FormulaT Racing: Designing a Game for Kinematic Exploration and Computational Thinking

Nathan R. Holbert, Uri Wilensky, Northwestern University, Evanston, IL
Email: nholbert@u.northwestern.edu, uri@northwestern.edu

Abstract: This paper describes the design and implementation of a prototype game, FormulaT Racing. FormulaT Racing is designed to be consistent with youth gaming culture while providing a thinking space for connecting intuitive notions of motion to everyday and formal representations of kinematics. A study with five children (age 7-13) revealed players engage with novel representations and construction tools in the game to develop complex computational strategies. We contend that the intuitive controls, alternate representations, and construction tools included in FormulaT Racing encourage players to consider the track as a collection of functional units – units of action made up of both track features and corresponding velocity changes – leading to an alternate encoding of embedded kinematic content.

Introduction

While a growing body of research shows a positive potential for video games as vehicles for learning (Barab, Thomas, Dodge, Carteaux, & Tuzun, 2005; Gee, 2003, 2007; Squire, 2005; Stevens, Satwicz, & McCarthy, 2008), there exists a tension between popular games created solely for entertainment purposes and educational games designed to teach content first and highlight entertainment second. In an effort to overcome this artificial dichotomy, our research agenda is to explore, create, and assess design principles that can be employed in popular commercial video games to enable players to connect intuitive experiences of embedded science content, to real world and formally taught representations. This paper describes a study of five children (ages 7-13) interacting with a prototype game, FormulaT Racing (Holbert & Wilensky, 2010), designed to encourage players to develop computational strategies to successfully navigate the physics embedded in this typical racing game.

There is a considerable amount of research literature examining children's understanding of motion. The overwhelming majority of this work has focused on "misconceptions" or children's tendencies to apply non-normative intuitive explanations to describe physical phenomenon (Carey, 1988; Duit, 2009; McCloskey, 1984). While science standards refer to Newtonian mechanics as "essential to understanding the natural world" (AAAS, 2002), research has shown an alarming number of high-school and college graduates fail to grasp these basic principles (McDermott, 1983). Researchers interested in physics education have begun to challenge the very notion of "misconceptions" and, in line with constructivist theories of cognition, suggest that learner's intuitive notions cannot simply be removed and replaced. Instead learning occurs most effectively when intuition is leveraged and refined (diSessa, 1993; diSessa & Sherin, 1998; Hammer, 1996). The importance of prior experience and salience of situational cues in this theory suggests that designs meant to help children make sense of Newtonian mechanics must consider common motion experiences. Drawing on this literature we argue that racing video games, a genre popular among youth (Lenhart et al., 2008), likely contributes to children's intuitive notions of motion and, as such, is both a potentially powerful means of intervention and an important context for conducting research on students' developing conceptions of kinematics.

Simply playing racing games, however, isn't enough. To transform racing video games into powerful kinematic thinking spaces we draw on the computational thinking literature. In the past few years there has emerged a consensus that it is important for 21st century students to be computational thinkers (diSessa, 2000; Guzdial, 2008; Resnick, 2001; Wilensky & Papert, 2010; Wing, 2006). The NRC has published a report clarifying the nature of computational thinking and its role in student learning (2010). While an official definition is still debated, we define computational thinking as the ability to translate or encode phenomena (real or imagined) into representations that leverage computational power. Often CT takes the form of utilizing abstractions to create algorithmic solutions to problems that can then be automated with computation.

Two core computational thinking practices on which we focus in this study are debugging and procedural thinking (Clements & Sarama, 1995; Noss, Healy, & Hoyles, 1997; Papert, 1980). Thinking procedurally involves chunking problems into smaller bits and recognizing patterns that can be effectively repeated (Papert, 1980). The NRC workshop on computational thinking (2010) suggests procedural thinking is about creating "a detailed step-by-step set of instructions that can be mechanically interpreted and carried out by a specified agent, such as a computer or automated equipment" (p. 11). Debugging involves systematic attempts to adjust a procedure or function in an effort to identify and correct the "bugs" or errors keeping a system from running properly. While games and software for building games have been proposed for teaching computational thinking (Kafai, 1995, 1996; Repenning, Webb, & Ioannidou, 2010), few have argued that simply playing video games can be an effective way to practice computational strategies. We believe the practice of computational thinking should be central in the design of video games.

In this paper we discuss the design and implementation of a prototype video game, FormulaT Racing, for connecting intuitive notions of kinematics to real world and formal representations of physics through the practice and refinement of computational strategies. Our intent is not to create a finished game for distribution, but instead to explore design principles that can be utilized by the gaming industry and included in commercially produced racing video games.

Theoretical Framework

FormulaT Racing (FTR) was designed specifically to tap into children's intuitive notions of kinematics and to connect these intuitions to formal representations while staying true to youth gaming culture. To be considered successful our design should look and feel like a traditional racing video game — one that participants could imagine sitting down to play after school, rather than in a classroom. However, we also intend FTR to be a game that participants will draw on in formal learning contexts as well as in common everyday experiences. Players may not become experts in kinematics by playing FTR, but they should be left with a sense that their experiences in the game are relevant to non-game motion experiences and players should be able to utilize qualitative foundational knowledge provided by the game to reason through more complex kinematic problems. To do this FTR foregrounds specific features of kinematics using tailored representations and controls embedded within typical racing game design, while also providing powerful construction tools that allow players to manipulate and debug these ideas in novel scenarios.

In a pilot study we found that traditional racing game design led to a one-to-one mapping of game action — instantiated by controller buttons — to discrete kinematic concepts (Holbert, 2010). In other words, specific controller buttons became synonymous with game actions (such as a "gas button"), which in turn stood in for isolated physics constructs (such as "velocity"). In FTR, we employ alternate designs that encourage the player to utilize computational strategies, ultimately leading to a more useful and flexible encoding of kinematic concepts. We refer to this new encoding as a computational encoding — by which we mean knowledge elements are relationally connected and function to describe and measure dynamic processes. We argue that a game that encourages this computational encoding should include the following set of design principles:

1. An interface connected to the player's intuitive and embodied understanding of physical phenomenon. (Barsalou, 2008; diSessa, 1993; Papert, 1980)
2. Representations that foreground the relationships between embedded content. (diSessa, 2000; Wilensky, 2006; Wilensky & Papert, 2010)
3. Opportunities to interact and create with these new representations. (Papert, 1980; Papert & Harel, 1991)

The following sections describe in more detail the theoretical underpinning of each design principle as well as how the principle is instantiated in the design of FTR.

1) Intuitive and embodied controls

Research in the Learning and Cognitive Sciences suggests much of our intuitive notions of motion are created through physical experiences out in the world (diSessa, 1993; Nemirovsky, Tierney, & Wright, 1998; Piaget, 1952; Roschelle, Kaput, & Stroup, 2000; Wilson, 2002). Work by diSessa and colleagues with physics students indicates that the richness of experiences in the physical world lead to dynamic, yet extremely salient, intuitive explanations for most common phenomenon (diSessa, 1993; diSessa & Sherin, 1998; Hammer, 1996, Sherin, 2006). A number of educational designs have also been introduced over the years showing that young children can be extremely effective at interpreting and constructing complex mathematical representations using motion-sensitive controls (Nemirovsky et al., 1998; Roschelle et al., 2000). Drawing heavily from theories of embodied, or grounded cognition (Barsalou, 2008; Wilson, 2002), these designs provide tools that allow learners to use physical movement in the world — movement that can be felt and experienced directly — to make sense of abstract mathematical principles.

FTR makes use of the *Nintendo Wiimote*, a commercial video game controller that includes multiple accelerometers, for controlling the player car. The controls allow for continuous (rather than discrete) adjustments of acceleration as well as heading, and serve as a metaphorical carrier for the player's idea of acceleration connecting it firmly to bodily experiences (Papert, 1980, p. 63). In other words, the player's natural bodily reaction to lean forward when wanting to "speed up" or backward to “slow down” changes the acceleration of the in-game car. In this way the control of in-game agents are naturally connected to conceptual “simulations” of motion (Barsalou, 2008).

2) Designing Restructurations

While representations in the world are often created with the intent to store, or embody some specific way of thinking, external representations also “become in a very real sense part of our thinking, remembering, and communicating” (diSessa, 2000, p. 6). Taking this theory of external representations seriously implies that alternate external representations may fundamentally change one’s thinking process. To this end, FTR was designed to enact what Wilensky and Papert (2006, 2010) call *restructurations* – changes in knowledge encoding as a result of a change in the representational infrastructure of a domain (2010, p. 2). In the case of FTR, by changing traditional representations of kinematics and the means of interacting with the player vehicle the game provides an opportunity for kinematic restructuration.

We have made two key design choices to facilitate this restructuration: including additional spatial representations of motion, and replacing discrete measures of velocity with formal representations that highlight change. FTR builds on the traditional “passing background” visual cue to indicate vehicle speed but adds a new “color-trails” cue. In this cue velocity is represented by a color-trail left by the player vehicle that changes as the player car’s velocity changes. These visual color-trails provide a means to connect ones changing speed to the structure of the track. In other words, players can more easily see how they slowed down around sharp turns or sped up on straightaways. In addition, FTR substitutes a velocity versus time graph for a speedometer to provide an early connection to formal kinematic representations and to highlight the importance of change, rather than static speeds. This velocity versus time graph is then color-coded to connect it firmly to the left behind color-trails.

3) Construction Tools

Finally, FTR also includes construction tools that fundamentally change the way the player *causes* motion further supporting kinematic restructuration. These construction tools are intimately connected to previously discussed controls and visual cues but are not explicitly introduced until the third phase of the game. This level was designed as a constructionist environment (Papert, 1980; Papert & Harel, 1991) allowing players to construct personal notions of motion by interacting with the representations of motion rather than the car itself. The player does this in one of two ways, either by painting the track different colors (that correspond to the color-trails they have become familiar with) or by constructing a velocity versus position graph.

In the “drive-by-paint” mode of the pit boss level the player utilizes the color palette of the color-trails to paint the track. The player can paint the track in any way they prefer, however, because each color corresponds to a particular velocity and the car’s ability to effectively turn is impacted by its current velocity, the choices made in painting the track determine whether or not the car will successfully complete the race. In the “drive-by-graph” mode, players construct a velocity versus position graph by accelerating points up and down the y-axis using the *Nintendo Wiimote*. Once the graph is constructed, the car “downloads” the data and drives around the track according to the velocities defined in the player-generated graph. In this way players directly connect the intuitive feeling of acceleration to formal graphic representations and can also explore how varying graphic features, such as sharp drops or plateaus in velocity correspond to particular track features.

We contend that the intuitive controls, alternate representations, and construction tools included in FormulaT Racing encourage players to consider the track as a collection of functional units — units of action made up of both track features and corresponding velocity changes. As players interact with and build vehicle motion using previously seen visual representations, and plan successful races by enacting computational strategies such as procedural thinking and debugging, kinematic concepts such as velocity and acceleration become functional — ideas that are no longer about category membership, but concepts that “do something.” In the following sections we will describe a study exploring children’s interactions with FTR. We argue, that rather than directly map game action to controller buttons, players of FTR utilized game controls, novel representations, and construction tools in functional units leading to a computational encoding of kinematic concepts.

Method

In this study, five children (ages 7-13), recruited from various informal organizations in a large Midwestern city, volunteered to test and provide feedback on a prototype video game, FTR. In a 15-minute pre-game interview session, researchers conducted a semi-clinical interview to gauge participants understanding of kinematics and their interest in video games. Two 45-minute game playing sessions were conducted a week later. In these sessions participants played FTR. Finally, a 15-minute post-game interview was conducted using the same prompts as the pre-game interview. Interviews and game play sessions occurred in the participants' homes or at an after-school program they were attending. All interactions with participants were videotaped and in-room recordings were synced with screen recordings of game play for analysis (Stevens, Satwicz, & McCarthy, 2008).

While we have done a larger analysis of FTR, this paper will focus on player interactions with the pit boss level. Here, video data was split into interaction units according to instances of strategy switching. In most cases the obvious point of strategy switching occurs after a failed run, occasionally however, verbal or physical cues from the player indicate a strategy shift between track resets. Interaction units were coded using a scheme emergent from the data informed by the computational thinking literature (Table 1). An independent researcher verified game-play codes. Conflicts were discussed and resolved resulting in agreement on 97% of video time.

Table 1. The following excerpt from a larger coding scheme was used to analyze video data of players interacting with the construction tools. These three codes were identified as "complex computational thinking" by the researchers. The full coding scheme is available upon request.

Code	*Description*	*Example phrases*	*Examples in-game*
Strategic	The player is painting the track in a strategic way. There is some indication that the player has an idea in their head they are trying to enact on the screen. There is a definite "plan." being enacted.		
Ordered	The player implements their plan in an ordered fashion, from beginning to end.	"First I need to… and then…"	The player constructs his idea starting at the beginning of the track moving towards the end and may follow along with the track image using their finger
Motif	The player has created a strategic pattern that they are repeating — not unlike a procedure that's being used at specific times.	"Every corner is a fast color and every line is a fast one"	Colors are clearly related to track features and repeated when the feature repeats. Peaks and valleys in the graph are clearly related to track features.
Debugging	Attempts are made to identify and fix a problem. Players may try to add or change colors (or graph points) in systematic, but small, ways.	"Maybe if I add some purple here…"	Player quickly adds or removes color in only one or two locations before running again. Points on the graph are just "changed" rather than rebuilt from scratch.

Results

Because FormulaT Racing is designed to provide a thinking space for players to explore and construct with kinematic concepts and representations, and not a game to *teach* physics, our analysis explores whether or not players engage with representations in complex and computational ways. Results suggest players develop systematic computational strategies to be successful in construction levels by leveraging game experiences and representations from previous levels.

Construction Tool Use

Players typically begin by testing uniform motion on the entire track, such as "painting" the track a color that causes the car to drive extremely fast. Gradually, players utilize intuitive knowledge of motion and in-game experiences to systematically debug constructions. Ultimately players begin to notice and reuse patterns of motion and track features to paint and graph successful solutions. Figure 1 shows the percentage of total time players enact a particular computational strategy while playing the pit boss level. While players spend some time simply exploring the model — painting the track all one color, "just to see what will happen," or to see how fast the car could go — players engage in sophisticated computational strategies (coded as strategic-ordered, strategic-motif, and debugging) 76% of the time.

A detailed analysis of each player's progression with construction tools shows evidence of not only computational thinking in action, but also paints a picture of computational strategy evolution. One of the youngest participants, Collin, struggled early to understand the mechanics of the construction levels. When painting the track, Collin was very strategic about his designs. When his construction would fail, Collin would work to understand what went wrong and systematically debug his design. He might add a fast color in a straightaway if he struggled to make it around the track in time or he may add a small strip of violet (a slow color) on a corner if he was crashing. However, if these small tweaks failed, Collin would often erase the entire track and claim, "I have another plan!" These early debugging attempts, such as putting only a small strip of violet in the exact location of a crash, indicate Collin had a disconnected understanding of acceleration and velocity — Collin knew

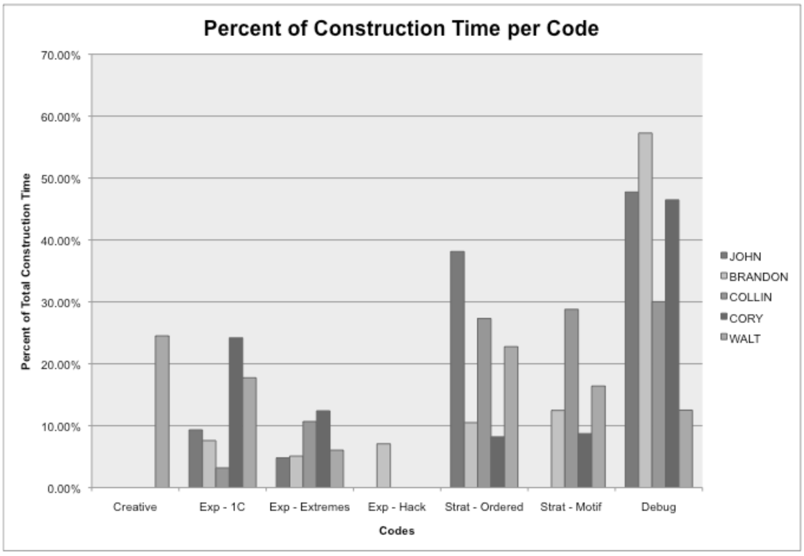

Figure 1. This graph shows the breakdown of time each individual spent engaged in the coded activities. 76% of time spent using the construction tools was spent engaging in complex computational thinking.

violet indicated a slow color, but he didn't take into consideration the acceleration that would be required to reach this speed. As Collin continued to interact with the construction tools, "*strategic motifs*" began to emerge. Before painting on a new track Collin thinks out-loud and states:

> **Collin**: Oh but that won't work because then I'll have to do it over and over again and it will crash… (pause) my idea is just going to make it crash again. (pause) Well, I'll test it.
> **Interviewer**: What's the plan?
> **Collin**: *Every corner is a slow color and every line like this is a fast one.*

At this point, shortly before constructing a successful run, Collin has begun to break his strategy down into small "procedures" (italics) that include multiple colors related to specific track features that he then used repeatedly at key track points (Figure 2). This procedural painting suggests Collin has begun to see acceleration as highly related to velocity and that together these kinematic concepts result in very specific types of motion relevant to different aspects of the race.

Figure 2. Collin's early and final attempts at painting the square track. His first attempt (left) was coded as "strategic - ordered." The final and successful version (right) indicates clear signs of "strategic - motifs" where slow colors are used in the corners and fast colors on the straightaways.

Collin's first attempt in the drive-by-graph mode made use of strategic motifs immediately. Rather than plot all 20 points on the x-axis, Collin only plotted eight points directly corresponding to the number of straightaways and corners. When presented with an error due to not "filling" the graph, pointing to different segments of the track Collin states, "Oh I see, I was going just like, uh...fast, slow, fast, slow." What at first looked like repeated spikes, or moments of high positive acceleration followed by negative acceleration, was Collin's reinterpretation of the track as a collection of repeated kinematic motifs rather than a continuous series of motion moments. After editing his graph to include all 20 points, Collin struggles with the scale making the car go as fast as possible as soon as possible resulting in a spectacular crash early in the race. Seeing his failure he asks, “How do I know how fast it is? Oh yeah! By using the other side [indicating y-axis labels]!” Collin, a participant that had asked to skip the graphing task conducted in the pre-game interview, not only identifies graphing errors from vehicle motion, but also constructs a new successful graph on the very next attempt.

Brian engaged in a variety of different computational strategies, but spent a large amount of his time in FTR debugging. Brian often began by painting the track one color, and then added and removed colors systematically. After being successful on a track the interviewer questions why he altered the paint at various points. Brian’s answers indicate a rich connection between the vehicle’s acceleration and the track features:

> **Brian**: Every spot that I picked blue, was all the spots where he crashed previously.
> **Interviewer**: Any idea why it crashed?
> **Brian**: *Maybe it moved too fast and didn't have enough time to turn*. So I slowed it down with some blue paint. And whenever it still crashes *I'll just make the blue paint larger. At least large enough for it to have enough time to steer.*

For Brian, the debugging process allows him to focus on the dynamic and time dependent nature of velocity as it relates to sharp turns and straightaways on the track.

The stories of the two FormulaT programmers show instances of computational strategies being employed and refined as they continue to interact with the game. As players progress in the pit boss level, insights gained early on in the painting version carried over into the graphing. As computational strategies become more sophisticated, player transcripts show evidence of a kinematic restructuration – players begin to talk about acceleration and velocity as interconnected units dependent on track features. In this new structuration, motion motifs continually interact with the previous and next motif resulting in a highly dynamic series of kinematic patterns.

Conclusions

Arguing for personal exploration in mathematics Confrey (1991) claims, “if mathematics is viewed as functional, the emphasis is not with mirroring some unknowable reality, but in solving problems in ways that are increasingly useful in one’s experience” (p. 136). Tools such as algebra and kinematics are simply designed artifacts that help us make sense of phenomena in the world. While it is likely that some representations are “better” at dealing with a wide variety of situations, such as formal physics conventions, these situations must be anchored in concrete experiences and embedded with personal meaning. Our work with FormulaT Racing suggests that popular video games may be able to support this meaning making for scientific domains by leveraging computational thinking. The evidence presented here suggests that players utilized complex computational strategies when interacting with construction tools and representations that they had imbued with kinematic meaning leading to an alternate, computational encoding of the embedded kinematic concepts.

References

AAAS. (2002). *Middle Grades Science Textbooks: A Benchmarks-Based Evaluation* from http://www.project2061.org/publications/textbook/mgsci/report/Sci_Ins/SIS_ps2.htm

Barab, S., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development*, *53*(1), 86-107.

Barsalou, L. (2008). Grounded cognition. *Annual Review of Psychology*, *59*, 617-645.

Carey, S. (1988). Reorganization of knowledge in the course of acquisition. In S. Strauss (Ed.), *Ontogeny, Phylogeny, and Historical Development* (pp. 1-27). Norwood, NJ: Ablex.

Clements, D. H., & Sarama, J. (1995). Design of a LOGO environment for elementary geometry. *The Journal of Mathematical Behavior*, *14*(4), 381-398.

Committee for the Workshops on Computational Thinking. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking.* Washington, D.C.

Confrey, J. (1991). Learning to listen: A student's understanding of powers of ten. In E. v. Glasersfeld (Ed.), *Radical Constructivism in Mathematics Education* (pp. 111-138). Netherlands: Kluwer Academic Publishers.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and instruction*, *10*(2 & 3), 105-225.

diSessa, A. A. (2000). *Changing Minds: Computers, Learning, and Literacy*. The MIT Press.

diSessa, A. A., & Sherin, B. (1998). What changes in conceptual change. *International Journal of Science Education*, *20*(10), 1155-1191.

Duit, R. (2009). *Students' and teachers' conceptions in science: A bibliography* from http://www.ipn.uni-kiel.de/aktuell/stcse/stcse.html

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.

Gee, J. P. (2007). *Good Video Games and Good Learning*. New York, NY: Peter Lang.

Guzdial, M. (2008). Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 27.

Hammer, D. (1996). Misconceptions or p-prims: How may alternative perspectives of cognitive structure influence instructional perceptions and intentions? *Journal of the Learning Sciences*, *5*(2), 97-127.

Holbert, N. (2010). Feeling fast: The role of intuitive thinking in video games. Poster presented at the annual meeting of the American Educational Research Association, Denver, CO, April 30-May 4.

Holbert, N., & Wilensky, U. (2010). FormulaT Racing. Evanston, IL: Center for Connected Learning and Computer-based Modeling.

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kafai, Y. B. (1996). Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact. In Y. B. Kafai & M. Resnick (Eds.), *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mahwah, NJ: Lawrence Erlbaum.

Lenhart, A., Kahne, J., Middaugh, E., Macgill, A. R., Evans, C., & Vitak, J. (2008). *Teens, Video Games, and Civics.* PEW Internet & American Life Project.

McCloskey, M. (1984). Naive theories of motion. In D. Gentner & A. Stevens (Eds.), *Mental models*. Hillsdale, NJ: Lawrence Erlbaum.

McDermott, L. C. (1983). Critical review of research in the domain of mechanics*. First international workshop research on physics education* (pp. 139-182). Paris

Nemirovsky, R., Tierney, C., & Wright, T. (1998). Body motion and graphing. *Cognition and Instruction*, *16*(2), 199-172.

Noss, R., Healy, L., & Hoyles, C. (1997). The construction of mathematical meanings: Connecting the visual with the symbolic. *Educational Studies in Mathematics*, *33*(2), 203-233.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Papert, S., & Harel, I. (1991). Situating constructionism. In S. Papert & I. Harel (Eds.), *Constructionism*. New York: Ablex Publishing.

Piaget, J. (1952). *The Orgins of Intelligence in Children*. Madison, CT: International Universities Press, Inc.

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools*. 41st ACM technical symposium on computer science education.* (pp. 265-269). Milwaukee, Wisconsin

Resnick, M. (2001). Revolutionizing learning in the digital age. In M. Devlin, R. Larson & J. Meyerson (Eds.), *The internet and the university: Forum 2001* (pp. 45-64). Boulder, CO: EDUCAUSE.

Roschelle, J., Kaput, J. J., & Stroup, W. (2000). SimCalc: Accelerating students' engagement with the mathematics of change. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in science and mathematics education: Advanced designs for technologies of learning* (pp. 47-76). Hillsdale, NJ: Earlbaum.

Sherin, B. (2006). Common sense clarified: The role of intuitive knowledge in physics problem solving. *Journal of Research in Science Teaching*, *43*(6), 535-555.

Squire, K. (2005). Changing the game: What happens when video games enter the classroom? *Innovate: Journal of Online Education*, *1*(6), 25-49.

Stevens, R., Satwicz, T., & McCarthy, L. (2008). In-game, in-room, in-world: Reconnecting video game play to the rest of kids' lives. In K. Salen (Ed.), *The ecology of games: Connecting youth, games, and learning* (pp. 41-66). Cambridge, MA: The MIT Press.

Wilensky, U. (2006). Complex systems and restructuration of scientific disciplines: Implications for learning, analysis of social systems, and educational policy. Paper presented at the *annual meeting of the American Educational Research Association*. San Francisco, CA.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulating knowledge disciplines through new representational forms*.* In J. E. Clayson & I. Kalas (Ed.), *Constructionism 2010*. Paris, France

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33-35.