

Codeclick : Rétroactions Personnalisées Basées sur les *Learning Analytics* pour Supporter l'Apprentissage Pratique de la Programmation

Zahi Hodeib^{1,2} (2^{ème} année de Thèse)

¹ Université de Lille, CNRS, UMR 9189, CRISTAL, F-59000 Lille, France

² Université Libanaise, faculté de technologie, RADIOCOM, Saida, Liban
zahi.hodeib@univ-lille.fr

Résumé. Cet article présente un nouvel environnement d'apprentissage de la programmation nommé Codeclick. L'environnement a pour objectif de fournir des rétroactions personnalisées et des recommandations destinées à améliorer l'apprentissage des novices. Il s'appuie sur une approche statistique de *Learning Analytics* basée sur les traces numériques des apprenants. Codeclick fournit des visualisations pour permettre aux apprenants de surveiller leurs niveaux de performance. Ces visualisations permettent également à l'enseignant d'identifier les problèmes de blocage et intervenir au moment opportun. Les résultats préliminaires de l'utilisation de Codeclick montrent une amélioration de la performance de chaque apprenant, en termes de nombre d'erreurs commises et de temps passé par activité.

Mots-clés. Programmation, rétroaction, analyse de l'apprentissage, EIAH.

1 Introduction

Au cours des vingt dernières années, l'apprentissage de la programmation est devenu essentiel dans le cursus de nombreuses filières académiques. Cependant, son apprentissage pratique représente une difficulté pour les novices [1]. Cela explique clairement les taux élevés d'échecs et d'abandon, surtout dans les cours d'initiation à la programmation [2]. Généralement, ces activités sont caractérisées par une densité d'apprenants, ce qui augmente la charge de travail des enseignants pour assurer un bon suivi [3]. Par conséquent, ces derniers seront incapables d'accompagner l'apprenant, en situation de blocage de code, en lui fournissant des rétroactions personnalisées. D'autre part, les Environnements Informatiques pour l'Apprentissage Humain (EIAH) se sont développés largement ces dernières années. Certains de ces systèmes offrent aux apprenants des rétroactions. Cependant, la qualité de ces rétroactions reste non satisfaisante vu qu'elles sont basées sur l'analyse d'une faible quantité d'informations collectées sur les apprenants [4]. Par conséquent, le contenu de ces rétroactions ne reflète pas clairement le niveau de performance atteint par l'apprenant.

Par ailleurs, l'immense quantité d'information, nommée également traces numériques des apprenants, résultante de l'interaction de ces derniers avec les plateformes éducatives, a ouvert la voie au développement de *Learning Analytics (LA)*. Les *LA* visent principalement à exploiter efficacement ces traces pour améliorer l'apprentissage [5]. La section suivante présente des travaux effectués pour exploiter ces traces dans l'objectif de supporter l'apprentissage pratique de la programmation.

2 État de l'Art

Selon la première conférence sur les *LA* intitulée *LAK (Learning Analytics and Knowledge conference, 2011)*, les *LA* ont été définies comme la mesure, la collecte, l'analyse et la communication de données sur les apprenants et leurs contextes, dans le but de comprendre et optimiser l'apprentissage et l'environnement dans lequel il se déroule [6]. Autrement dit, les *LA* utilisent les traces numériques des apprenants pour fournir aux enseignants des moyens permettant de réagir via des rétroactions [7]. Quelques approches utilisées par les *LA* d'après la littérature, la prédiction, les visualisations via des tableaux de bord et des techniques statistiques. De nombreux, outils, applications web et laboratoires intelligents utilisent les traces numériques pour assister les apprenants. Le Tableau 1 présente certaines de ces technologies.

Tableau 1. Travaux proposant des rétroactions basées sur les traces numériques

Technologies	Publications	Rétroaction	Données collectées
Applications web	WebToTeach [8]	Alertes et Messages	Code
	Data2U [9] CodeWrite [10]	Tableau de bord Messages Instantanés	Interactions Code, Interactions
Outils	Exercism Retina [11]	Messages Recommandations	Code Temps et Erreurs
	Course Signal [12]	Rapport Email Personnalisée	Historique, Notes, Interactions
Laboratoires Intelligents	Smart Lab [13]	Visualisations Personnalisées	Interactions
	Lab4CE [14]	Messages Visualisations	Code, Interactions

Généralement, ces systèmes collectent différentes informations sur les apprenants. Cependant, beaucoup de ces systèmes fournissent des rétroactions sans tenir compte du besoin personnalisé de l'apprenant. De ce fait, l'apprenant se retrouve envahit par l'énorme quantité de rétroactions reçues, qu'il n'arrive pas à comprendre et reste bloqué dans son activité. Dans cet article, nous visons à identifier l'influence des rétroactions personnalisées et des visualisations générées via une approche *LA* sur l'apprentissage pratique de la programmation.

3 Présentation de Codeclick

Codeclick est un environnement d'apprentissage de la programmation, qui propose des activités destinées aux novices apprenants. Chaque activité est composée d'un ensemble de notions à évaluer. Il offre deux modes d'entraînement: Avec Rétroactions (AR) ou Sans Rétroactions (SR). Codeclick collecte et analyse les traces numériques des apprenants (erreurs, nombre de clics, temps passé par activité), les classe et les stocke dans une base de données. Ces traces subissent ensuite une phase d'analyse statistique basée sur des indicateurs pour évaluer la performance de chaque apprenant selon la formule (1) et par suite générer des rétroactions et des visualisations illustrant ce niveau de performance.

$$(1) \quad P = 100 - 2.5 * NC - 3 * NE - T$$

Avec
NC : Nombre Clics
NE : Nombre Erreurs
T : Temps en minutes/activité

Dans cette formule, la performance de chaque apprenant est associée à trois indicateurs collectés de l'activité, à savoir le Nombre d'Erreurs (*NE*), Nombre de Clics (*NC*) et le temps par activité (*T*). Nous avons attribué une pondération à chaque indicateur en fonction de son importance dans la réalisation de l'activité. Au cours de la réalisation d'une activité donnée Codeclick calcule la performance de l'apprenant et génère une rétroaction personnalisée. Il est inspiré des travaux de [11][12][13] avec une originalité des indicateurs.

4 Expérimentation et Résultats

Nous avons mené une première expérimentation sur deux groupes d'apprenants, formés chacun de sept membres, sélectionnés arbitrairement et pour la même activité. La durée de l'activité était fixée à 10 minutes et les apprenants étaient uniquement des garçons avec un âge moyen de 19 ans. Le groupe G1 a réalisé l'activité en mode AR et le groupe G2 utilise le mode SR. Notre but est d'évaluer le niveau de performance des apprenants afin de fournir des rétroactions personnalisées. Pour ce faire nous nous sommes appuyés sur des statistiques basées sur des indicateurs (*NE*, *NC*, *T*). Ces statistiques sont utilisées pour générer des rétroactions et visualisations. Les résultats préliminaires montrent que les apprenants du groupe G1 ont atteint des niveaux de performance plus satisfaisants en termes d'erreurs commises et de temps passé par activité (Fig.1).

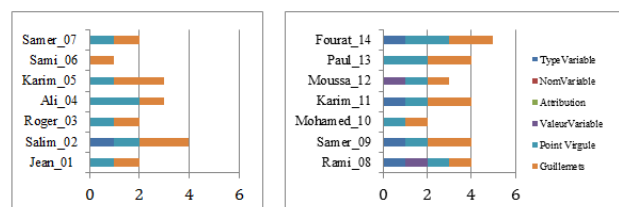


Fig. 1. Visualisations représentant le nombre et types d'erreurs des groupes G1 et G2

5 Conclusion

L'apprentissage pratique de la programmation nécessite la mise en place de systèmes capables de générer des rétroactions personnalisées basées sur les traces numériques des apprenants. Dans cet article, nous avons présenté l'environnement Codeclick, qui s'appuie sur une approche statistique de *LA* pour analyser les traces des apprenants afin de fournir des rétroactions personnalisées. Les résultats préliminaires montrent que les apprenants privilégiés par des rétroactions atteignent des niveaux de performance plus satisfaisants en termes d'erreurs de code et de temps passé par activité.

Références

1. Lahtinen E, Ala-Mutka K, Järvinen H-M. A Study of the Difficulties of Novice Programmers. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ITiCSE '05. New York, NY, USA : ACM, 2005 : 14–18.
2. Bennedsen J, Caspersen ME. Failure Rates in Introductory Programming. *SIGCSE Bull.* 2007 ; 39 : 32–36.
3. Gulwani S, Radiček I, Zuleger F. Feedback Generation for Performance Problems in Introductory Programming Assignments. *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. FSE 2014. New York, NY, USA : ACM, 2014 : 41–51.
4. Ferguson P. Student perceptions of quality feedback in teacher education. *Assessment & Evaluation in Higher Education* 2011 ; 36 : 51–62.
5. Pardo A, Dawson S, Dawson S. Learning Analytics: How Can Data Be Used to Improve Learning Practice? *Measuring and Visualizing Learning in the Information-Rich Classroom* 2015 ;
6. Siemens G, Long P. Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review* 2011 ; 5 : 30–32.
7. Villamañe M, Alvarez A, Larrañaga M. Supporting competence-based learning with visual learning analytics and recommendations. *2018 IEEE Global Engineering Education Conference (EDUCON)*. 2018 : 1572–1575.
8. Arnow D, Barshay O. WebToTeach: an interactive focused programming exercise system. *FIE '99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No.99CH37011)*. 1999 : 12A9/39-12A9/44 vol.1.
9. Khan I, Pardo A. Data2U: scalable real time student feedback in active learning environments. 2016 : 249–253.
10. Denny P, Luxton-Reilly A, Tempero E, *et al.* CodeWrite: supporting student-driven practice of java. *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11*. Dallas, TX, USA : ACM Press, 2011 : 471.
11. Murphy C, Kaiser G, Loveland K, *et al.* Retina: Helping students and instructors based on observed programming activities. 2009 : 178–182.
12. Arnold KE, Pistilli MD. Course signals at Purdue: using learning analytics to increase student success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*. Vancouver, British Columbia, Canada : ACM Press, 2012 : 267.
13. Alammery A, Carbone A, Sheard J. Implementation of a Smart Lab for Teachers of Novice Programmers. 2012 ; 123 : 10.
14. Broisin J, Venant R, Vidal P. Lab4CE: A Remote Laboratory for Computer Education. *International Journal of Artificial Intelligence in Education* 2017 ; 27 : 154–180.