



# Apprentissage augmenté: le numérique comme outil d'aide à l'apprentissage

Nicolas Stouls, Oscar Carrillo, Julien Ponge, Frédéric Le Mouël, Alexandre Claude

## ► To cite this version:

Nicolas Stouls, Oscar Carrillo, Julien Ponge, Frédéric Le Mouël, Alexandre Claude. Apprentissage augmenté: le numérique comme outil d'aide à l'apprentissage. 5è colloque " Pedagogie et Formation " Inter INSA, Mar 2017, Lyon, France. hal-01494867

**HAL Id: hal-01494867**

**<https://hal.inria.fr/hal-01494867>**

Submitted on 24 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Apprentissage augmenté :  
le numérique comme outil d'aide à l'apprentissage**

Nicolas Stouls, Oscar Carrillo, Julien Ponge, Frédéric Le Mouël et Alexandre  
Claude

(Prénom.Nom@insa-lyon.fr)

Univ Lyon, INSA Lyon, CITI, F-69621 Villeurbanne, France

**Résumé**

Dans le monde actuel, où la concentration est de plus en plus volatile, il devient important de permettre une meilleure immersion de l'apprenant afin d'exploiter son papillonnage dans le cadre de son apprentissage. Différentes démarches déjà. Dans cette communication, nous proposons une approche instrumentée numériquement, ayant pour but d'augmenter l'implication de l'apprenant. Si la *réalité augmentée* permet de superposer un monde virtuel par dessus la vision du monde réel, alors nous pourrions appeler *enseignement augmenté* un enseignement où l'étudiant a une interaction enrichie avec le monde réel.

**Mots-clés :**

Numérique, apprentissage, interaction.

## I INTRODUCTION

Commençons par une analogie télévisuelle. Aujourd'hui, regarder la télévision seule ne suffit plus. Les nouvelles générations utilisent les réseaux sociaux en *live* pour augmenter leur expérience, ou bien regardent leur programme en différé. Dans le premier cas, les échanges permettent d'augmenter l'expérience utilisateur du programme télévisé (lire qu'un autre spectateur a remarqué un personnage d'arrière plan qui a fait une bêtise, ou une incohérence dans l'histoire, permet de voir son divertissement sous un autre jour), tandis que, dans le second cas, regarder un programme en différé permet de choisir le moment le mieux adapté par rapport à son niveau de fatigue ou d'attention.

Nous pensons que, de la même manière, si on arrive à motiver un étudiant à réagir en *live* durant un cours, alors cela peut augmenter son implication et son appropriation. Ainsi, commenter ou se questionner par rapport au contenu du cours, devrait permettre soit d'avoir une réponse dès que possible, soit faire naître un questionnement dans l'esprit de ses camarades. Par ailleurs, nous pensons que permettre à un étudiant de faire des exercices quand il le souhaite est un point important pour l'aider à progresser. Mais il est alors nécessaire que l'étudiant ait un retour rapide sur sa production. Pour ce faire, nous nous sommes inspirés des MOOCs, dont l'un des enjeux forts est de conserver l'assiduité des apprenants et leur implication.

Nous proposons ici un petit retour d'expérience sur l'usage, dans un contexte d'enseignement traditionnel, de techniques utilisées dans les MOOCs. L'objectif est alors d'*augmenter* cet enseignement et de gagner en dynamisme, en implication dans la formation, en autonomie de l'apprenant et en contraintes de travail personnel. Les deux principales contributions, détaillées dans les deux sections suivantes, sont l'usage d'un canal de communication durant les amphis, et la réalisation d'une plateforme de correction automatique d'exercices fournissant des commentaires sur le travail fourni. Nous nous focalisons sur le cas particulier de l'initialisation à la programmation Java.

## II SALON DE COMMUNICATION (CHAT)

Il est vrai que l'attention dans un amphi peut être regagnée par l'usage de clickers et que cela est reconnu par les étudiants (Stevens et al., 2016), cependant leur usage peut impliquer des contraintes d'organisation et de temps.

Une autre approche consiste à utiliser Twitter (Othman et al., 2016) comme moyen d'échange. Ce qui est très intéressant, au delà de sa simplicité d'usage, c'est la possibilité d'exprimer une question ou une remarque. Mais le manque d'anonymat peut limiter son usage.

Une méthode entre les deux consiste à utiliser un canal de communication de type *Chat* sur les smartphones des étudiants pour favoriser les questions et l'interaction (Ball, 2016). Les apprenants redeviennent anonymes et sont motivés à pouvoir poser des questions en toute discrétion. Questions qui sont indispensables à l'enseignant pour appréhender en direct le niveau de compréhension de son auditoire et ainsi adapter le niveau de son discours.

Dans le cadre de l'enseignement PC-S1-IF-EM, l'outil Slack<sup>1</sup> a été utilisé comme canal pour poser des questions durant le cours. Le retour des étudiants a été relativement positif. Bien que seuls 30% des étudiants s'y soient connectés, seuls 5% des étudiants trouvent cela « inutile » et 29% y sont « indifférents ». Les 66% restants s'échelonnent depuis « il est rassurant de savoir cela disponible » jusqu'à « je trouve cela utile ».

Cet outil a commencé à être utilisé pour des échanges autres que les questions durant le cours (questions lors d'un travail en autonomie ou recherche de salle) et nous pensons que son usage ne demande qu'à être encouragé. Pour ce qui est des questionnaires, il est possible d'installer une application comme *Jeeves Poll*<sup>2</sup> pour réaliser des questionnaires remplaçant les clickers. Cette dernière étape n'a pas encore pu être testée in-situ.

Cet outil a également été déployé auprès des étudiants de TC, où des salons privés ou publics permettent d'interagir avec camarades ou enseignants.

### III EXERCICES AUTO-CORRIGÉS

Ces dernières années, les QCMs formatifs se multiplient au PC de l'INSA Lyon, dans différentes disciplines. Cette approche est très appréciée par les étudiants, qui peuvent avoir une correction de leurs résultats, généralement avec quelques commentaires. Cependant, ces évaluations sont généralement cantonnées à du contrôle de connaissance et n'ont pas la même forme que les évaluations notées. Plutôt que de changer la forme des évaluations pour aux QCMs d'entraînement, nous pensons qu'il faut réussir à proposer aux étudiants des exercices auto-corrigés et commentés qui soient plus proches des compétences attendues.

La correction automatique d'exercices avec génération de commentaires n'est pas une spécificité des cours de programmation. Cela est déjà utilisé dans les MOOCs, où différentes techniques existent, depuis la correction par les pairs (Huisman et al., 2016), jusqu'à la correction automatisée avec apprentissage (Pu et al., 2016).

Pour la réalisation d'exercices auto-corrigés en programmation, il existe différentes approches, mais le danger principal est l'exécution d'un code malveillant sur la plateforme. Ainsi, on pourra mettre en opposition les solutions qui se basent sur l'usage d'un sous-langage de programmation, interprété par l'outil, des propositions permettant d'écrire des programmes réels.

Le premier cas semble beaucoup plus simple pour gérer la progression des apprenants et les problèmes de sécurité, et existe notamment dans l'outil PLM (Quinson et al., 2015) ou algobox (Ponsonnet, 2011), tandis que la seconde est potentiellement plus stimulante pour les étudiants, car elle met en oeuvre un « vrai » langage de programmation. On peut notamment citer VPL (Thiébaud, 2015), qui permet d'exécuter un programme, soumis depuis Moodle, sur un serveur « bac à sable ». Cet outil très intéressant n'est cependant plus supporté.

Dans le cadre d'un projet étudiant, nous sommes actuellement en train de finaliser une première bêta d'un outil dont l'objectif est d'être installable sur n'importe quelle Raspberry-Pi et de permettre une exécution sécurisée de tout programme Java. L'exécution de cas de tests choisis parmi une famille de tests et associés à des catégories de messages permettra un retour clair vers l'étudiant, permettant de le guider, sans lui donner une solution toute faite. Une expérimentation est espérée au second semestre 2016-2017 au sein de du module PC-S1-IF-EM.

---

<sup>1</sup> <https://www.slack.com>

<sup>2</sup> <https://jeevespoll.com>

## IV AUTRES RESSOURCES ET OUTILS PLUS CLASSIQUES

### IV.1 Supports de cours synthétiques

Il peut être considéré comme important par les enseignants de fournir les diaporamas d'amphi, ainsi qu'un poly. Selon les démarches, ceux-ci peuvent d'ailleurs être laissés à trous pour forcer l'apprenant à suivre durant le cours et à identifier d'éventuels problèmes de compréhension.

Cependant, nos cours ont trop souvent le défaut de contenir le fond, ainsi que les explications, les anecdotes, les démonstrations et parfois les exercices d'application, ce qui en dilue la substance. Diffuser des supports mettant déjà en avant les points les plus importants ou des fiches déjà synthétisées (pouvant faire des renvois vers un support plus complet) semble un moyen intéressant d'aider l'étudiant dans sa démarche d'apprentissage des *connaissances* et de sa *compréhension* (Taxonomie de Bloom).

Cependant, à l'heure de youtube, certains apprenants préfèrent le format vidéo plutôt que le format du poly classique. De nombreux enseignants sont déjà auteurs de capsules vidéos bien appréciées par les apprenants. Cependant, contrairement aux « bonnes pratiques » parfois décrites, nous pensons que le modèle « youtube » est plus motivant : en plus d'un format court (chaque vidéo dédiée à un point précis du cours), il semble important de maintenir un certain rythme dans la vidéo, permettant à l'apprenant de ne pas avoir le temps de se disperser. Il doit être complètement immergé. Ayant la vidéo, il pourra rembobiner si nécessaire.

### IV.2 ENT / LMS

Un environnement numérique de travail adapté est souvent un atout intéressant pour rompre la linéarité de l'enseignement et permettre un parcours plus à la carte. Un LMS tel que Moodle permet ainsi de définir des supports qui apparaissent sous condition d'atteinte d'un certain résultat. Ainsi, il est possible de préfixer chaque exercice avec un petit QCM vérifiant le bagage théorique de l'apprenant (quitte à le forcer à réviser) avant de commencer un exercice. Par exemple, il serait possible de faire un programme de révision de vacances, à travailler en autonomie.

Ainsi, afin de permettre plus de travail en autonomie semi-encadré, arrêtons d'utiliser Moodle comme un support de PDF, et exploitons enfin ses capacités d'adaptation, déjà vantées dans les occurrences antérieures de ce colloques. Les difficultés sont alors qu'il faut mettre à disposition des apprenants des supports allant au delà du programme et qu'il faut utiliser Moodle, qui n'est pas la plateforme la plus ergonomique du monde.

## V BILAN

Tableau 1 : Bilan des usages numériques décrits et permettant une immersion de l'apprenant

Outils	CM	TD/TP	Travail personnel
Clickers	Permet de questionner la salle sur un point précis.	Peu adapté	
Canal de communication	Interactions avec l'enseignant ou entre apprenants. Permet de questionner la salle.	Interactions entre les étudiants ou avec l'enseignant.	
Exercices corrigés automatiquement		Utilisables pour occuper les plus rapides ou pour choisir des exercices plus simples pour les étudiants en difficulté. Fourni des retours commentés.	Apprentissage personnel. Utilisable pour faire des DMs à feedback rapide.
ENT/LMS		Permet de rompre la linéarité des exercices et d'avoir un enseignement mieux adapté au niveau de chacun. Porte d'entrée pour les révisions.	
Supports synthétiques		Utilisés pendant la séance pour aider à décoincer sur certains exercices	Permettent d'aider à la révision d'un amphi.

Le tableau 1 résume les différents éléments cités dans cette communication et rappelle ses principaux cas d'usage, en fonction de la modalité pédagogique.

Les principales contributions sont notamment le développement d'un outil de publication d'exercices auto-correctés bienveillant (ne mettant pas les autres apprenants en compétition et fournissant des retours constructifs), ainsi qu'un mode de communication apprenants-enseignant basé sur une application de communication par canal, permettant des échanges synchrones et asynchrones, ainsi que l'installation d'applications de sondage.

## VI RÉFÉRENCES

- Ball, M. (2016). Using instant chat for fun and for profit to run a large class. *In Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 495–495. ACM.
- Huisman, B., Admiraal, W., Pilli, O., van de Ven, M., & Saab, N. (2016). Peer assessment in MOOCs : The relationship between peer reviewers' ability and authors' essay performance. *British Journal of Educational Technology*.
- Othman, T., Royan, A., & Fazal, N. (2016). Use of twitter in classroom attendance and participation in a microbiology & immunology course. *The Journal of Immunology*, 196(1 Supplement) :130–2.
- Ponsonnet, L. (2011). Initiation à l'algorithmique et à la programmation pour le lycée. *Ellipses Paris*.
- Pu, Y., Narasimhan, K., Solar-Lezama, A. & Barzilay, R. (2016). Sk\_p : A neural program corrector for moocs. *In Companion Proceedings of the ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications : Software for Humanity, SPLASH Companion*, pages 39–40.
- Quinson, M. & Oster, G. (2015). A Teaching System To Learn Programming : the Programmer's Learning Machine. *In ACM Conference on Innovation and Technology in Computer Science Education 2015*.
- Stevens, D., & Fontenot, G. (2016). Measuring clicker impact on student perceptions of course and instructor. *International Journal of Innovation and Learning*, 21(1) :21–34.
- Thiébaud, D. (2015). Automatic evaluation of computer programs using moodle's virtual programming lab (vpl) plug-in. *Journal of Computing Sciences in Colleges*, 30.6 :145–151.