



# Détection précoce des élèves rencontrant des difficultés d'apprentissage de l'algorithmique



Certains élèves n'arrivent pas à développer les compétences nécessaires pour penser les algorithmes informatiques. Qu'est-ce qui dans leur attitude en classe nous permettrait de les identifier de manière précoce ? Nous avons découvert que, quel que soit leur âge, les élèves présentant des difficultés d'apprentissage de l'algorithmique et qui ne réussissent pas à apprendre, ont un comportement similaire dès les premières leçons. Dans ce poster, nous présentons notre protocole d'observation et nous proposons de simples conseils pour aider l'enseignant à détecter ces élèves le plus tôt possible.

Christian Blanvillain  
Université de Patras, Grèce  
Christian.Blanvillain@hepl.ch

Bernard Baumberger  
Haute École Pédagogique de Lausanne, Suisse  
Bernard.Baumberger@hepl.ch

## INTRODUCTION

Si l'enseignant en science informatique souhaite étendre la portée de son action formative, il doit chercher à comprendre l'origine des difficultés d'apprentissage des élèves pour y remédier. Problème de motivation ? De confiance en soi ? De trouble d'apprentissage ? La question n'est pas toujours simple et pour certains élèves, elle est même plutôt complexe. Lorsqu'on propose à l'élève de résoudre un problème algorithmique, on va directement solliciter ses compétences cognitives nécessaires pour penser les algorithmes qui mobilisent essentiellement les intelligences pratique, créative et analytique. Détecter précocement les difficultés d'apprentissage des élèves permettra à l'enseignant d'effectuer des remédiations dans ces trois dimensions.

## MÉTHODOLOGIE

Pour élaborer notre grille d'observation, nous avons identifié des processus qui couvrent l'ensemble des attitudes visibles de l'élève en train de résoudre un problème algorithmique. Nous les avons regroupés dans cinq états suivants :

- [D/M] **Donne** de l'aide ou une explication, **Montre** la solution à d'autres élèves.
- [R/C] **Réfléchit**, essaye de **Comprendre** le problème posé ou son propre code.
- [F/M] **Fait** ce qui est demandé et **Manipule** le dispositif didactique pour coder.
- [E/D] **Écoute** les instructions ou une solution, **Discute** sans parler du problème.
- [J/A] Hors tâche : **Joue**, ou bien simplement **Attend** en faisant autre chose.

Nous avons filmé 17 élèves débutants de 10-12 ans et 15 élèves débutants de 20-24 ans résolvant des problèmes identiques durant tout un semestre. Au fil du semestre, nous avons identifié les élèves en difficulté d'apprentissage. Nous sommes alors revenus aux vidéos des premières leçons et avons comparé l'activité des élèves qui, au terme de la formation, ont réussi à apprendre malgré leurs difficultés, avec l'activité des élèves qui n'ont pas réussi à apprendre.

## RÉSULTATS

Nous avons sélectionné quatre séances emblématiques sur l'ensemble des vidéos analysées. Les deux premiers cas (figures 1 et 2) présentent des élèves en difficulté d'apprentissage qui ont réussi à apprendre au fil du semestre avec l'aide de l'enseignant. Les deux derniers cas (figures 3 et 4) présentent des élèves en difficulté d'apprentissage qui n'ont pas réussi à apprendre malgré l'aide prodiguée. Les diagrammes d'activité ont été réalisés avec le logiciel *ActoGraph*. Nous comparons l'attitude générale des élèves sur toute la durée de la séance. Ce sont ici les mêmes exercices tutoriels du dispositif *human processor!* [1] qui sont résolus.

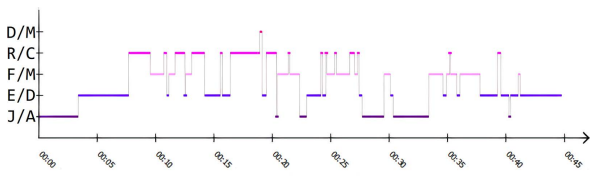


Fig. 1 - Activité d'un élève de 22 ans, en difficulté d'apprentissage, qui réussit à apprendre

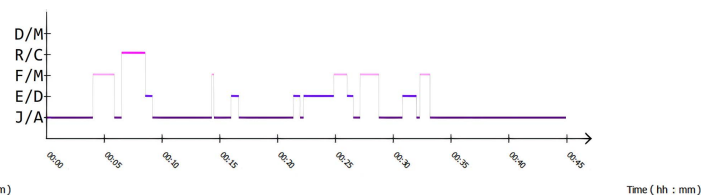


Fig. 3 - Activité d'un élève de 22 ans, en difficulté d'apprentissage, qui ne réussit pas à apprendre

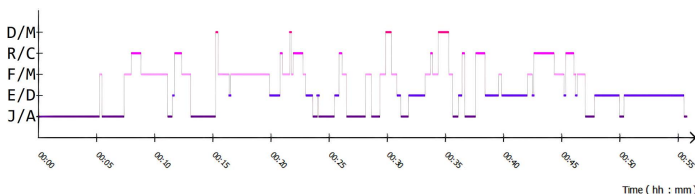


Fig. 2 - Activité d'un élève de 11 ans, en difficulté d'apprentissage, qui réussit à apprendre

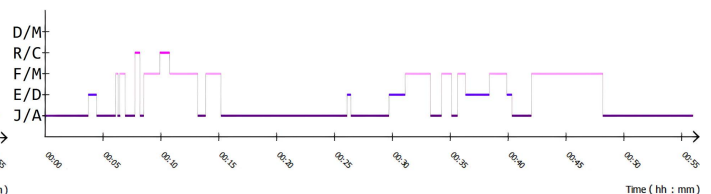


Fig. 4 - Activité d'un élève de 11 ans, en difficulté d'apprentissage, qui ne réussit pas à apprendre

## DISCUSSION

Nous constatons que, quel que soit l'âge de l'élève, les comportements en classe sont similaires. Les élèves en difficulté d'apprentissage qui réussissent malgré tout à apprendre au fil du semestre avec l'aide de l'enseignant, ont un diagramme d'activité beaucoup plus riche que les élèves qui n'arrivent pas à apprendre sur ces premiers exercices tutoriels. Ils alternent fréquemment entre manipuler les pièces du dispositif didactique débranché utilisé pour apprendre à programmer et les phases de réflexion. Parfois même, on les voit expliquer ce qu'ils ont compris aux autres élèves. Tandis que les élèves qui n'arrivent pas à apprendre sont dans l'action, mais ne mobilisent quasiment jamais les phases de réflexion et de compréhension du problème à résoudre. Ce qui fait qu'ils n'élaborent pas, sur ces exercices du début du cours, une image mentale interne efficiente ni une représentation complète de la machine notionnelle sous-jacente. Lorsque les exercices deviennent plus difficiles, il est alors trop tard pour rattraper ces lacunes et ces élèves décrochent. Être attentif à ces comportements permet de détecter les élèves qui n'arriveront pas à apprendre au fil du semestre. Découvrez d'autres conseils pour l'enseignant dans l'article qui accompagne ce poster, grâce au QR code ci-dessous.

## CONCLUSION

Détecter de manière précoce les élèves en difficulté d'apprentissage qui n'arrivent pas à apprendre, permet à l'enseignant d'apporter, dès les premières leçons, l'aide nécessaire pour éviter un décrochage en algorithmique. Pour aider ces élèves, l'enseignant devra identifier les fonctions cognitives défaillantes mobilisées dans l'acte de trouver des solutions algorithmiques aux problèmes proposés. C'est une remédiation délicate qui requiert du temps et de l'expertise. S'il n'est pas toujours possible de travailler sur ces fonctions cognitives avec un élève en particulier au sein du groupe classe, il peut être envisagé d'accorder un temps de soutien spécifique en dehors de la période de cours, pour renforcer les aptitudes cognitives des élèves présentant des troubles d'apprentissage de l'algorithmique. L'enseignant pourra alors expliciter les processus et les stratégies cognitives qui aident à résoudre un problème algorithmique [2], ce qui leur permettra de développer leur intelligence algorithmique pour réussir à penser les algorithmes.



<https://bit.ly/PosterDidapro9>

## RÉFÉRENCES

- [1] Blanvillain, C.: Human Processor! Communication présentée à Ludovia#CH, Yverdon-les-Bains, Suisse. (2020). <https://hdl.handle.net/20.500.12162/5376>
- [2] Blanvillain, C.: Apprendre à penser les algorithmes. Communication présentée à APIMU EIAH, Fribourg, Suisse. (2021) <https://hdl.handle.net/20.500.12162/5377>