Know Your Enemy: Learning From In-Game Opponents

David Weintrop
Northwestern University
2120 Campus Drive, Suite 332
Evanston, Illinois 60628
dweintrop@u.northwestern.edu

Uri Wilensky
Northwestern University
2120 Campus Drive, Suite 337
Evanston, Illinois 60628
uri@northwestern.edu

ABSTRACT
In this paper we present a novel approach to the design of game-based learning environments in which the content to be taught is embodied by the opponents the learners compete against as they play. By providing the player with the resources to make sense of the concepts exemplified by their opponents, as well as the tools needed to incorporate the concepts into their own gameplay strategy, players are challenged to learn from their opponents in order to advance in the game. This paper introduces RoboBuilder, a blocks-based, program-to-play game that uses this design strategy to introduce programming novices to core computer science concepts. Along with more fully developing this design principle, we provide evidence from a preliminary study conducted with RoboBuilder of players learning from their opponents to create winning strategies that use the concepts designed into the opponents they are facing.

Categories and Subject Descriptors
D.1.7 [Visual Programming]. K.3.2 [Computer and Information Science Education]:Computer science education.

General Terms
Design, Human Factors, Languages

Keywords
Game-based Learning, Visual Programming, Design, Computer Science Education

1. INTRODUCTION
The advice to "know your enemy" was a strategy prescribed by Sun Tzu, who, in his book The Art of War, stated that by knowing your enemy, "you will not be imperiled in 100 battles". Fast-forward 2,500 years, moving from ancient Chinese battlefields to the modern, virtual battlefields of video games and the advice still holds true. A key strategy for having success in many modern games is to decipher the behavior of your adversary and then use that knowledge to ultimately defeat it. In many games, the strategies employed by in-game opponents have little use beyond that specific game. But what if the strategies employed by in-game opponents did have value beyond the context of the game? By carefully designing in-game opponents to embody and demonstrate specific concepts, a conventional video game context can be transformed into a powerful learning environment.

This paper presents RoboBuilder (Figure 1.), a program-to-play game that uses in-game opponents to embody computer science concepts. To play the game, learners must develop and implement strategies in the form of simple programs using a custom designed graphical programming language [14]. The game was designed so that players are provided with the tools and resources necessary to understand and potentially apply the concepts their opponents embody. As learners advance in the game, they are introduced to more sophisticated opponents and more complex ideas. In taking this approach, we strive to make the process of learning congruent with existing youth gaming norms.

We begin this paper by more fully developing our design approach and discussing design features that support players in learning from their opponents, using RoboBuilder as an example of one such environment. We then continue with a discussion of work that informed our design, then return to RoboBuilder and present preliminary findings from a study investigating how programming novices learned from their in-game opponents. We conclude the paper with a brief discussion of the strengths and challenges of applying this design strategy.

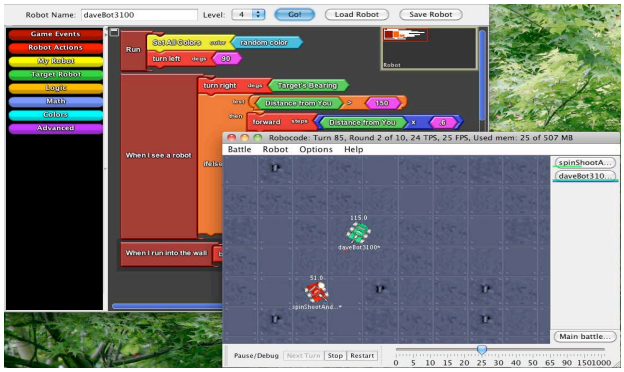

Figure 1. The RoboBuilder game.

2. LEARNING FROM OPPONENTS
The Learn From Your Opponents (LFYO) design approach uses in-game opponents as the mechanism through which learners are introduced to the content being taught. By having in-game opponents embody, demonstrate, or in some other way introduce the desired concepts, they can serve as opportunities for learners to interact with, explore and ultimately come to understand the concept. By designing a gameplay interaction that supports and encourages the practice of learning from opponents, in-game opponents become objects-to-learn-from, providing the designer a novel way to introduce content into game-base learning environments. With this strategy, learners are introduced to the concepts by seeing them in use, as opposed to encountering them

IDC '13, June 24 - 27 2013, New York, NY, USA

ACM 978-1-4503-1918-8/13/06…$15.00.

in abstract, decontextualized ways. In seeing a concept in use, players can develop a sense of its potential applications and see an example template for how to incorporate the concept into their own gameplay strategies as they progress through the game.

2.1 Features Of These Environments

In order to successfully employ this design strategy, there are a number of features of the environment that must be present. First, it needs to be clear to the learners that there are advantages to attending to and learning from their opponents with respect to accomplishing the in-game challenges. Second, to support learners in engaging with concepts introduced through their opponents, it is critical to provide learners with an appropriate and accessible set of tools or resources [11]. This includes an environment to explore and interact with the concepts, tools to recreate and employ the ideas being taught, and encouragement for interactions that enable learners to deeply engage with the material being taught [17]. By providing players a venue to test out and experiment with the concepts presented, learners are given the opportunity to figure out how these new concepts or strategies can be employed to accomplish the in-game task set before them. Finally, it is important that these environments provide feedback to the learner on the outcome of their attempts to employ the ideas being introduced by the opponents. This feedback serves as valuable debugging information and can provide clarification on the properties and characteristics of specific concepts. The process of experimenting with new concepts and then receiving feedback from the game provides players with the opportunity to learn from their mistakes, a powerful learning strategy [10].

2.2 Meet RoboBuilder

RoboBuilder is a game-based learning environment we designed that employs the LYFO design strategy. RoboBuilder is a program-to-play game that challenges players to design and implement strategies to make their on-screen robot defeat a series of progressively more challenging opponents in one-on-one battles [14]. The game is a constructionist video game designed to introduce players to basic computer science concepts and practices [13]. The objective of the game is for players to defeat their opponents by giving their robot instructions to locate and fire at their opponent while avoiding incoming fire; the first robot to make its opponent lose all its energy wins. Unlike a conventional video game, where players would control their robot directly during battle, in RoboBuilder, players must program their robot using a game-specific, graphical programming language before the battle begins. All of the opponents in the game are built using the same set of language primitives that are provided to the learner, a fact made clear to the player at the outset of the game. Because this is true, players can use the language primitives as a means to make sense of the observed behaviors and know it is possible to recreate observed behaviors [15]. Each of the opponents was designed to embody a specific concept such as conditional logic, iterative logic, and state.

3. LITERATURE REVIEW

In this section we present a brief review of the literature that informed the LFYO design approach, as well as work that was influential in the design and implementation of RoboBuilder.

3.1 Game-based Learning Environments

Game-based learning environments have grown in popularity in recent years. Part of this is due to the increasing presence of video games in youth culture [7]. Another contributing factor stems from the fact that a growing body of research has found video games, and game-based learning environments more generally, to be an engaging and effective medium for teaching and learning [2]. This includes both commercial video games as well as games designed specifically for learning [4] Along with learning through playing games, a great deal of work has examined the learning that can happen through constructing games [5].

3.2 Constructionist Learning Environments

Some key aspects of the LFYO designs strategy stem from the constructionist tradition [3]. First, this design approach requires that learners be able to employ the concepts being introduced as they play the game, with the ability to experiment and test-out their ideas being critical. The construction of, experimentation with, and feedback from the act of constructing in-game artifacts is essential to successful learning in these types of environments. Second, by giving learners an opportunity to interact with and explore the concepts in a way that the learner finds compelling or interesting, the act of learning becomes an individual, personalized process driven by the learner's own actions [10]. Finally, we see learning environments that employ this design strategy as a subgenre of computational microworlds, which are "computer-based interactive learning environment[s] where the prerequisites are built into the system and where learners can become the active, constructing architects of their own learning" [10]. Our proposed game-based learning environments fit nicely in this category as the concepts, and tools with which to explore them, are both embedded in the game.

3.3 Theoretical Framework

A central idea that informs this work is the recognition that learning is personal and that the context in which learning takes place is central to the understanding that develops for the learner. This view recognizes that the tools and resources with which the learner explores and expresses the ideas under investigation, the physical setting where the learning environment is used, and the social supports available to the learner, all play a critical role in the learning process [6, 12, 16]. In applying this theoretical frame to environments that employ the LFYO design strategy, we can see how the design of the opponents situated within the environment, and the tools and resources provided to support learners, influence the resulting understanding of learners. In their work on mathematical meaning making in computational microworlds, Noss and Hoyles [9] developed the construct of situated abstractions to capture the fact that meaning is situated within the context in which it is employed. In developing this idea, they emphasize how the network of resources provided by the environment itself, what they call webbing, is essential to the resulting understanding that develops. In this way, learners are "abstracting within, not away from, the situation" [8]. Given this central role of context, how in-game opponents embody the concepts of interest and the design of the resources and tools provided are critical decisions in the creation of these types of learning environments.

4. Methods

Having completed the design of our game, we conducted a study to see if and how learners use in-game opponents as objects to learn from? While RoboBuilder was designed as a learning environment for late middle school/early high school students, we recruited participants more broadly as we were interested in how programming novices, regardless of age, responded to this design feature. Fifteen subjects were recruited to participate in this initial

study, this group included: one middle school student, seven high school students, and seven university students. The university-aged participants were students at a large Midwestern university. Two of the younger participants were recruited through university connections, while the remainder of the participants were recruited through a community center in a large Midwestern city that serves a predominantly African-American, low SES community. Each participant played RoboBuilder for at least 40 minutes, resulting in a total of roughly 18 hours of interview and gameplay footage and over 200 robots being constructed.

The data presented below were collected through one-on-one interviews in which a researcher sat alongside the participant as he or she played the game. The interviews proceeded in a three-phase protocol. First, the interviewer initiated conversation with the players about their strategies. Next players were given the chance to implement their strategies using the blocks-based language while the interviewer observed. Finally, the players watched their robots compete, with the interviewer asking them to describe what they observed and whether or not it matched their expectation. After each battle, players were asked to explain how they wanted to proceed with their robot design, thus restarting the interview protocol. Through the discussions of the battles and hearing the players verbally articulate their goals, we gained insight into their developing understanding of the concepts embedded into the game and could identify how and when players attended to the opponent's strategy. Each interview was recorded using both screen capture and video-capture software, which served as the primary data source for our analysis.

5. FINDINGS

In this section we describe episodes from our interviews that provide evidence of learners attending to the in-game opponents either as a source of inspiration for their own strategies, or as a way to advance their own understanding of a concept. This is meant to serve as a proof-of-concept of the LFYO design approach; a more comprehensive analysis is forthcoming.

5.1 Beth Learning From Her Opponents

In this section we describe two episodes from a single participants' RoboBuilder experience. Beth, a vocal performance undergraduate student with no prior programming experience who proved to be one of our more thoughtful participants, was about 15 minutes into her RoboBuilder session when we first saw her shift her attention from her own robot's behavior to her opponent. The level-one robot remains motionless until its energy drops below 50, at which point it begins to move. As Beth watched her robot battle, she saw her opponent's energy drop below 50 and start to move; she then said: "*so that robot must have something built into it when it reaches 50. Oh! There we go! So that's what the, that's what the other boxes are for, so like if you reach a certain health level, you can change the actions*." Here we see Beth attend to her opponent's strategy and by doing so, she encounters the two concepts it embodies: conditional logic and robot state. Figure 2 shows the blocks that control the level-one opponent (Note: these blocks were not visible to Beth).

Figure 2. The blocks that control the level-one opponent.

Beth's comment: "*if you reach a certain health level*", shows not just that she has identified the two concepts that are involved, but she correctly uses them to explain her opponent's behavior. Knowing that her opponent's behavior was created with the same blocks she had to work with, the challenge of explaining this behavior became that of mapping the observed movements onto the tools provided. We see this in her reference to *"the other boxes",* which refers to the Robot State blocks, such as My Energy, that allow you to get information about your own robot (this fact is made clear in the utterance that followed the quote above). In this episode, we can see how by attending to the in-game opponent's behavior and using the language primitives provided, Beth was able to make sense of the concepts designed into the opponent she was facing.

Our second example of Beth encountering new ideas by learning from and using the strategies introduced by her opponent can be seen in the approach she took to defeat the level-seven opponent. The level-seven robot moves in a circle, adjusting its gun as it moves to keep it focused on its opponent. In this way, the robot has coordinated two different aspects of its own state to work in concert to produce an effective strategy. Beth's plan to defeat this new robot was to have her robot remain stationary and track its opponent with its gun. When asked why she was pursuing this strategy, she confessed it was not for tactical reasons, but instead that it would be easier than coordinating states like her opponent. Having said this aloud, she paused, then said: "*You know, but maybe I could make a robot that is exactly like it, except going the opposite way. You know? 'Cause it is going clockwise and it's gun is turning left, what if I made a robot that was going counter-clockwise and it's gun was always turning right?...I kind of want to see what happens.*" After a few iterations of tinkering with turning angles and step distances, Beth had successfully recreated her opponent's behavior and ultimately was successful in using the level-seven robot's strategy against it. By attending to, then replicating, her opponent's strategy, Beth was able to implement a strategy that she initially thought would be too difficult.

5.2 The Level-Six Opponent

The level-six opponent's strategy is to oscillate forwards and backwards, spinning its gun and firing at the player's robot each time it changes direction. In this way, the robot demonstrates how iterative logic can be incorporated into a robot strategy. Here we present an analysis of how different players responded to the level-six robot and the oscillating, iterative behavior it embodied.

Of the 15 players who participated in this study, eight players competed against the level-six opponent, four middle and high school students and four university students. We coded all of the level-six battles looking for how each participant perceived the oscillating strategy and the steps they took to try and defeat it. Four of the eight participants verbally commented on the forward-backward repeating pattern while watching their own robots compete against it, saying things like: *"It just goes forward then backward, over and over"* and *"It moves back and forth…are those the only place it goes? I think so"*. These comments show the players noticing the iterative nature of the robot's strategy. Of these four, three participants incorporated iteration into their own strategy. One university student articulated his plan to defeat the level-six robot this way: "*OK, so what I'm trying to do now is, I'm trying to do a little bit of his action, and some of my own, like kind of an amalgam of what he was doing and what I was doing to see if I can generate a better result*." After stating this, he proceeded to add iterating forwards then backwards logic to his robot strategy. The behavior was similar, although less explicit with our middle school participant. During an early battle with the level six opponent, he commented that its behavior was *"annoying",* but

ultimately decided to copy the forward-backward pattern, at one point saying that he was going to give his opponent "*a taste of his own medicine*". Of the four players who did not attend specifically to the iterative pattern of the level six opponent, two of them had already developed their own oscillating, iterative strategies earlier in the game, and thus were able to defeat the level six opponent by altering their existing strategy. In total, of the eight players who competed against the level six opponents, six of them verbally articulated or used iterative logic during their encounters with it.

6. DISCUSSION

6.1 Concepts Embedded in Gameplay

Over the course of gameplay, every player wrote a working program, and all but one, implemented a robot strategy that accomplished the first goal designed into the game. As they faced new opponents, learners were exposed to core computer science concepts and provided examples of how they could be used in the game. By introducing each concept alongside the tools with which to make sense of it, and by providing an accessible way of engaging and experimenting with the new concepts, learners able to utilize the resources to develop an understanding of the concepts embedded in the game. By using in-game opponents as the mechanism by which new concepts are introduced to the learner, the game convention of having players advance to more challenging levels as they progress in the game provides a natural way to build in a learning trajectory for the learner to follow. Through a careful sequencing of the opponents, designers can introduce basic, more accessible, or more foundational concepts early in the learning experience, laying the foundation for the more advanced concepts that follow.

6.2 Content in Context

Research in the learning sciences has found that creating a meaningful context has a positive impact on learners [1]. In RoboBuilder, learners encounter concepts and develop strategies situated within a context that encourages them to use the concepts or strategies as they are encountered or discovered. The language primitives develop meaning for the player through the iterative, construction process central to gameplay. By providing such rich contexts "these meanings become reshaped as learners exploit the available tools to move the focus of their attention onto new objects and relationships" [9]. The LFYO design strategy provides an avenue for supporting the learner in making sense of the concepts being introduced within the context of the game.

6.3 Challenges of this Approach

While we think there is great potential in utilizing this design strategy, it is not without its challenges. A central challenge we encountered was ensuring that players did in fact attend to and grapple with the concepts that each robot embodied. In some cases, robots proved too easy for the learners and thus they did not notice the concept on display, while other robots were too opaque, making it difficult for learners to figure out what concept the robot embodied. A second challenge we faced was ensuring the sequencing of the opponents and the concepts they embodied was appropriate. Just because a concept is deemed more accessible or does not necessarily mean an opponent embodying that idea will be easier than an opponent demonstrating a more difficult idea.

7. CONCLUSION

Learning from and adapting to an opponent during a competition is a common and effective strategy for achieving victory. As designers of learning environments, there is an opportunity to leverage this practice as a way to engage learners in content by designing games in which the opponents embody the concepts to be taught. In this paper we presented one such game-based learning environment that uses this approach to give learners the opportunity to encounter and employ core computer science ideas. We see the Learn From Your Opponent design strategy as having great potential for producing engaging and effective learning environments that challenge players to learn as they play.

8. REFERENCES

[1] Bransford, J. et al. eds. 2000. *How people learn: Brain, mind, experience, and school*. National Academies Press.

[2] Gee, J.P. 2003. *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan.

[3] Harel, I. and Papert, S. eds. 1991. *Constructionism.* Ablex Publishing.

[4] Holbert, N.R. and Wilensky, U. 2011. FormulaT Racing: Designing a Game for Kinematic Exploration and Computational Thinking. *Proceedings of the 7th Games, Learning, & Society Conference* (Madison, WI, 2011).

[5] Kafai, Y.B. 1994. *Minds in Play: Computer Game Design As A Context for Children's Learning*. Routledge.

[6] Lave, J. and Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge Univ Pr.

[7] Lenhart, A. et al. 2008. Teens, video games and civics. *PEW Internet & American Life Project*. (2008).

[8] Noss, R. et al. 1997. The construction of mathematical meanings: Connecting the visual with the symbolic. *Educational Studies in Mathematics*. 33, 2 (1997), 203–233.

[9] Noss, R. and Hoyles, C. 1996. *Windows on mathematical meanings: Learning cultures and computers*. Springer.

[10] Papert, S. 1980. *Mindstorms: Children, computers, and powerful ideas*. Basic books.

[11] Robertson, J. and Good, J. 2005. Children's narrative development through computer game authoring. *TechTrends*. 49, 5 (2005), 43–59.

[12] Vygotsky, L. 1978. *Mind in society: The development of higher psychological processes*. Harvard Univ Press.

[13] Weintrop, D. et al. 2012. Redefining Constructionist Video Games. *Proceedings of the Constructionism 2012 Conf.* (Athens, Greece, 2012).

[14] Weintrop, D. and Wilensky, U. 2012. RoboBuilder: A Program-to-Play Constructionist Video Game. *Proceedings of the Constructionism 2012 Conf.* (Athens, Greece, 2012).

[15] Weintrop, D. and Wilensky, U. 2013. Supporting Computational Expression: How Novices Use Programming Primitives in Achieving a Computational Goal. *Presented at the Annual Meeting of the American Educational Research Association* (San Francisco, CA, USA, 2013).

[16] Wertsch, J.V. 1991. *Voices of the mind: A sociocultural approach to mediated action*. Harvard University Press.

[17] Wilensky, U. and 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 (2006), 171–209.