

Chapter 14

Video Games: A Potential Vehicle for Teaching Computational Thinking



Sue Inn Ch'ng, Yeh Ching Low, Yun Li Lee, Wai Chong Chia
and Lee Seng Yeong

Abstract Previous studies in computer science education show that game playing is negatively correlated with success in introductory programming classes in (Wilson, Shrock ACM SIGCSE Bulletin, vol 33, pp 184–188, 2001). However, informally, we observed that students who have previous gaming experience take to programming tasks easier than those without gaming experience. There have also been recent studies that show that playing strategic video games can improve problem-solving skills which is an essential skill in program design. This chapter presents the findings of our study to identify if a correlation between previous gaming experience (game playing) and individual computational thinking (CT) skills exists. To achieve this, a survey was administered on undergraduate students undertaking an introductory computing course to collect data on their gaming history and an individual assignment on Scratch. Each project was subsequently analysed to determine the level of mastery of core CT skills. Cochran–Armitage test of trend was then executed on each CT skill category with respect to the coded gaming experience. The results obtained during our analysis shows a correlation between gaming experience and specific categories of the CT skills domain particularly in the area of abstraction and problem-solving and user interactivity. The outcome of our study should be beneficial as ways to leverage on students' gaming experience in the classroom will also be discussed.

Keywords Computational thinking · Game-based learning · Games and learning · Scratch assignment · Introductory programming

S. I. Ch'ng (✉) · Y. C. Low · Y. L. Lee · W. C. Chia · L. S. Yeong
Sunway University, Subang Jaya, Malaysia
e-mail: sueinn@sunway.edu.my

Y. C. Low
e-mail: yehchingl@sunway.edu.my

Y. L. Lee
e-mail: yunlil@sunway.edu.my

W. C. Chia
e-mail: waichongc@sunway.edu.my

L. S. Yeong
e-mail: leesengy@sunway.edu.my

14.1 Introduction

Computational thinking has been touted as a twenty-first century skill that is as important as reading, writing and arithmetic (Wing, 2006). The term computational thinking was initially coined by Seymour Papert in his book *Mindstorms* (p. 182) (Papert, 1980) and further elaborated in (Papert, 1996; Barba, 2016). It was only in the communication by Wing (2006) did the term become popularized. In this communication, Wing described computational thinking as ‘solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science’. Since then, different researchers and technological groups have suggested that computational thinking involves a number of subskills; with each entity adding their own interpretation as to the key skills that encompass computational thinking. Many educational programmes are then devised around these skill definitions to introduce computational thinking to children. However, many of these programs use programming tools and environments to expose students to computational thinking bringing about the misconception that computational thinking is equated to computer science and subsequently equivalent to ‘programming’. As highlighted in (Fletcher & Lu, 2009), teaching computational thinking is not as simple as repackaging CS1, or CS0, and teaching it at an earlier stage. It is also acknowledged that without problem analysis skills, students who are proficient at programming languages would fail to create their own solutions (Koulouri, Lauria, & Macredie, 2015). However, problem-solving skills is not something that can be developed by every student over a short period of time particularly a semester long class. It requires students to experience and reflect on the consequences of their actions. This takes time, practice and effort that sometimes a semester-long course does not permit.

Digital games have been around since the creation of computers and long before computational thinking was popularized and labelled as an important skill. The advent of mobile computing platforms allows digital games to be easily obtained via the internet and gaming to be done anywhere and anytime. Video games which were once confiscated in classrooms are now being adopted by educators as a key teaching tool (Shreve, 2005). This comes as no surprise because digital games contain interactive, engaging and immersive elements that have educational affordances (Frazer, Argles, & Wills, 2008; Gee, 2005). According to (Klopfer, Osterweil, & Salen, 2009), the use of games in formal education can take the form of two approaches: (a) adoption of commercial, off-the-shelf (COTS) games or (b) the application of games in the traditional classroom setting. For the field of CT education, works reported for the former approach are rare whereas the latter approach can take the form of game design assignments or use of serious games. Game design assignments are assignments whereby students are given the task to design and create games to demonstrate the application of learnt technical concepts (Basawapatna, Koh, & Repenning, 2010; Leutenegger & Edgington, 2007; Monroy-Hernández & Resnick, 2008). Computer science (CS) educators have also explored the use of specially created games, also known as serious games, incorporated into traditional lesson plans so that students

learn technical concepts through gameplay (Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012; Liu, Cheng, & Huang, 2011; Muratet, Torguet, Jessel, & Viallet, 2009). Despite studies (Becker, 2001; Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012; Liu, Cheng, & Huang, 2011) reporting an improvement in student engagement and motivation towards the CS content, these studies do not investigate the adoption rate of these games as leisure activities at the end of the course or the effects of extended usage of the designed serious games on students' problem-solving and/or programming skills over time.

On the other hand, children and adults learn best by playing. The work by (Ch'ng, Lee, Chia, & Yeong, 2017) delineates the gameplay elements possessed by popular COTS games for different game genres that support key skills in computational thinking. However, the study did not determine if there is indeed a correlation between playing video games and the mastery of key computational thinking skills. In this chapter, we therefore present our research design and findings to answer the research question on whether past gaming experience does influence specific computational thinking skills. If so, video games can be used as a vehicle to train students to think logically in a fun environment as an alternative to the forceful use of serious games or soldiering through the learning of a programming languages to teach students computational thinking.

14.2 Computational Thinking Skills

The main idea behind the computational thinking movement is that knowledge and skills derived from the field of computer science has far-reaching applications that can be beneficial to other fields too. However, since its conception, there have been different definitions about the skills that encompass computational thinking skills. Some of these definitions can be tightly coupled to programming while others are more loosely defined and general. For example, the CT skills listed by Moreno-León, Robles, & Román-González (2015) is more closely related to programming while the definitions provided by (Lee, Mauriello, Ahn, & Bederson, 2014) is general in nature with little reference made to programming. Barr and Stephenson (Barr & Stephenson, 2011) provided examples on how the nine core¹ CT concepts and capabilities may be embedded in different discipline activities. Table 14.1 lists the different definitions of CT skills by different parties. A look at these definitions shows a repetition and overlap in some of the skills defined such as abstraction, algorithm design and problem decomposition.

It is a common practice for researchers (Berland & Lee, 2012; Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012) in the field of CT education to usually formulate their own definition of CT skills by rationalizing from literature and existing defini-

¹Definition proposed by the American Computer Science Teacher Association (CSTA) and International Society for Technology in Education (ISTE) for use in K-12 education.

Table 14.1 Table of comparison listing the difference skills encompassing CT defined by different parties

Organization/Researchers	CT skills	Definitions
Google for education (Google, 2018)	Abstraction	Identifying and extracting relevant information to define main idea(s)
	Algorithm design	Creating an ordered series of instructions for solving similar problems or for doing a task
	Automation	Having computers or machines do repetitive tasks
	Data collection	Gathering information
	Data analysis	Making sense of data by finding patterns or developing insights
	Data representation	Depicting and organizing data in appropriate graphs, charts, words or images
	Decomposition	Breaking down data, processes or problems into smaller, manageable chunks
	Parallelization	Simultaneously processing or smaller tasks from a larger task to more efficiently reach a common goal
	Pattern generalization	Creating models, rules, principles, or theories of observed patterns to test predicted outcomes
	Pattern recognition	Observing patterns, trends and regularities in data
	Simulation	Developing a model to imitate real-world processes
BBC Bitesize (BBC, 2018)	Decomposition	Breaking down a complex problem or system into smaller, more manageable parts
	Pattern recognition	Looking for similarities among and within problems
	Abstraction	Focusing on the important information only, ignoring irrelevant detail
	Algorithms	Developing a step-by-step solution to the problem, or the rules to follow to solve the problem
Barr and Stephenson (2011)	Data collection	The process of gathering appropriate information
	Data analysis	Making sense of data, finding patterns and drawing conclusions
	Data representation	Depicting and organizing data in appropriate graphs, charts, words or images

(continued)

Table 14.1 (continued)

Organization/Researchers	CT skills	Definitions
	Problem decomposition	Breaking down tasks into smaller manageable parts
	Abstraction	Reducing complexity to define main idea
	Algorithms and procedures	Series of ordered steps taken to solve a problem or achieve some end
	Automation	Having computers or machines do repetitive or tedious tasks
	Parallelization	Organize resources to simultaneously carry out tasks to reach a common goal
	Simulation	Representation or model of a process. Simulation also involves running experiments using models
Berland and Lee (2012)	Conditional logic	Use of 'if-then-else' construct
	Algorithm building	Set of instructions
	Debugging	Act of determining problems in order to fix rules that are malfunctioning
	Simulation	Modelling or testing of algorithms or logic
	Distributed cognition	Rule-based actions
Lee, Mauriello, Ahn, and Bederson (2014)	Algorithmic thinking	Logical steps required for constructing a solution to a given problem
	Decomposition	Process of breaking down a large problem into smaller sub-problems or details
	Pattern recognition	Identification of similar phenomenon
	Pattern generalization and abstraction	Solving problems of a similar type because of past experience solving this type of problem
	Unarticulated instances	Performing an action as a result of thought process that could not be articulated
Kazimoglu, Kiernan, Bacon, and MacKinnon (2012)	Problem-solving	Decomposing the problem to generate alternative representations and to determine if the problem can be solved computationally
	Building algorithms	Construction of step-by-step procedures for solving a particular problem
	Debugging	Analysing problems and errors in logic or in activities
	Simulation	Demonstration of solution to proof that it works

(continued)

Table 14.1 (continued)

Organization/Researchers	CT skills	Definitions
Moreno-León, Robles, and Román-González (2015)	Socializing	Brainstorming and assessment of incidents/strategies among multiple parties
	Abstraction and problem decomposition	The ability to break a problem into smaller parts that are easier to understand, program and debug
	Logical thinking	Use of instructions to perform different actions depending on the situation
	Synchronization	Use of instructions to enable the characters to perform in the desired order
	Parallelism	Use of instructions to make several events can occur simultaneously
	Flow control	Use of sequence and repetition blocks to control the behaviour of characters
	User interactivity	Allows players to perform actions that can provoke new situations in the project through the use of input peripherals such as keyboard, mouse or webcam
	Data representation	Ability to set information for all characters in order to execute the program properly

tions. For our work in this chapter, we will use the CT skills defined by (Moreno-León, Robles, & Román-González, 2015).

14.3 Methodology

Data was collected from 736 first-year undergraduate students taking an ‘Introduction to Computers’ course at a private university in Malaysia. This course teaches students the basic concepts of what computers are, how computers store and process information, communicate with each other and applications of computers in daily life. The course also covers a brief introduction to programming, more specifically software design lifecycle, basic programming constructs and the different types of tools that can be used to develop software applications. Students were taught how to use Microsoft© Office tools to solve problems and basic game design using MIT Scratch (MIT, 2016) during the practical sessions of the course. Scratch was chosen as the development platform so that students can focus on the design of the solution instead of the syntax of a particular programming language. The students were from two different schools—School of Computing and School of Business. Table 14.2 shows the composition of students from each school.

For the individual assignment, students were tasked to design a Catching Name Game—a game where the objective of the game is to collect characters that appear

Table 14.2 Gender distribution and composition of students from each school

Gender	Computing	Business	Total
Male	114	213	327
Female	41	368	409
Total	155	581	736

on the screen to spell out words. The students were given the freedom to determine the type of gameplay that they wish to submit for the assignment but the game must include components of their own name inside the game to reduce the possibility of students passing off someone else's work as their own. An online questionnaire was administered on the students after they have submitted their Scratch Assignment to collect information on their gaming habits. Through the questionnaire students were asked to self-report their gaming habits (now and when they were young) through multiple choice questions (starting age and frequency of play) and open-ended questions (name of the favourite video game), refer to Appendix 1 for the full questionnaire. Based on the premise that gaming is a memorable experience during the students' childhood or adolescence, students who truly played games and for those who have spent a sizeable amount of their time doing this would at the very least remember the name of the game that they have played and/or be able to describe the gameplay of that particular game.

The starting age at which students reported their first foray into games and the responses from the open-ended question was used as a reference point to check the validity of the responses. For example, if the respondents claim that they started playing Candy Crush at an age of less than 6 years old, this response would be deemed invalid because Candy Crush was only released in the year 2012. Responses that were incomplete or those who gave nonexistent/invalid games for either instance were ignored in the study. If the game title provided by the respondents at either point of times—young and current—is valid, the respondents were categorized as having 'previous gaming experience'. The assessment of computational thinking skills of the students' Scratch project was done using the free web-based tool Dr. Scratch. The tool analyses each Scratch project in seven CT dimension (Moreno-León, Robles, & Román-González, 2015). Each dimension was then given a score that ranges from 0 to 3 according to the criteria provided in Table 14.3. The addition of the partial score from each dimension yields a CT Score and, based on this score, different feedback was also provided by the website to provide the student's information and suggestions on improvement.

Since each of the CT dimension constitutes an ordinal variable and 'Gaming Experience' is a nominal categorical variable, Cochran–Armitage test of trend was utilized to investigate the relationship between each CT dimension and 'Gaming Experience'. All statistical analysis was conducted using SAS Enterprise Guide software.

Table 14.3 Description for each CT dimension assessed by Dr. Scratch (Moreno-León, Robles, & Román-González, 2015; Robles, Moreno-León, Aivaloglou, & Hermans, 2017)

CT dimension	Basic (1 point)	Developing (2 points)	Proficient (3 points)
Abstraction and problem decomposition	More than one script and more than one sprite	Definition of blocks (creation of custom blocks)	Use of clones (instances of sprites)
Parallelism	Two scripts on green flag	Two scripts on key pressed or on the same sprite clicked	Two scripts on when I receive message, or video or input audio or when backdrop changes to
Logical thinking	Use of 'If' blocks	Use of 'If ... else' blocks	Logical operations
Synchronization	Wait	Message broadcast, stop all, stop program	Wait until, when backdrop change to, broadcast and wait
Flow control	Sequence of blocks	Repeat, forever	Repeat until
User interactivity	Green flag	Key pressed, sprite clicked, ask and wait, mouse blocks	Webcam, input sound
Data representation	Modifiers of sprites properties	Operations on variables	Operations on lists

14.4 Results and Discussion

Based on the results obtained in Table 14.4, it was observed that there is a strong evidence (p -value = 0.0100) of an association between the CT dimension of Abstraction and Problem Decomposition and gaming experience of students. A plausible explanation for this correlation is that all games, regardless of game genre, have goals/missions and a reward mechanism that entices players to continue playing—an attribute which makes games engaging. Players would then try to find ways to maximize these rewards while minimizing damage to their game characters during the gameplay (Ch'ng, Lee, Chia, & Yeong, 2017; Gee, 2008). This feature in all games requires players to determine the problem that they are currently encountering and to devise new solutions based on whatever information, which may differ greatly depending on the game, that they have at hand. It was noted in (Adachi & Willoughby, 2013) that these were also the exact features that promote problem-solving skills. We posit that, perhaps, this is the attribute of COTs games that provide informal training to its players in the CT dimension of Abstraction and Problem Decomposition.

It was also observed that there is a weak evidence (p -value 0.0470) to support the hypothesis of an association between the CT dimension of User Interactivity and gaming experience. A possible explanation to this phenomenon is that when

Table 14.4 Cochran–Armitage trend test results

CT dimension	Cochran–Armitage z-statistic	<i>p</i> -value
Abstraction and problem decomposition	2.3274	0.0100
Parallelism	−0.3318	0.3700
Logical thinking	0.6346	0.2628
Synchronization	−0.2861	0.3874
Flow control	0.9793	0.1637
User interactivity	−1.6750	0.0470
Data representation	0.5875	0.2784

students are exposed to video games and through repeated play over the years, they will indirectly pick up the basic elements needed to interact with computer software such as use of keyboard to input text and mouse to make selections; compared to those who have minimal or no exposure. Further investigation needs to be conducted if the same observation applies to those who are exposed to repeated general software usage and not only video games to determine if this observation holds.

Since the *p*-values of the Cochran–Armitage z-statistic is greater than 0.05 for the other CT dimensions considered in our study, we conclude that there is insufficient evidence from our sample to support the hypothesis that there is an association between ‘Parallelism’, ‘Logical Thinking’, ‘Synchronization’, ‘Flow Control’, ‘Data Representation’ and gaming experience of the students, respectively.

14.5 Implications for Educators and Researchers

Our findings show that there is a correlation between previous gaming experience and the CT dimensions of Abstraction and User Interactivity. At this moment, we cannot tell which particular aspects of specific COTs video games that support the cultivation of these particular skills or why the correlation exists without further investigations. However, our findings does encourage the idea that COTs games possess the potential to cultivate skills as suspected by Klopfer, Osterweil, and Salen (2009), Shreve (2005). The question now is: How do we actually harvest it to make it work for CT education?

The report by Klopfer, Osterweil, and Salen (2009) has presented creative ways in which games can be incorporated into classrooms to support different aspects of learning. The most common approaches that are currently utilized for CT education is the use of games as programming and reflective systems via game development assignments and serious games respectively. However, the creation of serious games takes time and skills that most educators do not possess and pale in comparison

to the expertise possessed by the games industry. An alternative is to use COTs games instead of serious games. The online content provided by (BBC, 2018) on thinking computationally utilizes a simple platform game to illustrate each aspect of CT skill without any programming. It should also be possible for educators to do the same in the classroom. For example, students can be tasked to pick and play their own choice of game from a pool of games (Blizzard, 2018) and subsequently asked to share about the challenges that they faced while trying to clear particular levels of the game (abstraction and problem decomposition) and how they overcome those challenges (algorithm). Another approach is to group students according to the choice of games that they have previously picked to have smaller discussions and knowledge exchange on how particular levels of the game are solved. A simple worksheet is included in Appendix 2 for students to work on before the face-to-face session on introduction of CT concept Abstraction and Problem Decomposition. The students' work can also be used as examples for discussion on the CT component of Algorithmic Thinking.

Another issue that educators face with incorporating games into the classroom is the struggle to cover the mandated curriculum and have games within the same block of time. An unorthodox solution to this problem is to leave play outside the classroom, in its original place, so that students have the freedom to play (explore, experiment and fail without penalties). The limited face-to-face time within the classroom can be used to facilitate discussions on what students did and to guide them to reflect on their own observations or actions during the play. Class activities that can be done within the classroom may be something as simple as free discussions of their gaming experiences of popular game titles. When conversations are started by students and led by students, students are given the opportunity to identify their own strengths and build their confidence in communication and social skills. The study by (Berland & Lee, 2012) observed that players exhibit skills to identify problems and build solutions based on their own observation and actions during gameplay. The same attribute can probably be observed on online gaming forums where players virtually gather to discuss issues that they face while trying to complete levels of a video game. However, with this approach, it is difficult to standardize the learning outcome or control the actions of the students since it is not within the teachers' power to control students' actions outside the classroom as it is within the classroom. A solution to this is to assign students the task to create a piece of work (video/written piece/blog) at the end of the semester demonstrating how they identify and solve problems that they face while playing their favourite games. For those who are unenthusiastic about gaming, contrary to common beliefs that everyone plays video games, the act of gaming can also be replaced with their own hobbies—crafting, sports, collecting items.

The idea of incorporating video games into education is an idea that has been around for some time (Annetta, 2008). In fact, video games have been reportedly used to teach city planning (Terzano & Morckel, 2017) and English as a Second Language (Miller & Hegelheimer, 2006) with positive feedback. Video game players were also found to fare better at surgical skills in the study conducted by (Rosser et al., 2007) and were noted to have better graduate attributes than non-gamers in

(Barr, 2017). For the field of CT education, we believe that video games have the same potential to be used as an effective tool for teaching and learning within and outside the classroom. The results reported in our study provide preliminary proof of this. In the future, we plan to obtain concrete evidence advocating the use of video games as a potential vehicle for learning and teaching computational thinking by implementing the ideas put forth in a real classroom environment.

Appendix 1: Survey—Video Game Experience

We are conducting a research about the relationship between previous gaming experience and game development. Your response to these questions will not affect your grades for this subject. Please answer the following questions.

1. Scratch Project URL: _____
2. Gender
 - Yes
 - No
3. Do you play video games when you were younger?
 - Yes (Go to question 3.)
 - No (Go to question 6.)
4. How old were you when you started playing video games?
 - Less than 6 years old
 - 7–9 years old
 - 10–12 years old
 - 13–15 years old
 - 16–18 years old
5. Name your favourite video game when you were young.

6. How often do you play this game back then?
 - Rarely
 - Occasionally
 - Frequently
7. Do you actively play video games over the past two years?
 - Yes
 - No
8. Name your current favourite video game. _____

9. How long do you spend each day, on average, playing your favourite video game?

- <1 h
- 1–2 h
- 3–4 h
- 3–4 h
- 5–6 h
- >6 h

Appendix 2: Homework Exercise—Describing My Favourite Game

Your task in this exercise is to describe the steps that you take to play one of the games that you frequently play at home. Games in this case can be any type of games ranging from board games, video games on the personal computer, mobile phone or television consoles or even physical activity game.

Components	Instructions
Game title	<i>Name of the chosen game</i>
Game description	<i>Provide a short description of the chosen game</i>
Goal of the game	<i>State the main goal of the game that players must fulfil to win the game</i>
Game strategy	<i>Describe the steps that you take as a player to achieve the goal of the game. You can include screenshots or images to aid your explanation. Be as detailed as possible in your description so that another new player can use your description as a walkthrough to complete the game on his own</i>
Sub-objective(s) of the game	<i>[Optional] Some games have mini-games/missions embedded inside the game itself. If the game has many mini-games/missions, select one and state the goal that players must fulfil in order to win or complete the mini-game/mission</i>
Mini-game/Mission strategy	<i>[Optional] Describe the steps that you take as a player to achieve the goal of the mini-game/mission within the game. You can include screenshots or images to aid your explanation. Be as detailed as possible in your description so that another new player can use your description as a walkthrough to complete the same mini-game or mission on his own</i>

References

- Adachi, P. J. C., & Willoughby, T. (2013). More than just fun and games: The longitudinal relationships between strategic video games, self-reported problem solving skills, and academic grades. *Journal of Youth and Adolescence*, 42(7), 1041–1052.
- Annetta, L. A. (2008). Video games in education: Why they should be used and how they are being used. *Theory into Practice*, 47(3), 229–239.
- Barba, L. A. (2016). Computational thinking: I do not think it means what you think it means. Retrieved January 11, 2018, from <http://lorenabarba.com/blog/computational-thinking-i-do-not-think-it-means-what-you-think-it-means/>.
- Barr, M. (2017). Video games can develop graduate skills in higher education students: A randomised trial. *Computers & Education*, 113, 86–97.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.
- Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 224–228). ACM.
- BBC. (2018). BBC—Introduction to computational thinking. Retrieved from <https://www.bbc.co.uk/education/guides/zp92mp3/revision/1>.
- Becker, K. (2001). Teaching with games: The minesweeper and asteroids experience. *Journal of Computing Sciences in Colleges*, 17(2), 23–33.
- Berland, M., & Lee, V. R. (2012). Collaborative strategic board games as a site for distributed computational thinking. *Developments in Current Game-Based Learning Design and Deployment*, 285.
- Blizzard. (2018). Blizzard entertainment: Classic games. Retrieved January 24, 2018, from <http://eu.blizzard.com/en-gb/games/legacy/>.
- Ch'ng, S., Lee, Y., Chia, W., & Yeong, L. (2017). Computational thinking affordances in video games. *Siu-Cheung KONG The Education University of Hong Kong, Hong Kong*, 133
- Fletcher, G. H. L., & Lu, J. J. (2009). Education Human computing skills: Rethinking the K-12 experience. *Communications of the ACM*, 52(2), 23–25.
- Frazer, A., Argles, D., & Wills, G. (2008). The same, but different: The educational affordances of different gaming genres. In *Eighth IEEE International Conference on Advanced Learning Technologies, 2008. ICALT '08*. (pp. 891–893). IEEE.
- Gee, J. P. (2005). Good video games and good learning. In *Phi Kappa Phi Forum* (Vol. 85, p. 33). The Honor Society of Phi Kappa Phi.
- Gee, J. P. (2008). Learning and games. In K. Salen (Ed.), *The ecology of games: Connecting youth, games, and learning* (pp. 21–40). MIT Press.
- Google. (2018). Google for education: Exploring computational thinking. Retrieved January 11, 2018, from <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>.
- Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9, 522–531.
- Klopper, E., Osterweil, S., & Salen, K. (2009). *Moving learning games forward*. Cambridge, MA: The Education Arcade.
- Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, 14(4), 26.
- Lee, T. Y., Mauriello, M. L., Ahn, J., & Bederson, B. B. (2014). CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction*, 2(1), 26–33.
- Leutenegger, S., & Edgington, J. (2007). A games first approach to teaching introductory programming. In *ACM SIGCSE Bulletin* (Vol. 39, pp. 115–118). ACM.

- Liu, C.-C., Cheng, Y.-B., & Huang, C.-W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, 57(3), 1907–1918.
- Miller, M., & Hegelheimer, V. (2006). The SIMs meet ESL incorporating authentic computer simulation games into the language classroom. *Interactive Technology and Smart Education*, 3(4), 311–328.
- MIT. (2016). Scratch—Imagine, program, share. Retrieved from <https://scratch.mit.edu/>.
- Monroy-Hernández, A., & Resnick, M. (2008). FEATURE empowering kids to create and share programmable media. *Interactions*, 15(2), 50–53.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 46, 1–23.
- Muratet, M., Torguet, P., Jessel, J.-P., & Viallet, F. (2009). Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, 2009, 3.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95–123.
- Robles, G., Moreno-León, J., Aivaloglou, E., & Hermans, F. (2017). Software clones in scratch projects: On the presence of copy-and-paste in computational thinking learning. In *2017 IEEE 11th International Workshop on Software Clones (IWSC)* (pp. 1–7). IEEE.
- Rosser, J. C., Lynch, P. J., Cuddihy, L., Gentile, D. A., Klonsky, J., & Merrell, R. (2007). The impact of video games on training surgeons in the 21st century. *Archives of Surgery*, 142(2), 181–186.
- Shreve, J. (2005). Let the games begin. Video games, once confiscated in class, are now a key teaching tool. If they're done right. *George Lucas Educational Foundation*.
- Terzano, K., & Morckel, V. (2017). SimCity in the community planning classroom: Effects on student knowledge, interests, and perceptions of the discipline of planning. *Journal of Planning Education and Research*, 37(1), 95–105.
- Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. In *ACM SIGCSE Bulletin* (Vol. 33, pp. 184–188). ACM.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

