Projet
conception d'un robot mobile

Professeur : Valentin Gies

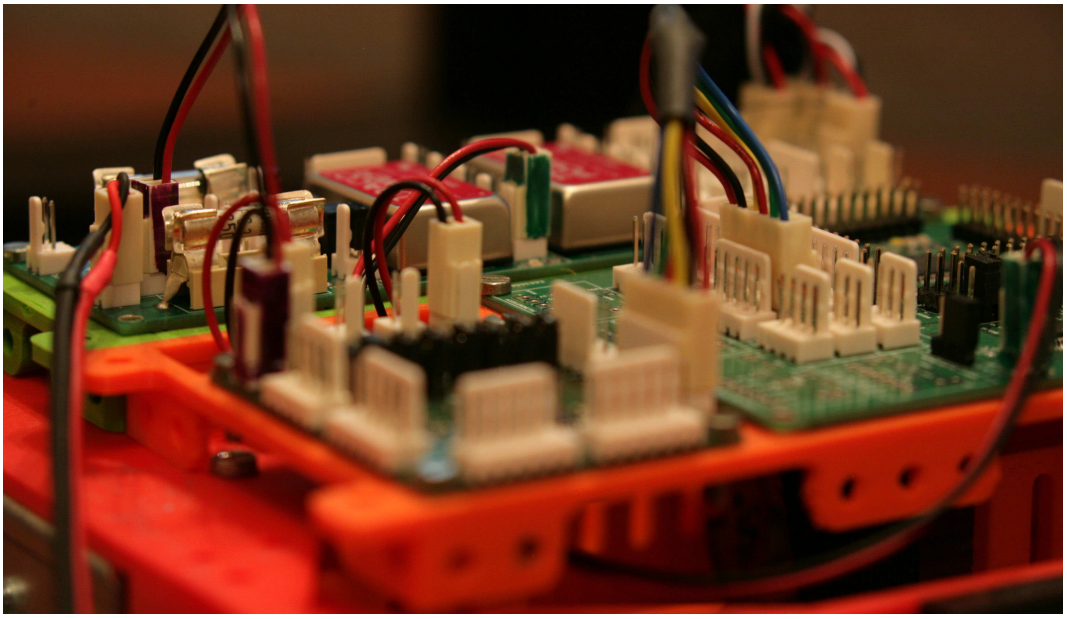

Table des matières

0.1 Utilisation d'un outil de développement collaboratif : GitHub

Commençons par les outils de développement collaboratifs, puisque sans eux, il sera difficile de gérer de gros projets en travaillant à plusieurs. L'un d'entre eux se nomme GitHub. Libre de droits dans son utilisation, il permet à des utilisateurs de partager des données dans des *repository*, sortent de dossier dédiés à un (ou quelques) projet. L'avantage est que l'historique des modifications effectuées par chacun des utilisateurs est consultable, que l'on peut revenir en arrière après une modification douteuse, et que l'on peut travailler à plusieurs sur un même projet, voir un même ficher simultanément.

Si ce n'est déjà fait, vous allez donc ouvrir un compte GitHub à votre nom de manière à pouvoir créer des *repository* pour vos projets. pour cela :

$\Longrightarrow$ Allez sur http://github.com, et créez vous un compte, si possible avec un *Username* évoquant clairement votre nom et prénom...

$\Longrightarrow$ Une fois votre compte créé, identifiez-vous dans GitHub et créez un nouveau repository en cliquant sur le bouton *New* en respectant les instructions suivantes :
- Le nom de ce repository doit suivre le code suivant : *SeaTech_Nom_Prenom* ou *IUT_Nom_Prenom*.
- Il doit être public.
- Ajouter lui un *.gitignore* ayant un template de type Visual Studio. Ceci permet d'éviter que les fichiers intermédiaires générés à la compilation par Visual Studio ne soient archivés. Cela simplifiera beaucoup les opérations de fusion ultérieures.

$\Longrightarrow$ Une fois votre projet créé, nous allons le partager avec un compte GitHub utilisé dans la salle de robotique afin d'y avoir accès facilement de chacun des ordinateurs de la salle. Pour cela, Allez dans *Settings* $\rightarrow$ *Manage Access* $\rightarrow$ *Invite a collaborator* et ajoutez le compte *iutgeiitoulonE105*. Prévenez le professeur quand c'est fait pour qu'il vous confirme le partage via son email.

$\Longrightarrow$ Une fois que l'accès à votre compte est partagé, il est possible de l'utiliser depuis chacun des ordinateurs de la salle de robotique depuis le logiciel *GitHub Desktop* que vous allez lancer. Pour y accéder, vous allez commencer par cloner localement le *repository* stocké en ligne. Pour cela :

— Allez dans *File* → *Clone Repository* → *Onglet GitHub.com* → *iutgeiitoulonE105/Nom de votre projet.*
— Spécifiez le local path suivant : *C :\GitHub*

Une fois la procédure de clonage terminée, GitHub devrait vous demander si vous voulez *contribuer au projet parent* ou *travailler pour vous*. Ne changez rien et laisser l'option *contribuer au projet parent* par défaut.

A ce point, un répertoire contenant votre projet doit être créé sur le disque du PC que vous utilisez dans le répertoire *C :\GitHub*. Nous allons à présent tester le fonctionnement de *GitHub*.

⟹ Ajoutez un fichier texte dénommé *test.txt* au répertoire du projet, avec ce que vous souhaitez à l'intérieur. Une cela effectué, regardez dans votre repo sous *GitHub Desktop* si le fichier est bien apparu dans la liste des *Changes*.

⟹ Si c'est bien le cas, vous allez archiver localement (*commiter*) le repo dans cet état, en ajoutant un nom au commit dans la ligne située au dessus de *description* et en cliquant ensuite sur *Commit to main*. Attention, à ce moment, le repo est sauvegardé localement uniquement, mais pas transmis au serveur, ce qui signifie que vous ne pouvez pas encore le récupérer depuis un autre PC.

⟹ A présent, nous allons transférer les modifications (*pusher*) vers le serveur. Pour cela, cliquez sur *Push origin*. Si vous allez dans l'historique du projet (*History*), vous devriez voir votre *commit* apparaître dans le haut de la liste des *commit* (il devrait d'ailleurs être le premier). Si ce n'est pas le cas, appelez le professeur.

⟹ A présent vous allez modifier le fichier *test.txt* créé pour y ajouter quelques mots et en retirer quelques autres, puis vous allez *commiter* et *pusher* votre projet vers le serveur. Une fois cela effectué, regarder dans l'historique, vous devries voir votre nouveau *commit* avec le détail des modifications fichier par fichier et ligne par ligne si vous cliquez dessus.

Vous savez à présent utiliser GitHub de manière basique. Il est possible de faire beaucoup plus avec, vous allez le découvrir au fur et à mesure des besoins et de votre utilisation. En particulier vous pouvez récupérer votre repo archivé depuis un autre PC en effectuant la même opération de clonage la première fois, puis en effectuant une récupération de données depuis le serveur (*Pull*) lorsqu'un commit a été effectué depuis un autre ordinateur. Pour aller plus loin, si vous effectuez un commit depuis deux ordinateurs différents sans avoir effectué de *Pull* entre les deux, vous allez dans ce cas devoir fusionner (*Merge*) les deux versions. Cela devrait se faire sans difficulté, si les modifications effectuées dans chacun des deux commit portent sur des fichiers différents (d'où l'importance du .gitignore qui permet d'éviter d'archiver les fichiers de compilation qui changent tout le temps...). Si elles portent sur des fichiers communs, alors il faudra effectuer un *Merge* manuel en sélectionnant dans les fichiers concernés ce qu'il faut garder ou pas. Dans ce cas, il est préférable d'utiliser un autre logiciel de gestion de GitHub dénommé GitKraken, à la place de GitHub Desktop. Arrivé ici, vous devriez être capables de vous en sortir seuls, avec l'aide de vos camarades ou de Google. Prenez le temps de faire pas mal de manipulation sur GitHub de manière à vous familiariser en profondeur avec l'outil.