

Learning Computational Thinking with Minecraft

Katherine Starr

Information, Organization & Learning Sciences
University of New Mexico
Albuquerque, NM, USA
starrk@unm.edu

ABSTRACT

Aside from Tetris, Minecraft is the most popular digital game of all time [1]. Analysis of its broad appeal and utilization of its allure is gaining serious attention from researchers and scholars. In the context of the call to provide all students access to learning “computational thinking” (CT) [2], Minecraft may be an unsung hero. Microsoft, who now owns Minecraft, recognizes the game’s appeal as a powerful learning tool, evidenced by their creating Minecraft: Education Edition and MakeCode for the Minecraft environment. These are valuable additions to the cadre of programs designed to engage children in CT, but it is remiss to think that Minecraft requires modification to be a valuable CT learning tool. The affordable, easily available, multi-platform game requires no modifications or adult contrivances to make it “educational”. This paper is an exploration of the idea that Minecraft is inherently valuable as a CT learning tool. To explore this hypothesis, I created a learning module for exploring CT in Minecraft.

CCS CONCEPTS

Applied Computing → **Education**; *Interactive Learning Environments*

KEYWORDS

Computational thinking; Video Game; Education; Minecraft; Digital Game

“After I went through my first night as a novice, I did not enjoy it so I got my 7-year-old nephews to show me how to play. We worked together and got it done. They have never played but they watch lots of online videos about the game.”

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission. Request permissions from starrk@unm.edu
© 2019 Copyright is held by the owner/author.

INTRODUCTION

Minecraft is an open-ended, procedurally generated sandbox video game with infinite creative possibilities. It is appealing and approachable to an astonishing range of ages and ability levels. Navigating with apparent ease and building elaborate structures, children as young as four and five find the game engrossing, stimulating, and satisfying. Adults are equally engaged, employing sophisticated electrical engineering and computer science concepts to create complex machines like automated item sorters, calculators, and even multi-core functioning computers, and a working smart phone. The creative building potential engages adult Minecrafters in projects like creating a precise replica of Tolkien’s Middle Earth or the entirety of the Game of Thrones’ Westeros map. The innate versatility and transparency of the Java Script code in Minecraft has inspired legions of “hackers” who modify the code to create custom modifications (“mods”) that provide an endless array of modified object behavior, special texture packs, custom downloadable maps, and hundreds of multiplayer servers featuring endless new ways to play and explore Minecraft.

The same qualities that invite the tinkering that results in modifications to the code—whether for “educational” purposes or “just for fun” —make Minecraft an environment rich in CT potential. Using the Moodle learning management system as an online delivery platform, I created a learning module designed to allow a wide range of learners to explore CT in Minecraft. The module guides the learner to explore Minecraft while paying attention to self-observation of CT processes. The module is broken into two major parts of Minecraft exploration: the first part is centered around Survival Mode and asks the learner to pay attention to the elements of CT that exist in the structure of the game that arise naturally during game play, and the second part is dedicated to Creative Mode and is focused on the learner directly applying computational concepts in the form of creating logic gates using redstone, the electrical component of the game.

What is computational thinking?

There is no single agreed-upon definition of computational thinking. The term is attributed to Seymour Papert in his seminal 1980 book *Mindstorms*, and popularized by Jeannette Wing's 2006 ACM article: "Computational Thinking" [3], [4], [5]. While the learning module does not attempt to define CT, it does briefly describe abstraction, patterns, efficiency, algorithms, logic, and coding, and alerts the learner to be observant of these concepts. A more thorough list of CT concepts includes parallel processing, recursive thinking, solving problems using reduction, transformation, and simulation, and "using abstraction and decomposition when attacking a large complex task or designing a large complex system." [4] So as not to cognitively overload the module participants I chose to define CT in simple terms instead of trying to be thorough. The goal of the module is to demonstrate to the learner that Minecraft contains opportunities for "logical thinking, abstraction, pattern identification, predicting, human computer interaction, hardware/software integration, algorithms and efficiency." [6]

Educators have been much quicker than parents to recognize the educational potential of Minecraft [7]. A 2016 article in the *School Community Journal* notes that playing Minecraft can "spark collaborative learning, critical thinking, and problem-solving skills" and that it enhances STEM/STEAM and language skills, and teachers can benefit by using the game "to increase academic engagement with students and reinforce parental involvement" [7]. While Minecraft: Education Edition is marketed to teachers and parents as the best way to bring Minecraft to the classroom, many educators are finding rich potential in unmodified, plain "vanilla" Minecraft. Educators have created custom mods, like Mathcraft [7], created elaborate maps for teaching microbiology [8], and—among countless other creative applications—used it as a source of inspiration for real-world construction, as in the case of a school-wide project for STEAM week at Bridgewater Primary School in Northamptonshire England [9].

The Hour of Code global movement by Computer Science Education Week and Code.org has reached "tens of millions of students in 180+ countries through a one-hour introduction to computer science and computer programming" [10]. Microsoft MakeCode includes a Minecraft environment that is popular with Hour of Code participants [11], [12]. It allows users to program in a Minecraft environment using drag and drop blocks of code, very much like Scratch. The JavaScript powering the

blocks of code is also viewable in MakeCode allowing more advanced programmers to tinker directly with it.

Although Minecraft has "proven efficacy as an educational tool" and "should not be dismissed as merely a game" [13], the value to the development of CT skills in playing Minecraft open-endedly, has not received the attention I think is warranted. After all, even though it was written in the context of teaching CT skills with creating apps, I think the following observation also applies to Minecraft: "The advantage of graphical programming is that logical and algorithmic skills essential to computational thinking can be taught in a way that does not intimidate students with unnecessary complexity" [13]. Even more absent from the body of research is analyzing the potential use of redstone, the electrical component of Minecraft, or command blocks, a programmable in-game component. I do not have any experience with command blocks, but the reputation for the complexity of redstone is well-acknowledged in the Minecraft community, as evidenced by the vast number of redstone tutorials on YouTube and discussions on social media platforms such as Reddit and Minecraft wikis. I suspect the lack of research regarding redstone is due to its adult-level complexity which perhaps does not reconcile well with the perception of Minecraft as a child's game. Despite being incredibly popular with children, an interesting feature of Minecraft is its broad range of appeal. There is a large community of adult Minecrafters, like software engineer Seth Hendrickson (a.k.a. SethBling), who have pushed the boundaries of what seems possible, engineering fantastically complex creations like a random video generator: (<https://www.youtube.com/watch?v=Jk1tuUUOnjo>), a computer that runs BASIC Programming Language and includes a LOGO-like Turtle: (<https://www.youtube.com/watch?v=t4e7PjRygt0&t=649s>), and even a game of Minecraft within a chest within the game of Minecraft:

(<https://www.youtube.com/watch?v=-j2wupurm4s>). My hope is that there will be more academic attention paid to widen the scope of serious consideration to the learning potential in Minecraft.

METHODS

This pilot study consists of a seven-part learning module hosted on a private Moodle server. Study participants were personally invited by me via email. Access to the module requires users to create a username and password and then enter the enrollment key provided in the email. More than 50 people were invited to participate in the study but only ten (N=10) enrolled and participated. The module is designed as an open-ended,



Figure 1: Screenshot of Sections in Moodle Module

self-paced study to encourage participation by a wide range of experience among learners. As there are no established parameters for testing computational thinking, evaluation procedures are highly subjective, which is why the learning module urges participants to play and explore at a leisurely pace and results rely on self-report. The wide range of experience desired for the study, the self-paced nature of the module, and the short amount of time for development made it impractical to test in a controlled environment, such as in a class with a set start and finish time. The result is that participants appeared in a slow trickle and fortunately, the first few participants were novices with whom I was able to communicate and provide guidance and their feedback informed adjustments I made to the learning module. The fact that I had never used Moodle or created content in a learning management system is one more obstacle that prevented a smoothly implemented and conclusive study.

The reason I directed the module toward a broad experience range of participants is twofold: Experienced players may find it easier to recognize patterns in gameplay that they may have noticed but not paid attention to. I do not explicitly state the algorithms and patterns I suspect the participant may notice because I do not want to influence user experience, but an example of the type of observation I am expecting is, for instance, noticing that the less safe the player is at night (i.e. not sleeping in a bed or not hiding where monsters cannot see) the more monsters spawn in the area and for longer periods of time. Repeated unprotected nights triggers the monsters to increase in number, appear earlier and stay longer. The novice player, feedback suggests, are too preoccupied with navigation and basic survival to notice much outside of how much they are struggling. The other

reason for targeting a broad range of participants is to evaluate the range of influence the learning module has on their attitude toward the game. In other words, I am anticipating that participants who are familiar with Minecraft are more likely to recognize CT elements, but they are more likely to be biased toward finding value in the game, whereas novices may be more likely to be skeptical of Minecraft's educational merit but may also find it considerably more time consuming and difficult to recognize CT elements due to the learning curve of the game.

The module consists of seven sections: Introduction, Getting Started, Survival Mode, Creative Mode, Redstone, Summary, and Feedback. The Introductory section introduces the participant to the basic navigational tools of the learning module and the hypothesis of the project, including an overview of basic "computational thinking" concepts. The Getting Started section focuses on guiding novices through the downloading, installation, and basic navigational features of Minecraft. The Survival Mode section contains instructions for all participants to play through three night and day cycles of the game, regardless of experience level, as an introductory activity to engaging with the game. Experienced players, who find this exercise extremely easy, are encouraged to do this with a "beginner's mind" approach. Early alpha novice testers found this experience so frustrating that I made significant changes to this portion of the module during the initial testing process. Before learners are done playing in Survival Mode they are encouraged to return to their gameplay for a "Deeper Dive". This is easy for experienced players and here is where I ask that all players note specific CT skills. I encourage players to spend enough time in game to accumulate a quantity of items that makes the basic storage chest system of Minecraft disorganized, and uneconomical or to

recognize the importance of developing “resource packs” of items to take on mining or other expeditions.

“You literally made ECE231 in Minecraft”

The Creative Mode section introduces examples of sophisticated structures experienced Minecrafters have built before participants receive detailed instructions for creating a featureless sandstone map that makes an ideal laboratory for experimenting with redstone. The Redstone section of the module consists of a “book” in Moodle where each page features a different logic gate, including a brief explanation, logic diagram, truth table, screenshots and video of a sample completed circuit from my own redstone laboratory map. The Logic Gates book consists of 13 pages: Introduction, Getting Started, Input/Output Gate, NOT Gate, OR Gate, NOR Gate, AND Gate, NAND Gate, XOR Gate, XNOR Gate, IMPLY Gate, More, and Additional Resources. “More” includes samples of repeaters and a rapid pulsar, as well as a suggestion that participants try making a NOR latch, NAND latch, and X-clock pulsar. The Summary section includes a brief overview, encouraging participants to consider the complexities of Minecraft, whether they continue to play or not, as well as some suggested challenges to try, like integrating logic gates into a functional item such as an item sorter or automatic farm.



Figure 2: Screenshot of Implied Gate

The final section, Feedback, is the third, and last, evaluation portion of the module. To evaluate participant experience, there are several feedback questionnaires within the module. The first is in the Introduction and it is designed to get a baseline assessment of prior exposure to the game and general opinion/impression of it (favorable, neutral, negative, etc.). The second feedback assessment follows the end of the Survival Mode section and asks participants to reflect on their observation of patterns and other CT elements, as well as report briefly about what they did (how many times did they die, did

they make a shelter, if so, out of what, etc.). The third assessment is the final part of the module. This feedback assessment is designed to get a general impression of the module as a whole, gauge whether or not interest in the game is changed and ask if any of the activities are something the participant has an interest in pursuing further. The results of the last assessment are flawed by originally being presented as a graded quiz rather than open feedback, so the first eight (out of ten) participants completed the feedback as a quiz that may have indicated they gave a wrong answer, which is not a significant problem, but not ideal, and not the intention.

CONCLUSION

The small sample size, the uncontrolled conditions and the numerous changes made to the learning module throughout this initial alpha testing phase make this an inconclusive pilot test, but the positive feedback from participants indicates that this is an interesting subject worthy of further study. Now that I am more adept at navigating in Moodle, I would like to get some more participation feedback to ensure that novices are able to get started more smoothly. I also think the module could be improved with feedback on the redstone section. It is unclear as to how participants reacted to the logic gates section. In fact, one participant reported that two of the gates were opposite of what she expected, so it is unclear if that is due to a flawed example, flawed instructions, or a misunderstanding on the learner’s part. All of the gates need to be reviewed by an expert for accuracy and the section may require more detailed instructions. The redstone section would be vastly improved if participants were learning about logic gates in context, or at least given a contextual example of each one in action, but I would need to collaborate with a subject matter expert because I am as new to working with redstone and understanding logic gates as my novice participants.

There is enough anecdotal evidence to suggest that players may be developing CT skills in normal Minecraft play to make is worthy of closer examination [6], and the cognitive challenge of working with redstone certainly requires CT skills. More study into how Minecraft can be used to develop CT skills is warranted. When children (or adults) are absorbed for hours on end in Minecraft, are they wasting time or are they learning? Or, is the quality and quantity of that learning directly correlated to *what* they are doing in Minecraft? Studies have shown that Minecraft can be used as an effective learning tool, but is open-ended play and creation in the game of equal, lesser or, perhaps, even greater value to the development of CT skills?

ACKNOWLEDGMENTS

The author would like to thank Dr. Leah Buechley, and the entire Learning Computing class at UNM for a fun, supportive, and incredibly informative and memorable learning experience.

The author would also like to thank Elizabeth Starr Harden for hosting the Moodle learning module and helping me and supporting me as I learned to navigate the site. The author is also grateful and would like to thank all the participants who generously took the time to try out the module and provide feedback, especially Elizabeth Starr Harden, Jessica Torres, Malachi Starr Stallings, Mackenzie Bailey, and Ezra Stallings.

The author would also like to acknowledge the following sources that influenced the thoughts and ideas that went into this project but were not directly quoted, thus not creating an in-text citation: [14], [15], [16], [17]

If you would like to participate in this project it can be accessed at: <https://www.esharden.com/learn/login/> Sign in or create an account and then navigate to >UNM >Learning Computational Thinking in Minecraft You will be required to enter a key: CTminecraft#1

References

- [1] J. Sirani, "Top 10 Best-Selling Video Games of All Time," IGN, 17 May 2019. [Online]. Available: <https://www.ign.com/articles/2019/04/19/top-10-best-selling-video-games-of-all-time>. [Accessed November 2019].
- [2] M. Smith, "Computer Science For All," Obama White House, 30 January 2016. [Online]. Available: <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>. [Accessed October 2019].
- [3] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, New York: Basic Books, Inc., Publishers, 1980.
- [4] J. M. Wing, "Computational Thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33-35, March 2006.
- [5] Wikipedia, "Computational thinking," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Computational_thinking. [Accessed November 2019].
- [6] Tracy, "How Minecraft Teaches Computational Thinking," TechAge Kids, 12 May 2016. [Online]. Available: <https://www.techagekids.com/2016/05/how-minecraft-teaches-computational.html>. [Accessed October 2019].
- [7] T. L. Ellison, J. N. Evans and J. Pike, "Minecraft, Teachers, Parents, and Learning: What They Need to Know and Understand," *School Community Journal*, vol. 26, no. No. 2, 2016.
- [8] C. Vicari, B. Joseph, B. Klimowicz, H. Jaris, S. Asseltine and J. Levin, "The Human Microbiome World: Using Minecraft to Enhance Microbiology Learning," *International Journal of Designs for Learning*, vol. 10, no. 1, pp. 116-130, 2019.
- [9] UkEdChat, "Feature: How One School Turned Minecraft into STEAM," 8 July 2014. [Online]. Available: <https://ukedchat.com/2014/07/08/feature-how-one-school-turned-minecraft-into-steam/>. [Accessed December 2019].
- [10] "Hour of Code," [Online]. Available: <https://www.khanacademy.org/hourofcode>. [Accessed December 2019].
- [11] "MakeCode," [Online]. Available: <https://www.microsoft.com/er-us/makecode>. [Accessed December 2019].
- [12] "Hour of Code Activities," [Online]. Available: <https://hourofcode.com/us/learn>. [Accessed December 2019].
- [13] J. F. Roscoe and S. Fearn, "Teaching Computational Thinking by Playing Games and Building Robots," in *Proceedings - 2014 International Conference on Interactive Technologies and Games, iTAG 2014*, 2014/12/16.
- [14] S. Papert and I. Harel, "Situating Constructionism," in *Constructionism*, Ablex Publishing Corporation, 1991.
- [15] D. Webb, A. Repenning, Grover and C. Brand, "Beyond Minecraft: Facilitating Computational Thinking through Modeling and Programming in 3D," *IEEE Computer Graphics and Application*, May 2014.
- [16] B. De Schutter, D. Black and H. H. Nap, "Teaching Older Adults to Play Minecraft," in *Foundations of Digital Games, At Pacific Grove*, 2015/01/01.
- [17] J. Jenson and M. Droumeva, "Exploring Media Literacy and Computational Thinking: A Game Maker Curriculum Study," *Electronic Journal of e-Learning*, vol. 14, no. 2, pp. pp111-121, 2016.
- [18] C. Hoadley and J. P. VanHaneghan, "The Learning Sciences: Where They Came From and What It Means For Instructional Designers," in *Trends and Issues in Instructional Design and Technology, 3rd Edition*, New York, Pearson, 2011.