

Exercices autocorrigés avec la plateforme INGINious

Anthony Gego

UCLouvain (ICTEAM/INGI)

Numéro thématique 4 / 2024



RÉSUMÉ INGINious est une plateforme open-source et libre d'accès permettant la conception d'exercices corrigés automatiquement. Elle supporte les exercices de programmation de type code, par bloc, interactifs en ligne de commande, l'analyse de trace réseaux ou encore les formules mathématiques. Les réponses de l'étudiant sont évaluées à l'aide d'une logique de correction intégrée ou spécifiée manuellement au moyen de scripts et d'une interface de programmation. L'enseignant dispose d'une interface web pour gérer son cours et consulter les soumissions des étudiants afin d'améliorer le feedback des exercices. L'atelier vise à prendre en main de la plateforme, la création de séquences de cours sur base d'un catalogue d'exercices existants ainsi que la conception de nouveaux exercices.

MOTS-CLÉS • *exerciceur – juge en ligne – lms - correction automatique – oer – lti - programmation - ligne de commande – réseaux*

The screenshot shows the INGINious interface for an exercise titled "String - toUpper [*]". On the left, there is a sidebar with navigation options like "Administration", "Liste des cours", and "Informations". The main area contains the exercise description, which explains the goal: to implement a method `toUpper` that returns a string with all characters in uppercase. Below the description is a code editor with a Java snippet:

```
public static String toUpper(String s) {  
    // à implémenter  
}
```

 A green notification bar indicates that the user's response passed all tests with a score of 100.0%. At the bottom, there is a "Soumettre" button and a footer with copyright information for INGINious.

Figure 1 : Exemple d'exercice de programmation Java sur la plateforme INGINious

Objectif

Depuis quelques années, une large partie de la société fait usage des technologies numériques tous les jours. Cette transformation a provoqué une augmentation de la demande du secteur

sur le marché de l'emploi mais également de l'intérêt des étudiants pour des carrières dans l'informatique, augmentant drastiquement le nombre d'inscrits dans ces filières. Simultanément, l'apparition de cours en ligne massivement ouverts (MOOCs) a également élargi le nombre d'étudiants par enseignants.

Cet accroissement du public génère de multiples difficultés pour les enseignants. Tout d'abord, ils ne sont plus en mesure de fournir une correction de qualité des exercices de programmation. Des exercices de type QCM ou à réponse courte, souvent employés pour les larges audiences, sont bien trop insuffisants pour enseigner l'informatique. Ensuite, ils ne sont plus capables d'assurer le même suivi des étudiants, puisque l'équipe éducative ne passe pas à l'échelle.

Afin de répondre à cette problématique, l'UCLouvain a développé la plateforme *INGInious* qui fournit aux enseignants en informatique un socle commun de conception d'exercices auto-corrigés, modulaire, permettant de fournir aux étudiants une note et un feedback élaboré, quelque soit les technologies utilisées ou la méthode de correction. Par ailleurs, les données collectées par la plateforme permettent un meilleur suivi des étudiants et ainsi d'améliorer le feedback existant en ligne mais également de mieux préparer les séances de restructuration.

Description de l'activité

L'atelier consiste dans un premier temps à prendre en main la plateforme *INGInious* en tant qu'étudiant au travers des exercices libres d'accès déjà disponibles.

L'étudiant se voit proposer une liste de cours auxquels il peut s'inscrire. Chacun d'entre eux est composé d'une série d'exercices, constitués d'un ou de plusieurs sous-problèmes potentiellement interdépendants. Les réponses aux différents sous-problèmes sont enregistrés dans une soumission unique qui donne lieu à un *feedback* automatique sur mesure.

Dans un deuxième temps, il est proposé à l'enseignant de créer son propre cours à partir d'un jeu d'exercices existant. Il dispose alors d'une vue administrative au moyen de laquelle il peut suivre le parcours des étudiants et consulter leurs soumissions, pour comprendre l'origine des erreurs commises et aborder les points problématiques en classe avec les étudiants. Par ailleurs, il peut définir des publics, afin de segmenter son analyse en fonction des disparités, ainsi que des groupes de soumission, permettant d'associer un travail à plusieurs étudiants.

L'enseignant peut modifier la manière dont les exercices sont distribués et choisir parmi une forme structurée, une distribution aléatoire, un concours type ACM ICPC ou encore un examen. Il est également possible d'intégrer directement les exercices dans un LMS existant comme *Moodle* ou *Canvas* grâce au support de la norme LTI.

Dans un troisième et dernier temps, l'atelier met en lumière toutes les possibilités de la plateforme en proposant à l'enseignant la création d'exercices simples. Plusieurs niveaux de modularité permettent de répondre le mieux possible aux spécificités de la matière à évaluer :

- les types de problème. Chaque exercice est constitué d'un ou de plusieurs problèmes de type programmation par code ou par bloc (*Blockly*), choix multiples, réponse courte, formule mathématique ou analyse de trace réseau. Chaque problème dispose d'une méthode pour représenter la réponse de l'étudiant sous une forme traitable par la logique de correction, et éventuellement d'une routine de correction pré-embarquée ;
- agents de correction. La correction d'un exercice s'effectue par un agent sur base des routines de correction pré-embarquées (QCMs, formules mathématiques, traces

réseaux,...) ou sur base de scripts spécifiés manuellement. Ces derniers sont alors exécutés dans un environnement conteneurisé dans lequel le code de l'étudiant peut lui-même être exécuté de manière isolée. Une interface de programmation permet de récupérer les réponses de l'étudiant et de générer le feedback. Un mode interactif, via SSH, permet également de demander des opérations en ligne de commande à l'étudiant ;

- environnements de correction. Définis à partir d'images de conteneur, ce sont les logiciels nécessaires pour effectuer la correction d'une famille d'exercices. Il peut s'agir d'une collection de compilateurs par exemple. Ces images peuvent également embarquer une logique de correction commune à cette famille d'exercices ;
- fichiers de l'exercice. Chaque exercice peut être accompagné de fichiers en plus de sa description dans l'interface web. Il peut s'agir de fichiers accessibles publiquement servant à la description de l'exercice (images, jeux de données,...), des configurations pour l'environnement de correction (tests unitaires, solution, ...) ou d'une logique de correction spécifique à l'exercice, alors exécutée prioritairement sur celle fournie par l'environnement de correction.

La plateforme *INGInious* et l'ensemble des ressources présentées sont distribuées sous licence libre et open-source. Des mécanismes de distribution d'exercices et des types de problème supplémentaires peuvent être implémentés sous forme de *plugins*.

A l'issue de l'atelier, l'enseignant dispose de jeux d'exercices utilisables avec sa classe de manière autonome, aussi bien sur la plateforme publique que sur une instance privée, et peut se porter volontaire pour contribuer au corpus existant.

Expérimentations réalisées

La plateforme *INGInious* est utilisée à l'UCLouvain depuis 2014 dans le cadre des cours d'introduction à la programmation pour les étudiants en ingénierie et sciences informatiques. Au fil des ans, son utilisation s'est étendue à plus de cinquante cours, tant en informatique qu'en électronique, mathématique ou mécanique, motivée par le retour positif des étudiants.

En effet, la plateforme constitue une importante plus-value pour ces derniers. Elle ne remplace pas les séances encadrées de travaux pratiques mais les complète et prolonge virtuellement l'accompagnement. Les étudiants peuvent ainsi travailler par eux même les exercices qu'ils n'ont pas pu faire en séance ou des exercices supplémentaires en bénéficiant d'une expérience plus enrichissante qu'un solutionnaire qui n'aura pas suscité réflexion.

Chaque année, plus d'un million de soumissions sont enregistrées et plusieurs examens de programmation sont organisés sur la plateforme via *Safe Exam Browser*. Plusieurs outils ont été mis en place afin de limiter le plagiat pour les travaux à domicile, tels que l'intégration du détecteur de plagiat *JPlag*, la distribution ou la paramétrisation aléatoire d'exercices.

Distribuée sous licence libre et open-source, la plateforme est déployée dans plusieurs autres institutions et supporte la localisation, exercices y compris.

Plusieurs ressources, en partie conçues et testées en interaction avec des enseignants du secondaire, sont mises à la disposition de tous, notamment en mathématiques, avec des exercices de préparation aux études supérieures, et en informatique, avec des introductions à la programmation par bloc, *Python*, *Java*, ou *C* et à la ligne de commande.

Liens avec la recherche

La modularité de la plateforme *INGInious* permet aux enseignants et chercheurs de prototyper aussi bien de nouveaux types de problème que de nouvelles logiques de correction.

Un retour sur expérience de Derval et al. concernant l'utilisation de la plateforme pour le *MOOC edX Louvl.lx* présente l'architecture de celle-ci, ses performances et le choix d'une logique de correction basée sur des tests unitaires. De Velp et al. cherchent à trouver la configuration de Docker optimale pour son utilisation avec la plateforme.

Bonaventure et al. ont conçu un type de problème permettant l'analyse de traces réseaux de manière détaillée et d'en compléter des champs masqués avant de construire un ensemble de ressources, disponibles librement, dans le cadre d'un cours d'introduction aux réseaux.

Dans le cadre d'un cours introductif à la programmation en *Python*, Ortegat et al. présentent une logique de correction basée sur des tests à données aléatoires, implémentée à l'aide de la plateforme.

Plusieurs travaux de fin d'études ont également pu être menés autour de la plateforme, tant pour la conception de nouvelles ressources que pour le prototypage de nouveaux types d'exercices. L'ensemble des soumissions stockées par la plateforme à chaque essai de l'étudiant constitue également un jeu de données important pour l'analyse de traces d'apprentissage.

Liens vers les ressources et point de contact

N'hésitez pas à contacter Anthony GEGO pour plus de renseignements (Zulip).

Références

- Bonaventure, O., De Coninck, Q., Duchêne, F., Gego, A., Jadin, M., Michel, F., ... & Tilmans, O. (2020). *Open educational resources for computer networking*. ACM SIGCOMM Computer Communication Review, 50(3), 38-45.
- Derval, G., Gego, A., Reinbold, P., Frantzen, B., & Van Roy, P. (2015). *Automatic grading of programming exercises in a MOOC using the INGInious platform*. European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCs'15), 86-91.
- Ortegat, P., Vanderose, B., & Devroey, X. (2022, November). *Towards automated testing for simple programming exercises*. In Proceedings of the 4th International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (pp. 33-36).
- Piraux, M., Taffin, L., & Bonaventure, O. (2020). *Towards a Collection of Packet Trace Interactive Exercises for Computer Networking Education*. ACM SIGCOMM CCR Series on Networking Education.
- de Velp, G. E., Rivière, E., & Sadre, R. (2020, December). *Understanding the performance of container execution environments*. In Proceedings of the 2020 6th International Workshop on Container Technologies and Container Clouds (pp. 37-42).