

Nombre de participant·e·s maximal pour l'atelier	25	Nombre de sessions de 45mn que vous pouvez réaliser	2
Matériel nécessaire (vidéo-projection, grande salle, autre ...)	Vidéo-projection, Eduroam		

Introduction à Caseine, plateforme Moodle et communauté d'enseignants, pour l'apprentissage de l'informatique

Denis Bouhineau

Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

RÉSUMÉ Présentation de la plateforme Moodle « Caseine » pour l'enseignement de l'informatique. Dans un premier temps, un exemple d'utilisation élémentaire du plugin Moodle « Virtual Programming Lab » (VPL) est proposé pour fournir un exemple de mise en place d'une activité de programmation par un enseignant, son utilisation par un étudiant avec ses tests et l'évaluation automatique de son travail et le contrôle que l'enseignant peut effectuer au cours de cette activité. Dans un second temps, des usages avancés de VPL et de la plateforme pourront être montrés pour illustrer comment une communauté d'innovation et de partage a pu se construire et s'est actuellement constituée autour de Caseine.

MOTS-CLÉS • Virtual Programming Lab (VPL), Programmation, Évaluation automatique, Innovation, Partage.

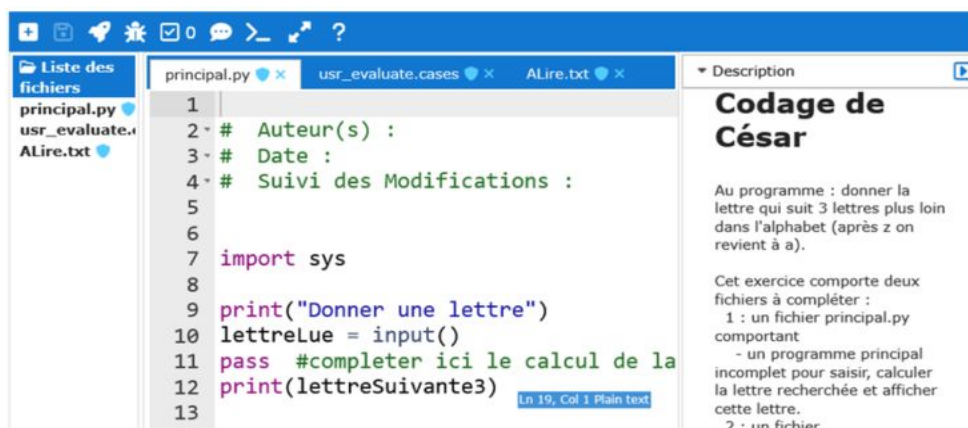


Fig. 1. Capture d'écran de l'environnement de travail de l'étudiant avec au centre le code à éditer, à droite l'énoncé, en haut, un bandeau avec les différents moments de son travail : édition, exécution, débogage et évaluation.

1. Déroulement de l'atelier

Deux objectifs seront poursuivis au cours de l'atelier : 1- montrer un exemple d'utilisation élémentaire de Moodle/VPL ; 2 – décrire et/ou démontrer des utilisations avancées de la plateforme à la base de la communauté d'innovation et de partage constituée autour de Caseine.

Ainsi, dans la première partie de l'atelier, trois temps de mise en place d'une activité élémentaire de programmation seront abordés lors d'une démonstration organisée de manière interactive pour que les enseignants présents à l'atelier puissent participer : création d'un exercice (par un enseignant-créditeur), utilisation de cet exercice en TP ou à distance (par un apprenant), suivi de l'activité de l'étudiant (par un enseignant).

Dans la seconde partie de l'atelier, selon le temps et les échanges, des usages avancés de VPL et de la plateforme pourront être montrés parmi ceux qui ont émergés, reprenant les différentes étapes de l'activité de l'apprenant et reposant sur des éléments structurels de la plateforme, pour illustrer comment une communauté d'innovation et de partage a pu se construire autour de Caseine. Ainsi, un plus ou moins grand nombre de ces éléments pourra faire l'objet d'une démonstration interactive, orientée innovation et/ou échange.

Consignes pour participer à l'atelier : avoir un exercice de programmation en tête, venir avec son ordinateur et s'inscrire sur Caseine (caseine.org).

2. Un mot sur Caseine, plateforme Moodle pour l'informatique

Caseine est une plateforme grenobloise active depuis 2014 pour enseigner l'informatique.

Initialement prévue pour l'université, elle est ouverte aux lycées et aux membres du réseau international eduGain. Reposant sur une infrastructure Moodle, elle intègre en particulier le plugin « Virtual Programming Lab » (VPL) permettant des activités de programmation avec évaluation automatique des productions étudiantes [1] et un suivi du travail étudiant par les enseignants.

Environ 10 000 étudiants issus d'une dizaine d'établissements l'utilisent régulièrement, au travers de plus de 200 cours [2].

3. Exemple de mise en place et pratique d'une activité élémentaire de programmation

Première partie de l'atelier, trois temps de mise en place d'une activité élémentaire de programmation sont abordés : création d'un exercice (par un enseignant-créditeur), utilisation de cet exercice en TP ou à distance (par un étudiant), suivi de l'activité de l'étudiant (par un enseignant).

Création d'une activité de programmation « élémentaire ». À minima, pour mettre en place une activité de programmation avec évaluation automatique VPL, l'enseignant -auteur de contenu- doit rédiger un énoncé, définir les fichiers requis pour réaliser son activité, avec éventuellement une trame de code pour préciser la structure globale, les bibliothèques autorisées et les entrées/sorties et fournir des cas de test entrées/sorties permettant une évaluation automatique. Souvent l'enseignant va au-delà et fournit également une solution, indique les métadonnées associées à son activité et vérifie si l'ensemble est cohérent (en particulier, si les tests valident bien la solution et que quelques erreurs courantes ne sont pas validées par ces tests. Ces différentes étapes peuvent être réalisées en restant dans l'environnement proposé par VPL, en un temps raisonnable (10-15 min). L'atelier pourra les expérimenter.

Travail de l'étudiant. Pour l'étudiant, l'activité commence avec la lecture de l'énoncé dans VPL puis un environnement de programmation, d'exécution, de débogage et d'évaluation de programme lui est proposé, voir Fig. 1. En particulier, tout au long de la phase d'édition, l'énoncé et les retours de différents temps de l'activité (compilation, exécution, évaluation) sont disponibles à droite du code proposé par l'étudiant. En général, l'activité se termine avec l'évaluation automatisée du code proposé par l'étudiant en reprenant les jeux de test proposés par l'enseignant et aboutie à une note. Selon les paramètres décidés par l'enseignant, cette évaluation peut être renouvelée ou pas par l'étudiant pour améliorer le code et la note. En prenant le rôle d'étudiant, les participants à l'atelier pourront expérimenter ces aspects.

Suivi de l'enseignant. Tout au long de l'activité, l'enseignant peut suivre le code sauvegardé (remis) par les étudiants, leur évaluation (ou effectuer une évaluation automatique enseignante), exécuter le code (dans un environnement enseignant), intervenir sur le code (dans la version étudiante ou une version enseignante) et éventuellement recherche des similitudes entre les codes proposés par les étudiants (c'était l'une des origines du projet VPL). Pour son suivi, des éléments de tableau de bord sont proposés à l'enseignant. Pour l'atelier, la démonstration de la mise en place d'une activité de programmation élémentaire se terminera donc par cette partie.

4. Pratiques avancées, communauté d'innovation et de partage

Plusieurs éléments ont permis la naissance d'une communauté d'innovation et de partage autour de la plateforme Caseine, d'une part des éléments propres à VPL (paramétrisation par script et éléments d'architecture matérielle et logicielle), d'autre part des développements propres à Caseine (plugins spécifiques pour VPL ou plugins généraux ; support spécifique aux innovations technologiques et au partage).

Paramétrisation par script. Chaque moment de l'activité élémentaire présentée dans la partie 1 (compilation, exécution, débogage, évaluation automatique) est défini dans la plateforme par un script VPL (.sh) mis à disposition des enseignants-créeurs pour l'adapter (et/ou le détourner pour l'adapter) à l'activité souhaitée. C'est ainsi que l'enseignant peut facilement ajouter un nouveau langage de programmation (dans la mesure où celui-ci est disponible sur le serveur d'exécution), ajouter une phase d'analyse statique du code et/ou d'exécution avec contrôle des ressources, remplacer l'évaluation par comparaison des entrées/sorties des tests par une comparaison sur des exécutions pseudo-aléatoires avec un code solution, etc.

Éléments d'architecture matérielle de VPL. L'architecture matérielle de VPL prévoit l'exécution de l'éditeur de code en client léger et les phases d'exécution, compilation, débogage, évaluation sur des serveurs linux (console ou graphique) en mode bac à sable éphémère (jail-server). L'aspect bac à sable limitatif, pour des raisons de sécurité, est contrebalancé par le support linux sous-jacent. Ainsi, le serveur linux peut servir de rebond pour intégrer une application de dessin de circuit logique (Logisim), utiliser un conteneur Docker, lancer un serveur Jupyter, etc. dans Moodle. VPL introduit un pont entre Moodle et Linux.

Éléments d'architecture logicielle de VPL. L'architecture logicielle de VPL permet le partage des scripts de configuration des activités élémentaires (compilation, exécution, débogage, évaluation automatique) par un mécanisme d'héritage (based on). Ainsi, les efforts de transformation des tests d'entrées/sorties en tests unitaires, ou d'intégration d'un LLM local, peuvent ainsi être partagés simplement entre enseignants (dans la mesure où leur auteur les partage, bien entendu)

Plugins Caseine spécifiques à VPL. Parmi les plugins spécifiques à VPL, citons Q-VPL (pour ajouter des questions de programmation dans un QCM Moodle), Défi de code VPL (pour organiser un concours de programmation), Vpl-For-VSCode, Moodle VPL Plugin for Eclipse, ... (pour rester dans son IDE préféré), l'écosystème iVPL, avec VplAdmin, VplOrg, etc.

Plugins Caseine pour le partage. Divers plugins, paramétrages ou développements ont également été mis en place pour permettre et faciliter le partage : définition de métadonnées pour les activités et ressources disponibles sur Caseine, espace de recherche de ces activités et ressources, cours ouverts, visitables et partageables pour observer et collecter in situ des activités et ressources via le panier de partage d'activités et de ressources pouvant aller de cours en cours.

Support Caseine à l'innovation technologique et au partage. Notons que nombres des développements spécifiques décrits ici n'ont été possibles que grâce au support fourni par Caseine pour le développement du code et l'ajout de packages sur les serveurs.

5. Innovation et vie de la communauté, retour d'expérience ?

Au jour le jour, un client de messagerie instantanée (Mattermost) est utilisé pour mettre en relation la communauté autour de Caseine (et les ingénieurs supports). Un espace enseignant (Tutoriels), des cours ouverts et partagés pour des enseignants visiteurs et une librairie de vidéos de présentations sont également disponibles.

Enfin, régulièrement, des moments de rencontre sont organisées ; à distance des 1/2h ou en présence des 1/2 journées. À l'occasion des 10 ans de la plateforme, un anniversaire a réuni une cinquantaine de participants, dont l'auteur espagnol de VPL.

Remerciements. Caseine est financée par l'UGA et Grenoble INP.

6. Références

1. Rodríguez-del-Pino, J. C., Rubio-Royo E., Hernández-Figueroa, Z. (2012) *A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features*. In proceedings of the 2012 International Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government.
2. Bizard, A., Bouhineau, D., Brauner, N., Cambazard, H., Catusse, N., Cortes Cornax, M., Lagoutte, A., Lemaire, P., Maillot, Y., Thiard, F. (2022) : *Accompagner l'enseignement de l'informatique avec Caseine*. In 1024 : Bulletin de la Société Informatique de France, No 20, pp.107 - 122.