

Atelier : Utilisation et création d'exercices de programmation visuelle sur une plateforme d'auto-correction

Olivier Goletti^{1,3}[0000-0002-1610-4985] et Kim Mens¹[0000-0003-0303-1630]

¹ ICTEAM/INGI, UCLouvain, Louvain-la-Neuve, Belgique
firstname.lastname@uclouvain.be

² LIACS, Leiden University, Leiden, Pays-Bas

Résumé L'apprentissage de la programmation, visuelle ou non, passe par la réalisation d'exercices. Cependant, la création d'exercices de programmation visuelle avec auto-correction n'est pas à la portée de tous les enseignants. Cet atelier présente la plateforme d'auto-correction INGI-nious et son support de séquences d'exercices de programmation visuelle. Des séquences d'exemples seront proposées pour illustrer et prendre en main l'outil. L'utilisation de l'exerciceur INGI-nious permettra aux enseignants de profiter d'un outil robuste pour proposer des exercices variés dans leur enseignement de la programmation visuelle.

Mots clés : enseignement de l'informatique · programmation visuelle · exerciceur · INGI-nious · auto-correction

1 Introduction

Comme l'enseignement de l'informatique prend de l'ampleur [4], notamment au collège en France [3] et arrive en Belgique francophone [4], les enseignants de ces matières vont avoir recours à des outils et des exerciceurs pour enseigner l'informatique.

2 Contexte atelier

Dans cet atelier, on propose de faire découvrir aux participants plusieurs séquences d'exercices de programmation visuelle intégrées à l'outil INGI-nious [1]. Cet exerciceur très versatile a été conçu pour l'auto-correction de multiples types d'exercices [2,3]. Depuis peu, les exercices de programmation visuelle sont disponibles grâce à l'intégration de la librairie Blockly [5] comme extension d'INGI-nious [6].

3. Programme scolaire en Technologie et Mathématique au Collège

4. Référentiel FMTTN

5. <https://developers.google.com/blockly/>

6. <https://github.com/UCL-INGI/INGInious-problems-blockly>

La structure générale d'un cours, constitué d'exercices rassemblés en sections, sera expliquée. Les exercices déjà disponibles sur la plateforme seront illustrés. Les différentes options de customisation d'une tâche de programmation visuelle sur INGINIOUS seront montrées et expliquées.

A titre d'illustration, les figures 1 et 2 montrent l'interface d'un exercice de programmation visuelle sur INGINIOUS. La figure 1 montre un exemple reproduit de Code.org [7] avec des illustrations différentes. Cet exercice avec une animation est principalement une démonstration de ce qu'on peut faire avec INGINIOUS. L'exerciceur permet en plus de créer des exercices génériques. L'apprenant peut donc tenter de résoudre en local (dans son navigateur web) une instance du labyrinthe et soumettre sa solution. INGINIOUS peut ensuite proposer une autre instance qui représenterait un cas de test pour lequel la solution n'est pas correcte.

La figure 2 illustre qu'on peut avancer dans les concepts de programmations sans nécessairement travailler avec une animation ludique. Il s'agit ici de travailler la manipulation des chaînes de caractères dans un contexte de bioinformatique. Cette variété d'exercices permet de s'adapter à différents publics.

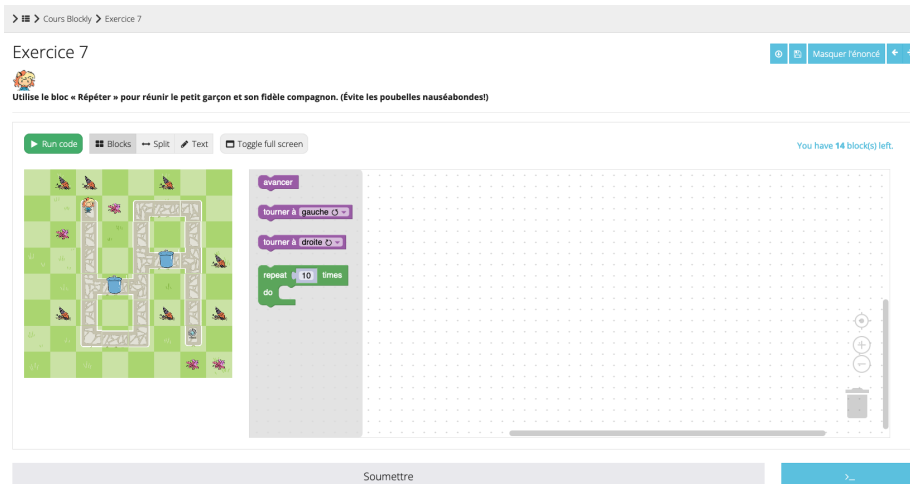


FIGURE 1. Exercice de déplacement dans un labyrinthe avec une animation liée au code.

Outre le contenu déjà disponible, on familiarisera les participants avec l'interface administrateur et l'utilisation générale qu'un enseignant est amené à devoir maîtriser pour utiliser l'outil. Cette première utilisation de la plateforme INGINIOUS permettra également de recueillir d'éventuels besoins pour des nouvelles fonctionnalités.

7. <https://studio.code.org/s/course3/lessons/2>

The screenshot shows the INGINIOUS interface for an exercise titled "Analyse de chaînes de caractères". The page includes a sidebar with navigation options like "Liste des cours", "Informations", "Administration", "Pour évaluation", and "Historique des soumissions". The main content area contains instructions and a visual representation of the word "BONJOUR" with its characters indexed from 1 to 7. Four numbered steps describe the tasks: 1. Use the 'length of' block to find the length of the string 'text' and store it in 'long'. 2. Use the 'get letter #' block to retrieve the character at index 17 from 'text' and store it in 'p'. 3. Use the 'find first occurrence of text' block to find the first occurrence of 'informatique' in 'text' and store the position in 'p'. 4. Use the 'get substring from letter # to letter #' block to extract the substring between indices 5 and 6 from 'text' and store it in 'secret'. Below the instructions is a "Utilisation des blocs" section with a drag-and-drop interface for code blocks. The blocks are arranged to implement the tasks: 'length of' (text: 'abc', output: '4'), 'get letter #' (text: 'text', letter #: '17', output: ' '), 'find first occurrence of text' (text: 'text', find: 'informatique', output: '3'), and 'get substring from letter # to letter #' (text: 'text', letter #: '5', to letter #: '6', output: ' '). At the bottom, there is a "Soumettre" button and a footer with version and copyright information.

FIGURE 2. Exercice de manipulation de chaîne de caractères.

Finalement, une séance d'idéation autour de la création de séquences pédagogiques sera proposée. On tentera de profiter des expériences du terrain des participants pour pouvoir récolter des idées de séquences de cours à développer. En effet, dans le cadre du projet Erasmus+ CAI⁸ (Communauté d'apprentissage de l'informatique), nous mutualisons les contenus en vue de soutenir les apprentissages et le développement professionnel des enseignants.

3 Détails pratiques

Matériel Pour cet atelier, l'idéal serait d'avoir un projecteur pour présenter des slides. Il faudrait que chaque enseignant ait accès à internet et à un ordinateur.

Participants Cet atelier s'adresse à des enseignants qui travaillent avec de la programmation visuelle. Pour permettre des échanges intéressants et pour pouvoir aider tout qui en a besoin, nous proposons de limiter la capacité de cet atelier à une dizaine de participants.

Remerciements

Le projet Erasmus+ CAI a été financé avec le soutien de la Commission européenne. Cette proposition d'atelier n'engage que son auteur et la Commission n'est pas responsable de l'usage qui pourrait être fait des informations qui y sont contenues.

Références

1. Derval, G., Gégó, A., Reinbold, P. : INGIInious [software] (2014), <https://github.com/UCL-INGI/INGIInious>
2. Derval, G., Gégó, A., Reinbold, P., Frantzen, B., Van Roy, P. : Automatic grading of programming exercises in a MOOC using the INGIInious platform. European Stakeholder Summit on experiences and best practices in and around MOOCs (EMOOCs'15) pp. 86–91 (2015)
3. Piraux, M., Taffin, L., Bonaventure, O. : Towards a Collection of Packet Trace Interactive Exercises for Computer Networking Education p. 3
4. Vahrenhold, J., Caspersen, M., Berry, G., Gal-Ezer, J., Kölling, M., McGettrick, A., Nardelli, E., Pereira, C., Westermeier, M. : Informatics Education in Europe : Are We All In The Same Boat ? (2017)

8. <https://cai.community/>