

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 INGINIOUS 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 INGINIOUS 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 · INGINIOUS · 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 INGINIOUS [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'INGINIOUS [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.

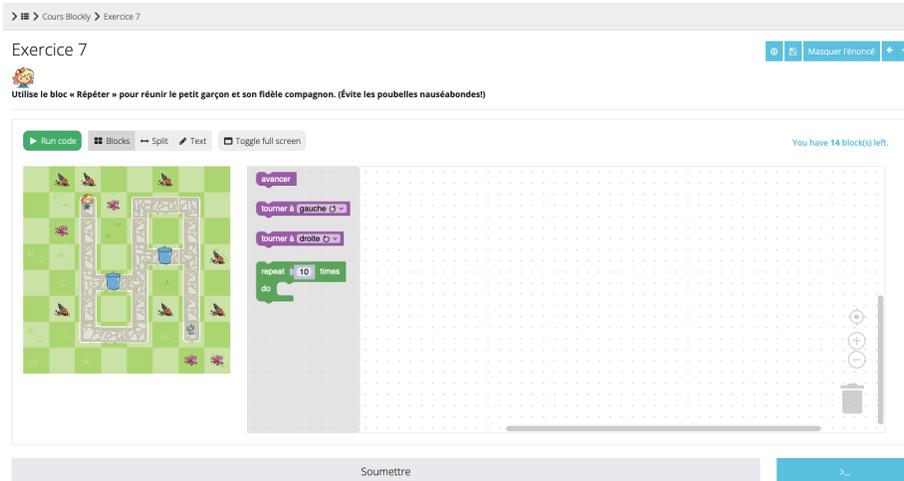


FIGURE 1. Exercice de déplacement dans un labyrinthe avec une animation liée au code.

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 récolter d'éventuels besoins pour des nouvelles fonctionnalités.

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

INGInious

Intuition à la bioinformatique > Analyse de chaînes de caractères

Informations

Date limite: Pas de date limite

Etat: Pas encore essayé

Note: 0.0%

Points de la note: 1.0

Nombre d'essais: 0

Limite de soumission: Pas de limite de soumission

Administration

Voir les soumissions

Editer l'exercice

Informations de débogage

Pour évaluation

Meilleure soumission

Pas de soumission

Partager mon résultat

Historique des soumissions

Pas de soumission

Analyse de chaînes de caractères

Les langages de programmation permettent de facilement manipuler des chaînes de caractères et d'en extraire de l'information. Cet exercice vous permettra de vous familiariser avec les principaux blocs de manipulation de chaînes de caractères. Ils sont de couleur verte.

Dans cet exercice, vous allez apprendre à utiliser les blocs de manipulation des chaînes de caractères.

- Le bloc `length` vous permet de trouver la longueur d'une chaîne de caractères. Quelle est la longueur de la chaîne de caractères contenue dans la variable `texte`? Sauvegardez cette longueur dans la variable `long`.

Une chaîne de caractères est une séquence de caractères. Chaque caractère est placé à une position dans la séquence qui est identifiée par un nombre. Dans la séquence `BONJOUR`, le caractère `B` se trouve en position 1, `J` en position 4, ...

1 2 3 4 5 6 7

BONJOUR

- En utilisant le bloc `get letter` ci-dessous, déterminez le caractère qui se trouve en 17ème position dans la variable `texte` et placez le résultat dans la variable `c`.

- La chaîne de caractères se trouvant dans la variable `texte` contient deux fois le mot `informatique`. En utilisant le bloc `find first occurrence of text`, trouvez la première occurrence de ce mot dans la chaîne se trouvant dans la variable `texte` et placez le résultat dans la variable `p`. Notez que le bloc `find` retourne la position du premier caractère du mot recherché et 0 si celui-ci n'est pas trouvé. Cela nous sera très utile par après.

- Enfin, en utilisant le bloc `get substring` extrairez le mot secret qui se trouve entre les positions 50 et 68 dans la variable `texte`. Placez ce mot dans la variable `secret` pour qu'il soit affiché par le programme.

Utilisation des blocs

Run code

length of

in text

in text

prompt for text

in text

```

def texte = " "
print "texte"
print "La bio informatique utilise l'informatique pour ..."
print "La chaîne de caractère contenue dans la variable ..."
set long = 0
print long
in text (texte) find [texte] occurrence of text
print position
print "Le caractère en position ..."
print "est ..."
set c = 0
print c
in text (texte) get letter #
print "Le mot informatique débute au caractère ..."
print p
set secret = 0
print "Le mot secret est ..."
print secret
                
```

Soumettre

Fourni par INGINIOUS v.0.8.6dev122-g13437114
 © 2014-2022 Université catholique de Louvain.
 Nihil in diebus suis invenit APT - l'histoire de confidentialité

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/>