4-1-2011

Collaborative strategic board games as a site for distributed computational thinking

Matthew Berland
University of Texas at San Antonio

Victor R. Lee
Utah State University

Recommended Citation

INTERNATIONAL JOURNAL OF GAME-BASED LEARNING

Table of Contents

Collaborative Strategic Board Games as a Site for Distributed Computational Thinking

Matthew Berland, University of Texas at San Antonio, USA

Victor R. Lee, Utah State University, USA

ABSTRACT

This paper examines the idea that contemporary strategic board games represent an informal, interactional context in which complex computational thinking takes place. When games are collaborative – that is, a game requires that players work in joint pursuit of a shared goal – the computational thinking is easily observed as distributed across several participants. This raises the possibility that a focus on such board games are profitable for those who wish to understand computational thinking and learning in situ. This paper introduces a coding scheme, applies it to the recorded discourse of three groups of game players, and provides qualitative examples of computational thinking that are observed and documented in Pandemic. The primary contributions of this work are the description of and evidence that complex computational thinking can develop spontaneously during board game play.

Keywords: Board Games, Collaboration, Computational Thinking, Computer Science Education, Designer Games, Group Processes, Multiplayer Games, Tabletop Games, Teams

INTRODUCTION

A great deal of interest has been expressed as of late in the complex reasoning that takes place during gameplay. This interest has developed for a multitude of reasons, including the inherent motivational aspects of gameplay, the awareness that there are millions of people across the country who are actively participating in games and gaming communities, and the extant design features of many modern-day games that foster learning (Gee, 2007; Nasir, 2005;

Steinkuehler, 2006). Often, these benefits are associated with video games and other highly interactive computational media. It is largely thought that the ability to foster a sense of immersion is a genuine strength of video games that distinguishes them from many other learning contexts (Shelton & Wiley, 2007).

Still, there are reasons to suspect that some of the generative potential of games is not restricted to those that take place on a computer platform. At their most base level, games are systems of rules in which players operate on representations. In a computer game, those rules are generally executed and strictly enforced by the game itself. Board games and other table-

top games, on the other hand, have no such inherent game rule management; it becomes incumbent upon the players themselves to know and execute the rules of the game. The players are doing the computation that would normally be the purview of the computer or console in a video game.

We consider a new genre of board games, of which *Pandemic* (Leacock, 2007) is an instance, to be an especially interesting context. These board games, which we often refer to as 'strategic' board games, involve complex coordinated play and highly motivating contexts. This class of games has also been referred to as German-style games, Eurogames, and designer games.[1]

In this paper, we show how this family of strategic board games can prompt novice game players to engage in relatively complex computational thinking. Through the gameplay we observed, players came to understand, 'debugged', and created global rules to guide their play of the game and their development of strategies. They did so in a socially distributed way; the players created rules together, they helped each other understand those rules, and they collaboratively built complex logics. To investigate this computational thinking, we created and deployed a coding framework for distributed computational thinking, which we present with examples.

In this study we recruited 3 groups of 3-4 college-age novice players. Each group played the selected board game Pandemic, (Leacock, 2007), at least once, and we video-recorded their gaming sessions. For this paper, we focus strictly on the first gaming session for these groups. We present three sources of evidence for the students' computational thinking: 1) quantitative analysis of the makeup of the students' computational thinking; 2) quantitative analysis of code counts for instances of 'global' and 'local' computational thinking; and 3) some descriptive examples of computational thinking.

Our work complements earlier findings with pen-and-paper role-playing games (Fine, 1983), in which players were found to do sig-

nificant mathematics in order to play strategy games. Our data suggest that this claim can be made stronger – players are doing more than simple math, they are doing computation.

THEORETICAL ORIENTATION: COMPUTATIONAL THINKING

This particular paper is guided by a mutually shared interest by the authors to understand the nature and development of computational thinking. Given the increasing role that computation plays in teaching and learning (Borgman et al., 2008), understanding how people both interact with computation and learn to think through the language of computation has become an area of interest for education and media researchers (National Research Council, 2010). Computational thinking has been discussed in detail beneath the larger umbrella of computational literacy (diSessa, 2000), the broad suite of practices associated with using computational media in our everyday and professional lives. Of specific interest to us here and most relevant to the study of computational thinking is what diSessa describes as the 'cognitive pillar' of literacy – how to use computation to think through hard problems. Papert (1980) calls this type of thinking 'procedural thinking' and his work focuses on students' problem solving with programmatic representations and symbol systems. According to the National Research Council (NRC) (2009), computational thinking is roughly defined as using the methods, language, and systems of computer science to understand a wide variety of topics. This can range from creating computational models of scientific phenomena to creating algorithms to plan one's day more efficiently. Board games are a relatively closed set of representational resources that are organized in a coherent, rule-like manner; as such, they are amenable to this kind of inspection.

We do wish to note that we are not the first to attempt to understand computational thinking in material and rule-based contexts. This endeavor has been undertaken in a number of

contexts, particularly with computer science education (Koedinger, 2001), and as far back at Newell and Simon's (1972) study of cognition in chess. Our contribution through this work is description of and evidence that complex computational thinking can happen spontaneously using non-traditional, non-computational media like strategic board games. In the discussion section, we tie the computational thinking more closely with the design of the game; we believe that further work will show that the reasoning that is developed through board game play can be leveraged for instruction through further design work and future research.

In our analysis, we will focus on five core aspects of procedural computational thinking that have been identified by Wing (2006). These include: conditional logic, distributed processing, debugging, simulation, and algorithm building. This subset does not encompass the full range of cognitive capabilities or processes that are involved in thinking computationally. However, this selection represents a modest start on our endeavor and also shows some of the clearest overlap between board game thinking and computational thinking.

Pandemic: A Collaborative Strategic Board Game

Board game play is a recreational activity common among groups of friends and family members and can involve very different sets of rules and playing styles. For example, one can classify the playing style of a board game as being competitive, cooperative, or collaborative (Zagal, Rick, & Hsi, 2006). Interest in board games appears to have experienced a recent resurgence (Kleinfeld, 2009) – for example, the web site 'boardgamegeek.com' has seen membership consistently increase over a period of 7 years from 929 accounts in 2002 to 240,623 accounts in 2009 (from S. Alden, co-creator of boardgamegeek.com, Personal Communication, March 25, 2009). This renewed interest in board games has even led some to advocate them as resources to include in public library collections for educational purposes because

they can tie into content learning or information literacy standards (Nicholson, 2008).

The increased interest in board games comes in part from a growth of German-influenced strategic board games in which resource management, short play times, elaborate themes, decreased reliance on chance, sustained participation of all players (e.g., no one is eliminated), and incentives to interact directly with peers are all designed into the game. The game we have studied in this work, *Pandemic* (Leacock, 2007) is one such board game. It is a collaborative game, similar to those studied by Zagal et al. (2006), in which 2-4 players share a common goal and either collectively win or lose the game. The cover story for the game is that there are four highly infectious diseases (designated by the colors red, yellow, black, and blue) that simultaneously appear and are spreading across the world. The players must combat the spread of the disease by moving player tokens to various cities and treating the infected populations while also gathering and exchanging 'information' (i.e., cards) that will lead to cures and/or vaccines for all the diseases.

During each turn of play, a player makes decisions about where in the world to travel (e.g., which lines to follow on Figure 1), whether or not to treat diseases (signified by wooden blocks placed on the cities) or if she should focus on other strategic decisions (e.g., knowledge sharing between players, passing her turn to enable a future action). Each player is limited in the number of actions they can take on their turn. Obstacles exist throughout the game in which disease spread begins to accelerate and penalties exist for delaying disease treatment, which may ultimately cause the players to lose the game. For example, one additional game obstacle is the presence of "Epidemic" cards, in which a drawn card will dictate that a disease should appear at full strength in a previously uninfected city. The precise rules are communicated in an 8-page guidebook that specifies rules, justifications for some of those rules, and even a sample turn for the game. Many of the same rules and procedures are also written on the game board

Figure 1. Approximation of the game board from Pandemic (Leacock, 2007). Each circle represents a major world city

itself, on rule cards that each player has for reference, and on individual playable cards that enable a specific sequence of events.

Players must coordinate moves and carefully utilize their resources to both prevent disease outbreaks (undesired spreading of disease to neighboring cities) and to discover cures. Cures are obtained by matching five cards of the same color at one of several locations on the board. The game relies on two decks of cards, one "player deck" which provides resources for a cure and another "disease deck" that results in new infections to different cities. Players are encouraged to speak to one another. Each turn allows a player to take up to four actions, which may involve activities such as movement, disease removal, card exchanges. The planning and decision-making with respect to which cards to use and where to move player tokens requires players to follow several logic chains, and the coordination among players to achieve a unified goal encourages parallel processing between players.

Coding for Computational Thinking

Any observation of *Pandemic* gameplay would reveal that there is a great deal of complex reasoning and inference that is taking place among the players. However, any effort to tie player behaviors to computational thinking is complicated by the lack of a concrete definition of computational thinking. Computational thinking has been the focus of several recent papers, studies, and reports (Haberman & Yehezkel, 2008; Sieg, 2007). Many of papers in this field have attempted to independently define it, and, as such, the construct is defined variously. Little of this recent work has satisfactorily operationalized computational thinking nor provided clear guidance on what we may identify in real interactions as computational thinking. As such, we created a working definition of computational thinking, drawing mostly from Wing (2006), Papert's (1980) concept of procedural thinking and the recent National Research Council's (2010) *Report from a workshop on the nature and scope of computational thinking*.

Our approach is empirically based, in that the data that we had motivated the categories that we created. We distilled computational thinking into a few categories and two stages that we thought might be relevant to our research topic. We did not expect to see the full breadth and width of what others may consider computational thinking; trying to operationalize all of computational thinking is outside the scope of this paper. The five categories we considered for this work included: conditional logic, algorithm building, debugging, simulation, and

distributed computation. Our two stages are local logic and global logic.

These categories are imperfect (at best), and perhaps their most serious problem is that they are overlapping and mutually dependent. Nonetheless, the categories present a scheme with which our discourse analysis coders were able to systematically analyze talk and behaviors. In the following sections and in Table 1 below, the categories are presented, with a description, a rationale from the literature, and a concrete example from the transcript.

Data Collection and Methods

As mentioned above, the data that we analyzed for this paper describe three complete runs of a game of *Pandemic* (Leacock, 2007) played by three groups of three-to-four different first year undergraduate students (ages 17-19) at a major university located in the Mountain West. We refer to these groups as "Alpha" "Delta" and "Lambda". Alpha was made up of 3-4 males (1 player could not make the game sessions consistently), Delta was made up of 4 females, and Lambda was made up of 4 males. The students had never played Pandemic before, nor had they played any related game. The students were encouraged beforehand to talk freely during the game; this is explicitly encouraged in the accompanying instruction booklet (Leacock, 2007). Each session lasted between 60 and 90 minutes.

We collected video recordings of all interactions, and we generated transcripts from the recordings. We segmented video excerpts into gameplay turns and then sub-divided them by utterance. Each of these excerpts was coded with respect to the rules that were being interpreted or the strategies that were ultimately developed. Iterative reviews of the video for the excerpts yielded narrative accounts for how the actions of different participants, their state of knowledge at the time, and the state of the game materials resulted in the ultimate strategies or understanding of game rules. These accounts were constructed independently by the two authors, then refined through competitive argumentation (Schoenfeld, Smith, & Arcavi, 1993) and extended discussion.

Using the interpretive analysis of specific excerpts, a set of codes related to computational thinking was developed (Table 1). These codes were then refined from multiple coding passes with data subsets. Our final set of codes included five categories of computational thinking: conditional logic, algorithm building, debugging, simulation, and distributed computation. Conditional logic involves using an "if-then-else" logic structure, and often involves players describing the chain of events that might happen, based on the games rules, should a particular action be taken. Algorithm building involves the construction of a plan of action, with the long-term goal being that the algorithm be robust enough that it can be reused in the future for unknown or unpredictable events. Debugging involves diagnosing errors in logic or behavior. It often involved clarifying rules or strategies during game play. Simulation involves the enactment of algorithms or plans in order to test the likely outcome. For example, a player who moves their token to various spots and declares the actions they would take without releasing the token (and thus committing herself to a set of decisions) would be engaged in simulation. Distributed computation is an inherently social aspect of computational thinking, in which different pieces of information or logic are contributed by different players over just a few seconds during the process of debugging, simulation, or algorithm building.

As stated above, this list is not exhaustive nor are the codes mutually exclusive. Because of the inherent interdependencies of different computational actions, we did not expect that we would be able to define explicit boundaries. Our approach is to identify each category by its exemplars. We validated those decisions by comparing our set of categories against computational activities already described in the literature related to computational thinking (Abelson, Sussman, & Sussman, 1984; National Research Council, 2010; Papert, 1980; Wilensky & Reisman, 2006; Wing, 2006).

Table 1. Summary of the code categories, accompanied by rationale from the literature and examples from the data corpus

Category	Description	Rationale	Example
Conditional logic	Conditional logic is the use of an "if-then-else" construct. It requires a student to think globally about the local consequences of the truth-value of a given statement.	Wing (2006), the National Research Council (2009), and a common introductory computer science textbook (Abelson, Sussman, & Sussman, 1996) all present the conditional logic construct as the simplest construct underlying all computation. Most machine language is evaluated conditional logic and simple variable use. We are categorizing variable use as "global logic." Effectively, a student using conditional logic with variables is doing computational thought.	"...if Milan gets one more, that means Istanbul gets one, and if Istanbul had 3, that means Istanbul would start infecting ones next to it, too, and it would be like a chain reaction."
Algorithm building	An algorithm is a data "recipe" or set of instructions. Fundamentally, computer programs consist of algorithms and data. Algorithms often contain sets of related conditional logic. In its simple form, it is the planning of actions for events that are taking place; in its complex form, it is planning for unknown events.	Though there is significant debate in our source literature on the relationship between programming and computational thinking, Papert's (1980) "procedural thinking" construct is about teaching students to abstract their concepts into algorithms. The National Research Council (2009) makes several references to procedural thinking as a core concept of computational thinking.	"...I could move ... here, that's 1. And then take out 1 there, then go to Tokyo, so 3. Wait, 1, 2 ... I could move here; and then just not do anything there; and then move to Tokyo; and then fly from Tokyo to where A is; and then give him this card so the beginning of his next turn ... he can play."
Debugging	Debugging is the act of determining problems in order to fix rules that are malfunctioning.	Papert (1980) describes debugging as a core "powerful idea" of procedural thinking. Wing (2006), the National Research Council (2009), and Abelson, Sussman, and Sussman (1996) describe debugging as central to both programming and computational thinking.	Alex[2]: "...but I think that might be only during epidemics." Brad: "Do you add them back to the top during epidemics? Cause I was reading here, whenever a player draws..." Alex: "Okay, so then I'll just leave it there."
Simulation	Simulation is modeling or testing of algorithms or logic. Simulation is used in debugging in order to determine problems, and it uses algorithm building to test a model. We are defining simulation as the enactment of algorithms or plans.	Simulation or model building underlies computation in the mathematical sense. Wilensky and Reisman (2006) define computational thinking as various aspects of model building or simulation.	"...Essen, I have [the Essen card], so I could fly, I could take care of that during my turn. [I could address] that London outbreak after I take care of that. 'Cause that would take one, then I can fly to Essen, then move there. And then I can take the rest of that."
Distributed computation	Distributed computation applies to rule based actions. For instance, if 3 people act together through a rule-based plan, this is distributed computation as considerations, contingencies, and strategy formation involve multiple parties with different knowledge resources.	The National Research Council (2009) describes distributed computational thinking as one social aspect that distinguishes computational thinking from computer science.	Patrick: "Okay, for my turn first off I'm going to cure Lima... And then I'm going to move LJ. ... I'll move you here because that way you're only two away." L.J.: "You can move me to one of your cards, and then I'll teleport there." Michael: "But you can only trade the card of the one you're standing in." L.J.: "Oh, that's right." Michael: "Just because you have one, you can't turn all of them in..."

Figure 2. Alpha-game code counts and frequencies
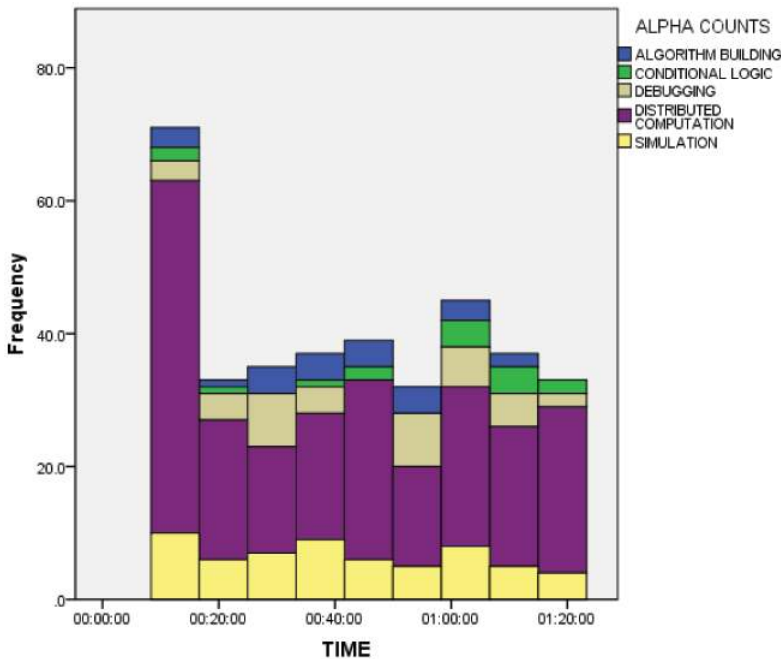
With this set of categories in place, codes were then assigned to the individual utterances in each of the three transcripts. Data coding was done while the video recordings were reviewed simultaneously, so that the players' meanings could be more accurately inferred. Having these codes in place served multiple purposes, as we illustrate below. First, these codes allowed us simply to determine the relative frequencies of each category relative to the larger corpus. Second, they allowed us to more precisely describe some of the game-turns in a systematic and consistent way.

Results and Examples

An initial analysis step in our work was to identify how frequently the different aspects of computational thinking appeared across the three groups. To do that, the coded transcript lines were automatically counted and plotted in five-minute intervals in the graphs shown in Figures 2 through 4. In total, there were 1711 utterances in the Lambda game (1 hour, 26 min), 1286 utterances in the Alpha game (1 hour, 23 min), and 869 utterances in the Delta game (1 hr 3 min). As talk was fairly continuous throughout the game, each five-minute interval should be thought of as roughly between 70-100 utterances. It is important to note that the amount of talk that indicated computational thinking is generally often less than half overall because of a high frequency of utterances in which players expressed agreement with one another ("Yeah", "Uh huh", "Okay"), stated directly the actions they were taking without specifying their thinking processes (e.g. "I'm going to move here, and now I draw a card"), or engaged in miscellaneous banter (E.g., "Dude, you're so lame", "What time is that party tomorrow?") with each other. Also, recall that the coding categories were not necessarily mutually exclusive, so there is some overlap between categories.

The first 10-20 minutes of each game involved group members reading the guidebook

Figure 3. Delta-game code counts and frequencies
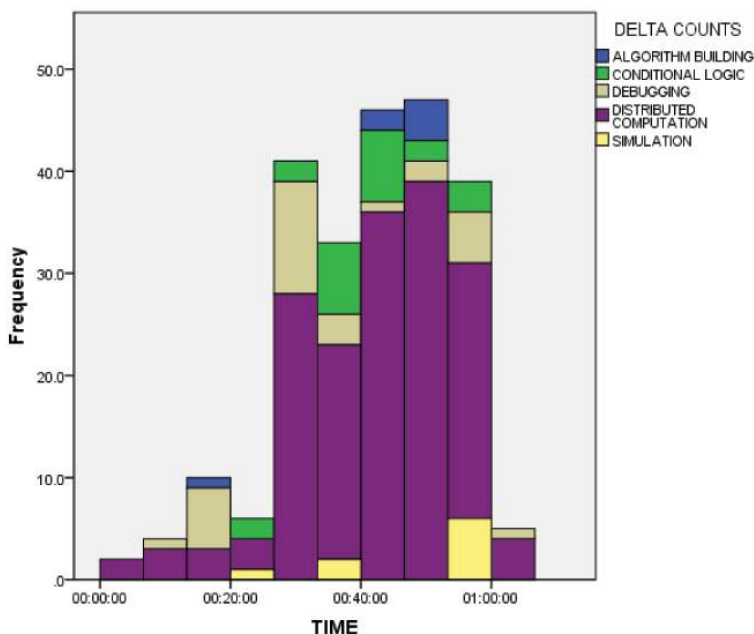
and deciphering the rules of the game, which explains the low amounts of computational thinking evidenced during the first five to ten minutes of each game. However, as we will discuss, dealing with or learning the rules were not exclusive of computational thinking. It was simply that much of those times involved one or two individuals reading out loud to the others. With those caveats in mind, it is apparent that the groups all varied with respect to each other and throughout the game in the kinds of computational thinking they expressed. Distributed computation was consistently the most frequently occurring computational discourse for all groups. The players were indeed engaging in a substantial amount of crosstalk and were collaboratively making sense of actions, algorithms, rules, and plans. In that respect, we see that *Pandemic* is successful at fostering collaboration.

The next most frequent type of observed computational thinking, other than distributed computation, depended on the group. For the Alpha group, it was simulation of future steps.

For Delta and Lambda, it was debugging. In the illustrative excerpts below, we will present examples of both.

Example 1: Debugging a Rule

Both the Lambda and Delta groups were actively involved in reviewing the rules of the game and resituating those rules into possible actions during individual turns. This took place during game play, rather than when the rules were first announced. For example, one rule introduced to all players early in the guidebook involves the conditions under which one can exchange cards with another player (Figure 5). Basically, two players may exchange a card if the card that is to be exchanged is of the city that both pawns presently occupy. The one exception to this is if a player draws a special card at the beginning of the game that exempts them from the requirement of having his and another player's pawn located in the city of the card that is to be exchanged. It is an exemption that only applies to the giving (not the

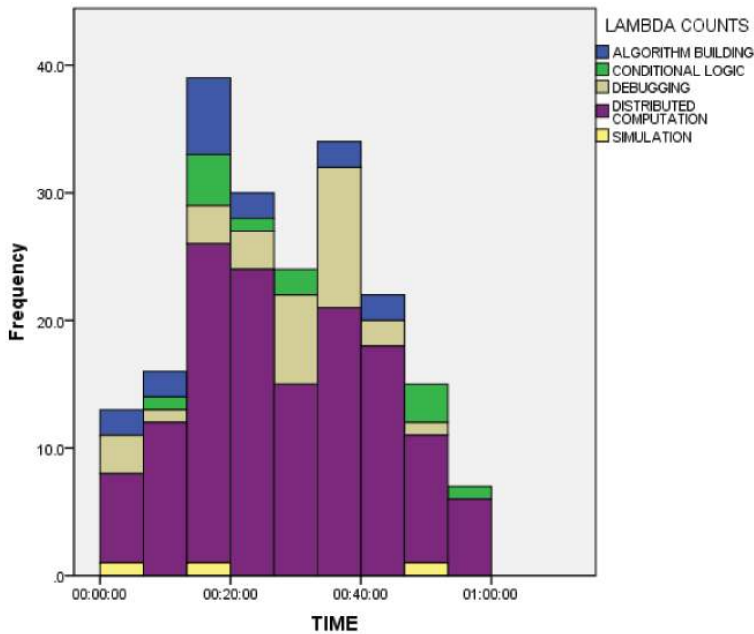*Figure 4. Lambda-game code counts and frequencies*

Figure 5. Card exchanges (a) when both players' tokens occupy the same space as the city card they wish to exchange are permissible at all times. On the other hand, (b) card exchanges in which a card different from the city occupied are not permitted unless a player has drawn a special card at the beginning of the game that permits such exchanges.
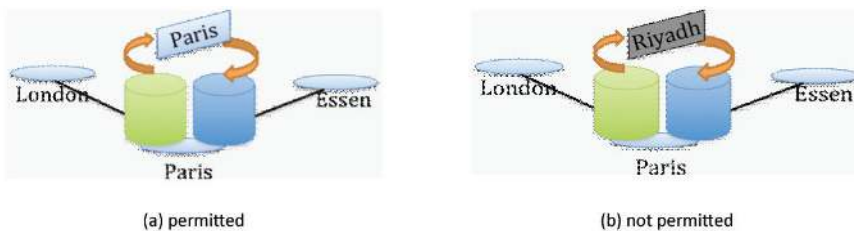
receiving) of a card by the exempt player. This proved to be a complicated point for Lambda; the group revisited it several times throughout the game. It involves establishing a number of preconditions being simultaneously met and also knowing an alternative path for following the card exchange action.

There was a substantial amount of time devoted by both the Lambda and Delta groups to determine what the preconditions were and whether or not they were met. To illustrate, we show some of the ways in which this played out for the Lambda group.

Twenty-two minutes into the Lambda group's game, Michael declared that he had the specific ability to exchange cards with other players without needing to be in he same city as the card that he is exchanging (i.e., the spe-

cial exemption). This then prompted Patrick to clarify how and when cards could be exchanged between players.[3]

1. Michael: I can give a player a card from my hands, for one action per card…That's my ability.
2. Jackson: Oh you can give me a card for an action point.
3. Michael: So it says both of our pawns must be in the same city, though it doesn't matter which city you're in.
4. Patrick: So since you're operations expert, your special ability.
5. Michael: Patrick, my special ability - I can give you cards as my actions. From my hand.
6. Patrick: They don't have to be in the city?
7. Michael: [Interrupting] Aahh, we have to be in the same city. But I can give it to you.
8. Patrick: Oh so, it doesn't matter like what city.
9. Michael: ...but we have to be in the same one.
10. Patrick: So doesn't matter like…
11. Michael: 'Cause when you like share knowledge…you have to be in the same city that's pictured on the card, or I can just give it to you.

Michael began by stating that he operated through the exception for card exchanges. Jackson then affirmed that the outcome for that operation would result in a card exchange and a decrease in one action point. Michael then clarified some of the preconditions. Patrick's subsequent interjection and discussion with Michael was one instance in which there were several utterances related to debugging (lines 6-9) because in those lines, the players sought to clarify whether the current card to be exchanged must be the same as the city that was currently occupied by the two player tokens. That debugging episode ends with Michael restating the main distinction between the regular rule and the exception that he can follow.

About fifteen minutes later, Patrick was relying on this knowledge of how cards were exchanged to suggest that LJ give Patrick two black cards. Michael agreed that could be a good plan, but Jackson interrupted to clarify (and debug what he saw as a possible confusion) around what was or was not possible given the rules.

12. Patrick: LJ, if you can give me the two blacks, we can build a research station in black, we could cure black.
13. Michael: Yeah, you're right in the, right next to each other.
14. Jackson: How come you just, he can't just give them to you though.
15. Patrick: No, I have to be in Mumbai or Chennai.
16. Jackson: So [Patrick] can get one of [either the Mumbai or Chennai card]

In the second excerpt, Patrick was coordinating with LJ on how to exchange two cards in a way that would help the group to make progress in the game. Their pawns were positioned near each other, but not in the requisite locations. Jackson interjected by stating that Patrick's cards did not follow the exception that applied to Michael, which Patrick agreed with (line 14-15). Jackson further sought to clarify the base rule that the exact city card that was to be exchanged must be the one that both LJ and Patrick player tokens occupied (line 16).

Twenty minutes later, on another one of his turns, Patrick was trying to figure out if he could receive a blue city card from Michael on the immediate turn. In order to straighten out what was or was not permitted, Michael had to refer to a written rule and translate that into permissible actions for card exchanges.

17. Patrick: My turn. Okay, so what we need to do.
18. Michael: Remember, you want me to give you my blues.
19. Patrick: So where are you?
20. Michael: Right there. In Sydney.
21. Patrick: So I need you to fly me, I'm going to fly you to LJ, and then you can, on your

turn, you're sure you can only give it to me on my turn, your turn?

22. Michael: Yes. [reading]You may give a player a card from your hand. One action per card.
23. Patrick: Okay.
24. Michael: So I have to spend my action to give it to you.
25. Patrick: Okay. But at least the next time around when it's my turn, I can cure blue.

That exchange involved clarification that the exception did not include any kind of exchange. That is, only during Michael's turn could an arbitrary card be given, rather than on Patrick's turn. As the non-exception rule permitted card-giving and receiving, this was another debugging of the rule that took place in the game, and ultimately prevented Patrick from executing his larger plan (i.e., curing the blue disease).

The above excerpts all represent instances in which debugging was making up a larger portion of the group's game play. Note that while these bugs could all be resolved by referencing written rules, these excerpts illustrate some of the same fundamental ideas shared in computation: special exceptions that must be handled (as is the case with Michael's special ability to exchange cards more freely) or sequences of actions must be reconfigured so that actions are permissible given previously established rules and conditions. These clarifications and exceptions were encountered under a larger goal structure that each individual player had. For example, Patrick often had the goal of making sure a specific player possessed a certain number of cards so a cure to a disease could be found.

Example 2: Simulation

While the activities of debugging were distributed in the above excerpts between multiple participants calling out exceptions or violations, distributed computation looked quite different in situations where conditional logic and simulation that takes place. To illustrate, we present a brief excerpt from the game with

group Alpha (Figure 6). This example arose as the players tried to decide what John should do during his next turn as they observed that the yellow disease and red disease were spreading on the game board. Note that in this excerpt, no pieces were being moved. They were only discussing hypothetical actions.

26. Aaron: That's the end of my turn. Right.
27. CJ: You should build the next one [research station].
28. Aaron: And this one goes here. So we're down two epidemics. So.
29. CJ: You should build the next one [research station] here so we can hurry and cure the yellow one [disease] before it gets all outbroken.
30. Aaron: Yeah, cause we got almost enough [cards]. We just need one more.
31. CJ: Actually
32. John: We need to stop this [the red spread] though. We need to prevent it.
33. Aaron: I can get over there. I'll fly into Bangkok and take care of that. That's 1, 2 [actions].
34. John: I know but
35. Aaron: Move over here and take care of that [yellow region].
36. CJ: Actually, you both-
37. John: Here's the thing. The next card is Ho Chi Minh, which means it's the fourth [disease marker] which means he'll get one-
38. CJ: Outbreak
39. John: -he'll get one, he'll, he'll get another. Which means these ones [cities] will all get one.
40. Aaron: Okay, so I can take care of that.
41. John: No, I have to take care of it right now.
42. Aaron: You going to take care of it? Okay. Oh yeah, huh.

Table 2 summarizes the features, rules and conditions, events, and outcomes of two "simulations" in this transcript.

During this excerpt, two simulations were being run (Table 2). One was being run primar-

Figure 6. Current game state for the red region of the board (East and Southeast Asia). Ho Chi Minh City and Bangkok have 3 disease cubes each and Taipei and Jakarta have 1 disease cube each. Referenced starting at line 32.
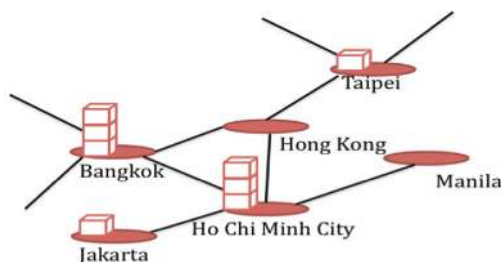
ily by Aaron, who suggested that the players focus on finding a cure for the yellow disease by using John's turn as an opportunity to establish a research station. That would then allow Aaron to cure the yellow disease, under the assumption that he would obtain a yellow card during his next turn. That was ultimately an offensive strategy against the game, in which the players would prevent the long-term spread of the yellow disease for the rest of the game. John ran the other simulation, which was more defensive in nature. John noted that at the end of each turn in the game, there were cards to be drawn with cities specifying where a disease infection will escalate next. If the Ho Chi Minh City card were drawn, then four disease tokens would infect Ho Chi Minh City, and if a city received four disease tokens, it would trigger a cascading outbreak. This was the point that John made when he announced, "we need to stop this" (line 32). He illustrated the implications of this hypothetical scenario by describing what would happen next in the game (lines 37, 39).

Aaron, who had already run his own offensive simulation, made a quick appeal that he could address the problem with the red disease during his turn (line 33). This was, effectively, an attempt to slightly modify John's defensive simulation. However, moments after he had a chance to fully understand the conditions under which John's simulation were being run (i.e., John would have to draw cards to increase the level infection at the end of his turn, and the next card could be Ho Chi Minh City), he accepted John's suggestion to treat Ho Chi Minh City immediately. CJ, who unsuccessfully tried to interject, these two simulations tacitly agreed to the defensive plan offered by John.

The utterances in this excerpt suggest that the players were using conditional logic to plan against unknown future conditions. They had internalized a set of rules and understand some starting conditions under which those rules can then be run over a period of time (characterized as player actions and player turns). By moving forward, one time-step after another, they were able to make some predictions about future outcomes and decide on a course of action, based on their game-based computations.

Local Nature of the Logic and Processing

The preceding examples show instances of computational thinking, but we have shown it only firmly situated within the game context. Action is based on immediate demands, and it is not always clear that any process of abstraction or generalization is taking place. While we expect that to some extent a player's computational thinking must be situated in the game context (Greeno, 1998), we would hope that the nature of that computational thinking would begin to involve some degree of generalization, as the 'programs' that students develop and run through their discourse should become more efficient over time. For example, we hope for players to develop new strategies (e.g., ignore

Table 2. The two simulations run during John's turn in the Alpha group game

	Yellow Cure Simulation (run by Aaron)	Red Infection Simulation (run by John)
Central Features of Current Game State for Simulation	Aaron possesses 4 yellow city cards	The Ho Chi Minh City space on the game board has 3 red disease cubes, as does the one for Bangkok.
Relevant rules and conditions for simulation	• A player must have 5 of the same color city cards to develop a cure of the disease of that color (e.g., James must have 5 blue cards in order to develop a cure for the blue disease) • A disease cure can only be developed on a city that has a *research station* built upon it • A research station can be built in a city by a player occupying that city as one of their actions during their turn. • New city cards for each player to use for cures or transportation are drawn at the end of each turn • A player can remove a single disease cube from a city they occupy as one of their four actions in a turn	• A disease cube of a pre-designated color must be added to each of the cities drawn from the infection card pile at the end of each player's turn. • Should the number of disease cubes of a given color at a city location be greater than three, then no new disease cube is placed on the current city. Instead, an additional disease cube is placed on each neighboring city. • A player can remove a single disease cube from a city they occupy as one of their four actions in a turn
Simulated events	*(based on transcript lines 27, 29, 30, 33 35)* **Step 1:** John builds a research station within reasonable proximity to Aaron's player token **Step 2:** John begins to eliminate yellow disease cubes from yellow cities. **Step 3:** Aaron begins to eliminate some red disease cubes from red cities. **Step 4:** Aaron obtains an additional yellow city card on his next turn, **Step 5:** After CJ and John complete their next turns, Aaron moves to the research station to cure the yellow disease with 5 yellow city cards.	*(based on transcript lines 32, 37, 39, 41, 42)* **Step 1.** Regardless of what John does during his turn, he draws two cards indicating cities that must be infected. **Step 2.** One of those cards is Ho Chi Minh City. That requires him to place an additional disease cube on Ho Chi Minh City. **Step 3.** Should no new action be taken prior, Ho Chi Minh City will already have three disease cubes, and therefore will not increase in total number of disease cubes. Jakarta, Bangkok, Manila, and Hong Kong will each receive an additional red disease cube. **Step 4.** Bangkok will also already have 3 disease cubes, and thus will not receive the new disease cube from Ho Chi Minh City. Instead, it will cause it's own neighboring cities to increase their number of red disease cubes by 1.
Simulation outcome	Yellow disease is cured in 5 player turns	Red disease level increases by a net of 6 cubes after 1 player turn.

disease levels until they reach 3 on a given city, always designate a single player to handle each disease) or principles or produce some generalizations (e.g., disease levels greater than 3 always trigger outbreaks) that would help them to control the game.

We describe the kinds of logic and processes in the two examples from the Alpha and Lambda games as 'local' logic. Local logic relates directly to immediate actions being taken and the structure of the logic or processes is not identified as applicable to a future course of action. A statement such as "let's move to Chennai and cure the diseases there" is an obvious example of local logic. An alternative is abstracted (or 'global') logic, in the sense that it involves "higher order" relationships (as per Newell & Simon, 1972). For example,

Table 3. ANOVA for changes in Alpha's use of global logic during game play. Alpha increasingly used more global logic statements later in the game

ANOVA[b]						
Model		Sum of Squares	Df	Mean Square	F	Sig.
1	Regression	.427	1	.427	5.355	.022[a]
	Residual	11.407	143	.080		
	Total	11.834	144			
a. Predictors: (Constant), TIME						
b. Dependent Variable: GLOBAL_LOGIC						

global logic may involve reasoning discourse such as: "we need to make sure to cure when cities have three disease cubes or else we have an outbreak risk." Global logic and processes require players to make a set of predictions for potential actions or decision in the game given past evidence, and to act on those predictions. This is roughly equivalent to (multi-agent) programming, in that players design plans for multiple actors given novel data.

Global logic and processes, as we refer to them, can be identified by using second order predicate logic operands (e.g., 'there exists some X', 'all X such that Y', or 'every X') and their synonyms in spoken English. For instance, at the simplest level, clarifications of the game's rules often take the form of global logic (e.g., Alex: "So you can use an action point every time to move me."). This form of logic recurs with increasing frequency as the games progress, as the players learn how to play the game. This occurs most clearly in the Alpha transcript, as shown in Table 3. As another example, consider the Lambda game[4], in which global logic statements appeared once in the first 500 utterances of the game, but then 9 times during the second 500 utterances. While we have not yet examined the interactions between local and global logic in great depth, we expect that the increased number of times when more global logic is used would increase over future games with the same group. We hope to devote more time to analyzing the transition toward abstraction in future papers.

Discussion: Computational Thinking and Tabletop Gaming

In light of our analyses and from our own observations of these groups of student players, we suspect that the quantity and quality of computational thinking in our data occurs because the players were required to: 1) internalize a set of rules and 2) devise strategies for optimizing behavior given the set of rules. In terms of computation, using conditional logic and debugging more often occurred as students internalized the rules of the game. None of the groups understood the rules by reading through the guidebooks without attempting to play through the rules. Behavior optimization required running those rules as part of a simulation or developing algorithms that will lead to desired outcomes. We have attempted to illustrate through the examples and analyses in this paper that the processing and reasoning that takes place in a collaborative strategic board game is complex and computational in nature.

Based on this work and our own observations as amateur tabletop gamers, we expect similar computational processes also take place in non-collaborative strategic board game play. However, they are less visible as players attempt to obscure their actions and motivations (in order to compete more effectively). We believe that the emphasis in many of these board games on resource management still engenders the same forms of conditional logic and simulation activities. For example, in the farm-themed strategic

board game, *Agricola* (Rosenberg, 2007), a player must consider how many turns will be required before he acquires enough material to build fencing for their livestock and the possible implications on a future score of making a pastures of a particular size or shape; the game requires some simulation activity in order to win. The research advantage to focusing on a collaborative strategic board game like *Pandemic* lies in the requirement that players must externalize their thinking processes and collaborate on their actions. Because coordination is necessary to win the game, it makes the development of rule understanding and group strategy formation an important part of the thinking and reasoning that takes place; distributed computation is explicitly verbalized. Those features make Pandemic uncommon among contemporary strategic board games, although there are others that are similarly collaborative such as *Lord of the Rings* (Knizia, 2000) or *Shadows Over Camelot* (Cathala & Laget, 2005), but it also makes *Pandemic* a promising site for future research (Zagal et al., 2006).

Thinking more broadly across board games, even beyond the contemporary, German-influenced ones that we have considered here, we expect there to be a degree of computational reasoning in gameplay. As we have said before, tabletop games are rule-based systems and require players to do the work that is often done by a computer in a video or console-based game. For example, we expect debugging to be the component of computational thinking most obvious across numerous game settings because it can be associated with the process of learning and internalizing rules. This was illustrated in Lambda group's work in debugging the card exchange rules. In that example, a player: 1) found a bug in his thinking through the enactment of his plan ('program'); 2) was flagged on his error by another player; and 3) revised his program so that it could conform to the actual game rules.

Algorithm building can occur across many different types of tabletop games, and we expect that it grows as an individual learns to better play a game. However, the reduced reliance on

chance is one distinguishing feature of contemporary, designer board games. In this genre of games, the player can build established routines fairly early on in their gameplay experience that can be reused several times in the future. The specific form of the randomness has a large effect on the types and amount of algorithm building that will occur. That is not to say that complex algorithms do not take place in games with high-randomness (e.g., *Scrabble*, in which the tiles one draws can absolutely influence the likelihood that an excellent player wins or loses (Fatsis, 2002)), but that designers can encourage complex computation through design features.

Having presented the argument that complex computational thinking takes place in one collaborative board game, we close with a bolder hypothesis that must ultimately be empirically verified. We suspect that many of these contemporary strategic board games could represent an important, and as-yet, under-considered foundation from which designers can intentionally develop computational thinking. This hypothesis invites new avenues for research. For example, we are exploring the instructional design issues associated with turning board game based computational thinking into a digital media computational literacy. The authors are also studying game-based computational thinking in its own right, focusing on understanding the complexities and design issues associated with the tabletop environment. One important next step is to more precisely connect the relationship between aspects of a game's design (e.g. turn structure, constraints, etc.) to the computational thinking that is ultimately elicited.

Thinking more broadly, there are also several social aspects related to strategic games that could also be explored. A number of contemporary board games have an inter-generational appeal to them; in our experience, we have observed families will often gather to play casually and local hobby stores frequently have a mix of teenagers and adults. An interesting issue to explore further is how this particular medium of play can increase participation in computational activities. Based on the work we have done in this paper and the possibili-

ties we see ahead, we remain optimistic that a promising arena for research has naturally emerged and is awaiting closer examination. Just as Gee (2007) shows how video games can be productive spaces for learning print literacy, we believe the same holds true and should be seriously considered for contemporary board games and computational literacy.

REFERENCES

Abelson, H., Sussman, G., & Sussman, J. (1996). *Structure and interpretation of computer programs* (2nd ed.). Cambridge, MA: MIT Press.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Borgman, C. L., Abelson, H., Dirks, L., Johnson, R., Koedinger, K. R., Linn, M. C., et al. (2008). *Fostering learning in the networked world: The cyberlearning opportunity and challenge.* Retrieved from http://www.nsf.gov/pubs/2008/nsf08204/nsf08204.pdf

Cathala, B., & Laget, S. (2005). *Shadows over Camelot*: *Days of Wonder.* Retrieved from http://www.daysofwonder.com/shadowsovercamelot/

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.

Fatsis, S. (2002). *Word freaks: Heartbreak, triumph, genius, and obsession in the world of competitive Scrabble players*. New York, NY: Penguin Books.

Fine, G. A. (1983). *Shared fantasy: Role-playing games as social worlds* (2nd ed.). Chicago, IL: University of Chicago Press.

Gee, J. P. (2007). *What video games have to teach us about learning and literacy*. New York, NY: Palgrave Macmillan.

Greeno, J. G. (1998). The situativity of knowing, learning, and research. *The American Psychologist*, *53*(1), 5–26. doi:10.1037/0003-066X.53.1.5

Haberman, B., & Yehezkel, C. (2008). Computer science, academia, and industry - an educational program for establishing an entry point into the computing community of practice. *Journal of Information Technology Education*, *7*, 81–100.

Kleinfeld, N. R. (2009). *Masters of the (tabletop) universe*. Retrieved from http://www.nytimes.com/2009/04/05/nyregion/long-island/05Rboard.html

Knizia, R. (2000). *Lord of the Rings*. Retrieved from http://www.fantasyflightgames.com

Leacock, M. (2007). *Pandemic*. Mahopac, NY: Z-Man Games.

Nasir, N. S. (2005). Individual cognitive structuring and the sociocultural context: Strategy shifts in the game of Dominoes. *Journal of the Learning Sciences*, *14*(1), 5–34. doi:10.1207/s15327809jls1401_2

National Research Council. (2010). *Report on a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Upper Saddle River, NJ: Prentice-Hall.

Nicholson, S. (2008). Modern board games: It's not a Monopoly any more. *Library Technology Reports*, *44*(3), 8–10.

Rosenberg, U. (2007). *Agricola*. New York, NY: Z-Man Games.

Schoenfeld, A. H., Smith, J. P. I., & Arcavi, A. (1993). Learning: The microgenetic analysis of one student's evolving understanding of a complex subject matter domain. In Glaser, R. (Ed.), *Advances in instructional psychology* (*Vol. 4*). Mahwah, NJ: Lawrence Erlbaum.

Shelton, B. E., & Wiley, D. (2007). *The design and use of simulation computer games in education*. Rotterdam, The Netherlands: Sense Publishers.

Sieg, W. (2007) The AProS project: Strategic thinking & computational logic. *Logic Journal of IGPL*, 359-368.

Steinkuehler, C. A. (2006). Why game (culture) studies now? *Games and Culture*, *1*(1), 97–102. doi:10.1177/1555412005281911

Teuber, K. (1995). *Settlers of Catan*. Retrieved from http://mayfairgames.com

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories -- an embodied modeling approach. *Cognition and Instruction*, *24*(2), 171–209. doi:10.1207/s1532690xci2402_1

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Zagal, J. P., Rick, J., & Hsi, I. (2006). Collaborative games: Lessons learned from board games. *Simulation & Gaming*, *37*(1), 24–40. doi:10.1177/1046878105282279

ENDNOTES

[1] Many of the most popular games originated from German designers, such as Klaus Teuber's *Settlers of Catan* (1995), and they have gained a large following in Europe. The term "designer game" has been used to acknowledge these games often prominently feature the name of the game designer on packaging materials.

[2] All names used here are pseudonyms.

[3] Transcript excerpts are numbered for the ease of reference in the text. These numbers do not necessarily indicate any order or sequence to the excerpts (e.g., utterance 12 happened several minutes after utterance 11).

[4] The absolute numbers of such logic statements in the Lambda and Delta games were too small to run meaningful statistical analyses.

Matthew Berland is an assistant professor in the Department of Interdisciplinary Learning & Teaching at the University of Texas at San Antonio. He received his Ph.D. in Learning Sciences from Northwestern University in 2008, studying computational literacy, systems literacy, and the design of constructionist learning environments. In 2009, he completed a postdoctoral fellowship in the Institute for Computational Engineering and Sciences at the University of Texas at Austin working on AI systems and human-robot interface design. His current projects include a mobile robotics game/learning environment; a computational thinking project using tabletop board games; a project to investigate the learning processes of novice programmers; and novel assessments for constructionist classrooms.

Victor R. Lee is an assistant professor of Instructional Technology and Learning Sciences at Utah State University. His research involves the study of visual representations, cognition as it takes place in face-to-face interactions, science education and instructional materials, and new technologies to support teaching and learning in K-12 settings. Current projects include a design investigation into the use of physical activity data devices for math and science learning, a qualitative evaluation of schools moving to an online-only science curriculum, and a study of how students represent informal learning experiences with digital photography. On the side, he is an amateur gamer who maintains a slight preference to tabletop games over console-based ones. Lee obtained his Ph.D. in Learning Sciences from Northwestern University in 2008.