

# C++ / Outils de développement

Robotique/UE 3.1 - TD 5 - Novembre 2019

Supports de cours disponibles sur  
[www.simon-rohou.fr/cours/c++](http://www.simon-rohou.fr/cours/c++)

L'objectif de ce TD est de prendre en main des outils de développement couramment utilisés en C++. On s'intéresse notamment aux outils de compilation, de tests, de documentation et de versionnage. Pour ce TD, on reprendra les sources des travaux précédents autour du labyrinthe.

## A. Créer un dépôt pour un projet

1. Installez Git sur votre machine, et configurez-le :

```
git config --global user.email "email@ensta-bzh.fr"
git config --global user.name "PrenomNom"
git config --global http.proxy 192.168.1.13:8080
```

2. Sur la plate-forme de gestion de développement de votre choix (par exemple, GitHub), créez un dépôt pour votre projet de labyrinthe.
3. Clonez ce dépôt sur votre machine (`git clone`).
4. Ajoutez-y vos sources localement (`git add`, `git commit`) et envoyez vos travaux sur le dépôt distant (`git push`).



## B. Utiliser CMake

5. Créez un environnement CMake vous permettant de compiler votre projet.
6. Configurez un logiciel d'intégration continue (par exemple, Travis CI) vous permettant de vérifier la bonne compilation de votre projet à chaque `commit` / `push`.

## C. Tester une implémentation

7. Avec Git, créez une nouvelle branche `dev` dans laquelle vous allez développer des tests.
8. À l'aide de la bibliothèque de tests de votre choix, construisez une suite de tests unitaires, par exemple pour valider l'ajout de cellules au graphe du labyrinthe. Les tests pourront être lancés avec la commande `make test`.
9. Configurez Travis CI pour automatiquement exécuter les tests à chaque `commit` / `push` sur votre dépôt distant.
10. Une fois la branche `dev` valide sous Travis CI, fusionnez la branche `master` avec `dev` (`git merge`).

## D. Documenter des sources

11. À l'aide de Doxygen, écrivez la documentation technique des classes du projet. On générera les pages web HTML associées à cette documentation.
12. Ajouter un fichier `README.md` à la racine de votre répertoire, décrivant le projet. Ce fichier sera interprété par la plate-forme de développement, qui le mettra en forme (voir le langage de balisage Markdown).
13. Ajouter dans ce `README.md` l'affichage d'une icône `build passing` (issue de Travis CI) renseignant la santé de votre projet.