VBOT: Collaborative Constructionist Learning using a Virtual Robotics Environment

Matthew Berland & Uri Wilensky
{m-berland,uri}@northwestern.edu
Northwestern University

Paper presented at the 2004 annual meeting of the American Educational Research Association. San Diego, CA.

Abstract

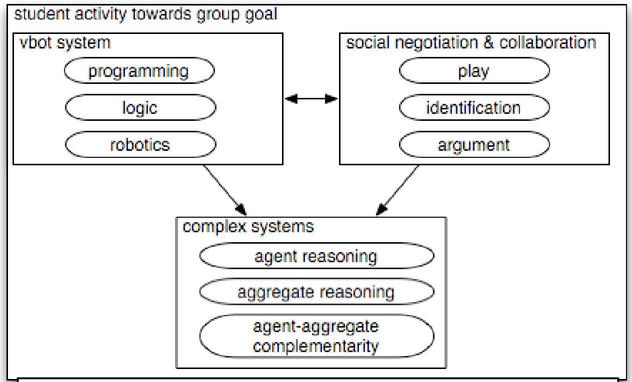
We report on a pilot study of student learning with a group of 9 middle school students using the VBOT system. The VBOT system is a set of "Virtual roBOTics" activities based in the HubNet networked learning architecture. The system and activities are designed for use by students in middle school or early high school. Using VBOT, each student programs his or her own virtual robot that interacts with other virtual robots in a shared environment. Considerable work has been done with robotics in constructionist learning environments in the last twenty years; this study builds on those findings with new insights on the role of virtuality and collaboration in educational robotics. The students build simple circuits with which virtual robotic agents can perform certain collaborative tasks (e.g., flock together near a virtual light source). The group activities were conducted with small groups in informal settings. We analyze the data from these activities and discuss the forms of reasoning that the students used during the activities.

1.0 Introduction

VBOT is an immersive collaborative robotics modeling and programming environment designed for use in middle school and early high school. The VBOT system provides an engaging environment for introducing middle school students to core ideas of both computer science and complex systems theory with a relatively low threshold of entry. VBOT consists of 1) an interface through which students build the control system of their personal virtual robot in real-time, 2) a shared space in which virtual robots exist, and 3) activities for students using the system. The system is built on top of the NetLogo multi-agent modeling environment (Wilensky, 1999) and the HubNet networked learning architecture (Wilensky & Stroup, 1999) used for participatory simulations. In these systems, participants interact in the social space (such as a classroom) and the virtual agents interact in a virtual space (typically on a server computer projected onto a classroom screen). The activities involve students in collaborative reasoning and negotiation about robot control.

VBOT is designed as a constructionist learning environment (see Papert, 1991) that directly addresses the learning of complex systems. By leveraging the kinds of creativity, motivation, and learning results shown in works by Martin (1996a, 1996b) and Papert

(1980), the aim is to help students learn about complex systems and engineering in an authentic, motivating context and social space.



VBOT Activities and Environment

Students were given a set of activities with which to play. In the course of these activities, students argued about, conversed about, programmed, tinkered, and strategized about their virtual robots. They built circuits both independently and in small groups. The students reasoned about complex systems, with complex systems, and as agents in a complex system. In this paper, we present salient examples of this reasoning. We analyze these examples in order to create a preliminary classification of the student's reasoning. This analysis is understood as the beginning of a more substantive and thorough exploration of group collaborative programming.

1.1 Overview of the System

During the course of the activities, students create a set of virtual robots (vbots). A vbot is made up of a set of circuits on a virtual breadboard (see Figure 3, section 1.3) that students can build individually or in groups. Building a vbot through the construction of circuits is an idea inspired by and adapted from Braitenberg (1984). He calls it "circuit intelligence". By "circuit intelligence," we mean that students build reactive "behaviors" for their individual agents through building simple, additive circuits matching input sources, such as light sensors, to outputs, such as virtual motors that drive their vbot around a virtual screen. They can modify these circuits *in situ* to explore them and adapt them to any number of individual, group, or classroom activities, such as simulations of a colony of ants finding food or a swarm of moths around a flame. Following Papert (1991) and Hancock (2001), we call this kind of circuit-modification "tinkering". Through building these vbots, students can come to understand the ways that an agent's behavior in a complex system affects the kinds of global phenomena that emerge.

1.2 How VBOT Works

The VBOT system is built on top of the NetLogo multi-agent modeling environment (Wilensky, 1999) and the HubNet participatory simulation architecture (Wilensky & Stroup, 1999b). A primary design goal of the NetLogo language is to make multi-agent modeling and programming accessible to a wide audience without sacrificing the ability to make detailed scientific models of complex systems. HubNet is a system built on top of NetLogo called a "participatory simulation environment" in which individual participants can each control an agent ("turtle") in a NetLogo model. For instance, in one HubNet model, Gridlock (Wilensky & Stroup, 1999a), each student controls a stoplight on a simulated city traffic grid. Each stoplight is an agent in a complex system from which the flow of traffic in the city emerges. The complex system here is one in which cars, lights, and roads interact to create a dynamic equilibrium. Individual students each have a set of controls, such as a button that switches their stoplight on the virtual screen between green and red.

Building upon the HubNet and NetLogo framework, the VBOT system enables each student to control one agent in a model, but instead of controlling the agent directly, as one might do in a HubNet simulation, or controlling the agents through general rules for classes of agents, as one might do in a NetLogo model, each participant sets rules for his or her own agent by building a simple circuit. Building these circuits is what we call "programming" in the VBOT system. VBOT includes a separate graphical interface in which the participant builds this circuit, and uses this as his/her interface to the NetLogo model (see Figure 3 in section 1.3 below).

1.3 One VBOT run

In the panels below, we show a short excerpt from a videotape of a classroom enactment in which students are using the VBOT system. The figures on the right accompany the text on the left. This section is provided as an overview of the process of working with VBOT.

The facilitator gives out a task, rule or game. In this case, "everybody flock in the middle." This is a photo of the group listening to the facilitator describing the task.	 **Figure 1**

This is a picture of the group screen. Each student is represented by a blue agent on the common background. The color of each background square represents the amount of "light" in the environment.

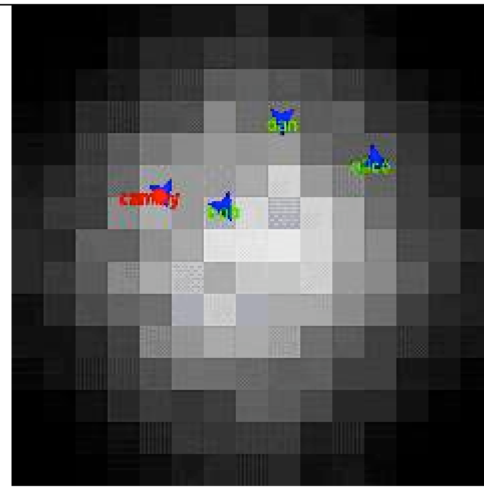


Figure 2

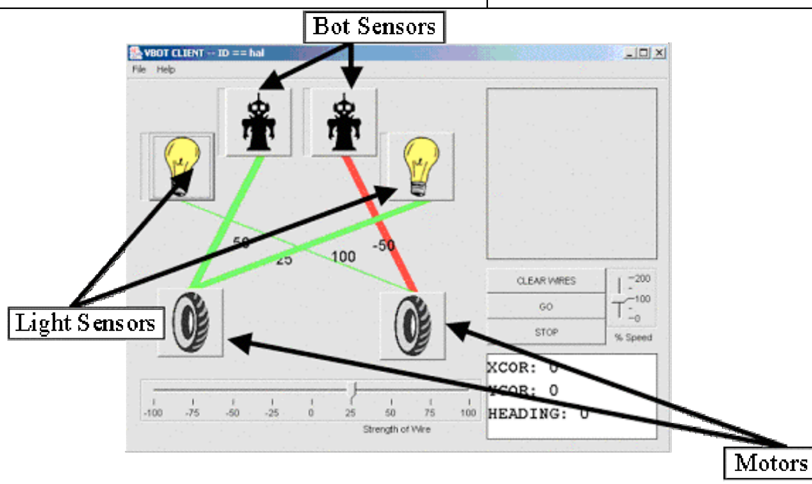


Figure 3

Each student controls his or her own VBOT circuit interface. The "bot sensors" each have a gauge reading the density of other vbots nearby on the left-front or right-front of the individual's vbot. The "light sensor" has a gauge reading the amount of "light" -- the lightness of the color of the patch-square to the front-left and front-right of the vbot. The motors drive the vbot around the NetLogo "screen" shown in Figure 2, changing both it's the vbot's speed and its heading.

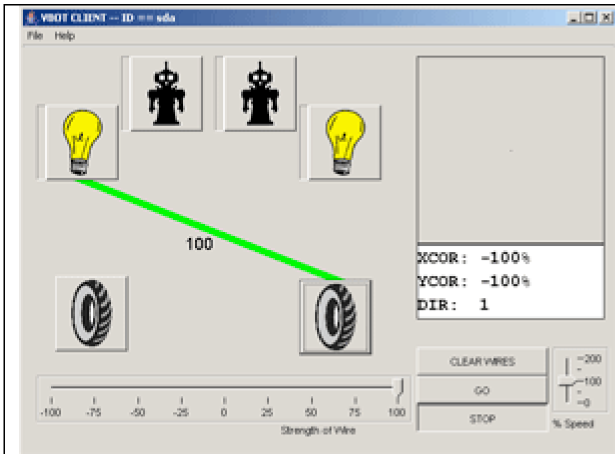


Figure 4

In Figure 4 at left, the left light sensor is connected to the right motor at 100%. This is represented by the bright green line with the number 100 next to it. In this scenario, an agent receiving light on its front-left would turn left, as light would cause the back right wheel to move more quickly. As the amount of light that the left light sensor increases, so does the speed of the back right wheel.

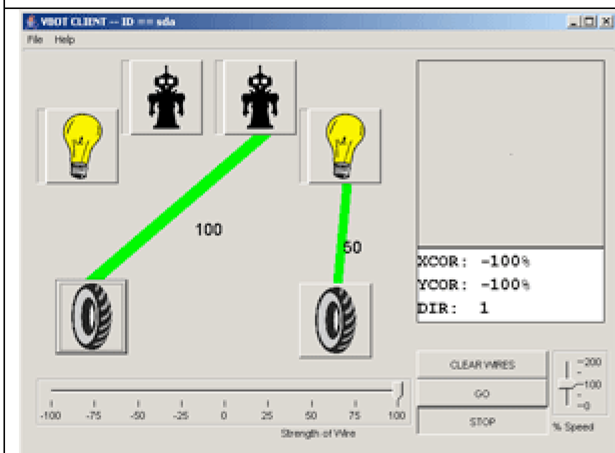


Figure 5

In Figure 5 at left, the right bot sensor is connected to the left motor and the right light sensor is connected to the right motor. In the case, as the agent sees more bots (fellow students' agents) on his right, it will turn left, and as it sees more light on the right, it will turn right. Note that the right light sensor is at 50% meaning that this sensor is only half as sensitive to light as the sensor in Figure 4, and, hence, given with same amount of light in the same place, would only drive the motor with half of the speed.

Each student then creates his/her own circuit to fulfill the given task. The circuits she creates will drive her vbot around the common screen. She can change this circuit while the activity is running.

Figure 6

Discussion happens alongside, and informs, the building of circuits.	 **Figure 7**

2.0 Design Rationale

Both programming and complex systems theory are widely understood to be hard to learn (for complex systems, see Mandinach & Thorpe, 1987; for programming, see Pea, 1987). However, as several studies have shown us, students can learn the methods and comcepts of complex systems theory as early as middle school (Wilensky, 1997; 1999; Wilensky & Resnick, 1999; Ioannidou, Rader, Repenning, Lewis, & Cherry, 2003; Centola, McKenzie, & Wilensky, 2000; Resnick, 1994), and students can come to understand central theses of programming and computer science as early as grade school (Papert, 1980; Harel & Papert, 1990). VBOT is designed to create a space in which middle school students can learn complex systems theory, computer science, and programming in an motivating collaborative space.

2.1 Design Challenges

2.1.1 Computer Fluency

Computer technology affects every aspect of our daily lives, but in many schools, it is less likely that students will be taught the basic building blocks of computers and how to author their own software today than it was 15 years ago (Harvey, 1994). VBOT is designed in part to foster computer fluency through collaborative engineering and programming. Computer literacy (Papert, 1980; diSessa, 2000) implies both the ability to use computer software and the ability to create and manipulate computer software (or hardware) to communicate and disseminate ideas. This definition of literacy parallels the generally understood meaning of print literacy. However, the term computer literacy has often been used to describe a very impoverished form of literacy in which the learners learn only how to run a few standard computer applications. To be considered print literate, one needs to be able to express oneself in writing. A similar facility is a desideratum for computer literacy. To distinguish this richer form of computer literacy from the more conventional, we follow Papert (1980) and refer to it as computer fluency. The expressive and authoring aspects of computer fluency are largely ignored in the general pre-collegiate curriculum. We argue that such computer fluency is essential for 21st century citizens as computers shape so much of our interaction and communication.

2.1.2 Complex Systems Fluency

Complex systems research has become increasingly important for understanding of scientific phenomena in general (Holland, 1995; Wolfram, 2002; Wilensky & Resnick, 1999). Scientists are using complex systems methods to model physics, social networks (Watts, 2003), and biological processes (Bar-Yam, 1997). Some have argued that complex systems theory is a new kind of science, one posed to usurp the mantle of scientific explanation from traditional equation-based science (Wolfram, 2002). Others have shown that modeling with complex systems is more comprehensible to high school students than traditional equation-based science (Wilensky, 1997; 1999; Wilensky & Resnick, 1999; Ioannidou, Rader, Repenning, Lewis, & Cherry, 2003; Centola, McKenzie, & Wilensky, 2000; Resnick, 1994). However, the generally applicable methods and theories of complex systems are still absent in school curricula. Why is it that these ideas are not more present? In informal conversations with teachers and principals, we have heard two reasons: they believe that these concepts are too hard for students to grasp and they think the cost of entry (training, resources) to introduce these concepts into school curricula must be prohibitively high. The findings in this study suggest that both of those assumptions are unfounded.

2.1.3 Play and Collaboration

Traditional school curricula rarely involve significant "play" time. As we endeavor to raise standards for teaching and learning in schools we have paid much attention to content standards. A neglected area in this effort has been the source of student motivation. However, a considerable body of research has shown that play can be a powerful motivator (Papert, 1980; Vygotsky, 1978; Dewey, 1913; Kafai, 1994). There is convincing evidence from both the social and cognitive streams of learning research that learning and transfer are more easily achieved when the students are motivated to work through activities (e.g., Dweck & Elliot, 1983; Ames & Archer, 1988; Pintrich & Schunk, 1996). Schank and Cleary (1994) show how intrinsic motivation can lead to more personally relevant, stable knowledge acquisition for many students.

Furthermore, a variety of studies have shown that enabling natural social interactions leads to cognitive and social phenomena (Vygotsky, 1978; Gutierrez, Rymes, and Larson, 1995). Vygotsky (1978) described the Zone of Proximal Development (ZPD) – that is, the level at which children can function in a social-help setting as opposed to an individualized one. The VBOT system and activities were designed by encouraging students to speak freely about how they solve problems during VBOT activities. Gutierrez et al. (1995) show that often the most productive and thoughtful interactions occur in informal spaces within the classroom, outside the direct view and control of the teacher. In the VBOT activities, we sought to create un-censored, informal spaces in the activity in which students could speak freely.

2.2 Design Analysis

The VBOT project draws on educational research on participatory simulations and complex systems (e.g., Wilensky & Stroup, 1999; Wilensky & Resnick, 1998); constructionism (e.g., Papert, 1980; Harel & Papert, 1990); and robotics (e.g., Martin, 1996a; Resnick & Ocko, 1991; Hancock, 2003). Whereas work has been done in each of these areas, there has been no work to this point that brings these four threads together as such. Many projects (e.g., Shaw, 1996) have taught computer fluency with issues of social literacy. Some of the educational robotics literature analyzes social interaction (e.g., Hancock, 2003). The literature on participatory simulations emphasizes social interaction and complex systems. However, there have been very few attempts, as yet, to teach complex systems through social, distributed programming environments like the VBOT system. We argue that by engaging the students through collaborative engineering, allowing the students to create their agents' behaviors together in the context of an emerging system, students will learn complex systems theory and methods organically. We hope to leverage these different gains, and see novel gains through the interface of social programming and complex systems.

There are several studies on programming with high school students. We use Hancock's Flogo system (2001, 2003) as a reference example of a constructionist robotics project that uses the circuit metaphor in programming. The Flogo system involves students working individually to create complicated circuits that are uploaded onto physical robotics systems such as LEGO Mindstorms (Martin, 1996b; Resnick & Ocko, 1991). Furthermore, there have been a wide variety of both systems and studies involving simple visual programming, ranging from those designed for the very young, such as ToonTalk (Kahn, 1996), to those designed as full featured simulation languages, such as AgentSheets (Repenning, 1993). However, the majority of these studies involve students programming individually. Indeed, there is very little previous work on students doing collaborative programming. Our work is part of a growing body of research on social programming (e.g., Bruckman, 1994; DeBonte, 1998). This study differs from some of the other recent research using the HubNet architecture (Wilensky & Stroup, 2004; Levy & Wilensky, 2004; Abrahamson & Wilensky, 2004) as it involved students programming together contemporaneously. They do so in a real-time social environment in which they would both compete and collaborate, whereas most participatory simulations to date have focused more on students' direct control of agents acting in complex environments and less on student programming.

From these studies, we have both gained an understanding of the relationship between the design of student projects and student reasoning using their design. Our design follows systems such as Martin's (1996b), Resnick & Ocko's (1991), and Hancock's (2003) above in analyzing the ways that students learn while creating autonomous agents. These studies clearly show the value of students constructing autonomous agents. Our direction stems from Wilensky & Stroup's work with participatory simulations (1999b) and uses the results of that work towards understanding learning in the content domain. Bruckman's analysis of her MOOSE Crossing system (1994) informed us in our design of the common social space for agent interaction.

3.0 Method

3.1 Participants

Our sample was a self-selected group of students from two schools, one public middle school and one religiously-affiliated middle school. The students were not reimbursed. The sample consisted of 3 girls and 6 boys (n = 9). All 6 boys and 2 of the girls were either 13 or 14 years old; one girl (Sally[1]) was 11 years old. This group was gathered in an informal setting, as this study was intended as a pilot test for the first generation of the VBOT system.

3.2 Procedure

The first author primarily facilitated the activities, with help from 3 colleagues. The classroom was arranged in a "V-shape" with students pointed at center-front of the room such that they could see one another, the screen in the middle of the classroom, and their individual computers.

The pilot ran around 90 minutes, with the first author giving group tasks, the students finding paths through those tasks, and the students discussing their approaches to the material.

All activities were videotaped using two handheld cameras. One camera was focused primarily on group interaction by videotaping the classroom as a whole, while the other cameras focused on specific student strategies and interactions. We also conducted pre- and post-interviews, but they will not be discussed in this paper.

3.4 Methods of Analysis

Our analysis builds on qualitative studies such as Cole's investigations with the 5th Dimension project (1996) and Hancock's paper on student learning with Flogo (2001). Both of their studies involved informal, unscripted activities for small groups of students in settings conducive to student discourse. Both of these studies used student discourse about technical activity as a unit of analysis. We captured this discourse, along with the student use of the system, on 2 digital video cameras. Our goal was to focus the camera on the instances where student discussion about the VBOT system informed student action with the system. This included all students in the study.

First we created a list of the relevant forms of reasoning from the prior research in the component domains. We then culled every segment in our data set at which significant student talk occurred. Our preliminary analysis showed that those segments corresponded well to the list of forms of reasoning. With this understanding, we matched transcript

[1] All names in the paper and excerpts (Appendix A) have been changed.

excerpts with forms of reasoning. Excerpts that show such student reasoning are presented in section 5.0 below.

4.0 The flocking activity

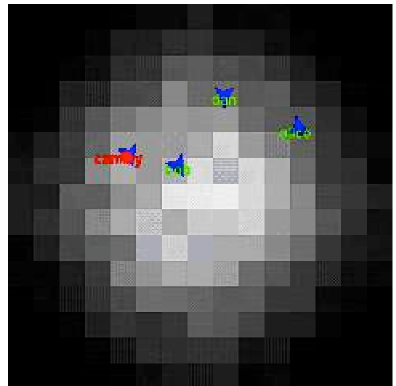


Figure 7

We present one example of a class activity with VBOT. This example is intended as a story of the process and execution of a single VBOT activity: the flocking activity.

Every activity begins with an instruction or goal given by the facilitator. In this activity, the first instruction given to the group was "everybody move to the middle." This is the first step of what we call the “flocking” activity, because the students’ vbots[2] “flock” together like birds. During this first group activity, each student had his or her own circuit screen (see Figure 7), which connects to a vbot on the shared, projected NetLogo screen (graphics window, Figure 6). Specifically, each student controlled how his/her vbot moved on the stage by building a circuit while his/her vbot is moving and progressively changed the circuit over the course of the activity.

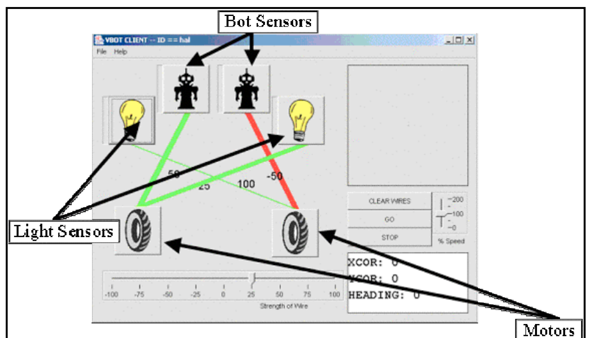


Figure 7

At this point, every student built a circuit to instruct his/her vbot to move to the middle of the screen. The middle of the screen is marked by a "light source" which radiates “light” outward from white to black. In this case, a correct[3] circuit is two crossed green (positive) wires connecting each light sensor to its opposite-side motor, each wire being at 100% (see Figure 7 for the circuit diagram). This means that if the maximal amount of light is received in the right light sensor, then the motor on the left will be activated at 100%. As the circuit uses crossed wires and is symmetrical, the same will be true of the relationship between the left light sensor and the right motor. I use these terms (such as 'sensor' and 'motor') to be evocative of robotics terms. Most students built this circuit, with some notable exceptions. One variation on this circuit involved uncrossed -100% negative, red (inhibitive feed-forward) wires, which will result in the vbot slowing down and circling the center of the stage.[4]

[2] A vbot (lowercase) is an individual student’s agent on the shared screen (see Figure 2).
[3] There are many different ways that a circuit can be correct, as there are many different circuits which could result in a system meeting given constraints.
[4] The -100% negative uncrossed circuit would mean that if the maximal amount of light is received in the right light sensor, it will totally inhibit the movement of the right motor. The same will be true of the relationship between the left light sensor and the left motor.

While this activity was fairly simple in both goal and process, it generated a wide range of discussion about the types of circuits to build. One example of a particularly interesting discussion involved the question of whether using robot sensors (sensors which read the density of other vbots in the vicinity) are helpful in the task (transcript to the right).

The students modified their circuits while they were discussing them in order to generate better results for the group. There were several students in this case who did not initially build circuits that brought their vbot to the light in a dynamic equilibrium. One student had a different idea to let his vbot wander randomly[5] until it found the middle then built a circuit that turned the vbot in tight circles around the light.[6]

The next instruction given was: "everybody flock together outside the light." In this case, the students should create a stable flock of vbots. Since most of them were already attracted to light from the first task, they were mostly bunched together in the middle of the screen. This provided some difficulty for the students, as they could tell that they needed a circuit that avoided light but was attracted to other vbots. However, if they simply created a circuit in which the light sensor from the same side (e.g., left light to left motor) and the robot sensor from the opposite side (e.g., left robot to right motor) were summed in each motor, a straight line or a very tight turn would often result. Vbots with this circuit mostly stayed near the middle (and, hence, near the light source) or wandered without reference to the task at hand. This led to a class discussion. Some students argued that a leader should leave the light, after which everybody else would be attracted to bots and not light. Some portion of the class tried this. However, they quickly found that the followers were simply heading back towards the clump in the middle, where a higher proportion of vbots were to be found. Another student suggested that everyone clear all of their wires to spread out, after which they could start looking for other vbots and against light again. Another suggestion was to wire up a circuit such that the attraction to bots was strong and the repulsion from light was weak. The last suggestion was similar, but instead of a strong attraction to other vbots, there was a weak one. As different students used their individual strategies, a flock was formed outside of the light. Then they discussed why certain strategies worked or did not work. We analyze their forms of reasoning in making these strategies in section 5.0 below.

5.0 Forms of Reasoning

Through the explanations and accompanying excerpts, we identify the existence of various forms of reasoning generated by the students in using the VBOT system and activities. We have broken these forms of reasoning into four overarching categories: complex systems, computer science, artificial intelligence/cybernetics, and identification.

---

[5] It wanders in a jagged line when no wires are connecting sensors to motors.
[6] This particular circuit was one wire, connecting the right light sensor to the right motor. The left motor was stopped, the right motor was at near 100%, so the vbot turned in a tight circle.

We have included the excerpts herein. All relevant excerpts can also be found in Appendix A. Time-codes are in the form hh:mm:ss:xx (where xx is hundredths of a second). Excerpts are temporally numbered.

We provide these forms of reasoning as a typology of the students' learning. These forms of reasoning are examples of the process of working through our component domains. For example, computer programming uses branching logic, and one must understand and utilize branching logic to program computers. We describe branching logic as it is defined in the VBOT activities and excerpt a quote from the transcript of a student using branching logic to work with her vbot. We are providing a set of examples that show the existence and type of reasoning that one can reasonably expect to see when working with the VBOT system. What we call "forms of reasoning" are identifiable categories of reasoning that the students used to describe a challenge that arose in using the VBOT system and activities.

Form of Reasoning	**VBOT Example**
5.1 Reasoning about Complex Systems	
5.1.1 Reasoning about the effect of individual agents on a system	Students try to figure out how the introduction of new vbots will affect a stable flock.
5.1.2 Reasoning about individual agents working within a system	A student tries to create a new circuit that will stabilize the flock of which her vbot is a member.
5.1.3 Reasoning about decentralization of control	Students argue about how the leadership roles in a flock might be distributed.
5.1.4 Agent-Aggregate complementarity	Students argue about the relationship between the flock and the agents.
5.2 Reasoning with Algorithmic Logic	
5.2.1 Reasoning using branching logic	Students differentiate between sets of discrete rules under pre-defined conditions.
5.2.2 Reasoning about circuits and electronics	Students must construct vbot circuits to successfully play the game.
5.3 Reasoning and Artificial Intelligence	
5.3.1 Reasoning using systemic noise	A student tries to understand how the randomness of the environmental noise affects her vbot's behavior.
5.3.2 Reasoning about sensing and knowing	Students discuss the difference between what their vbots "know" and what they "sense."
5.3.3 Reasoning about feedback	Students discuss how the instability of a discrete evaluation of a vbot's circuit results in overall in/stabilities in the vbot's behavior over time.
5.4 Reasoning Through Identification	
5.4.1 Body syntonicity	A student uses his arms and legs to interpret the movement of the vbot.
5.4.2 Character identification	Students are motivated by their identification with their vbot avatars.

5.1 Reasoning about Complex systems

We selected these different forms of reasoning about complex systems from work in the field, specifically Holland's (1995) and Wilensky & Resnick's (1999), in which learners and experts engage when discussing and using complex systems, then we codified and refined those for this context, and we found examples of those forms of reasoning in the given data.

5.1.1 Form: Reasoning about the effect of individual agents on a system

The students reasoned about the ways that many different sets of rules could affect the whole system. In the flocking activity described above, all of the students were reasoning about how to create a stable flock, and they eventually did so (and realized this) with a diversity of rules. By giving rules at the system-level, we were attempting to enable the students' intuitions about how the rules affected the evolution of the patterns.

Excerpt 6	
00:38:01:36 Daniel	Zip on robots because right now all the robots are clumped to his leg, so instead of moving away, they're all staying…
00:38:08:84 Arnold	I'm doing fifty on the robots and look at where I am.
00:38:11:81 Daniel	Because you're Schwarzenegger, there's something wrong with you… Terminator knows what's going on here… One at a time, everybody has to start following the robots because if you do it all at the same time, we'll all clump together somewhere.

Example: In Excerpt 6, Daniel says, "everybody has to start following the robots [at different times]." He is trying to enable the formation of a flock outside the light. He believes that only if individual agents leave in temporal succession can the flock be preserved without drawing back into itself those explorers that wish to leave the light.

5.1.2 Form: Reasoning about individual agents working within a system

Throughout the activity, students reasoned about how their individual vbot (and, hence, circuit) affected the system as a whole. To create a flock, students have to think about how their behavior related to the behavior of the flock and about how that behavior of the flock affected their own circuit. When they were trying to repulse the flock from the light, each student dealt with how his/her individual vbot affected this change.

Excerpt 7	
00:38:56:17 Jenny	Um… Only one person goes away from the light and everyone follows that person because right now nobody knows where they're going.
00:39:05:19 R: Dor[7]	Uh huh, one at a time you want.
00:39:07:00 Jenny	Yeah.
00:39:08:00 R: Dor	Why why is it important one at a time?
00:39:09:00 Jenny	Because everyone's clumped together in the middle now…
00:39:11:00 R: Dor	Uh huh…
00:39:12:89 Jenny	…and their motors are to go towards the other robots is more than to go away from the light so since there are so many robots, they're following the robots and nobody is moving away so… no one ELSE is moving away.

Example: In Excerpt 7, Jenny has to reason about the ways that the other vbots affect her ability to sense the world and how the vbots sense each other. In doing so, she must separate how the various scenarios of her compatriots' movements affect the robot sensor. As she tries to enable the formation of a flock outside the light in the center, she realizes that "since there are so many robots… nobody else is moving away." Using the robot sensor to attract oneself to the other robots will inevitably lead one back into the center of the group, as there are more robots in the center. As a result of needing to consider other robots to make a flock outside the light and yet not being able to simply use the robot sensor as an attractor, she decides, "nobody knows where they're going." The individual agents are generally following their own rules, but it creates a system in which people have trouble leaving the light but staying in a group.

[7] R: Matthew is "Researcher: Matthew Berland." R: Dor is "Researcher: Dor Abrahamson."

5.1.3 Form: Reasoning about decentralization of control

It was tempting for students in the study to think about these robots as needing a central leader, but repeatedly throughout the activities, students needed little recourse to leaders and, in fact, often did noticeably better without them. As shown in Wilensky and Resnick (1999), people often default to a deterministic-centralized leadership mindset even when there is no reason to do so. There is no clear leader in any VBOT activity, unless the students choose to elect one. They played all of the games without any clear leaders, and, generally, came up with mutually informing, but different, answers.

Excerpt 8	
00:41:13:35 Daniel	I know now there's a bunch of people in a clump in the top left corner if you are in that clump, stop moving but only if you're in that clump, because if everybody up there stops moving...

Example: In Excerpt 8, Daniel discusses the finer points of decentralization. When asked whether everyone should work at once, Daniel replies that people should follow his lead. They do not, however, follow his lead, but they do solve the problem. In trying to centralize control in himself, he is confronted with the difficulty of doing so in a robust way. When the flock starts to form outside the light without his direct control, he no longer tries to control his fellow students' behaviors.

5.1.4 Form: Agent-Aggregate complementarity

Agent-Aggregate complementarity (Wilensky & Stroup, 1999; Stroup & Wilensky, 2003) is a term we use to describe the enabling of thinking between "levels" (Wilensky and Resnick, 1999) of agent and aggregate representation. The students use their circuits to think about the behavior of their own vbots, but it can be difficult to transition from thinking about their own vbots to thinking about how the system of vbots works as a result of the individual behaviors. The complementarity between the levels of discourse, that of the agent and the system here, are enabled by the inherent social goals in the activity. Every agent follows the rules of his/her builder, but together they often form coherent systems.

Example: Daniel uses the VBOT system to reason about how his individual actions affect the system, and how the system affects his individual interactions. He says, "One at a time, everybody has to start following the robots because if you do it all at the same time, we'll all clump together somewhere." To come to this conclusion, he has had to think both about how the system affects the actions of the individual vbots (other than his own), and how the actions of individual vbots other than his own affect the system. He is working between the levels of agent behavior (the command: "start following the robots") and aggregate patterns (the movement of the flock) to accomplish his goal.

5.2 Reasoning with Algorithmic Logic

Algorithmic logic is a central facet of computer science. Students used forms of algorithmic logic to discuss and describe both their circuits and their strategies. Both branching logic and hardware circuits are central to algorithmic logic (Harvey, 1994).

5.2.1 Form: Reasoning using branching logic

Although there are no written "if-then" statements in a VBOT circuit, there are many decision points for the student at which s/he needs to make if-then statements. Indeed, the students had to use branching logic and logical forms fairly routinely to generate behaviors.

Example: Daniel thinking about all of the different behavioral possibilities given different inputs. He says, "OK OK OK. So if it sees light on the left, it sees light on the right, I want it to turn right so this should just… but if it sees light on the left then it turns a little, " while acting out the movement (see Appendix B for images of this movement). He has isolated distinct behavioral patterns for his vbot, and he is describing a conditional logic about those patterns.

5.2.2 Form: Reasoning about circuits and electronics

Circuit building is an engineering task that has fairly immediate real-world benefits. People with knowledge of the workings of electronics circuits can better understand the technological world around them, begin to understand how to fix electronics, get jobs, and understand modern mathematics and computers science more easily. Circuits are the structural building block of all electronics. By showing that kids can learn to understand circuitry fairly easily, we can show that it can be a helpful cognitive building block as well.

Example: Circuits are inherent to any control in a VBOT activity. The students' articulation on these circuits ranges widely in its degree of formality. Sally states her vbots' behavior informally: "Because if it sees it over here… because when it sees it, it'll turn the way that it sees the light." On the other hand, Daniel described the behavior of his vbot and circuit with more formal language: "Since it sees light on the left, it turns right, so it cancelled it out into a general back and forth action until it got towards the source of the light." Daniel is describing a circuit which is exhibiting a common robotics navigation issue called "chattering."

5.3 Reasoning and Artificial Intelligence (AI) and Cybernetics

Students used reasoning much like that found in the robotics and cybernetics community. By exposing the students to thinking in terms of sensing and knowing, we provide them with a basis on which to understand the future of computers. We draw these forms of reasoning from work by Brooks (1999).

5.3.1 Form: Reasoning using systemic noise

Students had to reason with the noise inherent in the system. In this system, the robots act much like the vehicles in (Braitenberg, 1984), but they exist in a noisy environment. The updating is not continuous, the light fluctuates, and the sensors do not always read perfectly. Once you happen upon a good circuit, it will not always work. This is inherent to robotics in general. Students learned to use the noise and accept the "wiggle" and use it to stabilize or transition the system.

Example: At about 16 minutes into the activity, Sally begins to realize that the circuits that she builds are not necessarily direct instructions ("It always turns towards that light. Now its going all squiggly in the center…"). While she was initially a little disturbed by the vbot "going all squiggly," she realizes that this is normal and uses it in her understanding.

5.3.2 Form: Reasoning about sensing and knowing

Robotics research itself underwent an epistemological break in the 1980's -- "soft" versus "hard" (Brooks, 1994; Brooks and Stein, 1994). The adherents of the "hard" philosophy (which was more classically acceptable) thought about robotics in terms of maximizing navigation equations and planning out the most correct route through a maze. The adherents of "soft" robotics believed in dealing with information on a need-to-know sensory basis. The use of sensors as the primary means of navigation is inherent to VBOT. The students independently reasoned using the distinction between what the vbot knew (e.g., how to move) and what it sensed (e.g., light).

5.3.3 Form: Reasoning about feedback

It was easier to tell that the control was not constant -- there was lag, jumpy movement, and update loops. The students figured out pretty quickly that there was "feedback" in the system. If they turned toward the light once, they would turn fast the next time, because there was more light coming into the sensors. Indeed, on a straight path to the light, they experienced what robotics researchers call "chattering." Chattering is the back-and-forth consistent overshoot of the target goal.

Excerpt 3	
00:20:59:04 Daniel	Since it sees light on the left, it turns right, so it cancelled it out into a general back and forth action until it got towards the source of the light. I still don't get it though.
00:21:23:00 Sally	This is going to be really weird… I go both ways at once whenever I see a robot.

Example: In Excerpt 3, Daniel figures out that the quickest way to get to the goal is by a consistent over-shooting of the direction towards the light. Sally was confused by the instability of the behavior of her vbot in certain circumstances, and she consequently created a circuit to minimize this instability.

5.4 Reasoning Through Identification

Papert (1991) describes construction as an identifying event both for individuals and groups. By encouraging identification with these technological objects, we make them more accessible to people who don't generally consider themselves computer fluent (see Turkle, 1995). We draw these forms of reasoning from Papert (1980).

5.4.1 Form: Body syntonicity[8]

Throughout the activity, we returned to one central movement metaphor -- that of swimming. The movement of the vbot is much like swimming with one's arms in front of oneself. The left arm are the left sensors, the left leg is the left motor. A crossed-wire positive setup means the more your right hand senses light, the more your left leg paddles, moving you to the right. The students could identify with this, and, indeed, used the metaphor unprompted when they were trying to figure out how build their wires. **Example**: Appendix B shows Daniel working through the problem by simulating his vbot with his own "swimming" body.

5.4.2 Form: Character identification

The students could identify with their vbot as a reflection, a servant, or a character. There was a strange co-occurrence of both "he is going left" and "I am going left" during the activity. Students either chose to "be the vbot" or "drive the vbot," but in both cases, they were trying to identify the vbot as their own as opposed to as themselves. There is also a rich literature in avatar-as-character (Turkle, 1995). The students could have their character act differently than they might, they can try out new circuit-personalities, and they can be someone else.

Excerpt 4	
00:32:59:00 Sally	Go towards the light and towards everybody else.
00:33:02:00 R: Matthew	So that's one suggestion… go towards the light and towards everybody else… everybody try to get to the center.
00:33:12:88 Arnold	Guys make all your robot wires 25 or 50 so that you're more towards the light and you'll follow someone else with… less strength.
00:33:22:93 Arnold	You see I'm going towards the center… look where Schwarzenegger is OK? He's towards the center… WOW.
00:33:36:00 Daniel	Terminator (Daniel's vbot) is confused!
00:33:36:27 Arnold	Schwarzenegger (Arnold's vbot) is NOT!
00:33:41:88 Jenny	I'm just further away than you…
00:33:41:93 R: Matthew	What were you saying?

[8] A term used by Papert (1980) to describe understanding virtual movement by thinking through it with one's own body.

00:34:03:35 Daniel	Abe's lost!
00:34:04:00 Arnold	Like usual! It's like Ms. W's class.
00:34:05:00 Abe	((laughs))
00:34:12:43 Jenny	(to Arnold) It's like… You said all the girls would get all the problems, but all the boys got all the problems! I love it! YES!

Example: Excerpt 4 shows Daniel and Arnold both taking credit and shifting blame for their respective vbots. Daniel's vbot ("Terminator") is "confused." Daniel does not see this as any personal problem and consequently changes his circuit. Arnold appears to be quite proud of the accomplishments of his vbot ("Schwarzenegger"). Jenny and Abe have more personal connections with their creations. Jenny says "I'm just further away than you…" Daniel then associates Abe with his vbot, saying that Abe's vbot is lost, much like Abe is in "Ms. W's class" to Abe's amusement.

6.0 Conclusion and Future Directions

We have presented the VBOT system, and we have proposed that this system is a new way to think about teaching kids about engineering and complex systems topics in a motivating, social environment. The system and activities scaffolded student reasoning in these domains. We gave examples and explanations of some forms of reasoning with which students navigated the activity. Students made connections between engineering, social, and complex systems topics. They were motivated to do so, and they did so freely.

This paper engages the challenges in teaching complex systems fluency and computer fluency using collaborative engineering of virtual robots. During the activity, students reasoned about complicated topics in complex systems and computer science. For example, students discussed the effects of individual agents on an emerging system by acting as agents within that system (Sec. 5.1.1) while also reasoning about the decentralization of control in the system (Sec. 5.1.3). They could use the building of their individual vbots to understand typical robotics issues like systemic noise (Sec. 5.3.1) while building several simple circuits (Sec. 5.2.2). They collaborated as a group and their collaboration led to different understandings of the material (shown in Excerpts 3, 4, 11, 12).

Learning systems such as VBOT represent new directions in thinking about programming in the classroom. By creating directed, motivating, open, and creative environments, we can help learners think more deeply about fairly complex ideas. The VBOT system is a jumping off point for similar technologies that can leverage the worlds of social video games and social information technologies (such as instant messaging) towards helping people think differently about complex systems and programming in the classroom. The most notable aspect of the VBOT project is that the students are both programming and collaboratively strategizing. The students are socially and virtually building social, virtual robots. The classroom interaction and activities contribute to the discourse on complex systems with the actual VBOT system. We note this to make a key point: this is not simply a paper about students learning to program for the first time in a new virtual environment. This is also a paper about students collaborating to build strategies using a virtual environment. While this is only the beginning of a set of studies on collaborative programming towards complex systems and computer fluencies, our analysis has shown that this system can foster the forms of reasoning important to building these strategies in targeted domains.

7.0 Acknowledgements

We wish to express our gratitude to everyone at The Center for Connected Learning and Computer-Based Modeling at Northwestern University, who supported us in the implementation of VBOT, in collecting data, in understanding the data, and in writing this paper. Special thanks go to Dor Abrahamson, Sharona T. Levy, Joshua W. Unterman, and Seth Tisue. A very special thank you goes to all of the student volunteers and their parents. The work herein was supported by NSF Grant REC-0126227.

References

Ames, C., & Archer, J. (1988). Achievement goals in the classroom: Student learning strategies and motivation processes. *Journal of Educational Psychology, 80,* 260-267.

Bar-Yam, Y. (1997). *Dynamics of Complex Systems.* Perseus Press.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MIT Press.

Brooks, R. (1999). *Cambrian Intelligence*. The MIT Press: Cambridge, MA.

Bruckman, A. (1994). *MOOSE crossing: Creating a learning culture*. Thesis for the degree of Doctor of Philosophy at MIT.

Centola D., McKenzie E., & Wilensky, U. (2000). Survival of the Groupiest: Facilitating Students' Understanding of Multi-level Evolution through Multi-Agent Modeling - The EACH Project. *The Fourth International Conference on Complex Systems*. Nashua, NH: New England Complex Systems Institute.

Cole, M. (1996). *Cultural psychology: A once and future discipline*. Cambridge: Harvard University Press.

DeBonte, A. (1998) *Pet Park: a graphical constructionist community*. MS Thesis. MIT, Cambridge, MA.

Dewey, J. (1913). *Interest and effort in education*. Boston : Houghton Mifflin Co.

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

Dweck, C.S., & Elliot, E.S. (1983). Achievement motivation. In P. H. Mussen & E. M. Hetherington (Eds.), *Handbook of child psychology* (Volume IV: Social and personality development, pp. 643-691). New York: Wiley.

Edelson, D.C. (2002). Design research: What we learn when we engage in design. *Journal of the Learning Sciences*, *(11)*1, 105-122.

Ericsson, K.A. & Simon, H.A. (1984). *Protocol Analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Gutierrez, K., Rymes, B., & Larson, J. (1995). Script, Counterscript, and Underlife in the Classroom: James Brown versus Brown v. Board of Education. *Harvard Educational Review, 65*(3), 445-471.

Hancock, C. (2001). Children's Understanding of Process in the Construction of Robot Behaviors. Paper presented at the symposium "Varieties of Programming Experience." *The Annual Meeting of the American Educational Research Association*, 2001, Seattle.

Hancock, C. (2003). *Real-time programming and the big ideas of computational literacy*. Unpublished doctoral dissertation, MIT.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 132.

Harvey, B. (1994). “Is Programming Obsolete?” [online] http://www.cs.berkeley.edu/~bh/obsolete.html

Holland, J. (1995). *Hidden Order: How Adaptation Builds Complexity.* Reading, Massachusetts: Perseus Books.

Ioannidou, A., Rader, C., Repenning, A., Lewis, C., & Cherry, G. (2003). Making Constructionism Work in the Classroom. *International Journal of Computers for Mathematical Learning*, 8(1), 63-108.

Kafai, Y. (1994). *Minds in play*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kahn, K. (1996). ToonTalk - An animated programming environment for children. *Journal of Visual Languages and Computing 7*(2): 197-217.

Kauffman, S. (1995). *At home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford: Oxford University Press.

Mandinach, E. B., & Thorpe, E. T. (1987*). The systems thinking and curriculum innovation project: Technical report, part 1. (TR-87)*. Educational Technology Center. Harvard Graduate School of Education. Nichols House, Appian Way. Cambridge MA 02138.

Martin, F. (1996a). Ideal and Real Systems: A Study of Notions of Control in Undergraduates Who Design Robots. In Y. Kafai, M. Resnick (Eds.), *Constructionism in Practice*. Mahwah, NJ: Lawrence Erlbaum.

Martin, F. (1996b), "Kids Learning Engineering Science using LEGO and the Programmable Brick," in the Proceedings of the Annual Meeting of the American Educational Research Association, April, 1996.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York, Basic Books.

Papert, S. (1991). Situating Constructionism. *Constructionism*. I. Harel and S. Papert. Norwood, NJ, Ablex.

Pea, R. D. (1987). Programming and problem-solving: Children's experiences with Logo. In T. O'Shea & E. Scanlon (Eds.), *Educational computing (An Open University Reader)*. London: John Wiley & Sons.

Piaget, J. (1952). *The Child's Conception of Number*. London: Routledge and Kegan Paul.

Pintrich, P.R., & Schunk, D.H. (1996). *Motivation in education: Theory, research, and applications*. Englewood Cliffs: Prentice Hall.

Raymond, E. S. (1999). *The Cathedral and the Bazaar*. Sebastopol, CA: O'Reilly & Associates.

Repenning, A. (1993). *Agentsheets: A Tool for Building Domain-Oriented Dynamic, Visual Environments*. PhD thesis, University of Colorado.

Resnick, M. (1994). *Turtles, termites and traffic jams. Explorations in massively parallel microworlds.* Cambridge, MA: MIT Press.

Resnick, M., and Ocko, S. (1991). "LEGO/Logo: Learning Through and About Design." In *Constructionism* (ed. by I. Harel and S. Papert). Ablex Publishing.

Schank, R. and Cleary, C. (1994) *Engines for Education*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Shaw, A. (1996). Social constructionism and the inner city: Designing environments for social development and urban renewal. In Y. Kafai and M. Resnick (Eds.), *Constructionism in practice: Designing, thinking, and learning in a digital world* (pp. 175-206). Mahwah, NJ: Lawrence Erlbaum Publishers.

Turkle, S. (1995). *Life on the Screen: Identity in the Age of the Internet*. New York: Simon and Schuster.

Vygotsky, L.S. (1978). *Mind in Society*. Cambridge: Harvard University Press.

Watts, D.J. (2003). *Six Degrees: The Science of a Connected Age*. Cambridge, MA: W.W. Norton.

Wilensky, U. & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education and Technology. Vol. 8* No. 1. pp. 3 - 18.

Wilensky, U. & Stroup, W. (1999a). Learning through Participatory Simulations: Network-Based Design for Systems Learning in Classrooms. *Computer Supported Collaborative Learning (CSCL'99)*. Stanford University: December 12 - 15, 1999.

Wilensky, U. & Stroup, W. (1999b). *HubNet*. http://ccl.northwestern.edu/netlogo/hubnet.html. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

Wilensky, U. (1999). *NetLogo*. http://ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

Wilensky, U. (2001). Modeling Nature's Emergent Patterns with Multi-agent Languages. *Proceedings of EuroLogo 2001*. Linz, Austria..

Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media, Inc.

Appendix A: Excerpts

00:16:28:15	R: Dor	Why did you cross wires?
00:16:28:00	Sally	Because if it sees it over here… Because when it sees it, it'll turn the way that it sees the light.
00:16:38:34	R: Dor	By crossing?
00:16:41:82	Sally	Yeah… Aw… Now its leaving the center… Now its staying in the center… Now it sees that light… It always turns towards that light. Now its going all squiggly in the center… It's not quite dead center but…

Excerpt 1

00:19:03:85	Daniel	OK OK OK. So if it sees light on the left, it sees light on the right, I want it to turn right so this should just… but if it sees light on the left then it turns a little. ((He puts hands over ears, thinking.))

Excerpt 2

00:20:59:04	Daniel	Since it sees light on the left, it turns right, so it cancelled it out into a general back and forth action until it got towards the source of the light. I still don't get it though.
00:21:23:00	Sally	This is going to be really weird… I go both ways at once whenever I see a robot.

Excerpt 3

00:32:59:00	Sally	Go towards the light and towards everybody else.
00:33:02:00	R: Matthew[9]	So that's one suggestion… go towards the light and towards everybody else… everybody try to get to the center.
00:33:12:88	Arnold	Guys make all your robot wires 25 or 50 so that you're more towards the light and you'll follow someone else with… less strength.
00:33:22:93	Arnold	You see I'm going towards the center… look where Schwarzenegger is OK? He's towards the center… WOW.
00:33:36:00	Daniel	Terminator (Daniel's vbot) is confused!
00:33:36:27	Arnold	Schwarzenegger (Arnold's vbot) is NOT!
00:33:41:88	Jenny	I'm just further away than you…
00:33:41:93	R: Matthew	What were you saying?
00:34:03:35	Daniel	Abe's lost!
00:34:04:00	Arnold	Like usual! It's like Ms. W's class.
00:34:05:00	Abe	((laughs))
00:34:12:43	Jenny	(to Arnold) It's like… You said all the girls would get all the problems, but all the boys got all the problems! I love it! YES!

Excerpt 4

00:36:49:22	Junior	I decided not to cross the lights, you know you can just decide to not cross the lights.
00:37:05:27	R: Dor	Junior, what will that give us what will you suggestion give us? What were you suggesting how should we do it?
00:37:16:18	Junior	Just keep it positive and go like this. It's already a positive so it's much easier, I think, instead of crossing and negative just go straight.
00:37:26:00	R: Dor	So is it a straight positive same as crossed negative?
00:37:28:00	Junior	Yep.
00:37:29:00	R: Dor	Are you sure?
00:37:30:00	Junior	Yeah.

Excerpt 5

[9] R: M is "Researcher: Matthew Berland." R: D is "Researcher: Dor Abrahamson."

00:38:01:36	Daniel	Zip on robots because right now all the robots are clumped to his leg, so instead of moving away, they're all staying…
00:38:08:84	Arnold	I'm doing fifty on the robots and look at where I am.
00:38:11:81	Daniel	Because you're Schwarzenegger, there's something wrong with you… Terminator knows what's going on here… One at a time, everybody has to start following the robots because if you do it all at the same time, we'll all clump together somewhere.

Excerpt 6

00:38:56:17	Jenny	Um… Only one person goes away from the light and everyone follows that person because right now nobody knows where they're going.
00:39:05:19	R: Dor	Uh huh, one at a time you want.
00:39:07:00	Jenny	Yeah.
00:39:08:00	R: Dor	Why why is it important one at a time?
00:39:09:00	Jenny	Because everyone's clumped together in the middle now…
00:39:11:00	R: Dor	Uh huh…
00:39:12:89	Jenny	…and their motors are to go towards the other robots is more than to go away from the light so since there are so many robots, they're following the robots and nobody is moving away so… no one ELSE is moving away.

Excerpt 7

00:40:56:73	R: Dor	What do you think about that? Arnold thinks that everybody at the same time, and you think one at a time what do you think about that?
00:41:13:35	Daniel	Everybody everybody there's a bunch of people… Nobody's listening… Yeah, whatever, I know now there's a bunch of people in a clump in the top left corner if you are in that clump, stop moving but only if you're in that clump, because if everybody up there stops moving...

Excerpt 8

00:41:56:56	Daniel	Matthew Matthew!
00:41:57:00	R: Matthew	Yeah?
00:41:58:00	Daniel	I'd just like to point out that even though it looks like the groups are in separate places, they're actually clumped… The upper left and the upper right, they're right directly by each other now. We're just waiting for the other people to catch in…

Excerpt 9

00:48:35:27	Abe	I noticed a problem, because whenever you run away from one bot, you crash into an infected one.

Excerpt 10

00:51:24:22	Sally	Early on it was harder to infect bots because there were less of us that were infected and then in the middle there were more of us that were getting infected, so it was getting easier and easier but then Edward was really good at getting away from us.
00:51:45:94	Junior	There were really few of us so it was hard to find those that weren't infected.

Excerpt 11

00:54:17:27	R: Matthew	Can anyone tell what the graph will look like once I add ten androids?
00:54:31:35	Jenny	Its going to go higher faster.
00:54:32:00	R: Matthew	Going to go higher faster…
00:54:48:66	Junior	I think that if more androids were put on the graph that it would run more smoothly, so it will sort of look the same, just more smooth and stuff.
00:55:06:71	R: Matthew	His suggestion… What do you think Ivan his suggestion is? That its going to look pretty much the same but its just going to be smoother with 10 androids?
00:55:13:18	Ivan	Its going to go up a lot faster… more people to infect, same amount of room.

Excerpt 12

Appendix B: Gesture

	Starting to move into position, his hands imitate sensors on the robot.
	He wiggles his right hand as a sensor receiving input.
	He taps on his right leg as the leg connected to the right hand/light-sensor.
	He simulates the machine as it moves with the changing input.
	He moves his left hand to make sure it's symmetrical.