

Documentation et outils

Robert Enzo
SUPINFO - TOURS

Sommaire :

| | |
|---|-----------|
| 1) Soutenance | 2 |
| a) Soutenance commerciale..... | 2 |
| b) Soutenance technique | 3 |
| 2) Logiciel : Doxygen | 4 |
| 3) Github et Github Desktop..... | 6 |
| a) Workflow GitHub n°1 : Features..... | 6 |
| b) Workflow GitHub n°2 : People | 7 |
| c) Utilisation de Gitmoji, des Review et gestion des Pull Request | 8 |
| 4) de Notion et Jira..... | 9 |
| 5) Plus d'informations..... | 11 |

1) Soutenance

a) Soutenance commerciale

Durant votre cursus à SUPINFO, vous allez à être amené à passer des soutenances commerciales.

Durant celles-ci vous allez devoir vendre votre production ainsi que vous mettre en avant, vous rentrez donc en « compétition » avec vos collègues de classes.

Votre jury sera composé de plusieurs personnes, certains membres de seront pas du même milieu que vous, ils n'auront donc pas les mêmes connaissances en programmation.

N'injectez pas de code dans vos slides, vous risquerez de perdre votre audience. De plus, il n'est pas pertinent de le présenter.

Exemple :

Quand vous regardez une affiche publicitaire, vous remarquerez qu'on ne met pas en avant la liste des composants du produit donné ais bel et bien le produit fini.

Voici donc un court exemple de ce que vous pourrez montrer, par un PowerPoint ou Keynote, en soutenance commerciale :

Slide de titre

(cérémoniales + phrase d'accroche débutant votre présentation orale)

Présentation de l'équipe

(que vous soyez seul ou en groupe, il est important de vous présenter. Montrant alors la répartition du travail par le rôle qui vous a été attribué ; exemple : Romain, Chef de projet – Backend)

Mise en situation

(rappel du cahier des charges,...)

Choix effectués

(fonctionnalités, structure ; Il est pertinent, ici, de répondre à la grande question : POURQUOI ?)

Démonstration de votre production

Commercialisation du projet

(qualité du produit, singularité. Faire un « devis » n'est pas une mauvaise idée)

Conclusion

(arrangement final de votre travail, mettez-vous en avant, ouvrez votre projet pour que le jury pose des questions auxquelles vous avez déjà les réponses)

Les soutenances commerciales à SUPINFO sont notamment utilisées pour les projets de fin d'année.

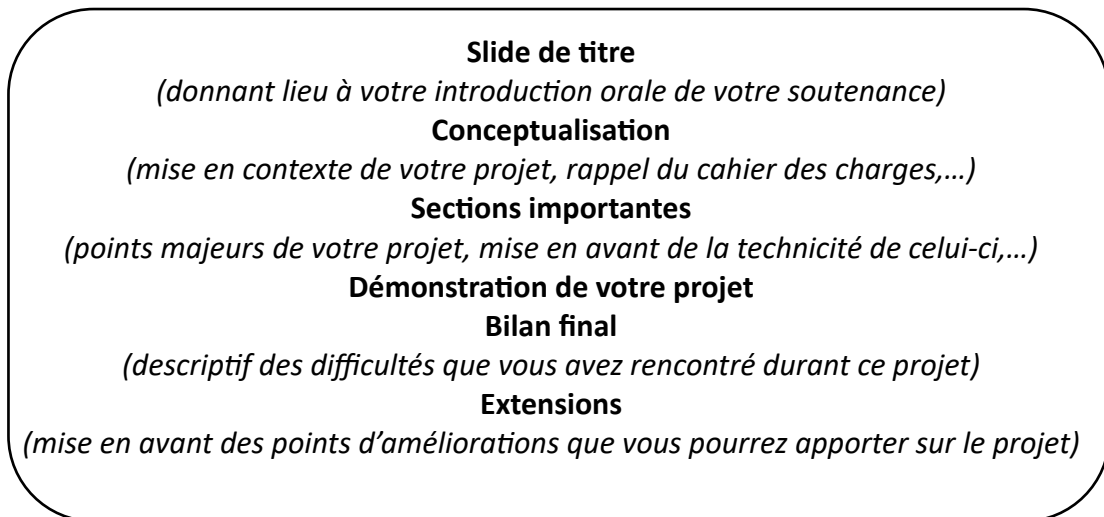
Comme vous le savez, il y a un autre type de soutenance : Soutenance technique.

b) Soutenance technique

Lors d'une soutenance technique, il est attendu que vous présentiez votre démarche de programmation.

Votre jury ne sera pas forcément du même domaine que vous, n'affichez donc pas de code durant votre PowerPoint/Keynote.

Voici un exemple du fil conducteur d'une Keynote type, que vous pouvez adapter, pour vos soutenance :



Pour ne pas mettre du code dans vos slides tout en l'évoquant, vous pouvez utiliser un logiciel de documentation technique tel que Doxygen, pour vous générer une documentation portée sur votre code via des commentaires que vous avez préalablement écrits.

2) Logiciel : Doxygen

Pour son installation, vous pouvez suivre le lien ajouté dans la rubrique « Plus d'informations ».

Une fois celle-ci faite, utilisez le Wizard que vous propose ce logiciel comme tel :

→ commande : **\$ doxywizard &**

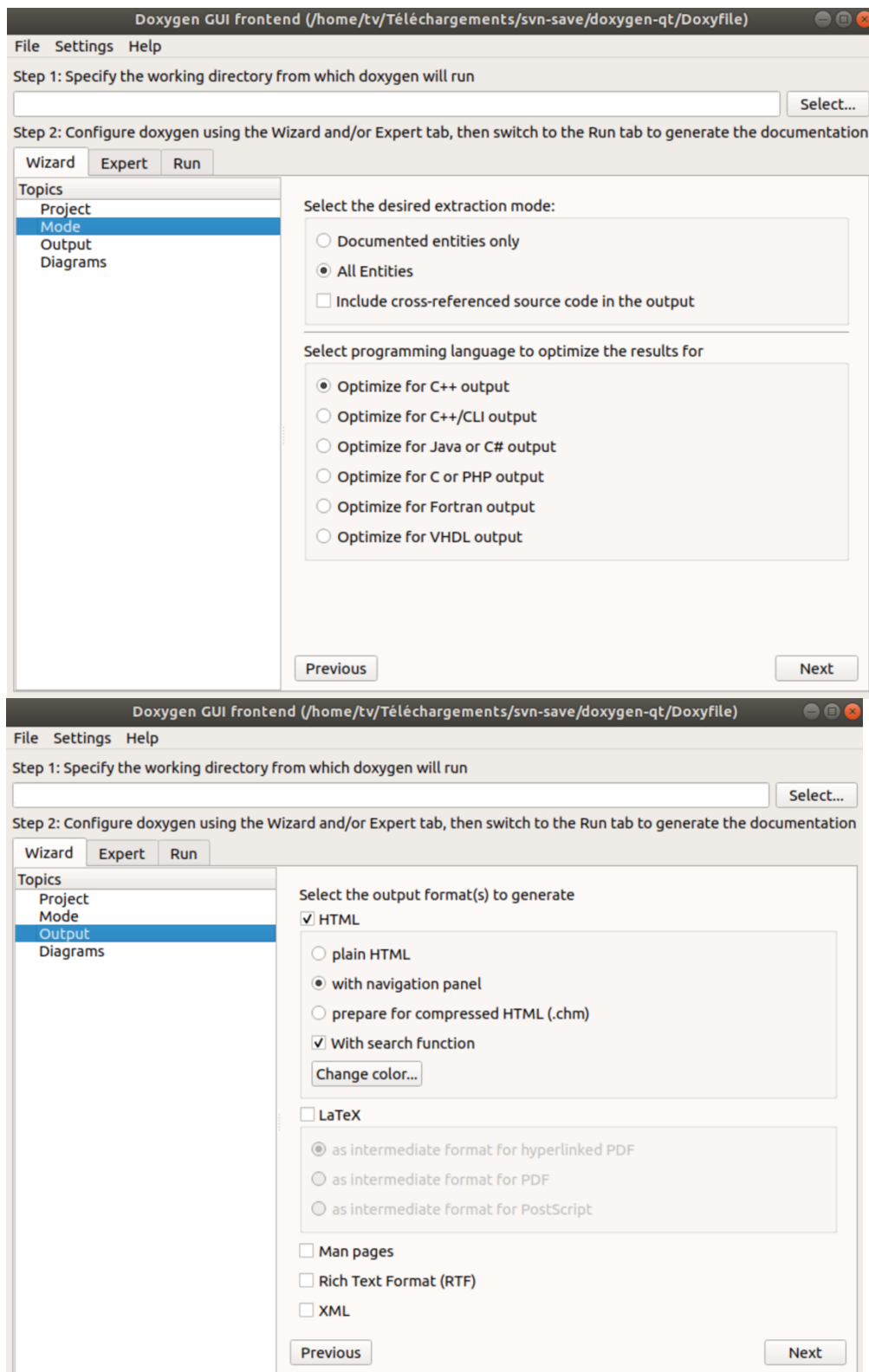
À savoir que l'onglet Wizard qui s'ouvrira vous servira pour :

- Créer votre projet.
- Sélectionner le format de génération de votre documentation (HTML avec ou sans frames, Latex, XML, PDF, ...).
- Sélectionner le fichier de destination.
- Générer des diagrammes.

Si votre code est déjà documenté, il ne vous restera plus qu'à lancer le logiciel avec le bouton « RUN ».

Voici donc un exemple de configuration :

The screenshot shows the 'Doxygen GUI frontend' window. The title bar indicates the path: `/home/tv/Téléchargements/svn-save/doxygen-qt/Doxyfile`. The menu bar includes 'File', 'Settings', and 'Help'. The main window is divided into two sections. The top section, 'Step 1: Specify the working directory from which doxygen will run', contains a text input field and a 'Select...' button. The bottom section, 'Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation', features three tabs: 'Wizard' (selected), 'Expert', and 'Run'. On the left of the 'Wizard' tab is a 'Topics' list with 'Project', 'Mode', 'Output', and 'Diagrams'. The main area of the 'Wizard' tab is titled 'Provide some information about the project you are documenting' and contains the following fields: 'Project name' (filled with 'Projet'), 'Project synopsis' (filled with 'BTS SNIR'), 'Project version or id' (filled with '0.1'), and 'Project logo' (with a 'Select...' button and a blue hat icon). Below this is a section 'Specify the directory to scan for source code' with 'Source code directory' (filled with '.') and a 'Select...' button, and a checked checkbox for 'Scan recursively'. The final section is 'Specify the directory where doxygen should put the generated documentation' with 'Destination directory' (filled with './doc') and a 'Select...' button. At the bottom are 'Previous' and 'Next' buttons.



L'exemple de configuration ci-dessus montre les différents « Topics » que vous pouvez modifier à votre guise pour créer votre projet selon vos préférences. Vous pouvez consulter différents sites pour vous documenter plus en profondeur sur ce logiciel.

Le mode expert est certes plus long et plus complexe à mettre en place mais reste très détaillé.

3) Github et Github Desktop

Lors de votre cursus à SUPINFO, vous allez être amené à collaborer avec des collègues de classes pour réaliser différents projets.

De fait, l'utilisation de GitHub pour le partage de fichier ainsi que la collaboration est donc pertinente.

Comme vous vous en doutez, GitHub est une plateforme en ligne de développement logiciel, utilisée pour stocker, suivre et collaborer sur des projets logiciels (exemple concret vos projets de module).

Son utilisation facilitera votre partage de fichier ainsi que votre vision globale sur le projet d'un point de vue purement axé sur la programmation.

Rappel sur la création d'un repository ; suivre le cours (qui est une introduction à GitHub) via ce lien :

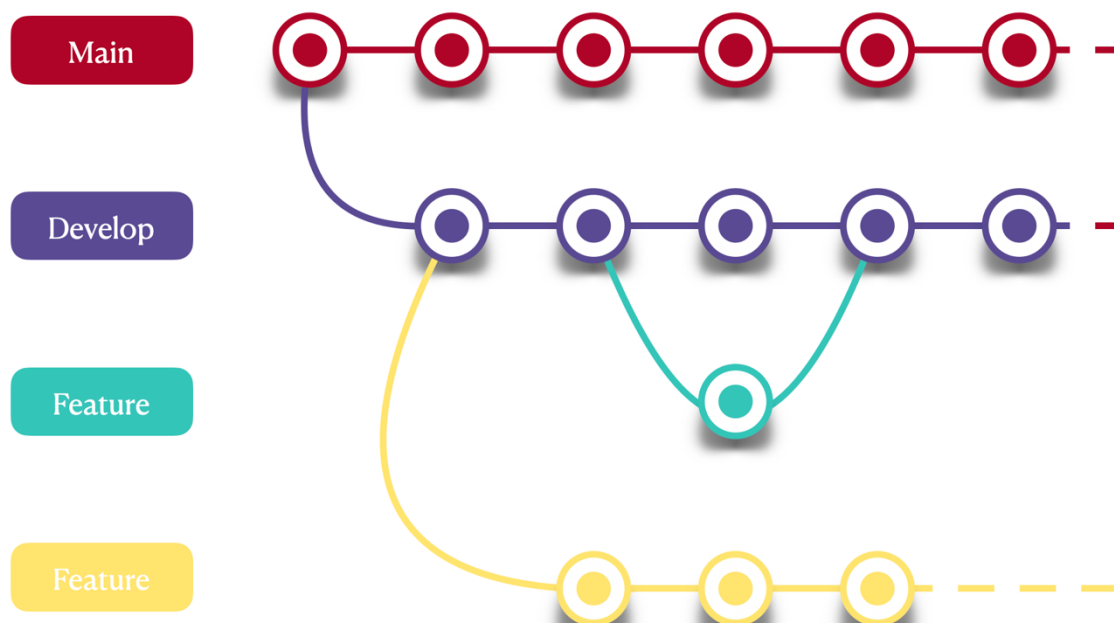
<https://github.com/skills/introduction-to-github?tab=readme-ov-file>

Pour vos différents projets, s'offre à vous deux Workflows :

→ Features

→ People

a) Workflow GitHub n°1 : Features

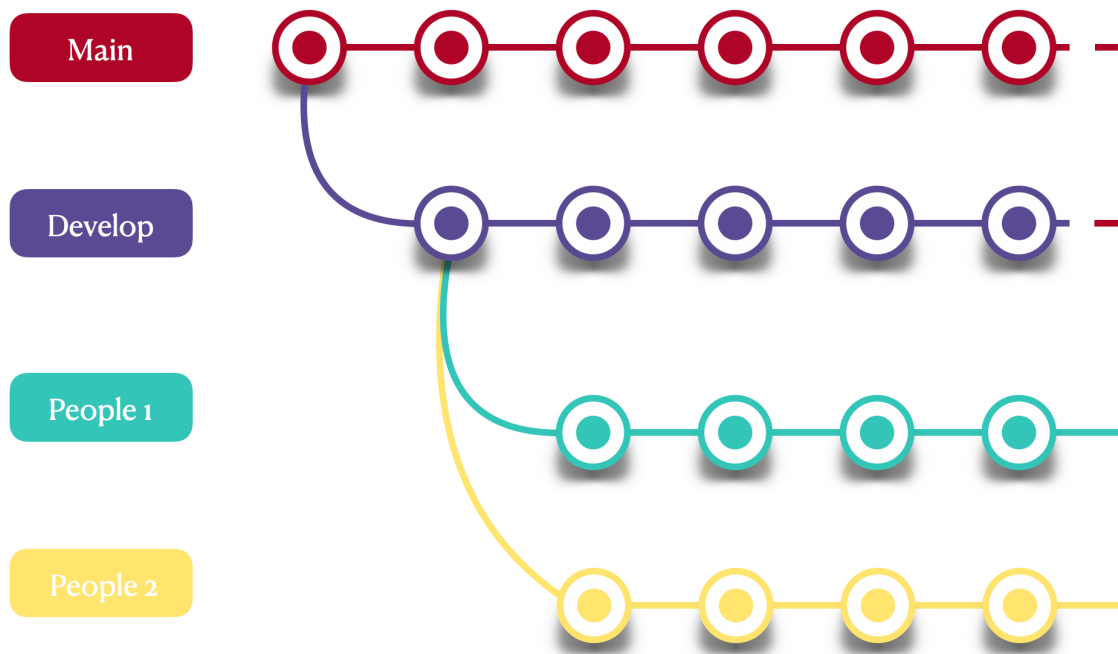


Représenté comme tel, on peut bien observer le fait qu'une seule branche est réservée pour un seul et même prototype de développement donné(feature).

Ce workflow est une bonne pratique que vous pouvez adapter, avec une autre branche partant du « Main » se nommant « Bug-Fix » ou avec d'autres alternatives.

Il est courant de voir ce style de méthode chez les développeurs mais il existe un autre workflow qui est bon à connaître ; People.

b) Workflow GitHub n°2 : People



Identifié comme ci-dessus, on distingue bien la différence avec le premier workflow. Dans le cas présent une branche n'est plus destinée à une seule et même fonctionnalités de votre projet mais directement à un développeur. Cette méthode fonctionne bien, mais peut être dépréciée par plusieurs développeur en raison de sa pratique « non-agile », car en utilisant ce workflow la répartition de vos tâches internes au projet sera différente et moins visible (au risque d'être brouillon).

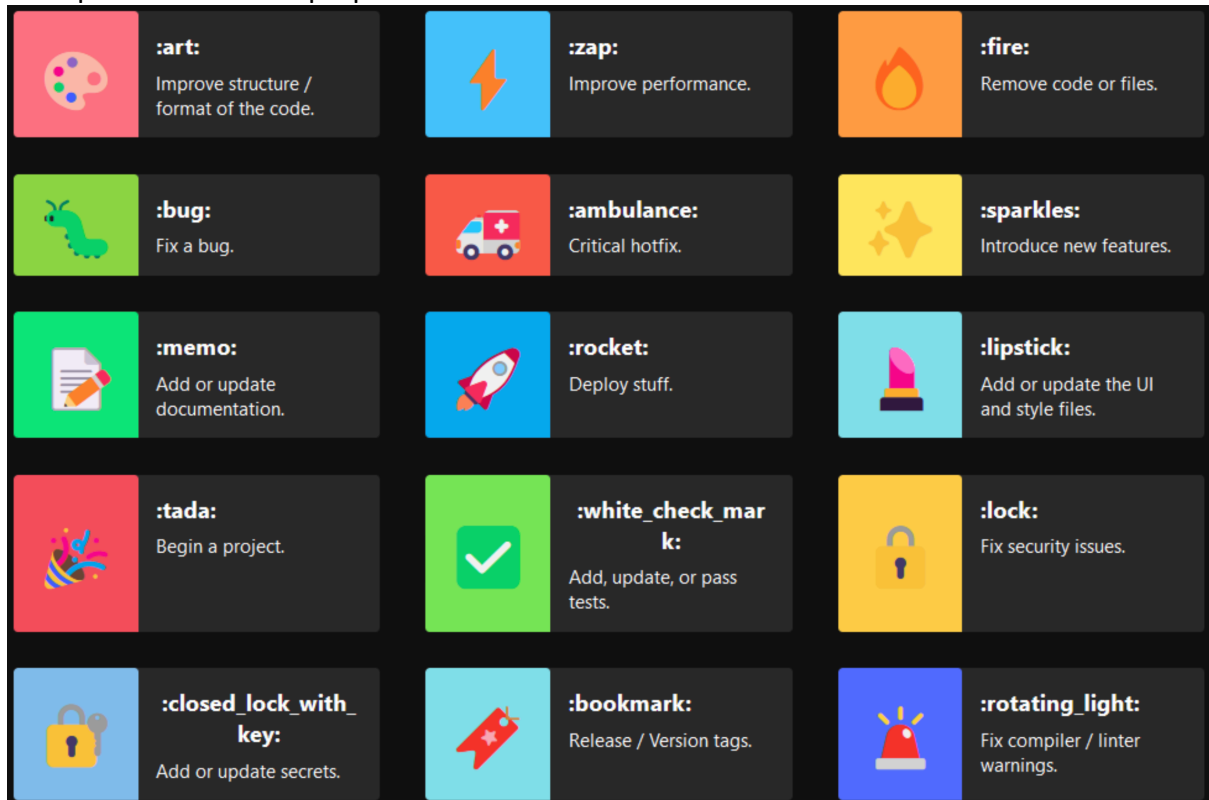
Il existe d'autre bonne pratique liant votre code et GitHub, notamment l'utilisation du Plug-in « Gitmoji » tout en faisant des Review avant la validation des Pull Request.

c) Utilisation de Gitmoji, des Review et gestion des Pull Request

Gitmoji est un plug-in à installer directement sur votre IDE.

Dès lors que vous ferez un « Commit », l'IDE vous proposera de mettre un Gitmoji :

Un emoji qui aura une fonctionnalité dans la coordination de votre projet. Voici quelques exemples illustrant ces propos.



Dans la gamme de Gitmoji qui vous est présentée, le **:sparkles:** sera celui que vous utiliserez le plus.

Il est d'usage de pouvoir se référer à ces différents emoji dans votre repository, facilitant alors la compréhension et la rapidité de la lecture s'il survient un problème en codant en équipe.

L'utilisation de ce plug-in se fera au moment du commit, après celui-ci viendra donc le moment de push, merge et de faire votre Pull Request sur la branche voulue (en général, sur la branche de développement).

Il est pertinent de faire des Review de code au moment des Pull Request par un autre membre du groupe que celui qui la fait.

Cela permet de bien comprendre le travail qui est proposé ainsi que limiter tous bug ou conflit.

Dans la section dédiée à l'utilisation des outils Jira et Notion, d'autres parties sur les Pull Request seront évoquées et mise en place.

Comme évoqué précédemment, GitHub est une plateforme en ligne, mais il existe aussi sa version logicielle se nommant Git Desktop.

Cette version va reprendre exactement les mêmes choses que la version initiale mais il n'y aura pas d'accès direct à l'entièreté du repository.

4) de Notion et Jira

Pour vos différents projets, professionnels ou personnels, l'utilisation d'outil de gestion de projet tel que **Notion** ou **Jira** est alors pertinente.

Notion sera plus axé vers la flexibilité et l'adaptabilité, ayant une interface conviviale ainsi que de nombreuses options de personnalisation (thèmes, couleurs, logo, ...). D'autre part, Jira va se distinguer comme un outil dit plus robuste, prisé par les équipes de développement de logiciels et **d'informatique**. Ce deuxième outil va mettre l'accent sur la gestion de projet agile et le suivi des problèmes.

Dans le cadre d'une utilisation de l'un de ces deux outils pour vos futurs projets, quelques mentions sont à connaître, telles que : dans les deux cas il est possible de lier votre GitHub à Notion comme à Jira.

De fait, vous allez pouvoir :

Avoir un visuel direct sur les différentes parties de votre code.

Mettre en place une répartition des tâches visible par tous : Une Pull Request pour une tâche par exemple.

The screenshot shows a Jira task card with the following details:

- Milestones:** Playable game
- Summary:** Vide
- N° Id:** PRO-46
- Type:** Task
- Date:** 22 janvier 2024 → 30 janvier 2024
- Assign To:** Enzo
- Pull Requests Git...:**
 - #24 [PRO-46] Prototype game UI (Merged)
 - #27 [PRO-46] :sparkles: Prototype UnitHealth and UI

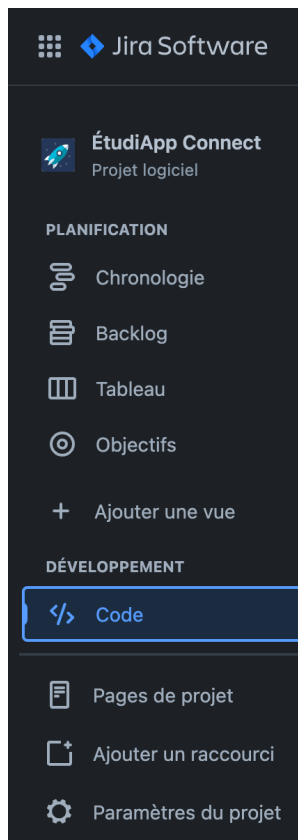
Prenons un exemple illustré par l'image ci-dessus.

Ici, il s'agit d'un projet (2PROJ) en lien avec SUPINFO. Un Notion a été préalablement créé ainsi qu'une base de données propre à Notion (Milestones).

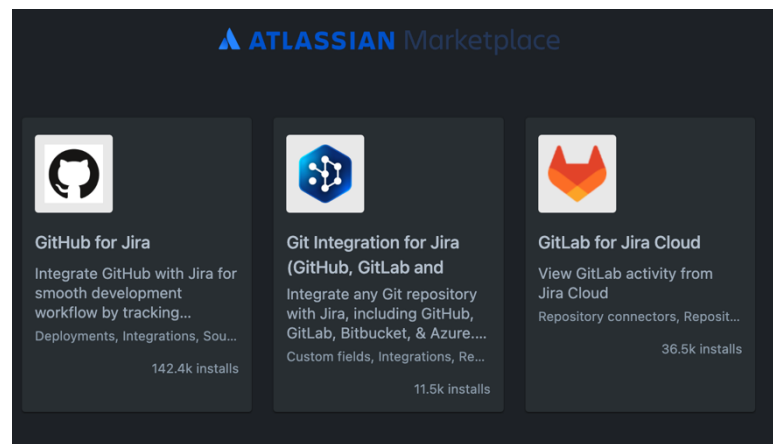
La tâche portant le nom donnée à la Pull Request est assignée à la section « Playable Game », avec une date/durée définies. Celle-ci a été assignée à un étudiant d'un groupe. Annotez qu'il y a deux fois la même Pull Request, car elle a été modifiée pour recevoir un Gitmoji (:sparkles: ici).

Pour ce qu'il s'agit de l'utilisation de Jira en groupe vous pouvez :

Dans un premier temps aller dans la rubrique « Code » se trouvant dans la section de développement, comme tel :



Et ensuite télécharger le plug-in approprié.



Autrement, vous pouvez aussi suivre ce lien qui illustrera de manière simple et explicite comme en créer un pour votre projet :

<https://support.atlassian.com/jira-cloud-administration/docs/create-and-edit-a-project/>

5) Plus d'informations

Pour votre culture personnelle ainsi que pour répondre à d'éventuelles questions qui ne peuvent être répondu par un simple document pdf, voici quelques liens que vous pouvez consulter pour en apprendre davantage sur les bonnes pratiques à avoir lors de vos prochains projets de module.

Apprendre à merge et rebase avec GitHub par l'IDE PyCharm :

<https://www.jetbrains.com/help/pycharm/apply-changes-from-one-branch-to-another.html>

Gérer et résoudre les différents conflits par l'IDE PyCharm :

<https://www.jetbrains.com/help/pycharm/resolve-conflicts.html>

Docstrings par PyCharm :

<https://www.jetbrains.com/help/pycharm/creating-documentation-comments.html#919f81c4>

Shortcut important : Context action Menu :

(pour générer la documentation par les commentaires: « alt + entrée »)

""" + entrée ; au-dessus de la classe ou fonction (**python**)

/** + entrée ; au-dessus de la classe ou fonction (**java, C et C++**)

/// + entrée ; au-dessus de la classe ou fonction (**C#**)

Format de documentation python par stackoverflow :

<https://stackoverflow.com/questions/3898572/what-are-the-most-common-python-docstring-formats>

Documentation générale de Doxygen :

<https://www.doxygen.nl/manual/docblocks.html>

Documenter un program python par des commentaires Doxygen :

<https://www.woolseyworkshop.com/2020/06/25/documenting-python-programs-with-doxygen/>

Templates Doxygen (usages et exemples) :

https://github.com/kcwongjoe/doxygen_theme_flat_design?tab=readme-ov-file

Template Doxygen (exemple type) :

<https://github.com/jothepro/doxygen-awesome-css>