Collaborative programming problem-solving in augmented reality: Multimodal analysis of effectiveness and group collaboration

Cheng-Yu Chung
Arizona State University

Nayif Awad
College of Sakhnin for Teacher Education

I-Han Hsiao
Santa Clara University

Although numerous studies have demonstrated different ways that augmented reality (AR) can assist students to understand the learning content via contextualised visualisation, less explored is its effect on collaborative problem-solving (CPS) in computer programming. This study aims to investigate how AR affects a CPS in a programming task. We designed a mobile app that could visualise computer programming in AR and non-AR 3D images. The app could involve two participants working together on a programming problem face to face in the same workspace. We conducted a within-subjects experiment to compare their AR experience to the non-AR experience and collected multimodal usage data about the task performance, verbal communication, and user experience. The analysis showed that the participants in the AR experience had higher task performance and more insightful communication than the non-AR. The participants also had positive attitudes toward the use of AR in classroom instructions. In a semi-structured interview, the participants reflected that AR helped them engage in the content and analyse the task easier. Based on this study, we discuss several challenges and implications for future instruction designers.

Implications for practice or policy:

- AR can improve student engagement in a collaborative problem-solving task.
- AR has the potential to promote and improve group communication in collaborative work.
- Instruction designers may need to carefully align the characteristics of AR with the task content especially when physical models are rarely used in the learning content.

Keywords: augmented reality, collaborative problem solving, programming practice, computational thinking, multimodal analysis

Introduction

Augmented reality (AR) has become a promising technology for educational purposes through its ability to integrate virtual information into the physical surroundings. AR can enhance the user's perception of the real world (e.g., a physical object on a table) by digital information related to the object (e.g., annotated color, visual effects) (Azuma, 1997). Researchers have suggested that the benefit of AR for education is in three dimensions: physical, cognitive, and contextual. In the physical dimension, AR provides intuitive interactions with the content (Santos et al., 2014). In the cognitive dimension, the integration of physical and virtual information may reduce the cognitive load when the user performs a task (Bujak et al., 2013). In the contextual dimension, AR may increase the personal relevance to the content via seamless interaction (Wei et al., 2015).

AR applications for educational purposes are sometimes called augmented reality learning experiences (ARLEs). ARLE has been used to generally describe learning content presented in AR visualisations (Institute of Electrical and Electronics Engineers [IEEE], 2020; Santos et al., 2014). Empirical studies have shown that ARLEs can enhance students' laboratory skills (Akçayır & Akçayır, 2017), understanding of complex concepts (Liou et al., 2017), and attitudes toward the learning subject (Fidan & Tuncel, 2019). For example, Liou et al. (2017) conducted a study to investigate the effect of ARLE in learning astronomy. The quasi-experiment with K-12 students showed that the AR group (the students perceiving the AR visualisation) not only performed better in a formal test but also achieved higher quality interaction than the non-AR group. The researchers pointed out that allowing the students to see the periphery and their peers in the classroom could be one reason for the positive outcomes. Another study from Fidan and Tuncel (2019) investigated an ARLE in problem-based learning, and found the AR group had significantly higher learning performance. The outcome was attributed to the "immersive and realistic contexts" (Fidan & Tuncel, 2019, p. 18) supplied by AR. In addition to the user-content interaction, one interesting finding from these studies is the improvement of the interaction between users.

Although various AR instructional designs have been explored and evaluated in different school subjects, especially those of STEM, several challenges still exist in designing an appropriate AR experience. For example, Ashtari et al. (2020) interviewed 21 content designers and summarised 8 key barriers in authoring AR or virtual reality (VR) applications, for example, designing "for the physical aspect of immersive experiences" (Ashtari et al., 2020, p. 10). How to design an AR experience that appropriately leverages the physical interactivity is another challenge especially for a school subject that seldom uses physical models in the instructions, for example, computer programming (Radu, 2014). Additionally, we noticed that the current literature focused less on collaborative problem-solving (CPS) which can also be considered a physical aspect in an instructional design. Considering that CPS is the cornerstone in modern workspaces (i.e., rarely is work done by a person alone), we believe there is a need for understanding how AR experiences affect the group dynamics in CPS. This study aimed to fill the research gap by investigating how and whether AR can support, or diminish, the performance of a programming CPS task. Specifically, this study was guided by the following research questions:

- How does AR visualisation take effect in terms of the performance of a CPS programming task?
- Does AR affect the content and efficiency of the verbal communication between the participants?
- What are the participants' attitudes toward the experience and usability of AR in the CPS task?

The originality of this study is threefold. First, this study focused on the use of AR in CPS programming tasks. Not only have there been few studies of AR in programming-related content, the confluence of AR and CPS is also less investigated. Second, this study proposed a feasible design of AR programming tasks following the affordances of AR and key concepts of CPS in the literature. This ensures the design is closely aligned with the existing evidence about the effectiveness of AR but not merely served as an alternative presentation of content. Finally, the analysis in this study was based on various data sources, for example, on-device task records, verbal communication, semi-structured interview to capture both in-situ user interaction in the AR application, and subjective feedback on the user experience. Different from the use of single data sources in most of the previous studies, this study provided a more comprehensive view of the users' reactions to the AR CPS task.

Literature review

Characteristics and designs of ARLE in STEM education

Although there are still many challenges in producing an ARLE (Ashtari et al., 2020), in the past decades, researchers have identified various characteristics (or features) of AR that can be used to support learning in STEM courses and training in workspaces (Bujak et al., 2013; Ibáñez & Delgado-Kloos, 2018). Also, the recent Institute of Electrical and Electronics Engineers (IEEE) standard for ARLE models provides a succinct modeling language that describes the essential characteristics of ARLE (IEEE, 2020). The language consists of two models: activity and workspace. The activity model is used to describe instructions for a learning event. The workspace model is used to describe the environment where the AR activity is carried out, including the physical objects, persons, and devices. This modeling language provided a framework for our review of instructional designs in ARLEs.

ARLE workspaces can be categorised as two types by the way they present the AR visualisation: image-based and location-based (Radu, 2014). Image-based AR uses a marker or image as the target to present the AR visualisation, for example, markers on top of a laboratory device (Radu & Schneider, 2019). Location-based AR uses the geolocation signal to render the AR visualisation, for example, locations of certain species of plants (Chiang et al., 2014). In a workspace, the AR device, either a handheld mobile device or a head-mounted display, is probably the major criterion that determines what kinds of instructions can be carried out effectively. Handheld devices are relatively affordable but provide very limited interactivity. Head-mounted displays usually provide better interactivity, for example, eye and head gaze input, but the cost may be too high to deploy in a large classroom.

According to Ibáñez and Delgado-Kloos (2018), commonly explored STEM subjects in ARLEs include physics, earth sciences, mathematics, and life sciences. Common instructional designs include unstructured/guided exploration, simulation tools, and augmented books (Martín-Gutiérrez et al., 2015). For example, Radu and Schneider (2019) demonstrated a head-mounted display image-based ARLE for electromagnetism education. Their instructional design was considered an unstructured learning event similar to a laboratory hands-on experience. They compared the participants' learning attitudes, outcomes, and collaboration in multiple experimental conditions. Although their analysis showed that the participants who saw the AR visualisation were more motivated and effective in understanding the learning content, they also found that too many AR effects could overwhelm the user. Fidan and Tuncel (2019) demonstrated an ARLE designed for handheld devices in the context of problem-based learning in a physics classroom. Different from (Radu & Schneider, 2019), the researchers evaluated the ARLE over an 11 week curriculum at a junior high school. They were able to deploy the AR app on multiple handheld devices at once. Their analysis showed that the participants in the AR group had better long term retention of the learning content than the non-AR group. The research implies that AR can be an effective instructional tool for engaging students in physics education.

ARLE for collaborative problem solving in computer programming

Collaborative problem-solving (CPS) refers to an exploration of solutions in a joint solution space where the participants socially interact with each other, share knowledge elements, and perform problem-solving actions to find solutions to achieve the goal (Roschelle & Teasley, 1995). The benefits of CPS in STEM education have been reiterated extensively in the literature of computer-supported collaborative learning. For example, a meta-analysis from (Jeong et al., 2019) reviewed computer-supported collaborative learning studies in the past 10 years and summarised the outcomes from the mode of collaboration, learner educational levels, domains of learning, technology types, and pedagogical strategies. The result suggested that computer-supported collaborative learning produces positive outcomes in STEM education. Specifically, the pedagogy of problem solving with an accepted effect size of around 0.2, has been used in computer science and achieved a positive outcome (Jeong et al., 2019).

Although there have been numerous studies focusing on ARLE for STEM subjects and CPS in computer programming, to the best of our knowledge there is little research on the confluence of these three aspects (i.e., ARLE, CPS, computer programming). There have been some studies focusing on AR representations of programming concepts, which have the potential to bring positive effects on CPS tasks. For example, an early work from Radu and Maclntyre (2009) demonstrated AR Scratch, that creates AR effects on top of the Scratch programming language (Radu & MacIntyre, 2009). Schez-Sobrino et al. (2021) proposed an AR visualisation of programming concepts for collaborative programming learning. Their analysis showed that the proposed notion was easy for students to understand. However, these tools have not been shown to be effective in CPS programming tasks in experiments.

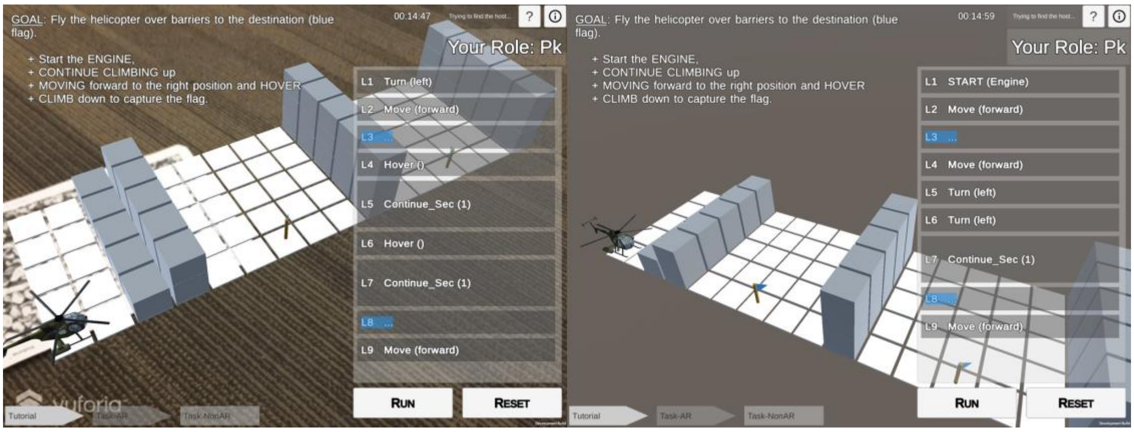
One work by Radu et al. (2021) is probably the closest to our research purposes. The team investigated the impacts of AR on collaboration in robot programming. They used head-mounted display devices, developed physical markers on top of a robot programming kit, and recruited 40 groups (dyads) of participants in a between-subjects design. The result showed that AR improved both group learning and collaboration and indicated that the ARLE has the potential to improve group dynamics, for example, communication. Indeed, there are many similarities shared between their work and this study. However, one major difference is that our study focused on programming tasks without existing physical manipulatives that have been used in the instruction. In Radu et al.’s (2021) work, AR was used to add an extra layer of information on top of already existing programmable robots, allowing the user can perceive more information in the programming task. In other words, their AR served as add-on information. On the contrary, this study focused on whether AR and non-AR presentations of a CPS programming task itself affected group dynamics and task performance. We investigated differences and effects in programming content presented in AR and non-AR.

Methodology

The major research purpose of this study was to investigate how AR visualisation affected users in a CPS programming task in terms of task performance, verbal communication with the partner, and their attitudes toward the use of AR in the CPS task. To the best of our knowledge, there was no publicly available AR-enabled programming language to help us design a CPS task. To address the research purpose, we first designed an AR-enabled app and CPS programming task that required two participants to collaborate and write a computer program. The app could present the same content in AR and non-AR visualisations, which allowed us to conduct a within-subjects control experiment to investigate how AR affects the user via paired *t*-tests. A hidden Markov model (HMM) was used to analyse the evolution of code snapshots and communication topics quantitatively. Furthermore, a semi-structured interview and a posttest questionnaire were used to assess the user's attitudes and experience after the experiment. In the following sections, we describe these methods and the collected data in detail.

AR application for CPS programming tasks

We chose to use handheld devices and image-based AR to design the CPS programming task. We adopted the concept of visual programming languages and made an app, Ogmented, which was a simple programming environment for small screens on mobile devices (Figure 1). The app has three components: (a) a program panel on the left for the user to modify the code snippet by touching and selecting a line of code, (b) a top panel showing the content of the current task, and (c) a viewport in the middle showing the result of the program. After clicking the run button, the user can observe the outcome produced by the script in the 3D scene which can be rendered in AR (the left screenshot in Figure 1) and non-AR (the right screenshot in Figure 1) visualisations. The left picture is the AR version, and the right one is the non-AR version. The only difference between the two versions is the visualisation. In the AR version, the app requires an image marker to render the task content. In the non-AR version, the task content is rendered as a 3D image on the screen. This app also allows the researchers to input different programming tasks as needed.



GOAL: Fly the helicopter over barriers to the destination (blue flag).

- Start the ENGINE,
- CONTINUE CLIMBING up
- MOVING forward to the right position and HOVER
- CLIMB down to capture the flag.

```
Turn (left)
Move (forward)
...
Hover ()
Continue_Sec (1)
Hover ()
Continue_Sec (1)
...
Move (forward)
```

GOAL: Fly the helicopter over barriers to the destination (blue flag).

- Start the ENGINE,
- CONTINUE CLIMBING up
- MOVING forward to the right position and HOVER
- CLIMB down to capture the flag.

```
START (Engine)
Move (forward)
...
Move (forward)
Turn (left)
Turn (left)
Continue_Sec (1)
...
Move (forward)
```

Figure 1. Screenshots of the CPS programming task in AR and non-AR versions

Although researchers have shown that CPS is an effective instructional tool, it has been noted that such an effect is "a function of the technology and the pedagogy" (Jeong et al., 2019, p. 14). That is to say, effective collaboration does not occur simply due to the coexistence of multiple participants in the same workspace. Both technology and pedagogy have to actively promote group collaboration. Considering this aspect, we also implemented a turn-taking mechanism that allows the users to share the control flow of the same script (Figure 2). This mechanism balances the amount of workload between the two users. To some extent, it can also reduce the likelihood of the case where one user dominates the whole CPS process, that is, they finish the whole task without collaboration. The CPS programming task in this study was designed to promote collaboration on the system level. The code snippets from the two devices synchronise in the runtime according to a special code block denoted by "…" in Figure 2. This mechanism was expected to reduce the probability that one user dominated the whole CPS process.

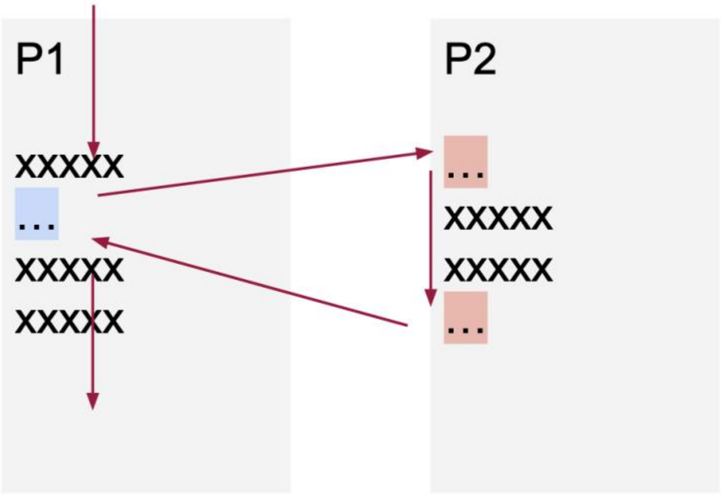


Figure 2. The turn-taking mechanism

In this study, the participants' CPS programming task was to fly a simulated helicopter across the 3D scene and collect all the flags. The design of this simulation considered that it involved temporal and spatial information which can be effectively presented in an AR visualisation. The two users were required to observe the scene and find out relevant information about the solution (e.g., as shown in Figure 3), break down the problem into sub-problems, observe and recognise patterns, and design an algorithm consisting of a series of command instructions. This CPS process is related to several concepts in computational thinking like abstraction, decomposition, pattern recognition, and algorithm design (Shute et al., 2017).

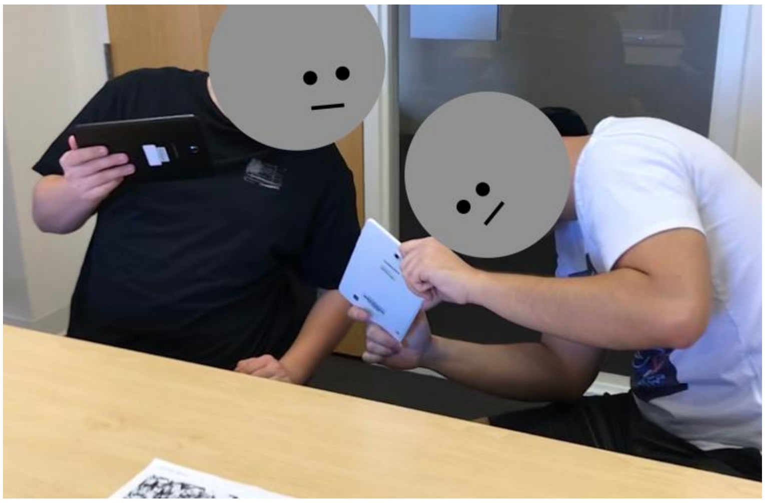


Figure 3. The participants adjust their posture to explore the task content.

The procedure of our within-subjects experiment

We conducted a within-subjects experiment to observe the difference in solving the task in the AR and non-AR versions of the app. To counter the carryover effect, a counterbalancing measure was applied by randomising the order of treatments, that is, the order of solving the task in the AR and non-AR versions. For the sake of simplicity, we used the terms AR and non-AN to denote these two treatments. One advantage of within-subjects experiments was to control individual differences because the participants received both the control and experiment treatments. This was important in this study because the participants may have had different levels of background knowledge and problem-solving skills.

The experiment was held in a synchronised workspace in a university meeting room. A session required a pair of participants who sit next to each other at the table. The participants were asked to solve a programming task collaboratively on two separate mobile tablets. They were allowed to move around and check the partner's screen if needed. However, they were told explicitly by the researchers that they must not take over and control their partner's device. In total, the participants spent around 60 minutes in the experiment. In the 60 minutes, the participants started with an introduction plus a tutorial (15 minutes), the task (15/10 minutes for the first/second treatments), a questionnaire about user experience (15 minutes), and a semi-structured interview (5 minutes). The allocated time and difficulty of the task were adjusted according to our pilot study with four pairs of participants from the same population. A snapshot of the experiment is shown in Figure 4 for reference.

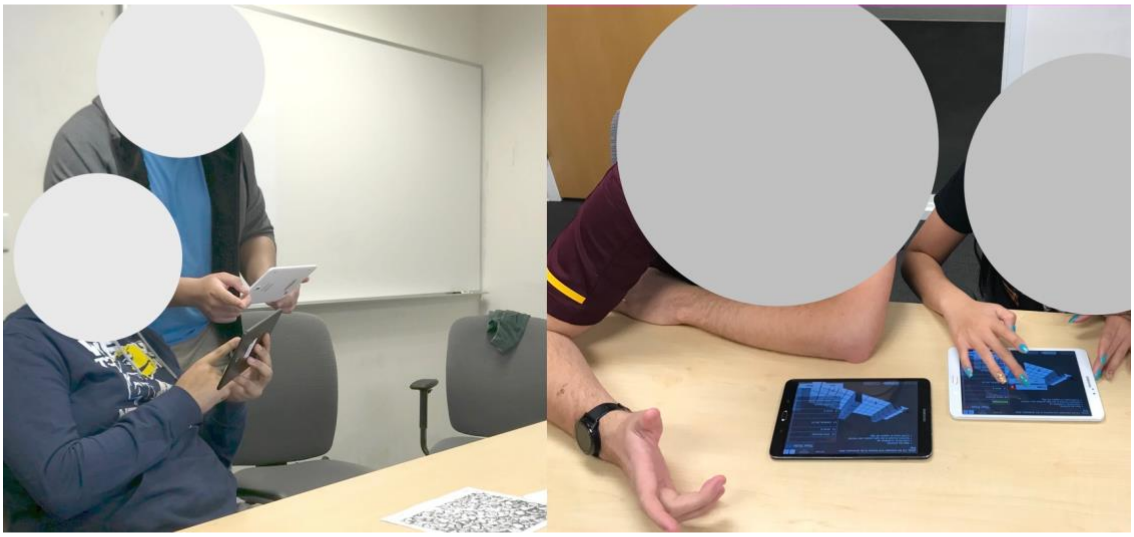


Figure 4. Examples of workspace conditions in the AR (left) and non-AR (right) versions.

Data collection

We recruited participants from an undergraduate course about introductory informatics that was offered in a 4-year bachelor program at Arizona State University. The research protocol was reviewed and approved by the institutional review boards at Arizona State University (Reference ID: STUDY00007968). We completed the experiment with 12 pairs (in total 13 males and 11 females). The data collection was based on learning analytics tools and multimodal data. The participants were also asked to fill a questionnaire and participate in an interview. Overall, there were four sources of data in the experiment: (a) task log data including code snapshots and execution outcomes, (b) audio recordings of verbal communication, (c) self-reported questionnaires about the user experience and system usability, and (d) a semi-structured interview.

Conversational coding schema for verbal communication

To understand the semantics of the audio recordings, we first manually transcribed the recordings into texts (each of which might contain one or multiple sentences) and processed the texts by adding semantic labels. The labels were adopted from a conversational coding schema for group collaboration (Fussell et al., 2000), shown as follows:

- TaskState: Related to task states and outcomes, for example, "It went too far."
- Inter: Related to internal states, for example, "I think it should be 'climb down' because…"
- Progress: Related to the progress of the task, for example, "I'll make it…", "Let's try…"
- Ack: Acknowledgement, for example, "Yeah.", "Okay."
- Other: Any other cases that cannot be categorised in the above categories.

Due to the large number of texts ($N = 3737$), we used a semi-supervised approach to cluster the texts into the semantic labels according to semantic and syntactic features. For short texts where the number of tokens was less than 3, we used keywords to categorise them as Ack (e.g., Okay) or TaskState (e.g., a programming command in the CPS programming task). The threshold of 3 was determined by the convention in English (where acknowledgment usually uses one or two words) and the maximum length of available programming commands in the task.

For long texts, we assumed that two texts can be assigned to the same semantic label if their semantic topics are similar in the vector space. First, we extracted topics from the texts by latent Dirichlet allocation (Blei et al., 2012), which is an unsupervised method that can summarise a given text into M number of topics. The M topics were categorised into the above semantic labels manually. This set was served as the ground truth in the process. Then, we used the hierarchical clustering algorithm to cluster the texts into K clusters according to the extracted topics and syntactic features. The syntactic features included term frequency-inverse document frequency (aka TF-IDF) and the number of part-of-speech tags (Shibani et al., 2017). Afterward, the texts in a cluster were tagged by the most frequent semantic label (based on the extracted topics). The values of M and K were determined empirically by a grid search according to the Silhouette coefficient (Rousseeuw, 1987), which is a method of validating the consistency within clustered data.

Questionnaire about the user experience and system usability

The user experience was evaluated by a questionnaire that consisted of 24 5-item Likert scale questions and 3 short-answer questions. The questionnaire aimed to assess the participant's motivation and attitudes in the AR CPS task (Glynn et al., 2011), the experience in the group collaboration, and the perception about system usability (Brooke, 1996). Specifically, there were 6 questions about the problem-solving process (problem-solving) that assessed the participant's motivation, interest, and curiosity in tackling problems and challenges in the CPS task; 5 questions about the programming content (task-content) that assessed the participant's desire to complete the task; 3 questions about group collaboration (collaboration) that assessed the participant's attitudes toward the collaboration with the partner; and 10 questions about the perception of system usability (usability). The three short-answer questions focused on the participant's opinions about using AR for the CPS task in programming learning, for example, "What do you think of pros and cons of AR in solving the programming task?" Because of the within-subjects design in the experiment, we adapted all the question texts to the formats, "Compared to the non-AR visualisation, the AR visualisation [describing a variable to measure]", which asked the participant to compare their experiences in AR and non-AR.

Results

The global and local error rates in the programming task

The participants composed and submitted computer code as an answer to the task. The task performance was measured by how much the code was different from the standard answer. We used *error rate* to denote this difference and adopted an edit-distance algorithm, with the longest common subsequence, for it. Longest common subsequence is commonly used in programming version control applications to measure the difference between code files. The error rate is 0.0 if an answer snapshot is identical to the standard solution and 1.0 if they are completely different.

Considering that the participants could solve the task by taking different paths (i.e., approaching the solution by trying combinations of code) and some of them could solve the task partially, we further derived two kinds of error rates from the longest common subsequence error rate: *global error rate* and *local error rate*. The global error rate (e_g) measures the error rate of high-level algorithmic designs, following the assumption that correct implementations share a similar high-level program structure. The local error rate (e_l) measures the error rate of implementation by comparing parts of the answer to their corresponding parts in the solution.

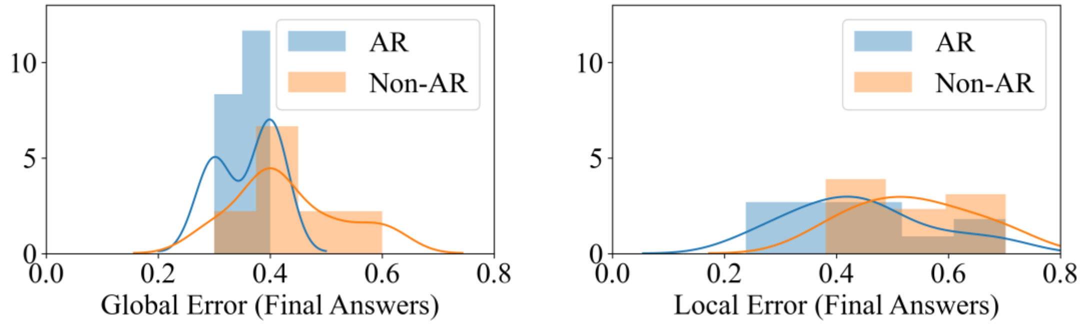


Figure 5. Distributions of the global error rate (e_g) and local error rate (e_l) of the final answers.

We computed e_g and e_l for the final answers submitted by the participants in AR and non-AR (Figure 5). A significant difference was found by a paired t-test in e_g between AR ($M = 0.36, SD = 0.05$) and non-AR ($M = 0.43, SD = 0.10$) ($t(11) = -2.28, p = 0.04$). There was no significant difference found in e_l. This result indicated that the participants solved the task more effectively in the AR version than in the non-AR version.

Effective and ineffective coding trajectories

The error rate did not reflect details about what the participants did in the CPS process, for example, the code-editing behavioru. We further analysed the code snapshots at a granular level: the code edits and their trajectories. A code edit is defined as a change which transforms one code snapshot to another one. For example, a participant changes a line of code from the command "moveForward" to "climbUp". This event, represented as a two-item tuple (moveForward, climbUp), is referred to as one code edit. Technically, two kinds of edits may occur in the task: *parameter tweaking* and *command tweaking*. The parameter tweaking *(P)* refers to the edit that changes the parameter part of a command. In this study, the parameter tweaking categorises the edit that changes the parameter of "continueSec" which works like a loop statement of programming languages that continues running a command for a while. Any other edit is categorised as command tweaking (C). The CPS process is essentially a sequence of code edits. To model such a sequence, we can focus on the transition between the code edits and use the hidden Markov model (HMM) to capture the dynamics in the participants' code-editing behavior. The result of HMM is referred to as coding trajectory in this study.

We applied HMM to the code sequences from the AR group and those from the non-AR group separately. The two HMMs were first initialised by sequences from all the participants following the assumption that all hidden states were independent. The parameters of the two HMMs were later learned separately by the expectation-maximisation (EM) algorithm. The analysis revealed five hidden states of code edits: PP (consecutive P's), CC (consecutive C's), PP-CC Mix (intermixed sequences of PP and CC), PC-CP Mix (intermixed sequences of PC and CP), and Others (any other sequences). The result models were visualised by their transitions with probability values as shown in Figure 6.

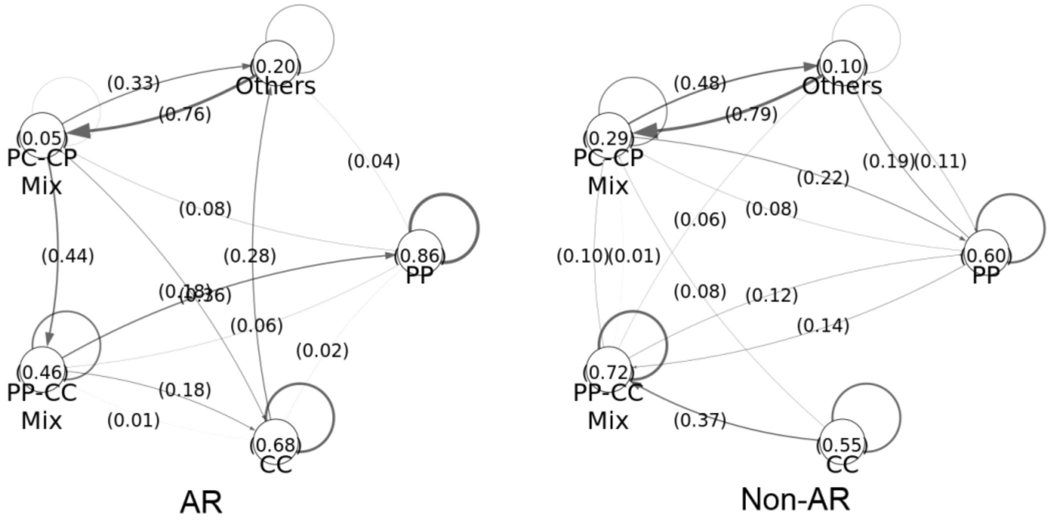


Figure 6. Coding trajectories from the AR and non-AR groups

One notable pattern in the AR group was a clear and unidirectional transition that traversed all hidden states along the path (Others, PC-CP Mix, PP-CC Mix, CC, Others). The non-AR group, however, did not reveal such a clear transition. Instead, we found that the transitions were more relatively random. This result showed that the participants in AR could edit their code steadily and more productively. In other words, the participants might proceed from muddling edits (e.g., PC-CP Mix) to relatively stable edits (e.g., CC, PP). On the contrary, such a clear coding trajectory could not be found in the Non-AR group. The relatively messy transition among the states indicated that the participants in Non-AR were not able to edit the code effectively. There could also be some random edits. To summarise, the comparison of coding trajectories of AR and non-AR from the HMMs suggested that the AR visualisation plausibly brought a positive impact on the participants' CPS process in the task.

Exploring semantics in the verbal communication

Similar to code edits, verbal communication can also be represented by a sequence of events. An event can be a word, a sequence, or a high-level topic (Dascalu et al., 2015). The sequence reflects the development of conversations and may reveal the CPS process at the cognitive level since the participants exchange ideas in a face-to-face discussion. We hypothesised that the participants could have different kinds of conversations that revealed their engagement and CPS process in AR and non-AR. To examine this hypothesis, we used the same HMM approach for the coding trajectory to model verbal communication.

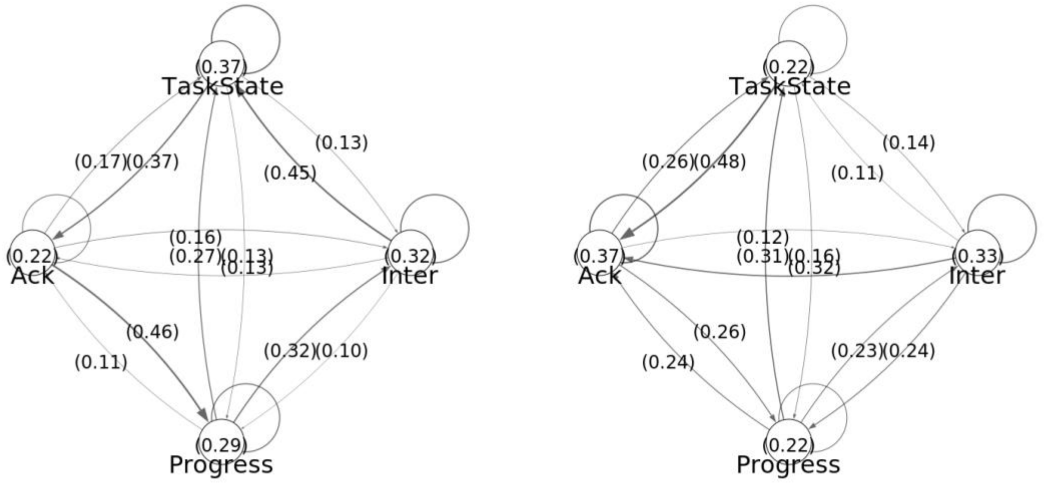


Figure 7. Communication trajectories from the AR and non-AR groups

With duplicates removed, there were 2920 unique sentences remained in the dataset, out of which were 472 (16.1%) sentences of TaskState, 856 (29.3%) of Inter, 1,099 (37.6%) of Progress, 250 (8.6%) of Ack, and 243 (8.3%) of Other. Our HMM analysis revealed four hidden states in the participants' verbal communication as shown in Figure 7. The states matched the four semantic labels in the conversational coding schema excluding Other. Comparing the transition diagrams, in the AR group we found a notable transition traversing along the cycle (TaskState, Ack, Progress, Inter, TaskState). Although the cycle also existed in the non-AR group, the probability was much lower than in the AR group. Similar to the coding trajectory, the existence of this cycle indicated that verbal communication between the participants in AR might be more efficient than in Non-AR. For example, the participants might have a conversation like this: a participant P1 mentioned, "...and your L8 it's going too far". The other participant P2 replied, "So [should it be the command] hover?".Then P1 said, "Yeah. I think [it should be] hover".

We also found that the transition from TaskState, Inter, Progress, to Ack was more likely to occur in the non-AR group than in the AR group. The tendency toward the state Ack indicated the verbal communication was probably less informative. In other words, the participants might not provide or perceive enough actionable feedback by which they could continue the discussion, for example, the transition from Ack to Progress. These findings provided another evidence showing that the CPS process was more efficient in AR.

User experience about the CPS process in AR

We evaluated the reliability of the questionnaire scores by Cronbach's alpha in each category. The result showed the level of reliability of the items was moderate to high with $\alpha(6) = .790$ in Problem-solving, $\alpha(5) = .779$ in Task-content, $\alpha(3) = .773$ in Collaboration, and $\alpha(10) = .630$ in Usability. The average of item scores was used to represent the summary score in each category. The result of both Cronbach's alpha and one-sample t-tests is shown in Table 1.

A one-sample *t*-test found that the average score of Problem-solving ($M = 4.205, SD = .701$) was significantly higher than the neural average with $t(25) = 8.762$ and $p = 0.00$. This result indicated that the participants had more positive attitudes toward the AR content and were more engaged in the CPS in AR than non-AR. This finding was cross-validated by their responses to the short-answer question about the benefits of AR in CPS. Some participants stated that the AR version was preferred, more useful, and comfortable to use than the non-AR one, mainly because they could see more details, for instance depth and height. Such details helped them observe the program outcome and improved their answers.

The analysis also found that the participants had a strong desire to finish the task in AR, as proved by the significant score in Task-content ($M = 4.000, SD = .754$; $t(25) = 6.756, p = .000$). Further, we found that the participants had positive attitudes toward their collaboration in AR in the Collaboration questions. The *t*-test showed that the score ($M = 3.961,\ SD = .896$) was significantly higher than the neutral score with $t(25) = 5.472,\ p = .000$. These findings suggested that AR was engaging for the participants, and they were willing to finish the CPS task in AR.

Table 1
The result of Cronbach's alpha and t-tests in all questionnaire categories

Category	Cronbach's alpha	*t*-test
Problem-solving	$\alpha(6) = .790$	$t(25) = 8.762; p = 0.00$
Task-content	$\alpha(5) = .779$	$t(25) = 6.756; p = 0.00$
Collaboration	$\alpha(3) = .773$	$t(25) = 5.472; p = 0.00$
Usability	$\alpha(10) = .630$	$t(25) = 7.050; p = 0.00$

System usability of the CPS task in AR

Although the AR was engaging and interesting to the participants, it was likely that the system was not easy to use. To measure the ease of use and the system usability of the AR app, we looked into the participants' responses to the Usability questions. The *t*-test found that the average score ($M = 3.727, SD = .526$) was significantly higher than the neutral score with $t(25) = 7.050,\ p = .000$. The value showed that the participants had positive attitudes toward the usability of the app. We noticed in the interview that the major concerns from the participants were about the stability of the app. Due to the state of networking, our app sometimes had delays in the runtime when synchronising the participants' code. Meanwhile, the AR visualisation required a certain amount of graphic processing resources to detect the image marker, which sometimes made the response speed even worse. However, we believe such an issue could be easily solved by better hardware.

Correlation analysis of the self-reported experience

To examine the relationships of the responses to different questionnaire categories, we computed the Pearson correlation coefficient for pairs of the categories (Table 2). The analysis found that Problem-solving was significantly, positively correlated with all the other categories. This result emphasised that the CPS process could be directly impacted by group collaboration, the content of the task, and system usability. For example, if a participant had a positive attitude toward their collaboration with the partner, it would be very likely for them to have a positive attitude toward the problem-solving process and vice versa.

It is noteworthy that Collaboration was not correlated with Usability nor with Task-content. First, this indicated that some defects in the AR app might not impact the participants' experiences in the CPS process. Second, this could mean that the design of AR for CPS should focus more on the task content and CPS steps.

Table 2
Pearson's correlation coefficients between the questionnaire categories. For readability, the upper triangle of the correlation matrix is omitted.

	Usability (U)	Collaboration (C)	Task-content (T)	Problem-solving (S)
U	1			
C	$r(26) = .226$ $p = .267$	1		
T	$r(26) = .558$ $p = .003*$	$r(26) = .323$ $p = .107$	1	
S	$r(26) = .464$ $p = .017*$	$r(26) = .526$ $p = .006*$	$r(26) = .600$ $p = .001*$	1

Note. *$p < 0.05$

Qualitative analysis of the interview

The participants were interviewed for 10-15 minutes at the end of the experiment sessions. The responses were transcribed and summarised into three categories about the students' experience: their behaviours, interests, and collaboration. Behaviours were about how the participants managed the task. Interest was about their desires to engage in the task. Collaboration was about their working conditions with the partners.

First, we examined the user experience related to behaviour. In the experiment, we observed that in 7 pairs out of 12, at least one participant voluntarily adjusted their posture to get a better picture of the task content. Although the participants were not explicitly told how they should use the AR app, they naturally learn how to use it to achieve a better result. In the interview, the participants attributed this to the feature of AR itself and said it made them adjust their posture unconsciously. For example, we received the responses: "Moving while using AR-enabled me to see the object more clearly.", "By going forwards and backward in AR, you can get greater focus." This result demonstrated that even without extra training, the AR visualisation did not put an extra load on the participants.

We also interviewed the participants about their working conditions in the CPS task with the support of AR. 19 participants out of 24 stated that they would prefer using AR despite having some concerns about system issues, for example, a delay in response time. They said that AR was more interesting and could catch the attention immediately. For example, participants reported:

- AR is more interactive… you can see and feel things… you flow with it.
- With AR, you easily become part of the game.
- AR is 3D ... you can move in-depth and clearly see what things actually look like.

Five participants did not prefer the AR version. Two stated that the AR and non-AR versions were mostly similar and they didn't feel evident differences. Three preferred the non-AR version and claimed that it was more stable while the AR version was more distracting.

Overall, the result indicated that AR could be used to engage most of the participants quickly in the activity. Nevertheless, some participants focused more on the efficiency and the CPS task. We considered this a design choice that instruction designers should ponder over according to the instructional objectives in the classrooms.

Finally, we interviewed the participants about their collaboration with and without AR. We asked more specific questions aimed at assessing their collaboration in the CPS task, for example, "Did you work together?", and "Please describe the way you collaborated with your partner and tackled the problem". Most of the participants suggested that they had similar roles in the CPS task, that is, having equal contributions in both the AR and non-AR versions. When asked about factors that facilitated or hindered their collaboration, many participants suggested the task design was important in supporting their collaboration. This result suggested that the presentation of the task content (AR or non-AR) might not directly affect the efficiency of collaboration. Instead, the effect of AR could implicitly emerge in enhancing task-relevant information.

Discussion

The analysis of task performance showed that participants using AR had a lower error rate of algorithmic design than non-AR. Also, using AR potentially helped the participants to compose their answers effectively as shown in the clear coding trajectories. In the CPS task, the easier-to-observe depth and height information can be one reason for the improvement. For example, when two participants work in the same workspace but from slightly different angles, the functionality of AR allows the participants to observe the task condition from different points of view (Dede, 2009). This feature may have helped the participants consume and supply information that may be ignored or missed when they work at a fixed angle. We believe this is one reason for the better task performance in the CPS programming task because the participants can easily spot potential flaws in their program via the alignment between the algorithm and the outcomes. This result adheres to findings in the literature, that is, researchers have shown that AR experiences in STEM learning can help students "consume information through the interaction with digital elements" (Ibáñez & Delgado-Kloos, 2018, p. 1).

The analysis also showed that participants had better group dynamics in terms of the topics discussed during the CPS task. When using the non-AR version, participants seemed to have less effective communication, for example, topics about making progress and interaction, than AR. We believe this effect can also be attributed to the better perception of information in AR as discussed above since the development of communication is highly correlated with the progress of CPS. This outcome was also reflected in the analysis of the post-test questionnaire, as well as the interview, where participants had more positive attitudes toward their collaboration and problem-solving in AR than non-AR. For example, some participants suggested that in AR they could see more detail and thereby better controlled the parameters in their program. Some participants also mentioned the immersive experience brought by AR, for example, "flying with it" that helped them obtain a better sense of the relationship between the task outcome and their algorithm. Overall, the findings of this study adhere to the literature where most studies show that AR improves group collaboration (Radu et al., 2021) and engagement (Ibáñez & Delgado-Kloos, 2018).

It should be noted that the findings may be correlated with the design of task content. Merely transforming digital content into the format of AR may not bring additional benefits other than perceptual differences. We identified several challenges and implications for future AR instructional designs as follows. First, the task design and how it utilises the characteristics of AR are more important than the AR visualisation per se. The instructional designer should focus on which AR feature can support a component in the learning content, instead of merely transforming whatever content into an AR visualisation. Second, the instructional activity may be enhanced by AR even though physical models are rarely used in the learning content. In this study, the task simulated a scenario where the depth information was meaningful. This design also made multi-angle observations in group collaboration meaningful.. We believe this design effectively utilises the characteristics of AR and, therefore, makes the participants engaged in the task. The design can also be used as a prototype for other ARLE designs, for example, tangible user interfaces (Radu et al., 2021; Radu & MacIntyre, 2009).

Overall, the findings imply that AR can potentially improve engagement, attitudes, and task performance in a CPS programming task. AR has the potential not only to enrich the content presentation to engage students further but also to promote efficient group discussion. These aspects are specifically important to programming learning because topics in this domain are relatively abstract, and students, especially younger students in K-12 education, may easily disengage from the learning activity. The use of AR can become one key component that improves students' perception and reaction to the content. AR can also be used to motivate interaction between students. This is aligned with social aspects of computing education promoted by some researchers (Kafai, 2016). We advise instruction designers who are interested in AR learning experiences to consider the potential of CPS programming tasks in their AR-enabled material.

Conclusions and limitations

This study investigated an AR instructional design for a CPS programming task. We designed an AR app where two users could collaboratively solve a programming task, and conducted a within-subjects experiment to evaluate the participants' task performance by multimodal data including error rates, code edits, and verbal communication. The results showed that the participants had lower error rates and more efficient communication when working in the AR version of the task than in the non-AR one. Furthermore, the analysis of the user experience suggested that the participants had positive attitudes toward the use of AR. The AR version was preferred and more engaging than the non-AR version.

There are several limitations in this study. At the hardware and technical level, the function and interaction of the AR app were limited due to the constraint of mobile tablets. Although the cost was much lower than advanced head-mounted display devices, the mobile tablets in this study could only provide image-based tracking and on-screen interaction with the content. These features limited the type of content that could be implemented in our app. For example, the image marker might affect how the participants collaborate in the task because the workspace was restricted to a small space.

At the data level, the sample in this study was relatively small (i.e., 24 students in 12 pairs) and from a specific group (i.e., college students in an entry level informatics course). This sample is not considered large enough to infer generalisation for the result. Since the data came from a within-subjects control experiment, whether or not the result remains the same or similar in a real-world classroom setting also remains unclear.

At the methods level, the analysis of this study focused on the specific task designed to investigate the use of AR for CPS programming tasks. The scope of the task content was limited in terms of computer programming. A future study might consider extending the content to other programming topics and examining whether the effectiveness of AR can transfer to not only broader topics but formal assessments in classrooms. In addition, the analysis of CPS did not involve much detail of interaction between the participants, for example: who initiated a discussion topic, how they responded to each other, or what the roles were during the collaboration. How to capture multiple transitions and their interactions requires further investigation.

References

Akçayır, M., & Akçayır, G. (2017). Advantages and challenges associated with augmented reality for education: A systematic review of the literature. *Educational Research Review*, *20*, 1–11. https://doi.org/10.1016/j.edurev.2016.11.002

Ashtari, N., Bunt, A., McGrenere, J., Nebeling, M., & Chilana, P. K. (2020). Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI*. 1–13. https://doi.org/10.1145/3313831.3376722

Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, *6*(4), 355–385. https://doi.org/10.1162/pres.1997.6.4.355

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2012). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(4–5), 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Brooke, J. (1996). SUS - A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4-7. https://doi.org/10.1201/9781498710411-35

Bujak, K. R., Radu, I., Catrambone, R., MacIntyre, B., Zheng, R., & Golubski, G. (2013). A psychological perspective on augmented reality in the mathematics classroom. *Computers & Education*, *68*, 536–544. https://doi.org/10.1016/j.compedu.2013.02.017

Chiang, T. H. C., Yang, S. J. H., & Hwang, G.-J. (2014). Students' online interactive patterns in augmented reality-based inquiry activities. *Computers & Education*, *78*, 97–108. https://doi.org/10.1016/j.compedu.2014.05.006

Dascalu, M., Trausan-Matu, S., Dessus, P., & McNamara, D. S. (2015). Discourse cohesion: A signature of collaboration. *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, Poughkeepsie, NY*. 350–354. https://doi.org/10.1145/2723576.2723578

Dede, C. (2009). Immersive interfaces for engagement and learning. *Science*, *323*(5910), 66–69. https://doi.org/10.1126/science.1167311

Fidan, M., & Tuncel, M. (2019). Integrating augmented reality into problem based learning: The effects on learning achievement and attitude in physics education. *Computers & Education*, *142*, 1–29. https://doi.org/10.1016/j.compedu.2019.103635

Fussell, S. R., Kraut, R. E., & Siegel, J. (2000). Coordination of communication. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, *Philadelphia*, *PA*, 21–30. https://doi.org/10.1145/358916.358947

Glynn, S. M., Brickman, P., Armstrong, N., & Taasoobshirazi, G. (2011). Science motivation questionnaire II: Validation with science majors and nonscience majors. *Journal of Research in Science Teaching*, *48*(10), 1159–1176. https://doi.org/10.1002/tea.20442

Ibáñez, M. B., & Delgado-Kloos, C. (2018). Augmented reality for STEM learning: A systematic review. *Computers & Education*, *123*, 109–123. https://doi.org/10.1016/j.compedu.2018.05.002

Institute of Electrical and Electronics Engineers. (2020). *IEEE standard for augmented reality learning experience model* (IEEE Std 1589-2020). IEEE Computer Society. https://doi.org/10.1109/IEEESTD.2020.9069498

Jeong, H., Hmelo-Silver, C. E., & Jo, K. (2019). Ten years of computer-supported collaborative learning: A meta-analysis of CSCL in STEM education during 2005-2014. *Educational Research Review*, *28*, 1–17. https://doi.org/10.1016/j.edurev.2019.100284

Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, *59*(8), 26–27. https://doi.org/10.1145/2955114

Liou, H. H., Yang, S. J. H., Chen, S. Y., & Tarng, W. (2017). The influences of the 2D image-based augmented reality and virtual reality on student learning. *Educational Technology and Society*, *20*(3), 110–121. https://www.j-ets.net/collection/published-issues/20_3

Martín-Gutiérrez, J., Fabiani, P., Benesova, W., Meneses, M. D., & Mora, C. E. (2015). Augmented reality to promote collaborative and autonomous learning in higher education. *Computers in Human Behavior*, *51*, 752–761. https://doi.org/10.1016/j.chb.2014.11.093

Radu, I. (2014). Augmented reality in education: A meta-review and cross-media analysis. *Personal and Ubiquitous Computing*, *18*(6), 1533–1543. https://doi.org/10.1007/s00779-013-0747-y

Radu, I., Hv, V., & Schneider, B. (2021). Unequal impacts of augmented reality on learning and collaboration during robot programming with peers. *Proceedings of the ACM on Human-Computer Interaction*, *4*(CSCW3), 1–23. https://doi.org/10.1145/3432944

Radu, I., & MacIntyre, B. (2009). Augmented-reality scratch: A children's authoring environment for augmented-reality experiences. *Proceedings of the 8th International Conference on Interaction Design and Children, Como*. 210–213. https://doi.org/10.1145/1551788.1551831

Radu, I., & Schneider, B. (2019). What can we learn from augmented reality (AR)? *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow*. 1–12. https://doi.org/10.1145/3290605.3300774

Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. O'Malley (Ed.), *Computer supported collaborative learning* (pp. 69–97). Springer. https://doi.org/10.1007/978-3-642-85098-1_5

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Santos, M. E. C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., & Kato, H. (2014). Augmented reality learning experiences: Survey of prototype design and evaluation. *IEEE Transactions on Learning Technologies*, *7*(1), 38–56. https://doi.org/10.1109/TLT.2013.37

Schez-Sobrino, S., García, M., Lacave, C., Molina, A. I., Glez-Morcillo, C., Vallejo, D., & Redondo, M. (2021). A modern approach to supporting program visualization: From a 2D notation to 3D representations using augmented reality. *Multimedia Tools and Applications*, *80*(1), 543–574. https://doi.org/10.1007/s11042-020-09611-0

Shibani, A., Koh, E., Lai, V., & Shim, K. J. (2017). Assessing the language of chat for teamwork dialogue. *Educational Technology and Society*, *20*(2), 224–237. https://www.j-ets.net/collection/published-issues/20_2

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Wei, X., Weng, D., Liu, Y., & Wang, Y. (2015). Teaching based on augmented reality for a technical creative design course. *Computers & Education*, *81*, 221–234. https://doi.org/10.1016/j.compedu.2014.10.017

Corresponding author: Cheng-Yu Chung, Cheng.Yu.Chung@asu.edu

Please cite as: Chung, C-Y., Awad, N., & Hsiao, I-H. (2021). Collaborative programming problem-solving in augmented reality: Multimodal analysis of effectiveness and group collaboration. *Australasian Journal of Educational Technology, 37*(5), 17-31. https://doi.org/10.14742/ajet.7059