

Projet PFITAXEL (M1 INFO)

Exercices en ligne pour PFITA

Érik Martin-Dorel (erik.martin-dorel@irit.fr)
Frédéric Migeon (frederic.migeon@irit.fr)

UPS/IRIT – 18 novembre 2017

1 Nature du projet : développement logiciel/web

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'UPS propose des devoirs maisons dont la correction s'appuie sur un outil d'évaluation automatique à base de tests unitaires et d'analyse statique. Cet outil génère un rapport HTML pouvant être envoyé aux étudiants pour leur indiquer leur erreurs ou les motifs de code à éviter.

Cet outil est composé d'un script, d'une version modifiée du moteur de MOOC learn-ocaml^{1,2} développé par OCamlPro, et enfin de codes OCaml dédiés pour corriger chaque devoir maison.

Le but de ce projet est de développer des évolutions de l'outil

- pour faciliter le travail des enseignants dans la conception de nouveaux sujets
- et pour pouvoir généraliser l'utilisation de cette plateforme aux TPs PFITA pour accroître l'autonomie et l'engagement des étudiants.

3 Description du travail demandé

Le travail se déroulera en 4 parties :

1. Prise en main de l'environnement technologique et identification de motifs de code typiques dans les scénarios de tests visant à évaluer automatiquement les exercices OCaml.
2. Développement de fonctionnalités pour faciliter la création de nouveaux sujets (éditeur intégré dans l'application en ligne learn-ocaml ET/OU ensemble de macros pour l'éditeur [emacs/tuareg-mode](#)). Il pourrait s'agir d'un éditeur de questions simplifié permettant de générer semi-automatiquement le code OCaml de l'évaluateur.
3. Expérimentation et utilisation de cette fonctionnalité en mettant en place des scénarios de tests automatisés pour la plupart des exercices existants dans le module PFITA.
4. Développement de macros pour emacs pour tirer profit en TP à la fois des fonctionnalités d'édition d'[emacs/tuareg-mode/merlin](#), et du feedback fourni via la plateforme learn-ocaml.

1. <https://try.ocamlpro.com/learn-ocaml-demo/>

2. <https://github.com/ocaml-sf/learn-ocaml>

Le code développé sera placé sous une licence libre comme BSD-3.

Des revues pourront être organisées avec les différentes parties prenantes en fonction du calendrier établi au début du projet.

4 Environnement technologique

- [ocaml](#)
- [js_of_ocaml](#)
- [emacs lisp](#)
- git et GitHub
- intégration continue avec Travis CI

5 Parcours de rattachement

Parcours DL ou IHM du Master Informatique

6 Taille de l'équipe recommandée

4 étudiants