

Sujet de stage M1 Info

« Développement logiciel pour learn-ocaml : maintenance corrective et évolutive »

Encadrant : Érik Martin-Dorel (erik.martin-dorel@irit.fr)

Janvier 2022

1 Informations pratiques

- Stage de M1 Info à temps plein
- Nature du travail : développement logiciel/web
- Proposition de dates : début mai → fin juillet (3 mois)
- Gratification de stage (taux horaire en vigueur depuis 2020) : 600,60 €/mois
- Établissement : [IRIT](http://irit.fr)
- Contact : erik.martin-dorel@irit.fr (joindre CV+LM pour candidater)

2 Description du contexte du sujet

Depuis l'année 2016-2017, l'UE PFITA (Programmation Fonctionnelle et Introduction aux Types Abstraits) de la Licence 3 Informatique de l'Univ. Paul Sabatier utilise un outil d'évaluation automatique à base de tests unitaires et d'analyse statique, et du moteur de MOOC learn-ocaml¹ développé par la fondation OCaml. Cet outil génère un rapport HTML indiquant les erreurs ou les motifs de code à éviter.

Depuis 2017, nous avons développé dans le projet PFITAXEL (auquel ont participé 10 étudiants de L3 ou M1 de l'UPS) et en collaboration avec la fondation OCaml :

- une seconde application web² pour faciliter le travail des enseignants dans la conception de nouveaux sujets de devoir maison et de TP ;
- un front-end Emacs³ pour accéder en tant qu'étudiant aux fonctionnalités de la plateforme learn-ocaml directement à partir d'un IDE OCaml standard de type « Tuareg » ;
- une extension de learn-ocaml permettant de gérer l'authentification par le protocole LTI⁴ ou login/mot-de-passe (en cours d'intégration).

3 Proposition de sujet de stage

Le dépôt principal du logiciel Learn-OCaml (comportant 1300+ commits) est muni depuis fin 2021 d'une chaîne de CI/CD complète comportant tests d'intégration, et livraison continue, à l'aide de l'outil release-please permettant de tirer profit de la spécification nommée *Conventional Commits*.⁵

1. <https://github.com/ocaml-sf/learn-ocaml>
2. learn-ocaml-editor : <https://pfitaxel.github.io/pfitaxel-demo/>
3. learn-ocaml.el : <https://melpa.org/#/learn-ocaml>
4. https://en.wikipedia.org/wiki/Learning_Tools_Interoperability
5. <https://github.com/ocaml-sf/learn-ocaml/blob/master/CONTRIBUTING.md#conventional-commits>

Il n’y a cependant pas de tests unitaires à proprement parler (ni de tests end-to-end pour les aspects front-end), et d’un certain point de vue, compte tenu des bonnes pratiques usuelles liées à la « pyramide des tests », ceci est un frein à la réalisation de certaines évolutions pouvant impacter des parties critiques de la plateforme.

Le but de ce stage est de travailler sur trois axes différents :

- faire un état de l’art des bibliothèques de tests unitaires et/ou de couverture de code pour OCaml, dans l’optique de pouvoir recommander une solution spécifique ajoutable à learn-ocaml ;
- mettre en application cette solution et l’automatiser (CI) en couvrant au moins un composant jugé critique ou emblématique ;
- être force de proposition et contribuer à la maintenance corrective et évolutive de Learn-OCaml : implémenter des User Stories telles que des « quality-of-life enhancements » suggérés par les enseignants, aider à prendre en charge des correctifs en collaboration avec l’équipe de développement.

Le code développé sera placé sous une licence libre permissive (MIT).

Le stage s’effectuera en contexte agile, en collaboration étroite avec Yann Régis-Gianas, co-mainteneur du projet Learn-OCaml, ainsi que l(es) autre(s) étudiant(s) recruté(s).

4 Environnement technologique

Environnement de développement : [Git](#) et [GitHub Actions](#), [GNU Emacs](#), [Tuareg](#) et [Merlin](#), [Docker](#)

Langages : [OCaml](#), [JavaScript](#)

Frameworks : [Lwt](#), [js_of_ocaml](#)

5 Pré-requis

Sont attendus : un intérêt prononcé pour la programmation fonctionnelle, une bonne expérience du langage OCaml (en particulier de son fragment fonctionnel pur), des connaissances solides en génie logiciel, en gestion de version décentralisée (Git) et en Web Services (REST), et de bonnes aptitudes de rigueur et de synthèse.

Non demandées : expérience préalable en virtualisation applicative avec Docker, ou connaissance des frameworks impliqués (Lwt, js_of_ocaml).