Future of Digital Education - An intelligent tutoring system for teaching and learning computational thinking through programming.

Computational thinking, problem-solving and programming

As we look to educate our young people who are digital natives and are born into a world with advanced technology, they are expected to be producers and not just consumers. The nature of computer science is constantly changing due to it being dynamic, flexible and innovative (Kalelioglu, 2015) which has resulted in a necessity to change education to adapt to developments in the field (Grout & Houlden, 2014). This view is amplified by the Computer Science Teachers Association (CSTA) in their 2017 standards where they view computer science and technology as being at the heart of the economy and the way we live our lives, and that students need to be prepared for a computing intensive society by having a clear understanding of computer science principles and practices.

Science, technology, engineering and mathematics (STEM) education is inter-disciplinary and is aimed at solving problems that can occur in everyday life (Gökçe & Yenmez, 2023). Wing (2014) defined computational thinking as the thought processes involved in formulating a problem and expressing its solutions in a way that a human or machine can carry out. Computational thinking is recognised as a unique problem-solving skill from computer science concepts that can be applied to solving a variety of problems (Shute, Sun, & Asbell-Clarke, 2017). The computational thinking process does not only prepare students for learning computer science and programming, but also provides them with skills and tools to approach and solve problems in different areas of knowledge (Werner, Denner, Campe, & Kawamoto, 2012).

Goals

This essay aims to plan a digital learning tool that will teach computational thinking skills to support problem solving through programming, with the long-term goal of educating young people to be producers and adaptive individuals in an advanced technological world. The learning tool will be planned for the future in the year 2028 and will incorporate new digital technologies to support the learning and environment. The digital learning tool will achieve its aims by completing the following goals which enable learners to:

- developing computational thinking skills
- learn programming concepts
- improve problem-solving skills
- acquire cognitive skills.

The digital learning tool will be targeted at secondary school students (aged 11-16) who are either beginning to learn computational thinking and how to program, or those who are undertaking computer science courses.

AI in education (AIEd)

The rapid development of artificial intelligence (AI) in all areas of our lives is beginning to have a major impact on education and has significant potential to support the achievement of the Sustainable Development Goals (SDGs) of the United Nations (Miao, Holmes, Huang, & Zhang, 2021). AI can be defined as computer systems that have been designed to interact with the world through capabilities and intelligent behaviours that we would think of as being human in nature (Luckin & Holmes, 2016). However, due to the ever-changing and inter-disciplinary nature of the area, there are many other definitions of AI. Loder & Nicholas (2018) define AI as computers which perform cognitive tasks that are usually associated with human minds, particularly learning and problem-solving.

AIED stands for Artificial Intelligence in Education. At the centre of AIED is a goal to “make computationally precise and explicit forms of educational, psychological and social knowledge which are often left implicit” (Self, 1999). Luckin & Holmes (2016) describe AIED as a powerful tool which gives us a deeper understanding of how learning actually happens, and this information can be used to develop future AIED software.

Intelligent tutoring systems

There are practical limitations for educators when providing individual feedback to large classes, such as time and workload (Buchanan, 2000). A study by Bloom (1984) found that 97% of students taught privately learned approximately two standard deviations more than students in a traditional classroom. An Intelligent Tutoring System (ITS) could address this problem as they are an individualised learning system which is similar to instruction by a one-to-one teacher and student situation (Hooshyar, et al., 2018). An investigation by Hooshyar, et al (2018) proposed a solution-based ITS to improve problem-solving and programming skills. It provided students with a personalised and adaptive environment for developing flow-charts to solve a programming problem. A Bayesian network handles the decision-making processes to enable the system to track the user’s knowledge and skills so the learning experience can be personalised for each individual. The results of the study showed the system contributed to both a higher level of achievement in problem-solving tasks as well as a boost to the learning attitude and interest of students. A criticism of this study is the small sample of participants (32) which could make it more difficult to claim the results are significant.

Ma et al (2014), define an ITS as a computer program that models learners' psychological states to provide individualised instruction. Their meta-analysis claims that ITS are effective tools for learning and are associated with greater achievement when compared to traditional methods of instruction. This outcome is reinforced by other studies that also claim ITS are more effective learning tools than teacher instruction and printed materials (Steenbergen-Hu & Cooper, 2014). However, we must question whether the positive effect of ITS on achievement over traditional instruction is due to similarities with other forms of computer-based instruction. To back-up this criticism, a meta-analysis by Tamim et al., (2011), which compared computer-based instruction with traditional instruction, found a similarly positive effect for using a computer to learn, possibly due to greater individual interaction with the program and the content or activities. The reviews in favour of ITS do however evidence a counter-argument for this criticism and conclude that using ITS is more beneficial than non-ITS computer-based instruction. Their reasoning for this is because ITS uses multidimensional modelling to individualise task selection for each student by matching their knowledge levels to the tasks in the knowledge domain. The system can monitor a student's progress, use the model to individualise learner-control options, provide more individualised learner feedback and interact with the learner as they construct answers (Ma, Adesope, Nesbit, & Liu, 2014).

Many examples of computer-based instruction are successful in delivering information to learners. The ITS plays a role in a more critical part of learning by involving the co-constructing of responses in specific tasks or activities such as solving problems, creating artifacts or answering challenging questions (Graesser, Hu, & Sottilare, Intelligent Tutoring Systems, 2018). To support this part of learning, the ITS needs to act similarly to a human tutor working one-to-one with a student, so a key part of the ITS are the interactions between the system and the student (EMT dialogue - Expectation and Misconception-tailored dialogue). Graesser et al. (2009) suggested a five-step dialogue frame the ITS must follow to support interaction and learning. After a question or problem is selected, the five-step process begins: (1) task is presented, (2) student constructs first attempt at task, (3) tutor gives short feedback on answer quality, (4) tutor and student interact to improve quality of answer, (5) tutor assesses student's understanding of correct answer and follows-up if required. Step four is an essential element in this framework to secure the student's learning and understanding. The ITS will have a list of good answers and a list of misconceptions in the knowledge base which it can utilise to provide scaffolding, prompts and hints to help the learner to solve the problem or question. The system will monitor the student's progress through this and use the data to help select the next task to personalise the learning to develop the student's learning.

Theoretical Understanding

Constructionism and constructivism

Constructionism is a learning theory which builds on the constructivist viewpoint that learners socially construct knowledge through creating objects (Vygotsky & Cole, 1978). Constructionism centres around the idea that, "learning happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity" (Papert & Harel, 1991). Papert (1980) encouraged a constructionist approach to learning Logo programming in schools that would help develop strong thinking skills in children, but on the contrary, a study by Kurland et al, (1986) found that programming using Logo did not improve children's thinking skills. The main element of constructionism which differentiates it from constructivism is that it is, "learning by constructing knowledge through the act of making something shareable" (Martinez & Stager, 2013).

The popularity of a maker movement or maker culture has increased in recent years with the development of technologies to support this. Halverson & Sheridan (2014) describe the maker movement as, “people who are engaged in the creative production of artifacts in their daily lives and who find physical and digital forums to share their processes and products with others”. The activity of problem-solving through programming involves applying computational thinking skills through the construction of artifacts (Resnick, et al., 2009). The Scratch programming language is a fine example of constructionism in the subject of computing that enables learners to construct their own programs to solve problems or create games for example. The intelligent tutoring system to support the learning of computational thinking through programming is underpinned by constructionist theory.

Kahn and Winters (2021) highlight a relationship between constructionism and AI, believing that teachers and students should be exposed to the positive and negative aspects of AI through constructionist approaches. There are many opportunities for younger students to explore and use AI in their projects such as the MIT App Inventor which enable students to embed trained classification models into their project and include AI tutorial lessons (MIT, 2023).

Collaborative learning

A collaborative problem-solving environment incorporated into the ITS is a form of Computer Supported Collaborative Learning (CSCL). Social constructivist theory argues that collaboration is promoted by CSCL which enables learners to co-construct knowledge and make meaning (Lazonder, Wilhelm, & Ootes, 2003). Vygotsky & Cole (1978) stated that learning will take place when the child is interacting with people in their environment and when cooperating with peers. Deal (2009) commented that using technology within collaborative learning can increase active participation and communication among students and help foster a more equal distribution of voices. Engaging individual learners to actively share their prior knowledge to make sense of new concepts will help to construct new knowledge and understanding (Du, Rosson, & Carroll, 2012).

The collaborative part of the ITS is an example of networked learning as it is a communication tool which allows for communication between learners. One theory that has grown in recent years is a community of inquiry where individuals engage in critical enquiry and reflection to help construct a personal knowledge and understanding (Garrison, 2022). This is a form of metacognition as the critical thinking and inquiry is affirmed by learners having the awareness and ability to take responsibility and control the construction of knowledge and meaning (Akyol & Garrison, 2011)

Metacognition

In recent years there has been a greater emphasis on the benefits of developing metacognitive /skills and reflective learning. Evidence suggests that it could be beneficial to include reflection activities within the ITS by asking the students to review and reflect on their programming experience. This will push learners to think about their programming process and develop computational practices and perspectives (Lye & Koh, 2014). For the ITS, a key design element will be to create reflection tasks that will have the greatest impact on…

A study into developing student metacognition through reflective writing was conducted by O'Loughlin and Griffith (2020). They examined data from undergraduate students in three semesters to assess the development of metacognitive skills of the students on a human anatomy course, using a reflective blog, where they had to assess points with regards to a patient problem. They concluded (p.691) that students "gained better awareness of their knowledge strengths and deficits, demonstrated greater self-confidence in their learning capability, and tended to have reduced anxiety and frustration regarding learning the material as the semester progressed”. They further commented that “late semester writing showed that students developed a more accurate assessment of their anatomic and medical imaging knowledge and skill, which provided evidence that the students were developing strong metacognitive processes”.

Some commentators have identified potential problems and challenges with blogging and reflective writing. The variety of reflective learning tasks can make it difficult for educators to choose which are appropriate for their particular students. It is difficult to implement blogging successfully when the students have very little experience which is often minimal and shaped by their blogging experiences and preconceptions (Kerawalla, et al., 2007). As a consequence of the potential for blogging in education combined with the acknowledged challenges, Kerawalla et al (2009) developed a framework to guide blogging in higher education that could also be used to facilitate reflection and cognitive skills. A scaffolded approach to student reflection was carried-out by Mair (2012) who utilised a spreadsheet to simplify a reflection process that had a structured approach to focus on how the students learn, not what they learn. The pilot study had some limitations but was successful in its aims. It concluded, “the structured approach was assessed as more systematic, less time consuming, and more constrained (focused) and accessible than the process of recording reflections in a traditional hard-copy journal or online blog.” (Mair, 2012; 164). The studies show that in order to achieve the benefits of a metacognitive approach in active learning, then specific tasks such as reflective writing need to be structured and scaffolded in order to have a positive impact in developing the understanding of concepts, otherwise they can just become meaningless assignments that do not result in positive transfer of knowledge and skills.

Focusing on the potential of reflective activities in the ITS for learning computational thinking through programming, the activities must lean towards a learner reviewing their own programming and learning performance. Robertson (2011) conducted a study whereby learners constructed their own program and blogged about the programming experience in Second Life. They discovered that a student's problem-solving capabilities in particular areas such as iteration and incrementation were supported by a blogging experience. From personal experience, asking students to revisit their code and comment on the key lines to explain the purpose of them helps to gain a deeper understanding of the programming and computational thinking within. Kyungbin and Jonassen (2011) also noted in their study that students who engaged in self-reflection were supported in testing and debugging their code.

The system

This section will describe the proposed solution to the learning goals identified previously. To achieve the learning goals, an intelligent tutoring system is proposed for teaching and learning computational thinking through programming. The web-based system will require users to login which enables the system to personalise the learning to each individual, as well as save their progress and data to their profile. The class teacher is provided with the data and summaries of student's progress. The ITS will have a built-in compiler which means that the users will not need to use a separate Integrated Development Environment (IDE) to run the code or view any errors in the code.

The ITS will cater for a few of the most common programming languages used, such as Python and Java. For each programming concept or topic area, there will be tutorials that utilise a combination of interactive screens and video to teach programming concepts and computational thinking skills. Within these sections will be a variety of review questions to check the student's understanding of the content as they progress.

Exercises and problem-solving challenges are created for each individual by the system based on their strengths and weaknesses in the topic area. Artificial intelligence automatically generates the personalised exercises and problems and works with the student to provide prompts and continuous feedback (such as helping to debug the code and find errors). The AI system is able to assess the student's submissions to exercises and problems, provide feedback on how the code could be improved, and utilise the student's work to personalise further learning to target specific areas of weakness.

It is beneficial here to explain the activities associated with the learning with an example of how the system will work. In this example the topic is the selection statement, often know as an IF/ELSE statement which is a basic programming function that enables simple decisions to be made. The student selects this topic and works through a series of navigable screens that provide instruction in the form of readable content, video tutorials and interactive examples to illustrate how the programming concept works. The student works through the content and completes review questions at regular intervals to check their knowledge and understanding of the concept.

The ITS knowledge base will contain a vast array of problem-solving exercises related to the selection statement. The system will provide the student with appropriately challenging problems to solve and they will type their program solutions using the interface provided. At this stage the conversational agent will begin to function and will converse with the student as they work and input their code. It may start by asking the student a question such as “Do you need to declare any variables?” which will prompt the student to think about what variables are required in the program. With the advancement of AI technology and machine learning, it should be possible for the agent to ask questions such as “How are you going to solve this problem?”. It will then listen to the student's response and react in a similar way to a human tutor. Eventually the student will complete the exercise and submit their program code for written feedback from the ITS. This process will continue with the system selecting more challenging questions to develop the student's understanding of the selection statement to the required level.

In terms of the technicalities behind the AI system, a range of advanced algorithms will be used to develop the machine learning capabilities of the system to ensure that the ITS is as intelligent as a human tutor, and also more efficient than a human tutor in all parts of teaching and learning. I would anticipate using the k-nearest neighbour algorithm for some parts of the system. This algorithm is commonly used in recommender systems used by major video streaming services, music streaming and online shopping websites. In a similar way to how it is used in these applications, the recommender system will provide content, learning resources and exercises or problems based on many factors such as their performance through the system content and their learner profile. It is necessary support the learners in moving through the content at an appropriate pace in order to meet the personalised learning goals of each individual student.

In order to address some possible limitations of the system when comparing with a human tutor, the technology available in the near future should allow the incorporation of a conversational agent to improve similarity of interactions between the ITS and student. The AI-based technology will enable the ITS and the student to have a purposeful dialogue with similarities to a conversation between a human tutor and a student by simulating an EMT (Expectation and Misconception-tailored) dialogue that evidently supports learning. This also links closely with developing metacognitive skills of the students by the system questioning the student's thought processes and encouraging them to reflect on their work or ideas. The conversational agent will be available when a student is working on a programming exercise and it will provide scaffolding to guide the student as well as appropriately timed continuous feedback to encourage and push the learner towards a correct solution, similar to the manner in which a human tutor would work one-to-one with a student.

The ITS can allow individuals to work collaboratively with other students by completing joint exercises or solving problems together. Again, the AI systems can utilise the data from all group participants to personalise the learning and provide feedback to the group. The conversational agent can interact with the group of students in a similar way to a human tutor in terms of scaffolding and providing immediate feedback to the group. The collaborative element of the ITS will also enhance learning further by taking advantage of the benefits of social learning mentioned previously.

Finally, the ITS will include reflection activities to help improve metacognitive skills and support a deeper understanding of the programming concepts. For example, after completing a problem-solving exercise the ITS will ask the student to comment on key lines of code to explain how they work, or it may ask the student to reflect on their performance when completing an exercise which the ITS identified them as having difficulty on, to see if they can reflect on the experience and use it to tackle future problems in a better way.

Benefits of the ITS

One of the main benefits of using the ITS is the fact that it saves time and effort for the teacher when providing feedback to a large number of students. Feedback will be personalised for each student to a greater level than if it was provided by the teacher as the ITS can quickly scan all of the data it holds about the student’s performance, processes, verbal responses, and reflections to identify their strengths and weaknesses.

The ITS will ensure content is not re-presented and the questions or exercises are unique to learner. This aspect reduces the chance of boredom due to exercises not being set at the appropriate level and also helps the student to make steady, continuous progress at the correct pace in order to achieve their personalised learning goals.

The ITS is likely to be better at assessing each student's knowledge, understanding and misconceptions. Human teachers are generally very good at assessing what each student does know and does not know, but rarely know the student's misconceptions and false beliefs (Chi, Siler, & Jeong, 2004). An ITS will be able to use a vast amount of data about the student to identify and address misconceptions.

The ITS will be able to select appropriate tasks more effectively than a human teacher. A study by Chi, Roy and Hausmann (2008) found that human teachers select tasks for students from a list that are sequenced in order of difficulty from those that are perceived to be more simple to those that are more challenging. The students then work through the sequence of tasks and their previous knowledge or misconceptions are not taken into account. The ITS will be able to use the student's data to provide tasks that are at the appropriate level to develop their knowledge and understanding of a concept.

How technology can overcome any disadvantages

It could be argued that human tutors are more effective with the timing of their feedback to support reasoning by monitoring their progress through a task or by asking the student to talk about how they are solving the problem. Human tutors can instantly help the student if they are having problems, and they can also ask the students to explain their reasoning as they construct a solution which enables them to quickly intervene if they hear or see something incorrect (Merrill, Reiser, Ranney, & Trafton, 1992). In terms of the ITS being able to replicate this level of human characteristic and interaction, a highly complex AI system potentially combined with an advanced conversational agent or chatbot would be needed to help identify any issues the student is having with a task at an earlier stage. There could be further issues with using a conversational agent to interact with the student while they are working through a problem, for example, how likely is a student to respond to an open question from the system asking about their thought process?

In a similar way, the ITS may struggle with helping to scaffold a student's reasoning and thinking. A human tutor is able to scaffold with a guided prompt that pushes the student towards a solution using their own thinking instead of giving direct feedback or more information (Chi et al, 2001). Humans tutors are able to quickly think and react to the student's responses, thoughts and ideas by asking questions to get the student to think about how to solve the problem based on what they already know. This again would be a challenging interaction for the ITS and would likely require an advanced conversational agent to accomplish a similar experience with the student.

One of the main challenges faced when designing and developing an ITS that teaches programming is the fact that a programming exercise can have many different solutions (Weragama & Reye, 2014). The ITS can take advantage of developments in big data to hold a vast knowledge base that utilises AI concepts to model many solutions to programming exercises. This will enable the system to assess a student's programs and provide valuable feedback.

Challenges

One of the major hurdles to the development and deployment of ITSs is the amount of time required and the large amount of funding needed to create them due to the sheer complexity of the systems (Graesser, Hu, & Sottilare, 2018). An ITS system like the one outlined in this essay, that promotes deep learning, incorporates collaborative learning and conversational agents, would take many years to develop and would cost millions of pounds. There will be a lot of difficulties in creating an effective ITS for this subject due to a range of reasons. It could prove difficult to combine the expertise of teachers with the skills of the system developers. In an ideal world the creators of the system would have the expertise in the system content but in reality this is highly unlikely. Focussing on the conversational agent, it would be extremely challenging to train the machine to be able to understand and respond to the learner's responses and reasoning, although with the development of AI-based chatbots, the agents are more likely to be able to simulate a realistic conversation, similar to one between a human teacher and student, by the year 2028.

Ethical issues with AIEd

As well as creating enormous opportunities for education, AI also carries some challenges which need to be carefully governed in an ethical and trustworthy way. Machine learning requires massive sets of data so it is essential that this private data is kept extremely safe and secure in line with data protection laws. Vast amounts of data are used to train machine learning systems. If the training data is biased in any way then this will result in algorithms and systems that are also bias that can lead to detrimental effects such as prejudice, discrimination or unrepresentative data. The algorithms are created by humans and humans have intentional or unintentional bias that will be reflected in the algorithms and systems they create (Baker & Smith, 2019). As well as the need to keep large quantities of data secure and private, there are concerns with surveillance. One such concern was identified in the report *Intelligence Unleashed* (Luckin & Holmes, 2016) which presented the chance of AI teaching assistants secretly monitoring the performance of the teacher. These are just a few brief examples of ethical issues with AIEd, there are many more that would require a far deeper analysis into the potential issues.

Conclusion

An Intelligent Tutoring System to teach computational thinking skills to support problem-solving through programming was proposed, and I believe the evidence suggests that the ITS will be successful in achieving the learning goals set out. It is clear that the activities used in the ITS will develop programming, problem-solving and computational thinking skills by using constructionist methods of teaching such as creating a program solution to a problem. Reflective activities through self-reflection and conversation with an AI agent will help learners to acquire and improve cognitive skills. The benefits of applying the latest AI-based machine learning technology to the system play a critical role in the personalisation of the learning experience to meet the needs of each individual student and the evidence explored highlights many key benefits to the learners.

I believe that in the near future, the ITS will have the potential to out-perform human teachers in effective teaching and learning. For this to become reality there will need to be a vast amount of research and funding to support a long-term strategy that will utilise the machine learning algorithms required to present a near perfect human-like interactive system. Critics will argue that students who are learning independently using an ITS will not learn as effectively as those learners who work collaboratively and experience the benefits of social learning. With the continuous rapid developments in AI technology and machine learning, the potential will be there to incorporate collaborative activities and challenges that enable groups of learners to work together to solve problems, whether that be in computational thinking and programming, or any other subject. The technology will be able to overcome all of the disadvantages that were identified.

Another key element of an advanced ITS like the idea proposed becoming reality are the challenge of combining the expertise of the teachers with the skills and knowledge of the software developers. Additionally a few ethical issues were identified and these only scratch the surface of an ever-growing register of possible threats in terms of data, privacy and security. We could question whether the governance of the latest technology is dangerously falling behind the rapid developments.

References

Akyol, Z., & Garrison, D. (2011). Assessing metacognition in an online community of inquiry. *Internet and Higher Education, 14*, 183-190. doi:10.1016/j.iheduc.2011.01.005

Baker, T., & Smith, L. (2019). *Educ-AI-tion rebooted? Exploring the future of artificial intelligence in schools and colleges.* London, UK: Nesta.

Bloom, B. (1984). The search for methods of group instruction as effective as one-to-one tutoring. *Educational Leadership, 41*, 4-17.

Buchanan, T. (2000). The efficacy of a world-wide web mediated formative assessment. *Journal of Computer Assisted Learning, 16*, 193-200.

Chi, M. T., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and instruction, 22*, 363-387.

Chi, M., Roy, M., & Hausmann, R. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science, 32*, 301-342.

Chi, M., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-533.

Computer Science Teachers Association . (2023, 06 03). *K-12 Computer science standards, revised 2017*. Retrieved from Computer Science Teachers Association : https://www.csteachers.org/page/standards

Deal, A. (2009). Teaching with technology: Collaboration tools.

Du, H., Rosson, M., & Carroll, J. (2012). Communication patterns for a classroom public digital backchannel. *Proceedings of the 30th ACM international conference on Design of communication (SIGDOC '12)* (pp. 127-136). New York: Association for Computing Machinery.

Garrison, D. (2022, June 11). *CoI Framework*. Retrieved from CoI: https://coi.athabascau.ca/coi-model/

Gökçe, S., & Yenmez, A. A. (2023). Ingenuity of scratch programming on reflective thinking towards problem solving and computational thinking. *Education and Information Technologies, 28*(5), 5493-5517. doi:https://doi.org/10.1007/s10639-022-11385-x

Graesser, A., D'Mello, S., & Person, N. (2009). Meta-knowledge in tutoring. In D. Hacker, J. Dunlosky, & A. Graesser, *Metacognition in educational theory and practice* (pp. 361-382). Mahwah, NJ: Erlbaum.

Graesser, A., Hu, X., & Sottilare, R. (2018). Intelligent Tutoring Systems. In F. Fischer, C. Hmelo-Silver, S. Goldman, & P. Reimann, *International Handbook of the Learning Sciences* (pp. 246-255). Abingdon: Routledge.

Grout, V., & Houlden, N. (2014). Taking computer science and programming into schools: The Glyndwr/BCS Turing Project. *Procedia – Social and Behavioral Sciences, 1141*(25), 680-685.

Halverson, E., & Sheridan, K. (2014). The maker movement in education. *84*(4), 495-504.

Hooshyar, D., Ahmad, R., Yousefi, M., Fathi, M., Horng, S., & Lim, H. (2018). SITS: A solution-based intelligent tutoring system for students' acquisition of problem-solving skills in computer programming, Innovations. *Education and Teaching International, 55*(3), 325-335.

Kahn, K., & Winters, M. (2021). Constructionism and AI: A history and possible futures. *British Journal of Educational Technology, 52*, 1130-1142.

Kalelioglu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in human behaviour, 52*, 200-210. doi:http://dx.doi.org/10.1016/j.chb.2015.05.047

Kerawalla, L., Minocha, S., Conole, G., Kirkup, G., Schencks, M., & Sclater, N. (2007). Exploring students' understanding of how blogs and blogging can support distance learning in higher education. *ALT-C 2007: Beyond Control: Association of Learning Technologies Conference.* Nottingham, UK.

Kerawalla, L., Minocha, S., Kirkup, G., & Conole, G. (2009). An empirically grounded framework to guide blogging in higher education. *Journal of Computer Assisted Learning*, 31-42.

Kurland, D., Pea, R., Clement, C., & Mawby, R. (1986). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research, 2*(4), 429-458.

Kyungbin, K., & Jonassen, D. (2011). The influence of reflective self-explanations on problem-solving performance. *Journal of Educational Computing Research, 44*(3), 247-263.

Lazonder, A., Wilhelm, P., & Ootes, S. (2003). Using sentence openers to foster student interaction in computer-mediated learning environments. *Computers & Education, 41*, 291-308.

Loder, J., & Nicholas, L. (2018). *Creating a people-powered future, for AI in health: confronting Dr Robot.* Nesta Health Lab. Retrieved June 08, 2023, from https://media.nesta.org.uk/documents/confronting_dr_robot.pdf

Luckin, R., & Holmes, W. (2016). *Intelligence Unleashed: An argument for AI in Education.* London: UCL Knowledge Lab.

Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behaviour, 41*, 51-61.

Ma, W., Adesope, O., Nesbit, J., & Liu, Q. (2014). Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis. *Journal of Educational Psychology, 106*. doi:10.1037/a0037123

Mair, C. (2012). Using technology for enhancing reflective writing, metacognition and learning. *Journal of Further and Higher Education, 36*(2), 147-167.

Martinez, S., & Stager, G. (2013). *Invent to learn: Making, tinkering and engineering in the classroom.* Constructing modern knowledge press.

Merrill, D., Reiser, B., Ranney, M., & Trafton, J. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *tutoring techniques: A comparison of human tutors and intelligent tutoring, 2*, 277-306.

Miao, F., Holmes, W., Huang, R., & Zhang, H. (2021). *AI and education Guidance for policymakers.* UNESCO.

MIT. (2023). *Artificial Intelligence with MIT App Inventor*. Retrieved from MIT App Inventor: http://appinventor.mit.edu/explore/ai-with-mit-app-inventor

O'Loughlin, V., & Griffith, L. (2020). Developing student metacognitino through reflective writing in an upper undergraduate anatomy course. *Anatomical sciences education, 13*(6), 680-693.

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas.* New York: Basic Books.

Papert, S., & Harel, I. (1991). *Constructionism.* Ablex Publishing Corporation.

Resnick, M., Maloney, J., M.-H. A., Rusk, N., E. E., & B. K. (2009). Scratch programming for all. *Communications of the ACM, 52*(11), 60-67.

Robertson, J. (2011). The educational affordances of blogs for self-directed learning. *Computers & Education, 57*(2), 1628-1644.

Savery, J. (2006). Overview of Problem-based Learning: Definitions and Distinctions. *Interdisciplinary Journal of Problem-based Learning, 1*(1).

Self, J. (1999). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence in Education, 10*, 350-364.

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

Steenbergen-Hu, S., & Cooper, H. (2014). A meta-analysis of the effectiveness of Intelligent Tutoring Systems (ITS) on college students' academic learning. *Journal of Educational Psychology, 106*, 331-347. doi:10.1037/a0034752

Tamim, R., Bernard, R., Borokhovski, E., Abrami, P., & Schmid, R. (2011). What forty years of research says about the impact of technology on learning: A second-order meta-analysis and validation study. *Review of Educational Research, 81*, 4-28. doi:10.3102/0034654310393361

Torp, L., & Sage, S. (2002). *Problems as possibilities: Problem-based learning for K-16 education* (2nd ed.). Alexandria, VA: Association for Supervision and Curriculum Development.

Vygotsky, L., & Cole, M. (1978). Mind in society: The development of higher psychological processes.

Weragama, D., & Reye, J. (2014). Analysing Student Programs in the PHP Intelligent Tutoring System. *International Journal of Artificial Intelligence in Education, 24*, 162-188.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. (2012). The fairly performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12)* (pp. 215-220). New York: Association for Computing Machinery. doi:10.1145/2157136.2157200

Wing, J. M. (2014, January 10). *Computational thinking benefits society*. Retrieved June 4, 2023, from Social issues in computing: http://socialissues.cs.toronto.edu/index.html%3Fp=279.html