Les types abstraits de données \(ADT\) - Emmanuel Delahaye](#)
-  [Notes sur les réseaux - Emmanuel Delahaye](#)
-  [Les Threads POSIX.1 - Emmanuel Delahaye](#)
-  [Comment se fabriquer des entrées solides en C - Emmanuel Delahaye](#)

III-B - Quels livres puis-je lire ?

-  [Le langage C - Norme ANSI, 2ème édition](#)
-  [Exercices corrigés sur le langage C - Solutions des exercices du manuel de langage C de Kerrighan et Ritchie](#)
-  [C en action](#)
-  [Maîtrise des algorithmes en C](#)
-  [Programmation système en C sous Linux : Signaux, processus, threads, IPC et sockets](#)


III-C - Quels outils supplémentaires sont à ma disposition ?

Dans cette section vous sont proposés des outils de *débogage* (*pratique indispensable pour avoir des applications solides et avec le moins de failles et bugs possibles*), des outils pour détecter des fuites mémoire, etc... Certains d'entre eux ne sont disponibles que sous un système, ils sont donc classés par OS :

 *Le débogueur phare de Microsoft et sans doute le meilleur sur ce système d'exploitation reste encore celui qui est fourni par défaut dans les solutions Visual Studio. La seule chose qui change ici, c'est que pour l'utiliser, il vous faut utiliser également l'éditeur Visual C++ (ici Visual C++ 2005 Express Edition) qu'on peut télécharger gratuitement sur Clubic: [Télécharger et installer Visual C++ 2010](#)*

Voici également un tutoriel vous permettant d'apprendre à utiliser ce débogueur:

 [Utiliser efficacement le débogueur de Visual Studio](#)

 *Sous Linux vous avez l'embarat du choix, voici sans doute le meilleur que l'on puisse faire à l'heure actuelle tout en restant le maximum dans la simplicité !*

 **DDD :**

Cet outil permet le débogage de vos applications. Ici plusieurs solutions s'offrent à vous, vous pouvez télécharger ce programme directement depuis le site officiel soit les liens suivants:

[Télécharger les sources](#)

[Télécharger les paquets Debian \(Ubuntu également\)](#)

[Télécharger les paquets RPM](#)

Le mieux reste sans doute d'utiliser l'outil de recherche/téléchargement/installation de votre distribution comme apt-get par exemple sous Debian et dérivés !

*Voici en complément, un tutoriel d'initiation à DDD: **Découverte du débogueur DDD***



Valgrind :

Ceci est une suite d'outils de débogage, il permet entre autre de détecter les fuites mémoires dans vos programmes.

Comme pour DDD, vous pouvez soit télécharger le programme par le biais des programmes de votre distribution ou bien télécharger directement depuis le site officiel:

Télécharger Valgrind depuis le site officiel...



GDB :

Cet outil de débogage est en fait le même que celui sous Linux et dont DDD (cité plus haut sur Linux) en est une interface graphique (le programme GDB d'origine est en ligne de commande !). Vous pouvez télécharger ce programme depuis le site officiel:

Télécharger GDB depuis le site officiel...



XCode :

Si vous préférez les interfaces utilisateurs (ce qui reste encore le plus pratique), vous pouvez également avoir à votre disposition XCode qui est un EDI pour Mac OS X et qui intègre un "Frontend" (une interface graphique pour un programme en ligne de commande).

Je ne vous propose pas de lien de téléchargement car ce programme est inclus dans le DVD d'installation de votre système !

IV - Où puis-je trouver des exercices à faire ?

Il est toujours dur de trouver des exercices relativement corrects, les liens qui vous sont proposés ci-dessous vont aussi bien vers des exercices en rapport avec les maths qu'au développement de petits jeux en mode console et divers problèmes à résoudre !

-  **Récréomath**
-  **TopDev**
-  **Franc - IOI**

Vous pouvez également entreprendre la création d'un petit jeu (*toujours en mode console*) comme par exemple :

- Mastermind (*deux versions peuvent être créées*) :
 - La première serait une version simple où le joueur devine la combinaison mise en place par l'ordinateur.
 - La seconde serait une version où c'est cette fois-ci, c'est l'ordinateur qui devine votre combinaison.
- Puissance 4
- Jeu de Dames


Ces exercices sont surtout intéressants du point de vue algorithmique et au niveau de l'IA (*Intelligence Artificielle*) !

V - Je veux aller plus loin :





Vous en avez assez des programmes en mode console ? Vous désirez faire des programmes avec une interface graphique (*GUI: Graphic User Interface/Interface Utilisateur Graphique*) ou voir même comment on peut créer des jeux vidéos en 2D et en 3D ? Regardez en-dessous !

V-A - Comment créer des applications graphiques ?

V-A-a - GTK+ (la GUI portable) :

La bibliothèque  **GTK+** permet de créer des programmes avec interfaces utilisateur tout en restant portable ce qui signifie que vous pouvez le faire tourner (*sous quelques conditions tout de même*) sur les systèmes comme Windows, Linux, et bien d'autres !

Voici des liens vous permettant d'apprendre à utiliser cette bibliothèque :

-  [Cours GTK 2](#)
-  [GTK+ par l'exemple](#)
-  [La page GTK+ sur developpez.com](#)
-  [La FAQ GTK+ sur developpez.com](#)


V-A-b - L'API Win32 (uniquement pour Windows) :

Contrairement à GTK+, l'API de Windows n'est pas portable et est sans doute un peu plus dure à prendre en main mais elle permet de faire des applications natives Win32 tout en ayant accès à toutes les fonctions de l'API Système ! Voici quelques liens pour vous familiariser avec cette API :





-  [Tutorial d'initiation à la programmation avec l'API Windows](#)
-  [Cours sur l'API Windows en C](#)
-  [La page Windows sur developpez.com](#)

V-B - Comment créer des jeux ?


V-B-a - La 2D :



La bibliothèque en vogue pour créer des jeux 2D (*ou autre application multimédia*) est la  **SDL** (*Simple Directmedia Layer*). Cette bibliothèque est en effet surtout spécialisée pour la création de jeux vidéos en 2 Dimensions, elle est très simple d'emploi et on la maîtrise assez rapidement.

Voici quelques liens pour vous familiariser avec cette bibliothèque :

-  [Tutoriaux SDL - Romain Perruchon](#)
-  [Introduction à la SDL](#)
-  [La page Jeux sur developpez.com](#)
-  [La FAQ SDL sur developpez.com](#)

V-B-b - La 3D :

Connue de tous, la bibliothèque de référence dans la création de jeux vidéos 3D est sans conteste  **OpenGL** !
Voici quelques liens pour vous permettre d'apprendre à l'utiliser :

-  **Cours d'introduction à OpenGL et GLUT**
-  **La page Jeux sur developpez.com**
-  **La FAQ OpenGL sur developpez.com**

VI - J'ai tout lu mais il y a des choses que je n'ai pas compris, que faire ?

Pas de panique ! Vous pouvez toujours poser vos questions sur les forums de developpez.com !
Voici les forums de developpez.com sur lesquels vous pouvez poser vos questions :

- **Forum : C**
- **Forum : Code::Blocks**
- **Forum : GTK+**
- **Forum : SDL**
- **Forum : OpenGL**
- **Forum : Programmation Windows**
- **Forum : Programmation Linux**

Sans oublier les FAQ des rubriques :

- **FAQ : C**
- **FAQ : GTK+**
- **FAQ : SDL**
- **FAQ : OpenGL**

VII - Conclusion

J'espère que ce guide vous aura permis d'y voir un peu plus clair et de commencer votre apprentissage du Langage C plus sereinement.

L'équipe C/C++/Jeux vous souhaite une bonne programmation !

VIII - Remerciements

Merci à **gege2061**, **Miles**, **Laurent Gomila** et **fearyourself** pour m'avoir donné leurs avis et suggestions.
Merci à **troumad** pour sa relecture et correction !