haute école
neuchâtel berne jura

Crawlect-GUI

Projet de fin de cours IHM

Présentation pour le cours IHM
Professeure : Rizzotti Aïcha

Présenté par : Yves & Alexandre
Neuchâtel, le 17.06.2025

Sommaire

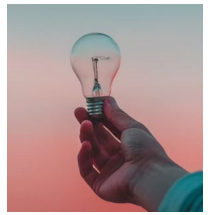
- Introduction
- Utilité de Crawlect
- Apports de l’interface graphique
- Démonstration
- Principe de développement
- Modèles de conception
- Améliorations
- Perspectives futures
- Conclusion

Introduction

Objectifs du projet

- Créer une IHM simple avec interaction
- Apprentissage de Swing

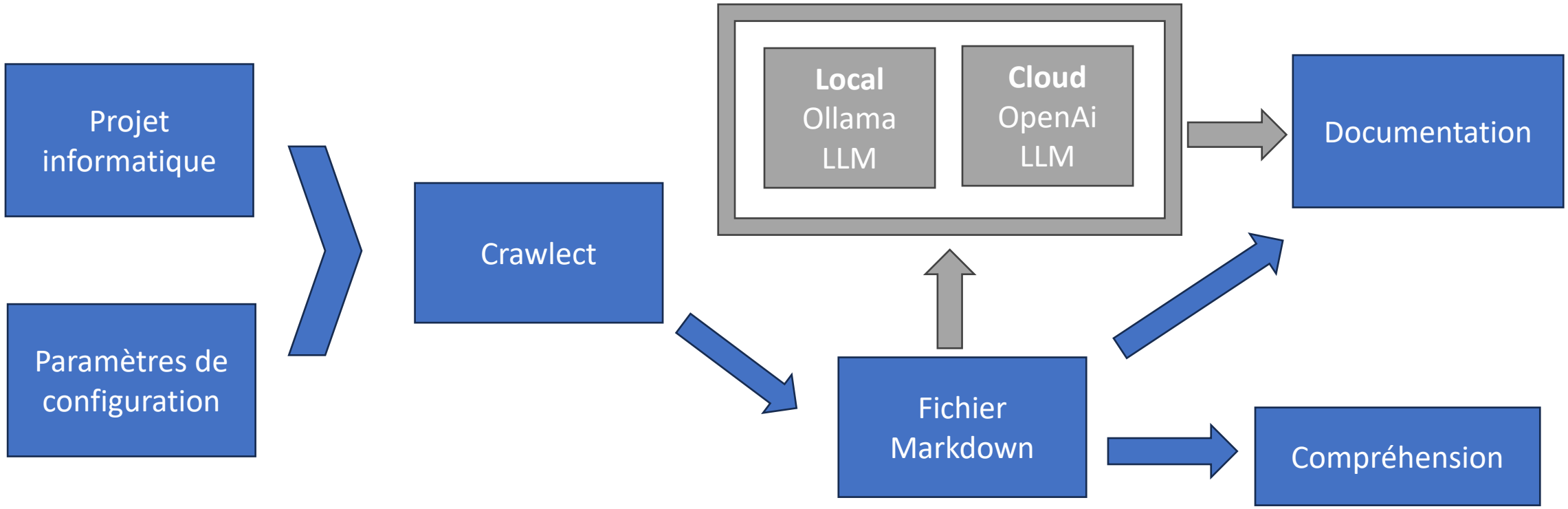
Motivations du sujet



- Reprise du projet de python
- Offrir une alternative à la cmd
- Faciliter l'accessibilité

Utilité de Crawlect

- Qu’est-ce que Crawlect ?



Apports de l’interface graphique

- Facilité d'utilisation.
- Flux d'utilisation plus rapide et moins d'erreurs.
- Remontée exhaustive et dynamique des paramètres disponibles.
- Répartition des responsabilités entre le cœur et l'interface.
  Les développements de *Crawlect* se concentrent sur les features alors que *Crawlect-GUI* se charge de la simplicité et de la validation.

Démonstration

haute école
neuchâtel berne jura

Démo

Principes de développement

- OOP
  - Encapsulation
  - Répartition des responsabilités
- Open/Closed
  - Faciliter l'extension
  - Eviter les modifications
- Autres:
  - Gestion des erreurs
  - Optimisation de processus couteux

Modèles de conception

Singleton et MVC:

- Model
  - ***CliSchemaParser*** : Partage des paramètres de l'application.
  - *CliOption* : model de paramètre.
  - *ComboItem* : Modélise les paires valeur / label pour les champs.
- View
  - ***MainWindow*** : Fenêtre principale.
  - *ShowMessages* : Affiche les message et alertes.
- Controller
  - ***MainController*** : coordination de (interaction, vue, logique).
  - ***PythonRunner*** : Prise en charge des appels Python.
  - ***Usersettings*** : Enregistrement des choix utilisateurs. *Glissant entre model et contrôle.
  - *CrawlectRunner :* formate et gère les appels à *Crawlect*.

Améliorations

- Allègement de la classe *MainWindow*,

  création d'une classe *UpdateOptionPallel* (amélioration MVC).

- Verrouillage de l'interface lors du traitement,

  avec une animation d'attente.

- Prévisualisation des *stderr* et *stdout*,

  dans une console embarquée.

- Rendre installable,

  tout en vérifiant automatiquement les prérequis et en installant les dépendances manquantes.

Perspective future

Crawlect-GUI

- Ajouter une prévisualisation du résultat dans une fenêtre
- Permettre à l'utilisateur d'exclure certain bout de code dans la prévisualisation
- Effectuer une ballade cognitive afin de valider ou modifier l'emplacement des boutons

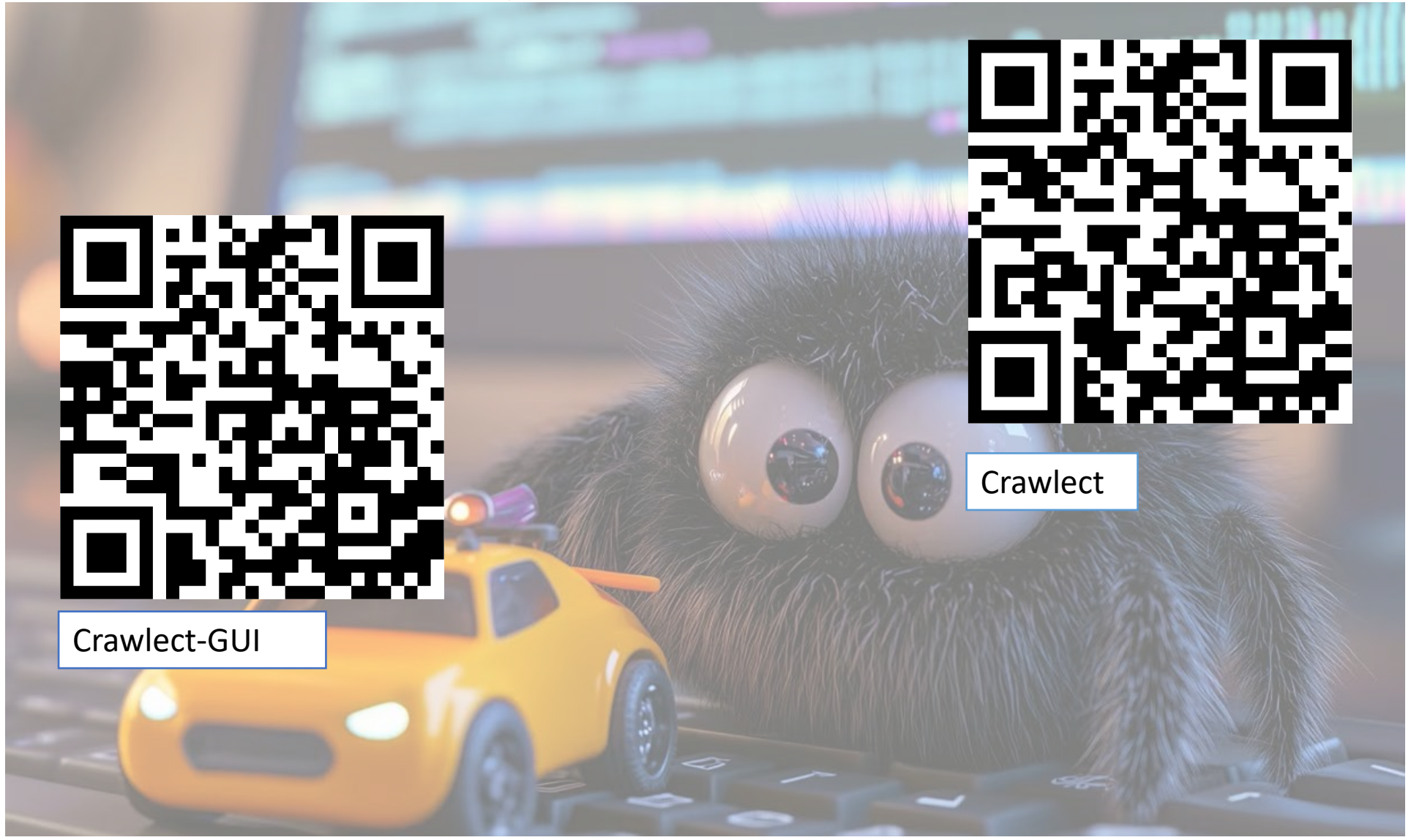
Crawlect + Crawlect GUI

- Afficher la proportion de chaque langage de code

Conclusion

- Architecture MVC
- Programmation orientée objet (OOP)
- Design Pattern Singleton
- Intégration Java / Python
- Rendu dynamique de l’interface graphique
- Parsing JSON avec Jackson
- Gestion des fichiers et validation des chemins
- Expérience utilisateur & design d’interface
- Sauvegarde persistante des préférences utilisateur
- Classes utilitaires et helpers
- Validation des entrées et gestion des erreurs
- Conception ouverte à l’extensibilité

Lien vers les repos



Merci de votre attention !

Avez-vous des questions

haute école
neuchâtel berne jura

Contact

yves.guillo@he-arc.ch

alexandre.jenzer@he-arc.ch