Kong, S.C., Hoppe, H.U., Hsu, T.C., Huang, R.H., Kuo, B.C., Li, K.Y., Looi, C.K., Milrad, M., Shih, J.L., Sin, K.F., Song, K.S., Specht, M., Sullivan, F., & Vahrenhold, J. (Eds.). (2020). *Proceedings of International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Developing Computational Thinking Through Tinkering in Engineering Design

Ashutosh RAINA[1*], Sridhar IYER[2], Sahana MURTHY[3]
[1,2,3] Indian Institute of Technology Bombay, India
raina.ashu@iitb.ac.in, sri@iitb.ac.in, sahanamurthy@iitb.ac.in

ABSTRACT

Artefact creation as part of constructionist approaches towards learning has seen an increase pertaining to the growth and ease of availability of design tools. Projects that involve artefact creation allows the learner to experience the problem solving process while being situated in a real-life context. Tinkering is one such approach to problem-solving. In this paper, we present a design of our tinkering intervention for teaching and learning of computational thinking. The intervention is a composition of four major components, namely the Pedagogy, Problem, Resources and Mentor. The proposed Explore-Solve-Evolve pedagogy incorporates aspects of constructionism, progressive formalisation, learning situated in a real-life context and immediate feedback for reflection. Lego Mindstorm is provided as a building resource, and an app seamlessly provides information about the resources. The mentor encourages the learners towards exploration and play with the resources in the problem space and scaffolds them with strategies to overcome challenges. A proposed study has been discussed to further understand the development of CT with tinkering. The paper is concluded with presenting the mapping between the phases of our intervention and the three dimensions of the CT framework.

KEYWORDS

computational thinking, tinkering, intervention, robotics

1. INTRODUCTION

Computational thinking has been defined as “The thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (Brennan & Resnik, 2012). Computational thinking has been taught not only through programming but also through activities like playing games, building a robot to solve challenges, creating e-textiles and range of activities that involve concepts of computational thinking. The idea is to be able to express yourself using computational artefacts which have been identified as an essential aspect of computational literacy. While developing artefacts, learners also deal with failure in physical components and compatibility issues that can be frustrating. However, they are an essential part of solving problems where one is often required to use of computational thinking, not limited to just writing code (which has been termed as the material aspect of CT). In addition to the material aspects of CT (which is the *how*), learning-environments that include artefact building as a part of the problem-solving process also focus on the social (which is the *where* and *whom*) and extends it to the cognitive aspects (which describe the *why*). Building artefacts to solve a given problem situates the problem-solving process in a physical context that is closer to an authentic scenario.

One such practice that includes artefact creation with problem-solving is tinkering. It has been considered as a novice and expert practice which sets it apart from most of the classroom practices (Danielak, 2014). It does not make tinkering better or worse but it does make it an authentic professional practice (Berland, 2016). Tinkering provides the opportunity to work in a realtime environment with immediate feedback on actions taken hence making it a potential means for developing computational thinking. We believe that tinkering with robotics kits like Lego Mindstorm provide a medium and opportunities for the development of computational thinking. We are interested in the ways that tinkering activities with programmable tangible robotics kits, like the Lego Mindstorm, can support the development of computational thinking in students in higher education which is highly dependent on learning of programming languages (Brennan & Resnick, 2012).

2. THEORETICAL BASIS

2.1. *CT Framework*

Computational-Thinking has further been classified into *CT Concepts* that learners develop while learning to program like loops, conditionals, sequences, parallelism, data structures, operators, event handling, procedures and initialisation. *CT Practices* that learners repeatedly demonstrate in the programming process like problem formulation, problem decomposition, abstracting and modularising, algorithmic thinking, reusing and remixing, being iterative and incremental, testing and debugging. *CT Perspective's* talk about the Learners' understanding of themselves and their relationships with others and the world of technology, also termed as Computational Identity (Brennan & Resnik, 2012). It also includes programming empowerment as well as provides a perspective of expressing, connecting and questioning with programming. The elements of CT as mentioned earlier in its three dimensions have also been included in the operational definition of CT for K-12 education by the International Society for Technology in Education and Computer Science Teachers Association (ISTE & CSTA, 2011).

2.2. *Tinkering Practice*

The growing availability design tools have led to a commitment to learning through design activities in a constructionist approach (Harel & Papert) to a level of learning that highlights the importance of young people engaging in the development of external artefacts (Kafai & Resnick, 1996). Besides, progressive formalisation (Bransford, Brown & Cocking, 2000) requires teaching to be designed to encourage students to build on their informal ideas in a gradual, structured manner that enables them to acquire the concepts and procedures of the discipline. Moreover Learning situated in a real-life context (Bransford, Sherwood, Hasselbring, Kinzer, & Williams, 1990) enables a better understanding of abstract concepts by establishing there need in a real-life context using everyday examples. In addition to situated learning, play becomes an essential tool for learning in real-life context as it allows experimentation with the available resources and one's ideas in the actual problem space with just in time feedback that enables reflection. It also allows one to take multiple perspectives on an action and its impact, which is an essential social skill for the development of the mind (Bailey, 2002). Tinkering has been addressed to be at the intersection of all the above (Roque, Rusk & Blanton, 2013). A definition of tinkering calls it as a playful, experimental, iterative style of engagement, in which people are continually reassessing their goals, exploring new paths, and imagining new possibilities (Honey & Kanter, 2013). Here play has been referred to as experimental play. Tinkering provides a multitude of possible paths taken progressively while situated in problem space working with immediate feedback.

2.3. Explore-Solve-Evolve Pedagogy

Based on our synthesis from the literature on tinkering for problem-solving, we have identified a few operational aspects of tinkering as Exploration, Play and Reflection. Exploration is used to determine the affordances or *can do's* of the available resources and possible solution or *want to do* for the problem at hand. Play is used to determine if a solution could emerge by mapping the *can do's* and *want to do's.* Reflection is used to overcome states of *stuck* and *fixation* that arise due to unexpected contingencies (exception violation (Schank, 1983)) or failure. Using strategies like questioning, repurposing, reflective strategies on productive failure (Kapur, 2008) provide the means to overcome such challenges. Reflection on the tinkering trajectories to enable modification of understanding and learning about the problem space.

We used the above operational aspects along with tinkering frameworks like Spark, Sustain and Deepen (Honey & Kanter, 2013), and Think, Make and Improve (Martinez & Stager, 2013) to derive a three-phase pedagogy named Explore, Solve and Evolve for taking a tinkering approach to computational thinking. The features of free exploration to capture intrinsic motivation have been incorporated in the explore phase. Progressive formalisation has been implemented in all the three phases of explore, solve and evolve. In explore learners start with small problems situated in context robotics, which requires them to interact with the physical space using the components of the robotics kits to solve the problem. In the problem given in the solve phase allows the learners to build their solutions with small component problems solved in the previous phase. This method also allows the reuse and iteration of previous solutions. Finally, in evolve, the learners frame and solve a problem to advance the solution they develop in the solve phase. The learning environment comprises of building resources and some pre-build solution of similar problems.

We believe the features of the pedagogical design and the element of the learning environment based on tinkering which has been aligned to the operational elements of CT aided with an explicit reflection on the action will lead to the development of CT among the students. The problems that have been chosen align to the High school curriculum of various educational boards in India.

Table 1. Summary of the pedagogy with its mapping to available resources and activities to be performed.

Pedagogy	**Problem**	**Resources**		**Activities**	
		Building	**Information**	**Learner**	**Mentor**
Explore	Small problems that are a part of the challenge for the next phase. *E.g. build a chases with wheels.*	With the focus on use of basic individual resources and their affordances. *E.g. Connecting motors and the EV 3 brick.*	Using the AR component view from the app for affordances of the individual resources.	Interaction with resources while solving problems to understand their affordances.	Encourage exploration and play with resources
Solve	One open-ended challenge that is derives from problems of “Explore” phase with opportunities for reuse. *E.g. build a wheeled bot that can move and turn.*	With the focus on combined use of the resources and their interactions with each other. *E.g. Mounting the EV3 on the chassis and building the turning mechanism.*	Additionally, information about the interaction of different resources and available use cases. Scaffolds for techniques for getting unstuck	Determining the sub problems and primary functional modules. Use pre-built solutions from previous phase	Additionally, provide prompts and scaffolds for techniques like reflection and productive failure.
Evolve	Additional challenge to increase the complexity of the previous challenge requiring the need of abstraction modularization and iteration. *E.g. Make the bot avoid obstacles*	Use of additional complex resources to enhance capability of the current build. *E.g. Adding IR, Ultrasonic sensors and building a parallel process of obstacle detection.*	Similar as above	Frame the new problem, choose the sub problems and address the sub problems while using techniques to overcome challenges	Indirect guidance using instances from the previous phases.

3. INTERVENTION DESIGN

The Tinkering environment for learning with CT comprises of the problem whose potential solutions derives from CT. Available resources allow free exploration, have a low floor and high ceiling and align to the constructs of computational thinking. Both the problem and the resources ensure the requirement of tinkerability (Resnick & Robinson, 2017). The pedagogy encompasses features like progressive formalisation, alignment to intrinsic motivation, guided reflection. Finally, a mentor provides scaffolds for the use of strategies like re-purposing, question-posing and reflection for working with expectation violation and productive failure. A summary of the entire intervention is as presented in table 1.

3.1. Problems

Though any problem with its corresponding resources could be provided in a tinkering based learning environment, we choose Lego Mindstorm Robotics kit and design a maze that would have to be solved as a part of the activity. This activity provides enough freedom to the learners for designing the robot as per their choice to solve a given maze. Keeping progressive formalisation in mind the problems are divided into two categories. The first category of problems is toy problems that help the learners to explore the resources available in Lego Mindstorm and get used to them. E.g. one of the problems requires the learners to determine the volume of the room given the Lego Mindstorm EV3 brick and the ultrasonic sensor. The objective of this problem is for the learners to understand the usage of ultrasonic sensors and also to be able to build a quick prototype and use the data representation features of the EV3 brick. Additionally, they are being exposed to the concept of input and output of data using physical sensors, or what we call they are getting a sense of the kind of output the sensor can provide. Though this question requires them to work with the ultrasonic sensor, the mentor encourages them to use all possible actuators and sensors to get a sense of the devices. Similarly, one of the problems requires the learners to build a two-wheel powered bot and a four-wheel powered bot to determine the use cases of each configuration. These problems are candidate sub problems to the bigger problem that the participants will solve in the next phase.

In the second phase “Solve” we provide them with a maze that their bot has to navigate. The maze is an NxN matrix where obstacles have been places, and the bot must follow the unblocked edges and reach the destination. The learners are given the maze along with the edges that will be blocked. This problem becomes a standard path traversal problem where the learner must sequence a set of instructions, and the sequence would determine the path that is traversed by the bot. The length of the edges are standard; hence the learners must determine the distance the bot would move and code it accordingly. Though the length is the same distance would vary based on the bot they have built or the motor parameters they are using. Thought a hard-coded solution is not the ideal solution for this problem, the problem the idea is to take the learners through this journey to understand the different solutions and challenges they pose and evolve them towards building using constructs to build better / dynamic/efficient solutions.

In the third phase named “evolve,” they are given a new challenge where they are to program and modify the robot in such a way that it could traverse the maze even if the obstacle locations have not been determined initially. They could add markers on the obstacles for the bot to identify and take action accordingly. The objective here is to allow the learners to understand the concept of functions and modularisation so their bot can take decisions based on the maker. This problem evolves the learners to thinking in terms of higher-order CT concepts while providing them with the freedom of incorporating their idea of how to implement them.

3.2. Resources

Resources in the learning environment refer to the components of the learning environment. These are divided into building resources and information resources. Building resources refer to raw building materials, fabricated building materials and electronic components. As an example, in our case, the building resources would consist of the Lego Mindstorm kit and a few other resources like tape cardboard etc. Further classification of the components could be done based on their nature of use and other characteristics.

Figure 1. Building Resources and Mobile Application

The information-seeking resource consists of repositories of information on a mobile application. The mobile app also has an interface to interact with the learning environment using Augmented Reality. The learners work in the problem space with the available tools and resources to find solutions to the problem at hand. Prior knowledge of affordances of tools and resources available for tinkering through a problem or ability to acquire such information in the time of need is a challenge for learners who intend to take a tinkering approach. Gathering this information from manuals and online resources frequently requires switch context, which inhibits or discourages explorations with the unknown components. Hence this app will enable the learners to seek information about problem statements, help them track their session, provide information about components. The app will have a different section for the different phases of the pedagogue. The app will also act as a platform where prompts and scaffolds will be presented. The apps also enable delivering just in time information by presenting information in an augmented manner to ensure seamlessness, as seen in Figure 2 below.

3.3 Pedagogy

The pedagogy has evolved from our explorations with tinkering and literature (Honey & Kanter, 2013) (Martinez & Stager, 2013). The initial motive is building curiosity into the mind of the learners by exposing them to various complex solutions and stories about solving them. The learners are guided to explore and play with the available solutions to build their understanding of the environment. One of the intended ways of doing it is by starting with candidate subproblems of the main problem that they will be solving in the second phase. These subproblems are introduced as primary problems for exploration with simple resources to interact with and gradually increase the complexity of the problems and the use of resources.

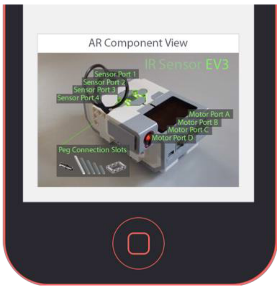

Figure 2. AR Component view of the Lego EV3 brick.

The motive is to encourage exploration of the resources for indented use. At the end of this activity, the students should have an understanding of the different components of the robot, their function, and how they can be arranged/combined to achieve a more significant function.

In the second phase, they are given a problem to solve within the same environment. Initially, the learners are mentored to find a starting point and then are left free to begin working on the starting point of their choice. Here the learners interact with the building resources based on their understanding from the previous phase of the problem. The disposition that learner should take is of experiencing what would happen than predict the outcome by observing or thinking about challenges. With practice, one may be able to predict the outcome by mentally experimenting with the problem space. This experience will later give rise to the needs of the solution or what kind of function/behaviour would be required by the solution. Another challenge they might face is of being stuck. Stuck is to be interpreted as the state when the participants are not able to ask the right questions. Being in one state but still being able to ask the right questions is still considered the state of flow. In the stuck state, the role of the mentor is to guide the participants to ask the correct questions. The app acts as the repositories of information about building resources and examples. Some necessary information maybe even augmented using the app on the resources for a quick understanding. The learns may record their progress on their app as a medium intended for logging. This can also be used by the learners to reflect and make decisions. The final part of this phase of the workshop is to enable reflection on the solutions the learners have built from the problem they were provided. The reflection would be triggered by posing questions regarding the requirements of the problem. The type of question to be posed. The learners will also be encouraged to use their logs to aid this reflection process. These reflections will be recorded by learns in the app. The objective of reflection is to make the explicit realisation of the CT elements and connect them to the activities performed by this. It ensures the development of an understanding of the use of CT as per the three domains.

This stage, the learners will evolve their solution to either enhance their capabilities or refine its function or performance. One of the objectives is to introduce them towards abstraction of primary function and their modularisation. Also, expose them to parallelism. In this phase, the mentor will be available only on demand as the mentor does not take an active role in the solution process. The purpose of the mentor would be to observe learners actions to monitor their approaching. The mentor may choose to intervene in some situations mentioned in the mentor's guide. The intervention would be limited to directing the learner's approach by asking broad questions. The objective of this phase is to determine if the learner can initiate explorations, make observations and ask questions about it. The mentor may choose to allow the learns to exit without building the refined solution if enough evidence of the objective is available. These will be available as guidelines to the mentor. In the final stage of the workshop, the mentor will trigger reflections among the learners on the entire actions to develop an understanding and use of the elements of CT.

3.4. Mentor

The Mentor is more of a facilitator to observe the activities and the process the students are following. The motive of doing this is to help them reflect on their actions. Guide them towards exploration and play. Guide them the overcome challenges by identifying the reasons. The reasons could vary from not being able to construct the intended the solutions, not being able to use the resources at hand, not being able to identify resources and/or the corresponding affordances or unpredicted behaviour. To direct the students to the flow state, the mentor themselves must become a genuine participant of the activity. They should try to figure out what is the problem. The mentor can probe using questions like what seems to be the challenge? What seems to be your approach? If the learner can answer mapping to solving a problem and changing the design, then the learner is actually in the flow state. To probe further, the instructors could explicitly ask "Which questions are you trying to answer?" if the participant shows signs of frustration or seems to have given up. These would be responses like I do not know what to do next, I have tried many things. It will not work. This cannot be solved. The Mentors could guide them by asking questions as stated above that would help them proceed with the approach. The participants could respond with answers that talk about the loss of interest or boredom like I am getting bored, and I do not feel interested in doing the same. I am not able to think more. The mentors could guide them towards skipping the current challenge and work on a different aspect or just ask them to take a break. If the mentor feels the participant is struggling due to lack of information, they may guide them towards the information. The objective is to make them realise that such information can be looked at.

The mentor should be able to take a multi-level view to weight between the more significant problem and the problem they are stuck. The criticality of the current problem for the more significant problem can help determine to solution approach. If it is critical, we need to find a way to work it out or if not, can we manage to solve the bigger problem without the problematic component at hand. The mentor should guide the students via open-ended prompts describing the behaviour of the component at hand. The prompts should target misconception or refer to some other project and explaining the function of the component and have them try it. Another way of doing this is by posing questions starting with What are they trying to achieve? Why are they doing it this way? How will it achieve what they intend to achieve? When and where does this help to solve the bigger picture?

3.5 Proposed Study

The study is targeted at High school students who have just started using programming languages and do not have exposure to Lego Mindstorm robotics kits. This version of the study will be done with one individual per kit. We plan to introduce elements of collaboration in later studies. The objective of the study is to explore the use of tinkering as a strategy for learning elements of CT. Will the tinkering learning environment designed with an alignment of CT elements lead to explicit learning of CT in its three dimensions? This will us with a deeper understanding of the alignment and the features that may or may not work as intended.

The study is based on the Explore, Solve and Evolve pedagogy and distributed over three days. On the first day, the learners will be introduced to Lego kit using the candidate sub-problems. They start with problems to introduce them to the EV3 brick along with the sensor and the motor functions. Similarly, they will be given problems that lead them to explore the construction blocks and beams. The learners are allowed to dismantle a few prebuilt bots. In the final part of the day, they would be given problems that would require them to code, either the prebuilt bots or the bots they have built. The day would end with the mentor asking the learners about the kind of bots they would want to build and making them reflect on their observations and understanding of the building resources. On the second day in the "Solve" phase, the learners would be provided with the challenge of solving a static maze. They could reuse the bots from the previous day or build new ones. At the start of this session, the learners will try to find out the essential requirement of traversing the maze. The bot will have to perform two functions which are moving on clear lines and turning to avoid obstacles. The mentors may lead the participants to play in the problem space to physically experience the problem by manually navigating the maze using a non-motorised bot. Once the participants have realised the essential functions, the instructor will facilitate the participants in realising the needs from the previous exercise and then try to translate them into functions and behaviour for their solutions. Once the desired behaviours have been achieved, learners can move forward to the next essential objective. Learners may perform as many numbers of trials on the maze and only when they determine or the time is done, they would require to demonstrate their solution. The learner determines the final demonstration beforehand. If the learner finishes before the time the mentor asks them to improve the efficiency in terms of time taken by the bot to complete the maze. The day ends with the mentors making the learners reflect on the solution trajectories. The reflection will be carried out through activities where the learner would be told about the CT elements, and they would map it to their solution strategies and later determine one use-case for their application.

On the third and final day called "Evolve," the learners are required to solve a similar maze, but they would not know where the obstacles would be placed. In this case, the obstacles would have a provision to place markers. The learner could use these markers to make the bot respond with a specific action like turning left or right. In this phase, the mentors will gradually reduce the scaffolds and prompt limited to making them recall things they did on the previous day, so they can make associations from what they learned. To increase the complexity of the problem, the standard length between the nodes may vary. The mentor facilitates reflection by having the learners talk about their experience and pointing out key actions they performed and having them articulate what they exactly did and what did they achieve. The mentor may ask learners to demonstrate the use of CT concepts that could be implemented if a given behaviour was to be achieved? Once the reflection session is over the learner are given scenario-based MCQ.

4. CT IN TINKERING

In this paper, we present the design of an intervention and a proposed study to explore the use of tinkering as a means for developing an operational level understanding of the different dimensions of CT. In the explore phase, activities that emphasise the interaction with the sensors and the EV3 brick help the learners to understand with programming is and empowers them with the opportunities of being able to program physical objects. Constructing small artefacts exposes them to CT concepts of operators, procedures etc. In the solve phase, the learners are introduced to sub-problem generation and encouraged to reuse and remix solutions from the previous phase adding a few more CT concepts. In the evolve phase, the learners are made to reflect on the iterative and incremental way of solving problems. The slight increase in complexity of the problem introduces them concepts of abstraction modularisation of the turning function. They also learn about parallel processing to achieve the motion and obstacle detection function. Table 1 below provides a summary of the mapping between the activities performed in the tinkering environment and operational elements of CT from the CT Framework. Table 2 also presents the distribution based on the three dimensional CT framework aligned to the essential phases of our tinkering pedagogy. We believe that by such an alignment of dimensions of CT with our tinkering pedagogue, the learners will be able to develop an operational understanding of using CT for solving problems.

Table 2. Activities done in different phases of the pedagogy and their mapping to dimensions of CT.

Phases	Activity	CT Concepts	CT Practices	CT Perspectives
Explore	Interaction of sensors with the environment Finding their affordances Making moving bots, right left turns Stopping and moving on obstacle	Operators, Procedures, Data structures	Problem Formulation, Questing	Programing empowerment, Perspective of expressing.
Solve	Use pre-built solutions from previous phase Determining the subproblems and primary functional modules	Sequencing, Event handling	Problem Decomposition, Algorithmic Thinking, Reusing Remixing	Connecting Questioning
Evolve	Using the learning from explore about sensors functions and bot motion Evolving the solution to a modular approach. Achieving obstacle detection while moving	Loops, Conditionals Parallelism	Iterative Incremental, Abstracting Modularising	Connecting Questioning

5. CONCLUSION

As present above, we proposed intervention for teaching computational thinking (CT) as a part of the high school curriculum. The first component of the intervention is problems that provide learners with opportunities to use CT. We have used problems with robotics. The second component of our intervention are resources to work with. We have chosen Lego Mindstorm and a few everyday materials for construction. Our application provides information about the resources textually, visually seamlessly using augmented reality. The third aspect of our intervention is that the Explore-Solve-Evolve pedagogy ensures a rich, authentic problem-solving experience for the learners. Reflections after each phase introduce the learners to the concepts, practices and perspectives of computational thinking. The mentor assumes the role of a noncontributing companion by scaffolding the learner towards exploration and play using strategies like question posing. They mentor learners with strategies to overcome challenges and reflection to ensure an explicit understanding of learns action.

The question that we pose to ourselves is that “Will the tinkering learning environment designed with an alignment of CT elements lead to the development of such an understanding of CT in its three dimensions?” Before we could aim at answering this question, this study will provide us with a deeper understanding of the alignment and the features that may or may not work as intended. With an evolved Tinkering enabled learning environment we plan to conduct more studies using techniques to evaluate the learning of CT as reported in the literature (Kong & Abelson, 2019) to be able to determine the impact of using a tinkering approach towards developing computational thinking.

6. REFERENCES

Bailey, R. (2002). Playing social chess: Children's play and social intelligence. *Early Years: An International Journal of Research and Development*, *22*(2), 163-173.

Berland, M. (2016). Making, tinkering, and computational literacy. *Makeology: Makers as learners*, *2*, 196-205.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How people learn (Vol. 11)*. Washington, DC: National academy press.

Bransford, J. D., Sherwood, R. D., Hasselbring, T. S., Kinzer, C. K., & Williams, S. M. (1990). Anchored Instruction: Why we need it and how technology can help. *Cognition, education, and multimedia: Exploring ideas in high technology*, 129-156.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (1)*, 25.

Danielak, B. A., Gupta, A., & Elby, A. (2014). Marginalized identities of sense‐makers: Reframing engineering student retention. *Journal of Engineering Education*, *103*(1), 8-44.

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

Honey, M., & Kanter, D. E. (Eds.). (2013). *Design, make, play: Growing the next generation of STEM innovators*. Routledge.

ISTE, I., & CSTA, C. (2011). Operational Definition of Computational Thinking for K–12 Education. *National Science Foundation*.

Kafai, Y., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking and learning in a digital world.* Routledge.

Kapur, M. (2008). Productive failure. *Cognition and instruction*, *26*(3), 379-424.

Kong, S. C., & Abelson, H. (Eds.). (2019). *Computational Thinking Education*. Springer.

Martinez, S. L., & Stager, G. (2013). Invent to learn: Making. *Tinkering, and Engineering in the Classroom. Torrance, Canada: Construting Modern Knowledge.*

Resnick, M., & Robinson, K. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT press.

Roque, R., Rusk, N., & Blanton, A. (2013). Youth Roles and Leadership in an Online Creative Community. *CSCL, (1),* 399-405.

Schank, R. C. (1983). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge University Press.