Using Board Games to Support Computer Science Learning

Shawn Reuter
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
reute078@morris.umn.edu

ABSTRACT

The goal of this paper is to show utilization of board games as a way of not only triggering students interest in computer science but also a way of building skills that students will continue to use throughout their jobs. The paper describes three studies that utilize board games in different ways. The first study describes elementary school students using a computational thinking board game and then introducing them to programming by having them implement the game. The second study describes the use of a board game about the Python programming language in a college classroom. The third study details the development of a board game for helping college students learn about working at a help desk. All three studies are examined from the perspective of the four-phase model of interest and self-determination theory.

Keywords

board games, Scratch, education, four-phase model of interest, self-determination theory

1. INTRODUCTION

Educators around the US are looking for ways to bring computer science (CS) and computational thinking into the K-12 classroom. However, not all schools have the funds necessary to buy the equipment needed nor do they have sufficiently experienced enough teachers to teach the curriculum. One way to approach these problems is to use board games as a way to motivate or excite students about the subject material. Because board games have already defined rules and procedures, they can be a supportive and structured way to introduce students to computational thinking [5]. Board games also offer schools a cost-effective solution, since one board game can occupy about four students rather than one computer per student in a class. This is especially important for schools that cannot afford to buy computers for each student, but can afford to buy a few board games that can serve a similar purpose as a way for students to collaborate using computational thinking [4].

Three different papers will be discussed that all use board games to learn in a way that is engaging. The first paper focuses on using a computer science board game called On the Brink as a tool for triggering interest in computer science and to introduce computational thinking [5]. The second paper focuses on designing a board game about the Python programming language which uses the board game as an engaging tool to increase students' knowledge of Python [4]. Finally the last paper dives into gamification and the creation of a Help Desk board game that trains students to work at a college help desk and troubleshoot problems as they arise [1]. Although the three papers describe studies that are quite different from each other, each study can be looked at in terms of how it aimed to increase interest or why it can be successful at engaging learners in improving their skills in computer science. The discussion at the end of this paper will be presented in terms of the four-phase model of interest as well as self-determination theory to further explore these studies through those lenses [3].

2. BACKGROUND

Using board games in classroom settings can provide an opportunity for students to collaborate and build communication skills, while also learning keywords on a subject matter in a meaningful context. A lecture-based class is similar in this respect, but often provides fewer opportunities for students to practice. Usually a teacher is lecturing on a specific lesson and asking questions as they proceed, as a way of trying to keep students engaged. Board games do a similar thing, except instead of students having interactions with primarily the teacher, students can instead have meaningful interactions with one another. In this way board games are being used as the centerpiece of discussion between students, providing an engaging way for students to practice and gain comfort with various context-specific ideas [2].

Board games have also been shown to increase motivation. In this paper, the increased motivation experienced by learners by using a board game for learning will be examined through two lenses: The *four-phase model of interest* and *self-determination theory* or SDT. The *four-phase model of interest* describes how individuals move from situational to individual interest, and *SDT* is a way of thinking about what drives individuals to move from one phase to the next [2].

2.1 Four-Phase Model of Interest

The four-phase model of interest was proposed by Hidi and Renninger in which a triggered situational interest can become a well-developed individual interest moving from an extrinsic to intrinsic interest (see Table 1 [2]). This model can be used to think about the utilization of board games in the classroom to trigger interest in a subject. The model

UMM CSci Senior Seminar Conference, November 2021 Morris, MN.

Four-Phase Model of Interest	
Category	Phase
Situational	1.Triggered 2. Maintained
Individual	3. Emerging 4. Well-Developed

Table 1: Table showing the four phases divided into two categories [2].

is divided into two categories: situational/extrinsic and individual/intrinsic motivation. The first category focuses on using a situation, in this case, a board game to trigger interest in a subject and then maintain that by continuing to play the game or talk about the subject in general. The second category focuses on the individual and if the individual has taken the steps prior and maintained an interest in a subject then that interest can develop into a Emerging Individual Interest and possibly eventually a Well-Developed Individual Interest.

Curiosity and motivation are separated into two categories, situational and individual. Situational interest being "focused attention and the affective reaction that is triggered in the moment by environmental stimuli, which may or may not last over time" [2] and Individual interest being "a person's relatively enduring predisposition to reengage particular content over time as well as to the immediate psychological state when this predisposition has been activated"[2]. Situational interest grabs the student's attention and encourages diving into the content again, with the ideal goal of shifting from situational to individual interest.

For example, consider a group of people playing a board game that is initially new to them. Before playing, the players may have no interest or possibly a slight interest. If they saw some game play or mechanics of the board game, players may have a Triggered Situational Interest in this board game, meaning that there was a short change in cognitive processing to spark interest. Once the group of people playing the board game have played it a few times, some may transition to a Maintained Situational Interest meaning that players have maintained this triggered interest for a prolonged period. After a while some players strive to master the board game being played and so the Maintained Situational Interest becomes an Emerging Individual Interest. As players transition from the second to the third phase, they have an interest in the subject that changes from an extrinsic motivation to intrinsic motivation. If players were to have a Well Developed Interest for the board game, they may have continued engagement over a long period of time. [2]. We can apply this model to the use of board games as a tool for instruction in the classroom.

2.2 Self Determination Theory

Self-determination theory or SDT is an explanation of what drives people. Everyone has three psychological needs those being autonomy, competence, and relatedness. People are drawn towards activities that fulfill these needs. If these needs are fulfilled a person is said to have a better well-being. Board games can easily fulfill all three of these needs. Autonomy is the need to have control over one's life or action. Competence is the need to be accomplished at something. Finally relatedness is the need for people to interact and connect with one another. Board games give players agency through their choice of actions, providing a sense of autonomy. Players develop competence through their ability to make mistakes and learn from them. Games also create an environment in which players talk and interact with one another, creating relatedness.

3. INTRODUCING CODING

In the first study described in this paper, an eight week unit was designed that would use an unplugged-to-plugged sequence that would include students learning the CS board game, On the Brink, and then trying to implement it in a Scratch shell. Scratch is a free programming language that was specifically designed to be used by kids as a way of creating games, stories and animations. The *Scratch shell* is just another way of talking about the interface that users use to create their game, story or animation.

Students learn the board game for the first two weeks ("unplugged") and implement the board game in Scratch for the following three weeks ("plugged"). For the next two weeks new levels were designed by students. During the last week, students share their implementation with teachers and classmates. In total the students had 6 preparatory lessons that each lasted for 10-20 minutes [5].

Three fifth-grade teachers implemented their units at the same time on different days of the week. Interviews with the teachers showed that they all had differing levels of experience with the unit. Teachers ranged from Shawn (who was familiar with block-based coding) to Shelly (who had not taught coding, but was confident in her capability to learn and teach), and finally Mandy (who was not confident in programming and technology in general). Data was collected in the form of video recordings: two cameras, one recording demonstration of the teacher and the other recording students doing hands-on activities.

3.1 On the Brink

Many computer science board games have been published in the table-top industry, often marketed with language saying that it will help teach kids (CS) concepts. These games may have some aspects of computational thinking but many times in explanation these aspects are not made explicit to the connection it has to the programming of code. The researchers use the board game On the Brink to introduce loops and conditions as an introduction to computational thinking. On the Brink was designed by Mark Engleberg and published by Thinkfun.

The story of On the Brink is about a robot that needs instructions to navigate a series of rooms, each with colored tiles arranged on a six-by-six grid. Each color of the tile is associated with a sequence of actions that the player specifies by playing cards on a control panel. The movement options include moving forward, rotating 90 degrees to the right or left, or doing nothing. Combinations or repetitions of these steps resemble functions or loops that can be programmed in Scratch. The researchers used a special Scratch shell with some aspects of the game already programmed so that the learners would have an easier time seeing how playing cards for control sequences in the board game and programming in Scratch's block coding environment are similar. Although the researchers did not publish results about how this setup impacted the learning of programming in this paper, they do describe the use of the special shell in Scratch as a way

	Pre		Post			Post-Pre
Teachers	**M**	**Med**	**M**	**Med**	**N**	**Z**
Shawn	4.45	5.00	3.45	3.44	29	-3.49***
Mandy	4.30	4.50	4.26	5.00	30	-0.93
Shelly	3.84	3.88	4.16	4.50	28	2.25*

Table 2: A pre and post-hoc test describing the intrinsic interest in computer science of the students who participated in the unit.

to make the board game and the Scratch environment more explicitly similar to help learners transfer their skills from the board game to a computer programming environment. For example, the game used color along with the control sequences to indicate which actions to take, and in Scratch the color of the background the character is on can be used to trigger certain conditional execution to take place.

3.2 Data Collection and Analysis

The primary goal of the eight-week unit is to increase intrinsic interest in Computer Science by helping students realize that information gained through play of a board game can be correlated to coding in Scratch. A pre and post survey was taken by all students. This survey contained 32 Likert scale items from one to seven that measured intrinsic and future interest in computer science (see Figure 2). From the initial level of intrinsic interest, the three classes did not differ significantly. However, in Shawn's class, the most experienced teacher with block coding, the students showed a significant decrease in intrinsic interest, shown in the figure by three stars (***) to denote significance. Students in Mandy's class stayed the same so there was no stars marking significance. but students in Shelly's class showed significantly higher levels of intrinsic interest than the other two classes as showed in the figure by the one star (*). One reason for this was students in her class having a lower initial mean of intrinsic interest in computer science compared to the other classes of students. Those who took the board game home also reported higher levels of intrinsic interest compared to those who did not. This means that taking the board game home was positively and significantly correlated with higher levels of support for increasing motivation to engage with computing.

3.3 Qualitative Analysis

A discourse analysis was undertaken to compare teacher utterances. The researchers' interest in what statements were made by the teachers as they taught a lesson and how their organization and information in their statements differed. Through analysis of teacher variation of teaching styles were seen and this contributed to the differences in student post-survey results.

3.4 Analysis of Teachers

Teachers' utterances were analyzed on direct instruction, connections, and narrating lesson trajectory. Shawn's class received the most direct instruction with 40 instances reported compared to the 23 and 24 Mandy and Shelly had. Shawn therefore, made nearly double the number of direct instruction utterances which often involved lengthy monologues, where in comparison Mandy and Shelly were brief.

These three teachers made an attempt to connect Scratch to an example relevant and familiar to the students. Mandy and Shelly both used Minecraft and Code.org in their explanations of Scratch. In contrast Shawn used Scratch to build a calculator program and explained how different buttons have their own respective commands and operations connected to them. Shelly's explanation was more playful and Shawn's was more advanced, complicated and abstract. Because of Shawn's greater knowledge of programming, he chose a complex example that, in the end was less engaging to the students [5].

In terms of narrating lesson trajectory, Shelly began the lesson by telling her students the activities that she would cover. This is important for any teacher to do when lecturing so that your audience knows where they are in the content and where exactly are they going with the lesson and even possibly how does that fit into the overarching lessons. Shawn narrated lesson trajectory 10 times with 40 instances of direct instruction compared to Shelly's 12 times with 24 instances of direct instruction. This means that Shelly explained activities and how each one connected with one another more clearly and as seen from the analysis of students' intrinsic interest in computer science the narrating the lesson had some contribution to higher intrinsic interest in the students There is no mention in the paper of how many times Mandy narrated lesson trajectory[5].

4. PYTHON THE BOARD GAME

The primary goal of this study was to explore the use of board games in a college classroom to improve students' knowledge of the Python programming language. Two research questions were developed to meet this primary goal: "Will students embrace board games in the computer science classroom to improve their knowledge of the Python programming language?" and "Will lecturers use board games in the computer science classroom as a tool to improve student's knowledge of the Python programming language?" [4].

4.1 Methodology of Development

The board game was developed in two action research cycles. Action research is a process that is carried out in five phases. Before the five phases, research is specified and defined. Once the research environment is determined, the five phases (diagnosing, action planning, action taking, evaluating and specifying learning) are executed repeatedly.

The board game that they developed is a game for two to ten players, but is best played with four to six players. The board consists of a circle divided into nine sections in the form of a pie chart. A spinner is mounted on the board and an app on a tablet accompanies the circle with a database that has 400 Python related questions (see Figure 1). These questions are color coded from easy (orange) to challenging (red).

4.2 Explanation of Play

The first player spins the spinner and reads the instruction of the segment where the spinner landed to the other players. Players answer the question using a pen and notepad independently except the player who spun the spinner. Once the time on the tablet has run out, player one gathers all the answers and records them using the 'Answer' button on the tablet. Players are given "bit dollars" for answering a question correctly. If a player answers incorrectly the group discusses the question in addition to correct answers

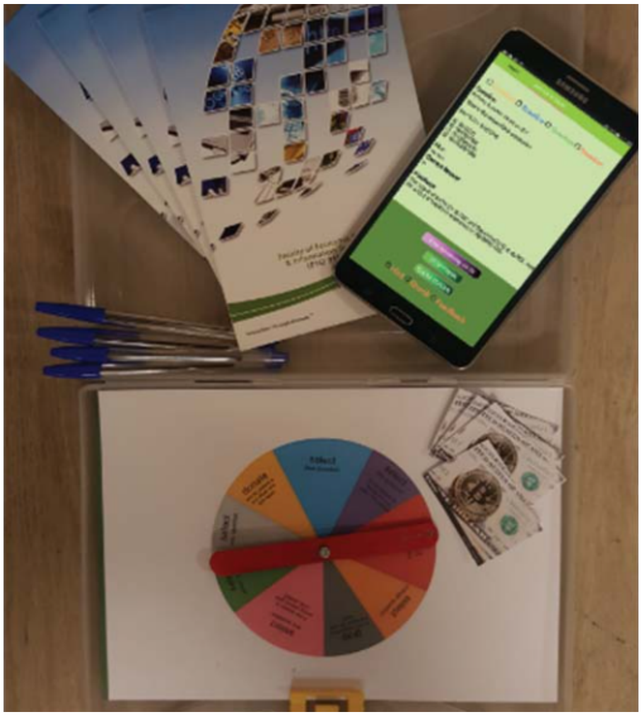

Figure 1: Pen and paper top left, tablet top right and spinner bottom of image [4]

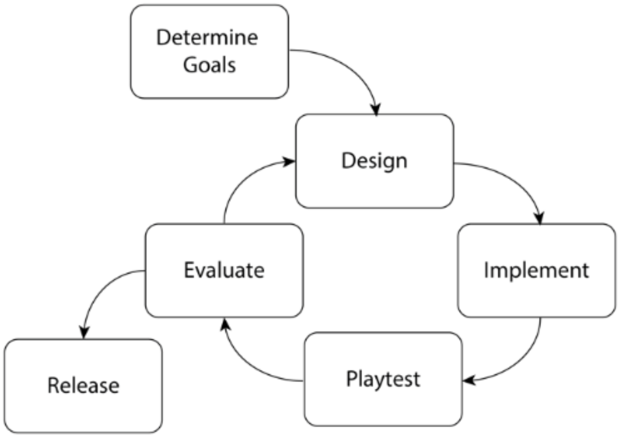

Figure 2: Iterative Model used in [1]

by using the discussion button on the tablet. Once everyone understands the correct answer, play proceeds with the next player. Players determine at what time to stop playing and the player with the most "bit dollars" wins [4].

4.3 Findings

In order to determine if the board game met the research questions and goal, a discussion was held with the participants who played the board game. Among the participants there were eight lecturers and eight students who were currently taking a course on Python. These sixteen participants were divided into four groups, two groups of four lecturers each and two groups of four students each. Each group played the game and then participated in a semi-structured interview. Groups were asked to discuss how the board game addressed the following issues: enjoyment, collaboration, communication, socialization, involvement, participation and subject content.

In the enjoyment category groups mentioned the fun and enjoyment they had playing the game and enthusiastically engaged themselves without any hesitation. Under the communication category players agreed that the option to discuss answers to questions could improve overall knowledge retention. Under socialization, players agreed that by playing the board game group engagement and spontaneous engagement will increase.

In summary students and lecturers agreed that the board game may increase their knowledge of the Python programming language in a fun and inexpensive way, thus answering both research questions that the board game developed was embraced by students and lecturers to use as a tool of improving their knowledge of the Python programming language.There was no particular test that was made in order to determine objectively if knowledge of the Python programming language was gained by the players.

5. HELP DESK GAME DESIGN

The final paper being discussed contains the process of designing a Help Desk board game in order to train students in a more engaging way. As a game designer it can be daunting in thinking of where to start: the iterative game design model can be a solution to this problem. The iterative model is one of the best for designing board games because it is fast and effective. The steps of the cycle in order are design, implement, playtest, and evaluate (see Figure 2). The cycle then either repeats or goes to release. In theory the more times a board game goes through this cycle the more refined the design of that board game should be.

5.0.1 Determine Goals

The first step in the model is possibly the most important of the steps even though it is not in the loop of model itself. That step is to determine the goals of the board game. Especially in a educational game there may be specific learning goals that the designer wants the players to walk away with. If game designers ever get lost in development of the mechanics or aesthetics of the board game it is always important to center around the original goals that were thought up at the beginning.

5.0.2 Design

After having a clear understanding of what ideas and concepts are being designed for, it is time to start thinking about what mechanics or aesthetics will align with the goals of the board game. Designers try to align their goals with mechanics that attach value to the actions that players are performing. All mechanics have inherent value that is attached to them; if designers ignore this value they can miss possible improvements that they could make to their game. In this case using a game mechanic that introduces randomness like rolling dice would be a poor choice for choosing how to interact, but may be suitable for choosing which client scenario to explore next since help desk employees don't get to choose what issues come their way.

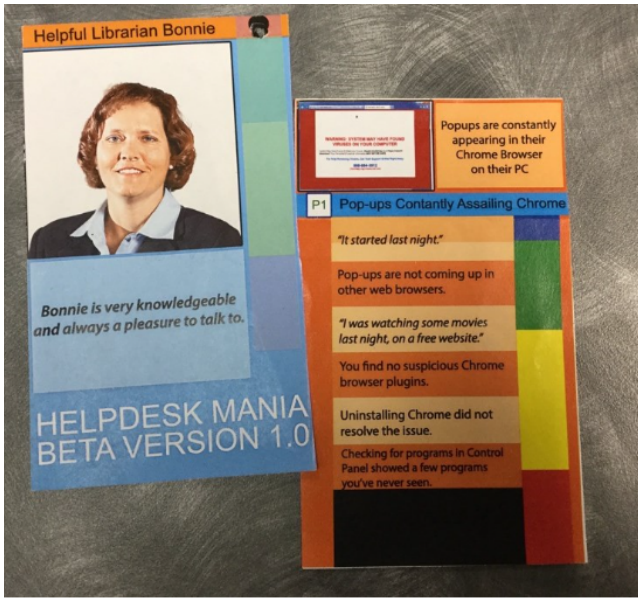
Helpful Librarian Bonnie
Bonnie is very knowledgeable and always a pleasure to talk to.
HELPDESK MANIA
BETA VERSION 1.0
WARNING: SYSTEM MAY HAVE FOUND VIRUSES ON YOUR COMPUTER
Popups are constantly appearing in their Chrome Browser on their PC
P1
Pop-ups Contantly Assailing Chrome
"It started last night."
Pop-ups are not coming up in other web browsers.
"I was watching some movies last night, on a free website."
You find no suspicious Chrome browser plugins.
Uninstalling Chrome did not resolve the issue.
Checking for programs in Control Panel showed a few programs you've never seen.

Figure 3: Client card on the top right, problem card on right [1]

5.0.3 Implement

After deciding what mechanics and aesthetics would be best to use for their board game it is time to construct designs and the physical objects that are used to play the game. A s a designer goes through the first few iterations it is important to keep things simple and not spend a insurmountable amount of time on the look and feel of each individual item, because designers do not even know if these mechanics will work for their game design yet.

5.0.4 Playtest

Playtesting is the time when designers have people sit down and play their game and this is the only way that game designers can actually tell if their design is any good, or not and whether the design meets the goals that they set.

5.0.5 Evaluate

Finally it comes to the evaluation step and when designers sit down with their playtesting notes and consider how to improve their game. Usually the easiest way is to start looking at issues with the mechanics and try to balance them from there. The last step is to determine if the game is ready for release or should go through more iterations for further game refinement [1].

5.1 Help Desk Mania - Goal Setting

In the initial goal setting phase it was determined that it would be advantageous to build the following skills in the players: troubleshooting problems in person and remotely, dealing with clients of different personality types, understanding that individual interactions with clients impact our larger IT organization, and finally problem solving, communication and teamwork skills among the players [1].

5.2 First Iteration

A decision was made to try and simulate a realistic help desk encounter so that the board game could be used as a tool for both new and returning students as way of building or sharpening their skills. Plans were drawn out on a whiteboard with clients and players being expressed as tokens and a desk between them. However, many of these tokens were unnecessary in order to express a customer service interaction. This resulted in the design moving from a board game to a card game.

The implementation of this card game utilized three components: client cards, problem cards and a satisfaction score. Cards were prototyped using index cards. The problem card included any information necessary in order to troubleshoot the problem. The client card included biography information along with an image of the client and a satisfaction gauge on the right side of the card. As the development team positioned the cards around one another an idea sprang up to use the client card to hide information on the problem card and every round would slide down the client card to reveal a clue about the problem. Eventually the card would slide off the problem card if players did not solve the question fast enough. Players gain reputation by how fast they solve a problem. After a few cards were developed it was on to the playtesting phase.

The development team played two rounds of the game with difficulty. Overall the system was playable but, did not feel great to play. They tried to find problems with the design to no avail, so they found some students to playtest the game in order to possibly find design problems. From this playtesting, they discovered that the restrictive behavior of the clues on the problem cards created problems. For example, a player would guess something but the card did not reflect their guess because it did not have that information on the card and as a result the player would only get partial information. Overall the game was clumsy and lacked a realistic back-and-forth experience that a person would get at an actual help desk. This resulted in the player walking away confused.

The first playtest feedback showed that the structure was confusing. Information being displayed on each card was too intricate for one card to contain. This defined the point in which the design shifted from a complete game that only needed minor faculty involvement to a game that could be utilized for help-desk training but required a facilitator. If the cards could contain only the necessary information that people would receive when they walk up, then a facilitator of the game could give each interaction meaning. Making this design change means that extra time will be needed for training facilitators, but overall it will benefit the game and students who play the game [1].

5.3 Second Iteration

The next iteration began with the creation of a manual for facilitators to utilize while guiding players as they encounter each interaction. To accomplish this task one senior student was tasked with developing typical problems that the help desk comes across along with a reasonable explanation of why those problems would occur. One interaction that can be seen in Figure 3 is the problem of popups appearing in a Google Chrome browser. As a result, a problem and solution set was created that problem cards could then be based upon. These problem cards could be given to facilitators with guidance on how they should respond to solutions that students are trying.

A set of formal options were developed for players to use on their turn to try and solve a problem. These options are as follows: "asking the client a question, performing a troubleshooting task on the client's device, looking up information online until their next turn, asking another player for help, escalating the problem to a professional staff member, and solving the issue by explaining the cause and solution for the issue [1]." These game play actions were all meant to mimic real-life actions that students could enact in their job. These actions also strengthen team-based game play and skills.

In the implementation step the development team looked to redesign the client and problem cards to only have the necessary information. They decided to design the client cards to how their SUNY ID cards look and design problem cards in to a similar structure as their campus ID cards. The satisfaction gauge on the client cards now was put horizontally on the bottom of the card and was broken up into the three different sections green, yellow and red each with different acceptable rates of problem resolution that range from green the fastest to red the slowest problem resolution rate. With the problem card being much smaller in size it would now slide along the bottom of the client card to show how many chances the player had remaining to guess. The development team were ready to move on to the playtest step as they were satisfied that this new mechanic would be clear and train skills that they wanted to cultivate in the students.

In the playtesting phase a student employee was given an overview of the rules and started the game by drawing a client and problem card. Troubleshooting the problem began right away but the student hit a roadblock in that the game allowed only a two-action limit each round. Feedback was given to developers that this was awkward and so play continued without the limit. Finally the student employee ran out of ideas to troubleshoot the problem. This gave the designers more feedback that they could apply in the evaluation step.

In evaluation of the playtesting step and after taking in feedback from students, the development team changed the game to use a cooperative round robin approach instead of individual turn-based gameplay. Using this new system players are able to try any number of troubleshooting tasks and ask as many questions as necessary.

6. DISCUSSION

With the results given from the three papers, not much can be said as only one study had results of any significance and those results were mixed at best. It is still early in the development of using board games in a computer science classroom so there are not many well developed studies in this field as of yet. However many things of importance can still be taken from these studies, those being: board games being a structured way of introducing defined rules or procedures while being cost-effective in the process.

Future work into this field would be to specifically develop board games that could fit into the curriculum of a computer science class whether they be at high-school or college level. such as Algorithms and Computability. While most classes may still be lecture based, the addition of a well designed and appropriate board game for the class may prove to be an effective tool for engaging students.

The four-phase model of interest can be applied on how each paper utilized a board game in order to trigger interest or be engaging and educate players on a subject material. The board game from the first paper On the Brink specifically was focused on triggering interest in computer science and computational thinking. This board game can help with maintaining interest especially, how great the color scheme of the board game and how it is similar to Scratch.The parallels that the researchers made explicit between the board game and the Scratch environment were an attempt to make the transition between the game and programming simpler. The latter two papers designed board games in which the focus was engagement with the subject material, but not necessarily triggering an interest. However, there are aspects of SDT that may help explain how learners might advance through the phases of interest.

From the perspective of SDT, players of the board games will be driven to engage with the games in ways that increase their sense of autonomy, competence, and relatedness. In the first paper, with the board game On the Brink, players are likely drawn to the game by their ability to control their own actions by choosing cards (autonomy) and learning from their mistakes in a playful way (which increases their competence). Since they are playing the game with others, they are all experiencing the same scenarios and building a shared experience, which likely helps increase their feelings of relatedness.

The findings from the second paper show similar utilization of the three psychological needs. Communication being the first among the issues discussed in which groups suggested that reiteration of question and answer would be helpful to retention of knowledge, but also as a way of building competence in the players. Socialization is another issue discussed in which group engagement was found to increase and through this relatedness can grow between the players.

In analysis of third paper Help Desk Board game, all three concepts can once again be applied. Players utilize their own autonomy by completing actions and gaining competence through troubleshooting problems in the game. Players also fulfill the final need relatedness by being solving problems together and getting to know one another.

7. CONCLUSIONS

Many different concepts and models have been seen throughout the papers. The first paper used the board game On the Brink to introduce computational thinking to students in an engaging way [5]. An analysis of teachers utterances concluded that it is important not only to use direct instruction and lesson trajectory but also to use examples that will connect and motivate students specifically

The second paper discussed the design of a Python programming language board game specifically to build students' competence of the Python programming language. The author concluded that "the use of board games should be seriously considered in the Computer Science class" [1]. However this conclusion was not one that is backed up by evidence of any analysis but merely based upon the experience that the board game brought to students and teachers that played it.

The third paper introduced crucial concepts such as the iterative model and through the use of this model, Babcock developed a board game to train students to work at a help desk in an innovative and exciting way [1].

In summary, using board games in and outside of the classroom are a way to engage students and adults while learning.

Simply put, board games are a medium of art in which any information can be expressed. Many board games are not inherently educational, however some board games are created to be educational while still being engaging and fun to the players. Then teachers and professors can have opportunities to use board games as a way of engaging their students in the classroom.

Acknowledgments

I thank my advisor K.K. Lamberty for helping me through this process and thank my mom for assisting me in revision of my paper. Finally I thank Morris alumni Charlot Shaw for reviewing my paper and providing helpful feedback.

8. REFERENCES

[1] C. Babcock. *Developing a Help Desk Board Game*. PhD thesis, New York, NY, USA, 2017.

[2] E. A. Dewey. *Triggering Student Interest in Classroom Subjects through the Use of Board Games*. 2020.

[3] S. Hidi and K. A. Renninger. *The Four-Phase Model of Interest Development*. PhD thesis, 2006.

[4] D. B. Jordaan. *Board Games in the Computer Science Class to Improve Students' Knowledge of the Python Programming Language*. PhD thesis, 2018.

[5] V. R. Lee, F. Poole, J. Clarke-Midura, M. Recker, and M. Rasmussen. *Introducing coding through tabletop board games and their digital instantiations across elementary classrooms and school libraries*. PhD thesis, 2020.