

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

Christian Blanvillain^[0000-0003-3806-8492], Bernard Baumberger

Haute École Pédagogique, Lausanne, Suisse
prénom.nom@hepl.ch

Résumé. Certains élèves n'arrivent pas à développer les compétences nécessaires pour penser les algorithmes. 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, avaient 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.

Mots-clés : algorithmique, difficultés d'apprentissage, protocole d'observation.

1 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 informatiques 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. Dans ce poster, nous proposons d'observer en classe le comportement des élèves afin de réussir à identifier le plus tôt possible les élèves qui vont présenter des difficultés d'apprentissage et qui ne réussissent pas à apprendre, de manière à pouvoir en chercher les causes et éventuellement apporter les remédiations nécessaires lorsque les conditions le permettent.

Pour pouvoir distinguer les phases de manipulation du dispositif des phases de réflexion des élèves, nous avons utilisé le dispositif didactique débranché *human processor!* [1] permettant de coder les algorithmes par assemblage d'instructions (plaques en bois), qui permet de rendre visibles toutes les étapes d'élaboration de la solution algorithmique et sa vérification par l'élève à l'aide de la trace des données en mémoire. Lors de l'analyse des vidéos prises en classe, nous avons identifié des processus qui couvrent l'ensemble des attitudes visibles et possibles de l'élève en train de

résoudre un problème algorithmique. Nous avons construit notre grille d'observation en identifiant dans le comportement des élèves les cinq états suivants :

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

Lorsque l'élève écoute les instructions, s'il est attentif, il se construit déjà une représentation mentale du problème à résoudre ; s'il n'est pas attentif, en revanche, ce moment ne lui sert pas à apprendre. Lorsqu'il ne manipule pas les pièces, à quoi réfléchit-il ? Est-il en train de penser au problème ou bien à tout autre chose ? Ce sont les états précédents et suivants qui nous aideront à comprendre le contexte et l'état d'esprit dans lequel se trouve l'élève. Grâce au dispositif didactique débranché et les captures vidéo, nous pouvons identifier les moments où l'élève est passif ainsi que les moments où il est actif sans mobiliser ses compétences à penser les algorithmes. Avant et après avoir manipulé les pièces, il y a un moment où l'on voit l'élève réfléchir. Hors contexte, si on l'observe juste à ce moment-là, il n'est pas certain qu'il réfléchisse au problème posé : il peut penser à autre chose. Mais en fonction de ce qu'il a fait avant et surtout de ce qu'il fera après, on peut savoir s'il réfléchit au problème à résoudre ou pas. Enfin, lorsqu'il communique sur le problème résolu, lorsqu'il explique comment il a trouvé sa solution ou lorsqu'il explique à un autre élève ce qu'il faut faire, il mobilise clairement la représentation mentale qu'il s'est faite du problème à résoudre et sollicite ses compétences à penser les algorithmes.

2 Méthodologie

Nous avons construit cette grille d'analyse des activités des élèves en même temps que nous conceptualisons ce que nous souhaitons observer, dans une alternance vertueuse d'observation/théorisation (méthode de la théorie ancrée). Nous avons utilisé le logiciel *ActoGraph* pour taguer les différents états des élèves et réaliser les diagrammes d'activité des élèves tout au long de la séance de cours. Quatre figures emblématiques sont présentées dans le poster (<https://bit.ly/PosterDidapro9>), mais nous n'avons pas pu les inclure ici faute de place.

Nous avons observé des élèves de 10-12 ans et des élèves de 20-24 ans, tous débutants en informatique, travailler sur les mêmes problèmes pendant tout un semestre à raison d'une période par semaine. Nous nous sommes focalisés sur les élèves en difficultés d'apprentissage qui, après un semestre de formation, n'ont visiblement pas réussi à apprendre à penser les algorithmes. Nous les avons comparés avec les élèves en difficultés d'apprentissage qui, avec l'aide de l'enseignant sur plusieurs semaines, ont réussi à apprendre malgré leurs difficultés initiales. Il est simple d'identifier un élève qui a réussi à apprendre à penser les algorithmes : c'est un élève qui parvient à résoudre les problèmes proposés avec le dispositif didactique débranché et éventuellement un peu d'aide. Grâce aux enregistrements vidéo, nous avons pu remonter aux

premières leçons et observer spécifiquement le comportement de ces élèves. La question étant de tenter de comprendre ce qui permettrait de distinguer très tôt les uns des autres, en observant leur attitude, sur les problèmes simples du tutoriel. Est-ce uniquement un problème de motivation ou de confiance en soi ? Ou bien y a-t-il un réel trouble d'apprentissage lié à des difficultés de développement de leur intelligence algorithmique ? Et à ce moment-là, comment les aider ?

3 Résultats

Pour illustrer nos travaux, nous avons sélectionné quatre diagrammes d'activité représentant des séances emblématiques sur l'ensemble des vidéos analysées. Les deux premiers cas présentent l'activité d'élèves en difficulté d'apprentissage qui réussissent quelque peu à résoudre les exercices avec du temps et de l'aide. Les deux cas suivants sont des élèves en difficulté d'apprentissage qui ne réussissent pas du tout à faire les exercices, même avec du temps et de l'aide. En comparant les diagrammes d'activité élaborés à l'aide de notre grille d'observation, nous avons identifié un profil de comportement caractéristique des élèves en difficulté qui ne réussissent pas à apprendre à penser les algorithmes par rapport aux comportements des élèves en difficulté qui réussissent, malgré tout, à apprendre. Les élèves qui ne réussissent pas à apprendre à penser les algorithmes n'entrent jamais dans une phase de réflexion ou de compréhension de ce qu'ils font ou de ce qu'il faut faire (état [R/C]) et jamais, bien sûr, on ne les verra aider un autre élève ou bien discuter de leur solution (état [D/M]). Ils restent intellectuellement passifs durant toute l'activité et n'apprennent donc pas grand-chose. S'ils programmaient derrière un écran au lieu de programmer sur un dispositif didactique débranché, ce serait encore moins détectable puisqu'ils seraient probablement physiquement actifs à manipuler l'interface tout en restant intellectuellement passifs et donc sans apprendre.

Pour apprendre à coder, il ne suffit pas de lire et de comprendre des explications ou du code solution. Il faut s'entraîner à penser le code de manière à réussir à se construire un modèle mental de la machine notionnelle correspondant au dispositif didactique utilisé, c'est-à-dire une représentation mentale efficace et cohérente pour pouvoir développer sa pensée algorithmique. Le fait que certains élèves mobilisent leur intellect durant l'acte d'apprentissage ou pas, est visible à posteriori dans leur attitude grâce aux enregistrements vidéo et au dispositif didactique débranché, mais l'est hélas de manière beaucoup moins évidente en temps réel dans la classe ou lorsque les élèves sont derrière un écran.

4 Discussion

Nous avons observé qu'il n'était pas bon signe qu'un élève reste trop longtemps dans l'action de codage ou dans un semblant de réflexion. Il est certainement en train de rêvasser ou de penser à autre chose tout en jouant avec les pièces. S'il était en train de raisonner sur le problème en programmant, il faudrait en effet qu'il s'arrête de manipuler les pièces à un moment donné, pour réfléchir à nouveau avant de se remettre à

tester ses idées. Ainsi, ce qui est important d'observer pour l'enseignant, c'est la fréquence de l'alternance des états de l'élève. Un élève qui est en train de réfléchir, teste ses idées fréquemment et communique sur celles-ci lorsqu'il trouve quelque chose, ou bien va demander de l'aide pour comprendre si ce qu'il a fait fonctionne ou pas (et sera alors parfaitement attentif au moment où il recevra ces explications). Un élève qui manipule ses pièces pour coder de manière continue durant une trop longue période a juste besoin d'aide.

Les élèves qui présentent des difficultés d'apprentissage et qui ne réussissent pas à apprendre ont, dès les tout premiers exercices du début de la formation, une attitude en classe qui ne favorise pas le développement de leurs compétences à penser les algorithmes. Une chose remarquable est que, quel que soit leur âge, leurs comportements sont similaires : ils n'entrent jamais dans des états mobilisant leur intelligence algorithmique. Le détecter permet de diagnostiquer ces élèves de manière précoce, c'est-à-dire avant qu'il ne soit trop tard et que l'élève soit en décrochage.

Si les élèves travaillent sur écran, la seule manière pour l'enseignant de voir une différence dans le comportement des élèves en difficulté est d'être attentif aux moments où l'élève va spontanément communiquer sur ce qu'il a réussi à faire. Une bonne pratique consisterait alors à favoriser les échanges en faisant des interruptions fréquentes au cours de l'activité de résolution pour leur demander individuellement d'expliquer le problème, ce qu'ils sont en train de faire, mais aussi d'expliquer ce qu'ils ont déjà réussi à faire. L'enseignant pourra alors détecter assez tôt les élèves qui ne communiquent pas sur leurs travaux et qui n'arrivent donc pas bien à apprendre.

5 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 et 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.

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>