Supporting Interactive Storytelling with Block-Based Narrative Programming

Andy Smith [0000-0002-6577-7764] 1, Danielle Boulden [0000-0003-4833-7115] 1, Bradford Mott[0000-0003-3303-4699]1, Aleata Hubbard-Cheuoua[0000-0002-7310-1326]2, James Minogue[1], Kevin Oliver[0000-0001-8463-0621]1, and Cathy Ringstaff [0000-0002-2856-1120]2

[1] North Carolina State University, Raleigh NC 27695, USA
{pmsmith4, dmboulde, bwmott, jminogu, kmoliver}@ncsu.edu

[2] WestEd, San Francisco, CA 94107, USA
{ahubbar, cringst}@wested.org

Abstract. Recent years have seen growing interest in utilizing digital storytelling, where students create short narratives around a topic, as a means of creating motivating problem-solving activities in K-12 education. At the same time, there is increasing awareness of the need to engage students as young as elementary school in complex topics such as physical science and computational thinking. Building on previous research investigating block-based programming activities for storytelling, we present an approach to block-based programming for interactive digital storytelling to engage upper elementary students (ages 9 to 10) in computational thinking and narrative skill development. We describe both the learning environment that combines block-based narrative programming with a rich, interactive visualization engine designed to produce animations of student generated stories, as well as an analysis of students using the system to create narratives. Student generated stories are evaluated from both a story quality perspective as well as from their ability to communicate and demonstrate computational thinking and physical science concepts and practices. We also explore student behaviors during the story creation process and discuss potential improvements for future interventions.

Keywords: Narrative-centered learning, block-based programming.

1 Introduction

Recent years have seen growing evidence that engaging students in digital storytelling activities is an effective tool for promoting meaningful learning across a variety of subjects [1–3]. While these activities have been shown to facilitate positive outcomes such as creative exploration of scientific phenomena [4, 5], studies have also shown that the benefits of digital storytelling can vary greatly between individual students. This highlights a critical need for a better understanding of how to effectively support students as they create and present stories.

Another barrier to large-scale adoption of digital storytelling activities in classrooms is lack of perceived alignment with existing curricular goals, specifically in areas such as computational thinking and physical science [6]. At the elementary level, many teachers have limited, if any, instructional time allocated to these topics, and may see digital storytelling activities as an inefficient use of that limited resource. However, there is a growing recognition of the similarities between the digital storytelling process and computational thinking and science competencies [7, 8]. Additionally, teachers who often lack significant training in science and computational thinking may be more comfortable supporting a more familiar task such as narrative construction.

To address these challenges, we have developed the INFUSECS narrative-centered learning environment, to engage students in deep, meaningful physical science and computational thinking learning through the creation of interactive narratives. INFUSECS utilizes a custom-built narrative programming environment, where students utilize a block-based programming interface to create, revise, and visualize interactive narratives. Building on research exploring design best practices for block-based programming for younger learners [9, 10], we created a learning environment to enable upper elementary students o create rich interactive narratives while also engaging with concepts and competencies from physical science and computational thinking. In this paper, we use INFUSECS to specifically focus on two research questions:

RQ1: How effectively are students able to use block-based programming to create interactive narratives, when evaluated from a story quality perspective?

RQ2: How do students exhibit knowledge of physical science and computational thinking concepts through their created interactive narratives?

To answer these research questions, we conducted a pilot study with students in the southeastern United States as part of an after school program. Initial results show students were able to effectively use the tool to create stories, while also demonstrating evidence of computational thinking concepts such as debugging and sequential execution. However, not all student generated stories met all story quality criteria, and all students struggled to integrated physical science concepts into their narratives.

2 Related Work

Narrative experiences offer an exceptionally promising tool for engaging students in computationally-rich problem solving. By leveraging narrative's unique ability to help us understand the world around us [11] and communicate conceptual understanding to others [12], storytelling has significant potential well beyond the traditional educational context of language arts. Specifically, digital storytelling has shown great potential to leverage the creativity and effectiveness of narrative for domains such as science [1, 13] through activities such as creating a multimedia presentation of a story. Other digital storytelling activities have shown positive effects for both cognitive measures of visual memory and writing skills [14], affective measures such as student engagement [2], and improved 21st century skills such as problem solving, argumentation, and cooperation [15].

However, the benefits of digital storytelling interventions are dependent on the ability of the student to construct a narrative. This has led to a broad range of research into how to best design and support digital storytelling interventions for students of a variety of age ranges, including approaches that seek to leverage the synergies between storytelling and computational thinking through block-based programming. A modified version of the popular Scratch environment was used to enable students to create animated stories involving placing and moving sprites, as well as including audio clips and other events responsive to user inputs [16]. Other research has investigated using block-based programming in non-traditional methods such as a tangible, sticker-based block language used as part of an interactive storybook [17], or as a method for introducing computational thinking strategies into English language learning [8]. This work extends these efforts by designing and investigating story quality and demonstration of conceptual knowledge using a block-based programming environment focused primarily on interactive storytelling augmented by rich visualizations.

3 Narrative-Centered Learning Environment

This study builds on a previous version of InfuseCS, which featured a block-based programming interface built with Google's Blockly framework that focused on the creation of text-based teleplays. The current version of InfuseCS is a WebGL-based application built with the Unity 3D game engine. In addition to supporting interactive multimedia content and simulation activities, this version also supports a greatly expanded narrative programming interface. The interface consists of two main parts: a custom block-based programming interface developed for the Unity 3D game engine, and an interactive visualization engine. Figure 1 below shows a short example story and the accompanying visualization.

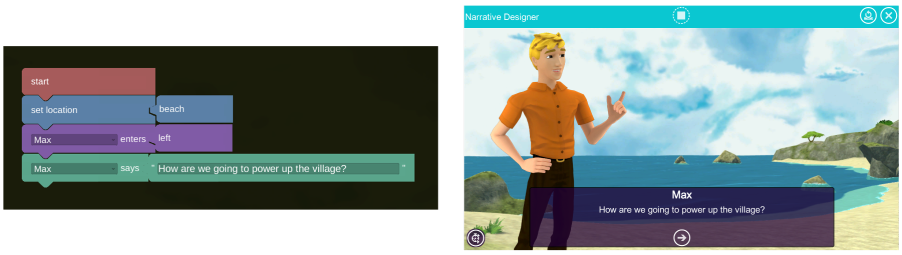

Fig. 1. Screenshots of a sample narrative displayed in InfuseCS's block-based editor and visualization of a line of dialog.

The narrative programming interface of InfuseCS utilizes four types of blocks designed to enable key aspects of digital storytelling including setting the location of the scene, arranging characters, enabling dialogue between characters, and supporting branching stories based on choices made by the viewer of the story. The custom blocks leverage design principles from previous research on block-based programming environments for upper elementary students [16, 18–20]. Specifically, the custom narrative blocks utilize distinct colors for each category to leverage visual affordances in the appearance of the blocks. Stories are executed sequentially starting from a *Start* block, avoiding event-driven programming. *Dialogue* and *Stage Direction* blocks utilize drop downs for character names and a starter story is provided to the students to encourage customization and editing rather than a purely generative activity. Finally, the language on the blocks is designed to limit complicated syntax and vocabulary.

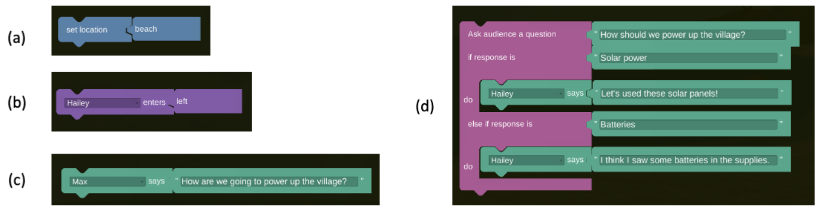

Fig. 2. Custom blocks for the INFUSECS narrative programming interface.

The set of custom blocks are separated into four categories corresponding to the main features of visual interactive narratives: describing a scene, moving characters in and out of the scene, character dialogue (including narration), and branching.

The *Scene* blocks, shown in Figure 2a, allows students to set the location where the events in the narrative take place. Location blocks are defined before the activity to fit with the motivating scenario framing a particular digital storytelling activity, and to align with art assets for the visualization system described below. This also allows INFUSECS to utilize type-checking, and prevent students from attaching incompatible blocks, i.e. text fields, a feature also afforded by the color-coding scheme of blocks.

The *Stage Direction* blocks allow for students to move characters in and out of the scene of their story. The *Enter* block, shown in Figure 2b, allows students to select which character is entering the scene, with fixed location blocks that can be attached similar to the locations for the *Scene* block. The other *Stage Direction* block currently implemented is the *Exit* block, where students can specify a character to leave the scene.

The *Dialogue* blocks, shown in Figure 2c, are the core blocks of the storytelling system, as the majority of the content in the student-generated stories is dialogue between characters or statements by the narrator. A line of dialogue is generated through the combination of a *Character* block and a *Text* block. The name property of the *Character* block is filled with a list of characters that have been previously introduced to the student earlier in the activity. Additionally, a *Narrator* option can be chosen to provide narration for the story.

The *Ask Audience* block, Figure 2d, is designed to allow students to incorporate branching into their narratives, while also providing them an opportunity to experiment with computational thinking concepts like conditionals and flow control. The

Ask Audience block contains three properties that must be defined by learners. The first property is the question that will be asked to the audience, and then the two possible responses the audience can respond with to the question. Below each response, students can then place story blocks that will execute if that choice is selected.

At any point in the story authoring process, students can press a run button at the top of the editor to play an animation of their story. First, the block-based representation is converted into Ink script (https://www.inklestudios.com/ink/), a narrative scripting language developed primarily for game applications. Next, the generated ink script is passed to the visualization engine, which displays it using graphical assets in Unity. The visualization pauses after each dialogue utterance to give the audience time to read the story, and advances when a button is pressed. At the completion of the visualization, the student returns to the narrative editor where they can continue to iterate and revise their story until they are satisfied.

4 Pilot Study

To understand how INFUSECS supports digital storytelling and computational thinking in elementary classrooms, a pilot study was conducted at an after-school program in the southeast United States. Participants in the pilot study included 6 fourth grade students ages 9-10, including 4 males and 2 females. Four of the students were Black or African-American, one was White, and one was Multiracial. The 6 students worked independently, and students reported a range of previous experience with block-based coding environments with 1 student reporting no prior experience, 2 students reporting some prior experience, and 3 students reporting that they frequently used block-based coding environments.

The pilot study took place over 4 days, with approximately 30-45 minutes spent each day. Students and coordinators for the after-school program were together in person, while research staff observed remotely using video conferencing software. The first day students were given a brief introduction to the activity, completed a demographic survey, and watched an introductory video introducing the motivating scenario of a group of scientists who were shipwrecked on a remote tropical island. On the second day, students were presented with the task of solving how the scientists would power their makeshift village and engaged with physical science content focusing on types of energy and energy conversion. After completing the science content, students were given a planning worksheet to assist them with planning their story. On the third day, students completed their planning worksheet and began creating their stories using the narrative programming interface, and on the fourth day the students spent the entire time period working on their stories.

For all students, a brief starter story was provided in the block-based programming interface, which provided an example of each type of block. The learning environment logged detailed trace logs of student actions in the environment, as well as logging their story workspace each time the story was visualized.

All 6 students were able to use the system to create short narratives in the allotted time. Each student effectively used at least one of each type of block with stories ranging from 12-26 “lines of code” (i.e. two connected components of a *Dialogue* block counting as 1 “line”), consisting mostly of narration and dialogue between characters. Overall student stories averaged 17.3 total blocks (M = 17.3, s.d. = 5.64), 4 scene blocks per story (M = 4, s.d. = 1.78), 4.2 *Stage Direction* blocks per story (M=4.2, s.d. = 1.94), 1.8 *Ask Audience* blocks (M=1.8, s.d. = 0.98), and 7.3 *Dialogue* blocks (M=7.3, s.d. = 1.97). Additionally, students used an average of 3.33 characters in their story. This represents a large increase compared to previous versions of the system containing a similar set of story blocks but lacking the visualization functionality of the current system [21].

We also evaluated the student stories based on a set of story quality metrics, as well as for evidence of curricular competencies from both physical science and computational thinking. We first investigate the research question:

RQ1: How effectively are students able to use block-based programming to create interactive narratives, when evaluated from a story quality perspective?

To measure story quality, we devised a scoring rubric to evaluate the students' stories based on the Common Core State Standards (http://www.corestandards.org/) for 4th grade English Language Arts Writing. Common Core State Standards are a set of standards for various subjects such as writing and mathematics that have been adopted by many states in the United States. Students' stories were evaluated based on six criteria in alignment with the standards and one additional criterion we added specifically focusing on how branching was utilized in the story to facilitate its organization and development. These criteria are described in Table 1 below. Each criterion was rated on a 0-2 scale, with 0 as having little to no evidence, 1 as having some evidence, and 2 as having sufficient evidence of the given criteria.

Table 1. Story evaluation criteria.

Criteria	**Description**
Criteria 1	Establishes a situation that introduces reader to characters / setting
Criteria 2	Presents / establishes an organized event that unfolds naturally (including a distinct beginning, middle, and end)
Criteria 3	Includes dialogue and description that develops experiences and events or shows the responses of characters to situations
Criteria 4	Uses a variety of transitional words and phrases to manage the sequence of events
Criteria 5	Uses concrete words and phrases and sensory details to convey experiences and events precisely
Criteria 6	Provide a conclusion that follows from the narrated experiences or events
Criteria 7	Integrates branching to organize and develop the story

Analysis of student narratives using these criteria is shown below in Table 2. Overall, the custom blocks facilitated Criteria 1, 2, and 3 for all students. Student stories performed worst on Criteria 5 and 6. Summing across all criteria yielded an average of 6.67 per story, with a high score of 12 and a low score of 4.

Table 2. Story evaluation scores.

Student	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	Criteria 6	Criteria 7
Student 1	1	1	1	1	0	1	1
Student 2	2	2	2	2	1	1	2
Student 3	1	1	2	2	1	0	2
Student 4	1	0	1	1	0	0	1
Student 5	1	1	1	1	0	0	1
Student 6	1	1	1	0	0	0	1

RQ2: How do students exhibit knowledge of physical science and computational thinking concepts through their created interactive narratives?

To evaluate physical science content, we used frequency counts to measure how many science concepts from the physical content were referenced in their stories. Overall, the number of science concepts students utilized in their stories ranged from 0 to 3 with four of the students not integrating any science concepts in their stories.

For computational thinking we looked at both the final story, as well as behaviors exhibited while creating the story. Using the K-12 Computer Science Framework (https://k12cs.org/) we identified concepts and practices that overlap with the story creation process. In addition to the concept of *Creating Computational Artifacts,* students also exhibited evidence of *troubleshooting/debugging* behaviors through the iterative running and modifying of their stories as they were created. Overall, the 6 students averaged 42 runs of their story (minimum = 12, maximum = 61). Students showed evidence of understanding *control structures* and *sequential execution* of their stories through their usage of the *Ask Audience* block. Multiple students stored segments of code on the workspace not connected to the *Start* block, showing evidence of their understanding of the single thread of execution.

5 Discussion

Overall, the INFUSECS learning environment enabled students to create interactive narratives incorporating features such as dialog, setting changes, and branching. Compared to a previous version of the system, the new system with a revised set of custom blocks and a story visualization system appears to have effectively supported longer, higher quality stories, as well as encouraged desirable debugging and troubleshooting behaviors. This was particularly noticeable in the students' improved understanding and usage of the *Ask Audience* block to incorporate branching into their narratives.

A noticeable shortcoming was the lack of physical science content included in the stories. Discussions with students suggested adding props representing the science content to the block palette may encourage their inclusion in the resulting narratives. Similarly, more explicitly including science concepts in the planning process may also encourage their usage in the resulting narratives. From a computational thinking perspective, further analysis of the students' debugging and troubleshooting behaviors could help ensure that the visualization functionality is being used in a productive manner, rather than a distracting one. Inclusion of more complex command structures, such as loops or event-based execution, could provide more opportunities for students to demonstrate richer understanding. Finally, including more traditional assessments (i.e. validated survey measures) could help determine if lack of understanding, or a shortcoming of the interface is responsible for the omission of concepts included in the narratives.

Finally, there are two key limitations to these findings. First, we acknowledge the small sample size makes it difficult to generalize the findings too broadly. Secondly, while researchers were able to observe the activities via video conferencing, lack of in person observations due to the ongoing pandemic makes qualitative analysis of the story creation process more difficult.

6 Conclusion and Future Work

As evidence of the transformative potential of digital storytelling continues to grow, so does the need for understanding how to effectively integrate and support digital storytelling in different educational contexts. The structured format and intuitive design of block-based programming environments provides a promising modality for both enabling rich interactive narrative creation, while at the same time providing an intriguing activity for integrating computational problem solving in a variety of subjects.

With this as motivation, we investigated student story creation utilizing block-based programming. Findings show that the custom story blocks enabled upper elementary students to create structured narratives, incorporating traditional storytelling concepts such as dialogue between characters and branching based on viewer input. Analysis of student stories highlighted how the storytelling activity can be used by students to demonstrate their conceptual understanding of physical science and computational thinking concepts and practices.

Future iterations of the learning environment will focus on revising the set of storytelling blocks, with a focus on encouraging students to communicate more science and computational thinking concepts and practices. This focus will also extend to the visualization portion of the environment, allowing for richer interactions to be authored by the students. We will also investigate how to improve storytelling through a more robust and structured planning phase as well as additional support to augment the story creation and revision process.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grants DRL-1921495 and DRL-1921503. 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. Robin, B.R.: Digital storytelling: A powerful technology tool for the 21st century classroom. Theory Into Practice. 47, 220–228 (2008). https://doi.org/10.1080/00405840802153916.
2. Smeda, N., Dakich, E., Sharda, N.: The effectiveness of digital storytelling in the classrooms: a comprehensive study. Smart Learning Environments. 1, 1–21 (2014). https://doi.org/10.1186/s40561-014-0006-3.
3. Yang, Y.T.C., Wu, W.C.I.: Digital storytelling for enhancing student academic achievement, critical thinking, and learning motivation: A year-long experimental study. Computers and Education. 59, 339–352 (2012). https://doi.org/10.1016/j.compedu.2011.12.012.
4. Henriksen, D.: Full STEAM Ahead: Creativity in Excellent STEM Teaching Practices. The STEAM Journal. 1, 1–9 (2014). https://doi.org/10.5642/steam.20140102.15.
5. Henriksen, D., Mishra, P., Fisser, P.: Infusing creativity and technology in 21st century education: A systemic view for change. Educational Technology and Society. 19, 27–37 (2016).
6. Tan, M., Lee, S.S., Hung, D.W.L.: Digital storytelling and the nature of knowledge. Education and Information Technologies. 19, 623–635 (2014). https://doi.org/10.1007/s10639-013-9280-x.
7. Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L.: Computational Thinking for Youth in Practice. (2011).
8. Parsazadeh, N., Cheng, P.Y., Wu, T.T., Huang, Y.M.: Integrating Computational Thinking Concept Into Digital Storytelling to Improve Learners' Motivation and Performance. Journal of Educational Computing Research. 59, 470–495 (2021). https://doi.org/10.1177/0735633120967315.
9. Hill, C., Dwyer, H.A., Martinez, T., Harlow, D., Franklin, D.: Floors and flexibility: Designing a programming environment for 4th-6th grade classrooms. In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE). pp. 546–551 (2015). https://doi.org/10.1145/2676723.2677275.
10. Weintrop, D., Hansen, A.K., Harlow, D.B., Franklin, D.: Starting from scratch: Outcomes of early computer science learning experiences implications for what comes next. In: Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER). pp. 142–150 (2018). https://doi.org/10.1145/3230977.3230988.
11. Bruner, J.: Acts of meaning. Cambridge, MA: Harvard University Press (1990).
12. Avraamidou, L., Osborne, J.: The role of narrative in communicating science. International Journal of Science Education. 31, 1683–1707 (2009). https://doi.org/10.1080/09500690802380695.
13. Robin, B.R.: The power of digital storytelling to support teaching and learning. Digital Education Review. 17–29 (2016). https://doi.org/10.1344/der.2016.30.17-29.

14. Sarica, H.Ç., Usluel, Y.K.: The effect of digital storytelling on visual memory and writing skills. Computers and Education. 94, 298–309 (2016). https://doi.org/10.1016/j.compedu.2015.11.016.
15. Niemi, H., Multisilta, J.: Digital storytelling promoting twenty-first century skills and student engagement. Technology, Pedagogy and Education. 25, 451–468 (2016). https://doi.org/10.1080/1475939X.2015.1074610.
16. Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., Weintrop, D., Harlow, D.: Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. In: Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE. pp. 231–236 (2017). https://doi.org/10.1145/3017680.3017760.
17. Horn, M.S., AlSulaiman, S., Koh, J.: Translating roberto to omar: Computational literacy, stickerbooks, and cultural forms. In: Proceedings of the 12th International Conference on Interaction Design and Children. pp. 120–127 (2013). https://doi.org/10.1145/2485760.2485773.
18. Franklin, D., Hill, C., Dwyer, H.A., Hansen, A.K., Iveland, A., Harlow, D.B.: Initialization in scratch: Seeking knowledge transfer. SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education. 217–222 (2016). https://doi.org/10.1145/2839509.2844569.
19. Weintrop, D., Wilensky, U.: Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In: Proceedings of the 11th Annual International Conference on International Computing Education Research (ICER '15). pp. 101–110 (2015). https://doi.org/10.1145/2787622.2787721.
20. Dwyer, H., Hill, C., Hansen, A., Iveland, A., Franklin, D., Harlow, D.: Fourth grade students reading block-based programs: Predictions, visual cues, and affordances. In: Proceedings of the 2015 ACM Conference on International Computing Education Research (ICER). pp. 111–120 (2015). https://doi.org/10.1145/2787622.2787729.
21. Smith, A., Mott, B., Taylor, S., Hubbard-Cheuoua, A., Minogue, J., Oliver, K., & Ringstaff, C. Toward a Block-Based Programming Approach to Interactive Storytelling for Upper Elementary Students. In Proceedings of the International Conference on Interactive Digital Storytelling. pp. 111-119 (2020). https://doi.org/10.1007/978-3-030-62516-0_10