Travail de Bachelor 2025
Filière ISC
Orientation ISCL

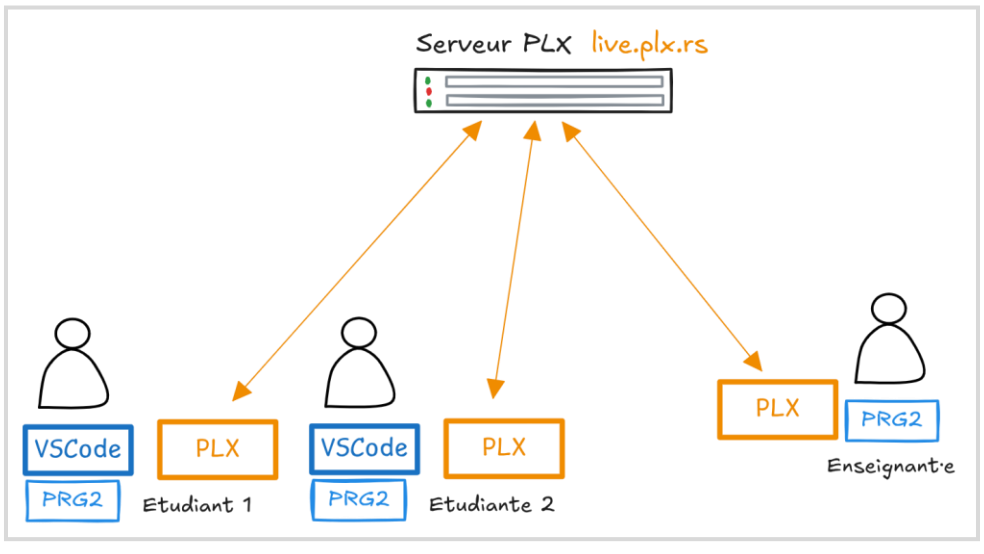
Concevoir une expérience d'apprentissage interactive à la programmation avec PLX

Contexte
La programmation est un domaine particulièrement abstrait et complexe à apprendre. Les sessions théoriques en université sont souvent données sous forme magistrale et les étudiant·es ont rarement la possibilité d'être actif·ves. Lors des rares sessions d'exercices, les enseignant·es peinent à savoir si les concepts de l'exercice sont acquis et les étudiant·es n'ont pas de feedback sur leur code.

Objectifs
Ce travail de Bachelor poursuit le développement de PLX, une application desktop qui accompagne les étudiant·es dans leur apprentissage de l'informatique.

Le projet a été étendu pour

- **Permettre aux enseignant·es d'accéder au code des étudiant·es durant des sessions d'entrainement en classe.** Un serveur central sert à transférer le code et les résultats des vérifications automatiques (les checks) vers le tableau de bord de l'enseignant·e.
- **Faciliter la rédaction des exercices de programmation en inventant une syntaxe concise**, facile et rapide à taper, pour décrire les informations du cours. Au lieu d'utiliser un format répandu comme le JSON, YAML ou TOML, la syntaxe DY a été inventée pour réduire au minimum la complexité de rédaction. Cette syntaxe se trouve à mi-chemin entre le Markdown et le YAML et intègre une vérification du document.



Architecture clients/serveur de PLX

Résultats
Un serveur en Rust a été implémenté en utilisant le protocole WebSocket. Les messages JSON entre le client et le serveur ont été spécifiés dans un protocole et des tests automatisés ont permis de vérifier le fonctionnement du protocole.

Un parseur de la syntaxe DY a été créé dans une librairie Rust appelée *dy* et son intégration dans PLX permet d'extraire les données d'un cours, de compétences et d'exercices. Des tests unitaires ont permis de valider le fonctionnement général et la génération d'erreurs.

Le tableau de bord des enseignant·es a pu être développé sur l'application desktop de PLX et une session live peut être suivie de bout en bout, sur plusieurs exercices. Un CLI a été développé pour démarrer le serveur de session live et utiliser le parseur depuis son terminal.

Conclusion
Les enseignant·es de programmation ont de nouveaux outils à disposition pour rendre leurs cours dynamiques, donner du feedback durant des exercices en classe et très rapidement créer de nouveaux exercices dans des fichiers textuels.

```
exo Salue-moi
Un petit programme qui te salue avec ton nom complet.

check Il est possible d'être salué avec son nom complet
see Quel est ton prénom ?
type John
see Salut John, quel est ton nom de famille ?
type Doe
see Passe une belle journée John Doe !
exit 0
```

Exemple 1: Exercice PLX rédigé en syntaxe DY.

```
> plx parse exo-error/exo.dy
Found 1 item in exo-error/exo.dy with 3 errors.

Error at exo-error/exo.dy:3:0
check Salue la personne donnée en argument
| Missing required key 'see'

Error at exo-error/exo.dy:5:0
args John
^^^^ The 'args' key can only be used once at this level
```

Exemple 2: Erreurs générées par le parseur et leur présentation via le CLI

Auteur: Samuel Roland
Prof. responsable: Bertil Chapuis
Sujet proposé par: Samuel Roland

Hes·so
Haute Ecole Spécialisée de Suisse occidentale