Promoting Computer Science Learning with Block-Based Programming and Narrative-Centered Gameplay

Wookhee Min
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
wmin@ncsu.edu

Bradford Mott
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
bwmott@ncsu.edu

Kyungjin Park
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
kpark8@ncsu.edu

Sandra Taylor
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
smtayl23@ncsu.edu

Bita Akram
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
bakram@ncsu.edu

Eric Wiebe
Department of STEM Education
North Carolina State University
Raleigh, NC, USA
wiebe@ncsu.edu

Kristy Elizabeth Boyer
Computer & Information Science & Engineering
University of Florida
Gainesville, FL, USA
keboyer@ufl.edu

James Lester
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
lester@ncsu.edu

***Abstract*—Recent years have seen increasing awareness of the need for all students in primary and secondary education to learn computer science (CS) concepts and skills. Educational games hold significant potential to serve as a platform for CS education because they integrate engaging problem solving with effective pedagogical strategies. This potential is especially high for narrative-centered educational games that embed learning activities within rich interactive stories. In this paper, we present an educational game featuring block-based programming challenges contextualized within an engaging narrative, designed to promote CS learning for middle school students (ages 11 to 13). In the game, students undertake problem-solving challenges that are aligned with the K-12 Computer Science Framework. Results from a classroom implementation of the game with middle grade students suggest that their perceived game control ratings are positively correlated with their progress in the game, which suggests the need for adaptively supporting students' game-based learning activities. Building on these findings, we discuss design implications for creating student-adaptive CS learning experiences in educational games that incorporate block-based programming enriched narrative-centered gameplay.**

Keywords—Game-based learning, Computer science education, Computational thinking, Block-based programming

I. Introduction

There is a growing recognition that students in primary and secondary education should learn core computer science (CS) concepts [1][2][3]. CS has become a foundational skill for students to thrive in a digital economy, with CS fluency being essential for many career paths, especially in Science, Technology, Engineering, and Mathematics (STEM) fields where modeling, coding, and analysis are an essential part of professional life [4]. To prepare students for future studies and professions in STEM, it is essential to ensure that they acquire a robust foundation in CS.

Digital games, which have been recognized as a promising educational tool [5][6], offer significant potential for creating engaging CS learning experiences. Educational games can provide students with rich, virtual worlds that have been specifically designed to develop problem solving, critical thinking, and other twenty-first century skills [7]. Systematic reviews examining the effectiveness of educational games have shown they significantly enhanced student learning over more traditional methods [7][8][9]. Narrative-centered educational games, a class of digital learning environments that contextualize problem solving within engaging interactive stories, have shown significant potential in supporting students' learning processes by tightly integrating instructional and narrative elements [10].

Incorporating block-based programming into narrative-centered educational games can provide engaging contexts for promoting CS learning [11]. Due to the intricacies of syntax in text-based programming languages, block-based programming languages have emerged as a tool with which novice programmers, including K-12 students who often have limited programming experience, can implement algorithms to creatively solve problems [1][12]. Deeply integrating block-based programming into gameplay driven by motivating narratives (e.g., rich settings, engaging characters, and compelling plots) holds significant promise for providing students with effective, engaging CS learning experiences.

In this paper, we present ENGAGE, a narrative-centered educational game developed to promote computer science learning and foster computational thinking for middle school students (ages 11 to 13) through engaging block-based programming gameplay aligned with the K-12 Computer Science Framework [4]. Findings from a pilot study at a public middle school in the United States suggest that it is critical to support students' game-based learning activities by providing

978-1-7281-4533-4/20/$31.00 @2020 IEEE

timely and contextually appropriate feedback tailored to individual students, as underscored by prior research [13][14]. We conclude with design implications for narrative-centered educational games that feature block-based programming.

II. Related Work

Narrative-centered educational games that incorporate adaptive scaffolding (i.e., personalized support given to students throughout the learning process, such as hints and prompts) are situated at the intersection of (1) digital games that increase students' motivation through rich settings, engaging characters, and compelling plots in virtual environments, and (2) intelligent tutoring systems that foster students' learning through tailored scaffolding and context-sensitive feedback [10][15]. Narrative-centered educational games build upon students' natural abilities for understanding stories and developing experiential knowledge from them [16]. These learning experiences are especially powerful since they tap into key principles of narrative comprehension [17].

Block-based programming languages have proven effective in fostering computer science learning for novice learners by decreasing cognitive load, promoting recognition of shapes over recall of syntax, and limiting the ability to create syntax errors [12][18]. In contrast to many of the block-based programming interventions (e.g., App Inventor, Snap!, Scratch) that have focused on introducing students to programming using environments that support the creation of games and interactive artifacts [19][20][21][22], Engage deeply integrates block-based programming within a narrative-centered digital game that provides engaging CS learning experiences to middle school students.

III. Engage Narrative-Centered Educational Game

We have undertaken a long-term effort to investigate the use of narrative-centered educational games to support middle school students' CS learning. Engage (Fig. 1) is aligned with the K-12 Computer Science Framework [4] with learning objectives that are developmentally appropriate for U.S. middle school students. In addition to the curriculum alignment change, the latest version of the game is designed for WebGL and tablet computer deployments, since Chromebooks, Android tablets, and iPads are quickly becoming the platform of choice in schools throughout the U.S.

In Engage, students assume the role of the protagonist of the story who has been sent to an undersea research facility to determine why all communication with the station has been lost and restore its server network. Unbeknownst to them, a rogue scientist has taken over the research facility. Through a series of highly interactive learning activities, students restore the server and stop the rogue scientist from causing further harm to the research facility. The narrative-driven, problem-solving challenges within Engage were designed to develop middle-grade computer science learning and computational thinking skills by supporting activities of creating, testing, and refining programming artifacts. A paired *t*-test comparing the pre- and post-content knowledge test scores of students in a study indicated that Engage delivers significant student learning gains ($t(184) = 12.18$, $p < .001$, $d = .70$) [23].

Engage includes three thematic levels, Algorithms, Variables, and Controls. We briefly describe the Algorithms and Variables levels, which are the focus of this paper. Through the Algorithms level, students learn about CS concepts that include "Algorithms and Programming" and "Computing Systems" as well as CS practices such as "Recognizing and Defining Computational Problems," "Creating Computational Artifacts," and "Testing and Refining Computational Artifacts" described in the K-12 Computer Science Framework [4]. In this level, students are introduced to two programmable devices: the *quadcopter* and the *crane*. Each device can be programmed using the IntelliBlox programming interface [24] that is integrated directly into the gameplay to drag and drop "blocks" that represent basic programming constructs to create programs. In the Algorithms level, students must develop algorithms to navigate the quadcopter device (Fig. 1), which they can ride on, around obstacles in the room to reach the room's exit, or program the crane device to move floor tiles around to fill gaps in the floor as well as move obstacles such as crates that block the player's path to the room's exit.

In the Variables level, we build on the concepts covered in the Algorithms level and extend the set of concepts by introducing variables, a key CS concept specified under "Algorithms and Programming" in the K-12 Computer Science Framework. To effectively scaffold student learning of variables, we developed two novel devices: the *scale* and the *laser lock*. For example, in rooms that include a laser lock device, students must find the correct PIN code by iterating over a range of numbers using a brute force searching method. Students use a *count with [variable] from [number] to [number] by [number]* block to iterate over the range of numbers, and progressively check if the value assigned to the variable in the loop can open the lock that blocks the player's path to the exit (Fig. 2).

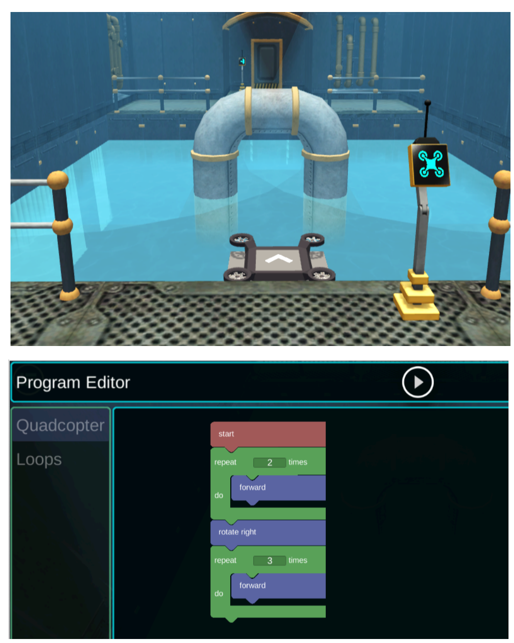

Fig. 1. (Top) Quadcopter device and (Bottom) sample quadcopter program written using IntelliBlox.

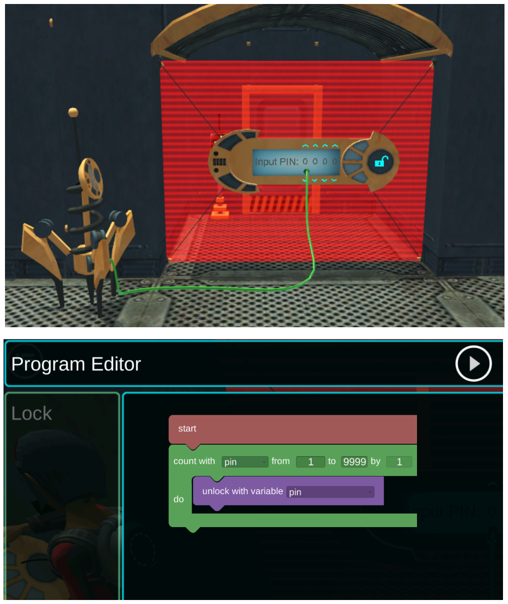

Fig. 2. (Top) Laser lock device and (Bottom) sample laser lock program written using INTELLIBLOX.

IV. Classroom Pilot Study

ENGAGE was deployed in a classroom pilot study conducted in a public middle school in the southeastern United States. The classroom's teacher and researchers from the project team led the two-day in-school implementation of ENGAGE, where students played the Algorithms level on Day 1 and the Variables level on Day 2. The researchers briefed students on the game and the objectives of the study (e.g., collecting feedback from students to improve the game) prior to the start of the learning activities. Each day, students played the game for roughly 30 minutes using either a laptop or tablet. On the second day of the study, we set aside 12 minutes before the end of class to conduct a post-survey with students that included a set of questions collecting data on student's demographics, student's engagement with the game, students' perceived user interface (UI) ratings (e.g., "Were the programming interface icons understandable by their appearances?"), and student's perceived game control ratings (e.g., "Was the programming interface easy to use for writing programs?"). The post-survey, which used a 5-point Likert scale (i.e., "strongly disagree" (1) to "strongly agree" (5)), was formulated to elicit students' feedback to inform the continued development of ENGAGE. Among 24 students who participated in the pilot study across the two days, 18 students attended both days and consented to the study (average age: 11.83, standard deviation: 1.06). Among those 18 students, 14 students completed the post-survey, which can be matched with their gameplay trace data.

Overall, students reported a mean post-game engagement rating of 4.04 out of 5, while the mean in-game task completion rates were 42.1%. We investigated if students' perceived difficulty of the UI and the game control are correlated with the task completion rates to identify areas to improve the game. Spearman nonparametric correlation tests indicated that there was a moderate, positive correlation between the task completion rates and the perceived game control ratings ($r = .54$, $p = .049$), and a positive, but not statistically significant, correlation between the task completion rates and the perceived UI ratings ($r = .46$, $p = .097$). These findings strongly suggest the need for providing students with adaptive feedback to ensure they make steady progress in game-based learning activities.

V. Discussion

Findings from the pilot study suggest the following three design implications for creating effective narrative-centered educational games that focus on middle-grade CS education. First, a robust, reliable student model should be devised, trained, validated, refined, and deployed in the learning environment. The predictive model's sequential inference results will inform pedagogical decisions for scaffolding at run-time. The model outputs (e.g., students' gameplay skills, affective states, and competencies for a fine-grained set of CS concepts and practices) should be reliably inferred by salient features captured from problem-solving behaviors (e.g., in-game interactions, block-based programming activities) as well as static aspects of students (e.g., student traits). As a promising approach, stealth assessment techniques can enable inferences about students' learning outcomes using competency models in an unobtrusive manner without conducting overt formative assessments [23][25][26].

Second, the scaffolding policy that connects student modeling with adaptive, run-time intervention (e.g., hints, feedback, narrative adaptation, task selection) during game-based learning should be carefully designed, empirically evaluated, and iteratively refined. While a policy can be induced using a set of rules, promising computational approaches include (1) using supervised learning where labels for effective intervention strategies are collected from human experts implemented in a Wizard-of-Oz study [27] and a scaffolding model can be devised and trained utilizing a labeled dataset, and (2) using reinforcement learning algorithms that can derive effective policies that maximize expected return (e.g., learning gains in CS content knowledge, engagement) [28][29].

Third, narrative-centered educational games offer a variety of intervention channels to realize adaptive scaffolding [30]. Examples include modifying the sequence of in-game challenges and introducing new challenges that can develop and reinforce students' mastery of CS concepts [31]. Another effective intervention channel is through dynamically adapting narratives such as personalized plots and dialogues with in-game virtual characters (e.g., pedagogical agents, learning companions), who can provide tailored hints and feedback at run-time to remediate students' knowledge and skills for the core CS concepts and practices.

VI. Conclusion

Narrative-centered educational games hold significant potential to serve as an effective platform for promoting students' learning and engagement in computer science education. We have introduced ENGAGE, a narrative-centered educational game that supports computer science education for middle school students based on the K-12 Computer Science Framework. ENGAGE features block-based programming enriched gameplay implemented with the Unity® crossplatform game engine and the INTELLIBLOX block-based programming toolkit [24]. Findings from the ENGAGE pilot study indicated that students' perceived game control ratings are positively correlated with their task completion rates. The results suggest that providing students with adaptive feedback could help ensure they make steady progress through the game, thereby promoting their engagement and learning. To achieve this goal, we discussed design implications including the importance of robust student modeling, as well as the need for inducing effective scaffolding policies that drive the realization of personalized intervention through multiple channels offered by narrative-centered educational games.

In future work, it will be important to train, validate, refine, and deploy CS competency-focused, stealth assessment models utilizing student gameplay data, to develop effective scaffolding policies, and to incorporate the adaptive scaffolding framework within the learning environment. A range of machine learning techniques (e.g., deep neural networks, probabilistic graphical models) along with feature engineering and fusion approaches should be investigated to identify models that most optimally promote computer science learning through the educational game. It will also be important to evaluate and iteratively refine the learning environment to create effective CS learning outcomes for students with diverse backgrounds.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grants CNS-1138497 and DRL-1640141. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] S. Grover, S. Basu, and P. Schank, "What we can learn about student learning from open-ended programming projects in middle school computer science," in *Proc. ACM Technical Symposium on Computer Science Education*, 2018, pp. 999–1004.

[2] J. Nouri, L. Zhang, L. Mannila, and E. Norén, "Development of computational thinking, digital competence and 21st century skills when learning programming in K-9," *Education Inquiry*, pp.1–17. 2019.

[3] K. Rich, C. Strickland, T. A. Binkowski, C. Moran, and D. Franklin, "K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals," in *Proc. ACM conference on international computing education research*, 2017, pp. 182–190.

[4] K-12 Computer Science Framework, https://k12cs.org/, 2016.

[5] M. Papastergiou, "Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation," *Computers & Education*, vol. 52, no. 1, pp. 1-12, 2009.

[6] D. Shaffer, "*How computer games help children learn*," New York, NY: Palgrave Macmillan, 2006.

[7] M. Qian and K. Clark, "Game-based learning and 21st century skills: A review of recent research," *Computers in Human Behavior,* vol.63, pp. 50–58, 2016.

[8] D. Clark, E. Tanner-Smith, and S. Killingsworth, "Digital games, design, and learning: A systematic review and meta-analysis," *Review of Educational Research*, vol. 86, no. 1, pp. 79–122, 2016.

[9] P. Wouters and H. Oostendorp, "Overview of instructional techniques to facilitate learning and motivation of serious games," *Instructional techniques to facilitate learning and motivation of serious games*, Springer, pp. 1–16, 2017.

[10] J. Rowe, L. Shores, B. Mott, and J. Lester. "Integrating learning, problem solving, and engagement in narrative-centered learning environments," *International Journal of Artificial Intelligence in Education*, vol. 21, no. 1-2, pp. 115–133, 2011.

[11] A. Bauer, E. Butler, and Z. Popović, "Approaches for teaching computational thinking strategies in an educational game: A position paper," in *Proc. IEEE Blocks and Beyond Workshop*, 2015, pp. 121–123.

[12] T. Price, Y. Dong, and D. Lipovac, "iSnap: towards intelligent tutoring in novice programming environments," in *Proc. ACM Technical Symposium on Computer Science Education*, 2017, pp. 483–488.

[13] C. Chen and V. Law, "Scaffolding individual and collaborative game-based learning in learning performance and intrinsic motivation," *Computers in Human Behavior,* vol. 55, pp. 1201–1212. 2016.

[14] M. Pedro, R. Baker, S. Gowda, and N. Heffernan, "Towards an understanding of affect and knowledge from student interaction with an intelligent tutoring system," in *Proc. International Conference on Artificial Intelligence in Education*, Springer, 2013, pp. 41–50.

[15] J. Lester, E. Ha, S. Lee, B. Mott, J. Rowe, and J. Sabourin, "Serious games get smart: Intelligent game-based learning environments," *AI Magazine*, vol. 34, no. 4, pp. 31–45, 2013.

[16] J. Bruner, "The narrative construction of reality," *Critical Inquiry*, vol. 18, no. 1, pp. 1–21, 1991.

[17] R. J. Gerrig, "*Experiencing narrative worlds: on the psychological activities of reading*," New Haven, London: Yale University Press, 1993.

[18] D. Bau, J. Gray, C. Kelleher, J. Sheldon, and F. Turbak, "Learnable programming: blocks and beyond," *Commun.,* ACM, vol. 60, pp. 72–80, 2017.

[19] A. Repenning, "Moving beyond syntax: lessons from 20 years of blocks programing in Agent Sheets," *Journal of Visual Languages and Sentient Systems,* vol. 3, no. 1, pp. 68–89, 2017.

[20] M. MacLaurin, "The design of Kodu: A tiny visual programming language for children on the Xbox 360," in *Proc. ACM Symposium on Principles of Programming Languages*, 2011, pp. 241–246.

[21] E. Rosenbaum, "Scratch for second life," in *Proc. International Conference of the Learning Sciences*, 2008, pp. 144–152.

[22] M. Tsur and N. Rusk, "Scratch microworlds: designing project-based introductions to coding," in *Proc. ACM Technical Symposium on Computer Science Education*, 2018, pp. 894–899.

[23] W. Min, M. Frankosky, B. Mott, J. Rowe, A. Smith, E. Wiebe, K. E. Boyer, and J. Lester, "DeepStealth: Game-based learning stealth assessment with deep neural networks," *IEEE Transactions on Learning Technologies*, vol. 13, no.2, pp. 312–325, 2020.

[24] S. Taylor, W. Min, B. Mott, A. Emerson, A. Smith, E. Wiebe, and J. Lester, "Position: IntelliBlox: A toolkit for integrating block-based programming into game-based learning environments," in *Proc. IEEE Blocks and Beyond Workshop*, 2019, pp. 55–58.

[25] K. Georgiadis, G. van Lankveld, K. Bahreini, and W. Westera, "Learning analytics should analyse the learning: proposing a generic stealth assessment tool," in *Proc. IEEE Conference on Games*, 2019, pp. 1017–1024.

[26] V. Shute, "Stealth assessment in computer-based games to support learning," *Computer Games and Instruction*, vol. 55, no. 2, pp. 503–524, 2011.

[27] S. Lee, J. Rowe, B. Mott, and J. Lester. "A supervised learning framework for modeling director agent strategies in educational interactive narrative," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 2, pp. 203–215, 2014.

[28] R. Sawyer, J. Rowe, and J. Lester, "Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning," In *Proc. International Conference on Artificial Intelligence in Education*, Springer, 2017, pp. 323–334.

[29] S. Doroudi, V. Aleven, and E. Brunskill, "Where's the reward?," *International Journal of Artificial Intelligence in Education*, vol. 29, no. 4, pp. 568–620, 2019.

[30] J. Plass, and S. Pawar, "10 adaptivity and personalization in game-based learning," *Handbook of Game-Based Learning*, vol. 263, 2020.

[31] K. Park, B. Mott, W. Min, K. Boyer, E. Wiebe, and J. Lester, "Generating educational game levels with multistep deep convolutional generative adversarial networks," In *Proc. IEEE Conference on Games*, IEEE, 2019, pp 345–352.