How to Repeat Hints: Improving AI-Driven Help in Open-Ended Learning Environments

Sébastien Lallé, Özge Nilay Yalçın, Cristina Conati

▶ **To cite this version:**

Sébastien Lallé, Özge Nilay Yalçın, Cristina Conati. How to Repeat Hints: Improving AI-Driven Help in Open-Ended Learning Environments. 24th International Conference on Artifical Intelligence in Education (AIED) 2023, Jul 2023, Tokyo, Japan. pp.760-766, 10.1007/978-3-031-36272-9_68 . hal-04142461

HAL Id: hal-04142461
https://hal.science/hal-04142461

Submitted on 27 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

How to Repeat Hints: Improving AI-driven Help in Open-Ended Learning Environments

Sébastien Lallé[1], Özge Nilay Yalçın[2], and Cristina Conati[3]

[1] LIP6, CNRS, Sorbonne University, Paris, France
[2] School of Interactive Arts & Technology, Simon Fraser University, Burnaby, Canada
[3] Dept. of Computer Science, University of British Columbia, Vancouver, Canada
sebastien.lalle@sorbonne-universite.fr

Abstract. AI-driven personalized support can help students learn from *Open-Ended Learning Environments* (OELEs). In this paper, we focus on how to effectively provide repeated hints in OELEs, when students repeat a sub-optimal behavior after receiving a hint on how to recover from the first occurrence of the behavior. We formally compare two repeated hint designs in UnityCT, an OELE that fosters *Computational Thinking* (CT) via free-form game design in K-6 education, with the long-term goal of providing AI-driven personalized hints.

Keywords: Open-Ended Environment, Real-Time Hint, Hint Design

1 Introduction

There is evidence (e.g., [4, 6, 8]) that AI-driven personalized support can help students learn from OELEs, namely, educational software that is specifically designed to let the students learn via exploration of the learning material with minimal constraints. However, delivering such help in an effective way is challenging for two reasons: 1) *student modeling*, namely, how to capture and recognize students' behaviors that indicate the need for help and 2) *how to deliver the help effectively* without interfering with the interaction with the OELE. Our long-term goal is to study how to design effective personalized help for OELEs that support game-based activities for K-6 students. This context is especially challenging from the point of view of delivering help that is effective and not intrusive. For this purpose, we leverage UnityCT, an OELE designed by UME Academy (*www.ume.academy*) to foster CT among young learners. CT, or the ability to express problems and their solutions computationally, has become an essential part of today's K12 curricula [1]. Previous work [7] addressed the challenge of building student models in UnityCT, by inferring from data non-obvious sub-optimal student behaviors that call for personalized help.

In this paper, we address the challenge of *how to deliver this personalized help effectively* once the need for it has been established by the student model. We focus on a specific form of help provision, namely how to effectively provide *repeated hints* when students repeat sub-optimal behaviors and they have already

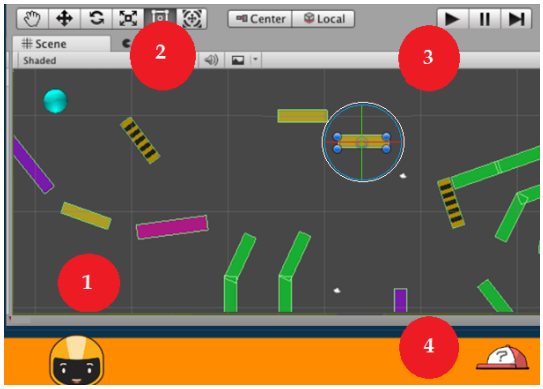

Fig. 1: UnityCT environment.

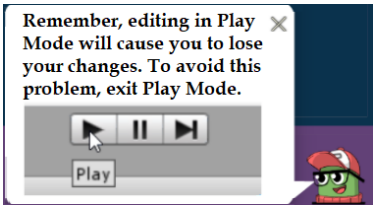

Fig. 2: New design of the repeated hints with combined texts.

received a hint on how to recover from the first occurrence of the behavior without interfering with the exploratory nature of OELEs. Some research on this issue has targeted OELEs for university students, by providing a more explicit justification as for why students should follow a hint when an error is repeated, with positive results on the students' experience [6, 4]. However, these findings with college students may not translate to younger K-6 students who are less experienced with computers and learning environments. Two works targeted OELEs for K-6 students, but they just repeated hints without changing their content or format, in an OELE for model building activities (e.g., modeling engine force) [2] and in UnityCT [8]. Here, we build on the results of [8] by exploring how to alleviate the issues of increased confusion and lack of impact on error correction they found with repeated hints. Specifically, we address these research questions (RQs): "Compared to [8], does changing the design of repeated hints in UnityCT: RQ1/ improve students' error correction behavior? RQ2/ reduce students' reported confusion with these hints? " The evaluation of the new hints was conducted in real-life settings aiming for high ecological validity, and our results show that the new design successfully answered both RQs.

2 UnityCT

UnityCT is built on the Unity game engine and used in online lessons to foster CT skills as suitable for young audiences. During each 1-hour lesson, an instructor teaches the CT skills covered in that lesson, and then assigns a 30-minutes challenge to design incremental game components that meet specific constraints. Students can freely explore how to build these components in the Unity scene (Fig. 1.1) in which they can manipulate game objects with different manipulators (e.g., move, rotate, Fig. 1.2). Students can enter into Play Mode by clicking the Run button (Fig. 1.3) to execute and test their game. The feedback from UME Academy is in line with research showing that game-based activities increase student engagement in CT classes [3, 5]. However, UnityCT introduces specific

challenges due to students having to both operate in the complex Unity interface and learn the CT material. Although students can ask the instructor for help, not all students do it. Instructors generally try to watch for students needing help, but this is challenging without continuously monitoring what each student is doing. Thus, UME Academy has been focusing on augmenting UnityCT with personalized hints delivered by an *Intelligent Pedagogical Agent* (*IPA* from now on), as reported in [8]. Fig. 1.4 shows the IPA when it is not providing a hint.

In [8], authors worked with the introductory lesson with UnityCT as a proof of concept focusing on errors commonly addressed by the instructors. Here, we focus on one of them, the *PlayMode* error, for which the hint repetition in [8] yielded some positive results. This error occurs when students enter "Play Mode" to execute their current game and make changes to their scene while in this mode. This is problematic because UnityCT does not record changes while in Play Mode, and can be corrected by pressing the Play button (Fig. 1.3) to exit the mode or by avoiding making changes to the scene. In [8], the PlayMode hint was designed to provide help at incremental levels of detail. This incremental support is consistent with how the UnityCT instructors provide help: they first flag the problem and why it exists (e.g., your changes will be lost because of editing in play mode), followed by an explanation of how to fix the problem if the student asks (e.g., exit play mode). This 2-step design was used to format the hints via speech bubbles. UnityCT and the hints are fully described in [8].

An evaluation of this work [8] showed that repeating the same PlayMode hint helped students make fewer errors and perform more correct behaviors, as compared to delivering the hints only once (1-Shot) or having no hints. However, hint repetition did not help students correcting their errors compared to 1-Shot and no hints, even though the hints explicitly showcase how to do so. The repeated hints were also judged by the students to be more confusing than the 1-shot hints, showing that there is room for improvement.Hence, we designed a new version of the repeated hint, described next.

3 User Study

To address the aforementioned limitations of previous repeated hints (*repeated-two-levels* version from now on), we designed a new version of these hints, in collaboration with UME Academy's UX/UI expert and three experienced instructors who used the previous IPAs. The key of this new design (*repeated-one-level* version from now on, see Fig. 2) is that the two levels of the repeated hint are combined into one bubble, as opposed to two bubbles in the previous study [8], so when students repeat the error they see a reminder of the nature of the error and how to correct it all at once. The rationale for this new design is two-fold. First, showing the solution right away when a hint is repeated is meant to increase error correction, especially for those students who, with a two-level hints, would choose to only see the first speech bubble and thus never see how to correct the error in the second bubble. Second, the instructors suggested that receiving the same speech bubbles twice may cause students to wonder why they

Table 1: Summary statistics (mean and st. dev.) of the corrrection measures.

Performance measure	Repeated-one-level	Repeated-two-levels	No-hint
Prop. of 2nd error correction	0.92	0.38	0.13
Rate of spont. error correction	0.92 (±0.14)	0.29 (±0.41)	0.12 (±0.16)

are receiving the hint again and generating the confusion with the repeated-hints IPA reported in [8]. Hence, changing the repeated hint so that it provides all the information at once may help reduce this source of confusion by diversifying it from the first occurrence of the hint. Importantly, these changes apply only to the repeated hints, i.e., we did not modify the first hint because the previous study [8] found that the students who got only one hint did not report confusion. The phrasing of the hints was finalized by the instructors and UX/UI designer.

The repeated-one-level version of the hints was deployed in 10 on-line classes for a total of 53 students. The study procedure and participant population was the same as in [8]: the classes covered exactly the same content, were open to students in grade 4 to 6 living in North America and used the same enrolling process. The inclusion of the hints did not alter the structure of the lessons, which took place in fully ecological settings; the students received no training nor specific instructions about the hints At the end of class, we collected the students' levels of perceived confusion with the hints using the same usability survey as in [8]. Confusion is rated on a 3-level Likert scale, coded as 0 for "Never", 0.5 for "Sometimes" and 1 for "Always". Maintaining the exact same study process allows us to compare the results of this evaluation of the repeated-one-level hints against the results obtained in the study in [8].

4 Analysis and Results

To answer our RQs, we compare the new *repeated-one-level* condition, against the *repeated-two-level* and *no-hint* conditions from [8] (each having 56 students).

Results for RQ1. We use the two measures of error correction from [8]:

M1. *Second error correction*: the proportion of students who corrected their second PlayMode error. For students in the conditions with an IPA, in which the second error triggers a repeated hint, this measure evaluates the *immediate effect* of the repeated hint on correcting the error they just made. For the no-hint condition, it measures if the students spontaneously correct their second error.

M2. *Rate of spontaneous error corrections* after the second error: the ratio of errors corrected over errors made after the second one. For students who received hints, it shows if the hints are successful in teaching the students to spontaneously detect and fix errors, when they don't receive hints anymore.

We retain students who made a second PlayMode error, namely 13 students from the repeated-one-level condition, 21 students from the repeated-two-levels condition and 38 students for the no-hint condition. Table 1 shows descriptive statistics for (M1-M2). We report statistical significance at the 0.05 level, and

use Cohen's d for effect sizes (reported as large for $d > 0.8$, medium for $d > 0.5$, small otherwise.) We include *class_id* as a random effect in all models, to account for potential differences in the classes involved in the study.

Second error correction. We ran a generalized linear mixed model (GLMM) with a logistic link with this measure as dependent variable, condition (repeated-one-level, repeated-two-level, no-hint) as a factor, class_id as a random effect and first error correction (corrected, non-corrected) as covariate. This model estimates the probability that the students corrected their second error depending on their hint condition. The covariate captures if students could correct their error after receiving the first hint. This GLMM revealed a significant main effect of conditions ($p = .02$, $d = .68$). Pairwise comparisons using Holm-adjusted t-tests reveal a significant difference between the repeated-one-level condition and the no-hint conditions ($p = .03$, $d = .6$). There was no other significant pairwise comparisons ($p = .2$, $d = .4$), i.e., only the one-level hints outperformed no-hint.

Spontaneous error correction. For M2, we look at those students who continued to make errors after the second one (7 in repeated-one-level, 14 in repeated-two-levels and 37 in no-hint). We fit a GLMM with spontaneous correction rate as the dependent variable, condition (three levels) as the factor, and class_id as a random effect. This GLMM revealed a significant main effect of condition ($p < .01$, $d = 2.02$). Pairwise comparisons with Holm-adjusted t-tests show that the repeated-one-level condition outperforms both the no-hint condition ($p < .001$, $d = 2.46$) and the repeated-two-levels condition ($p < 0.001$, $d = 1.32$) with large effect sizes, highlighting an additional benefit of our new hint design[4].

Results for RQ2. We examine the self-reported levels of confusion with the hints in the usability survey, which was completed by 19 students from the repeated-one-level and 23 from the repeated-two-level condition. We use Ordinal Logistic Regression (OLR) to compare the Likert-scale ratings among the conditions. The OLR shows that the repeated-one-level hints are rated as significantly less confusing ($p = .01$, $d = .74$) than the repeated-two-levels hints. This indicates that the new hints were successful in reducing the confusion seen in the repeated-two-levels condition, with an average rating of 0.13 and a mode of 0, which corresponds to "never confusing" in the survey, as compared to a mode of 0.5 for repeated-two-levels ("sometimes confusing"). In addition, the students' perception of the hints was very positive for all other usability items, with almost all students liking the hints and wanting to reuse them.

5 Discussion and Future Work

Our results provide evidence that the repeated-one-level condition can improve student error correction behavior (RQ1), while eliminating the confusion generated by the repeated-two-levels hints in [8] (RQ2). The repeated-one-level hints

[4] Note that we also confirmed that the repeated-one-level hints do maintain the positive results found for the repeated-two-levels hints in [8] as for fostering less errors and more correct behaviors (we did so with the exact same analysis as in [8].)

also maintain all of the positive results already found in [8] related to the performance and experience of the students. These results are encouraging as there has been no research on OELEs for K-6 students that formally investigated how to format a repeated hint, and here we propose a new promising design that we tested in regular classes, thus demonstrating high ecological validity. While we focus on one OELE, we see our study as a proof of concept for the value of personalized help to support young learners in game-based learning activities. This opens the possibility of testing the hint design in other game-based OELEs, which are getting popular to engage children in CT.

As a next step, we plan to leverage our IPA to address other problematic behaviors using the existing data-driven student models in UnityCT mentioned in the introduction [7]. We will focus on providing personalized support to the sub-optimal behaviors mined in [7] that are similar in nature to the PlayMode error targeted in this paper. In particular, in [7], they found that students failed to remediate other errors (e.g., deleting a key object, panning outside the scene), or did not perform important actions (e.g., not interacting with objects). Applying our repeated hint design to these behaviors would be a important step toward evaluating fully AI-driven hints in game-based OELEs for K-6 learners.

Acknowledgement. This work was supported by UME Academy Ltd. (*ume.academy*) and Mitacs (Grant #IT13336).

References

1. Barr, V., and Stephenson, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads 2*, 1 (2011), 48–54. Publisher: ACM New York, NY, USA.
2. Basu, S., Biswas, G., and Kinnebrew, J. S. Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User-Adapted Interaction 27*, 1 (Mar. 2017), 5–53.
3. Biswas, G., Segedy, J. R., and Kinnebrew, J. S. Smart Open-Ended Learning Environments That Support Learners Cognitive and Metacognitive Processes. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data* (2013), Lecture Notes in Computer Science, Springer, pp. 303–310.
4. Borek, A., McLaren, B. M., Karabinos, M., and Yaron, D. How much assistance is helpful to students in discovery learning? In *Learning in the Synergy of Multiple Disciplines* (2009), Springer, pp. 391–404.
5. Çakir, N. A., Gass, A., Foster, A., and Lee, F. J. Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education 108* (2017), 115–130. Publisher: Elsevier.
6. Kardan, S., and Conati, C. Providing adaptive support in an interactive simulation for learning: An experimental evaluation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2015), pp. 3671–3680.
7. Lallé, S., Yalçin, Ö. N., and Conati, C. Combining data-driven models and expert knowledge for personalized support to foster computational thinking skills. In *11th Inter. Learning Analytics and Knowledge Conf.* (2021), ACM, pp. 375–385.
8. Yalçin, Ö. N., Lallé, S., and Conati, C. An intelligent pedagogical agent to foster computational thinking in open-ended game design activities. In *International Conference on Intelligent User Interfaces* (2022), ACM, pp. 633–645.