UNDERSTANDING THE EVOLUTION OF STUDENTS' PROBLEM DECOMPOSITION IN GLOBALORIA CLASSROOMS

By

Luke Kane

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF ARTS

Telecommunication, Information Studies, and Media

2012

ABSTRACT

UNDERSTANDING THE EVOLUTION OF STUDENTS' PROBLEM DECOMPOSITION IN GLOBALORIA CLASSROOMS

By

Luke Kane

Today there is an increasing demand by companies, governments, and society for people who know how to think computationally (i.e., think critically, logically, and solve problems in innovative ways using computational tools) (Wing, 2006; National Academy of Sciences, 2010), in order to be competitive in the knowledge economy. Educational video game design has shown potential in helping to prepare youth with skills germane to computational thinking, and the so-called STEM disciplines whose practices heavily rest on computation (Games, 2010; Hayes and Games, 2008).

The potential has been recognized by the White House's efforts (White House, 2009) to support educational video game design, including sponsoring national game design contests and encouraging programs that teach computational thinking. This study examined one such program, Globaloria, whose goal is to foster computational thinking and STEM skills and concepts in middle school students by immersing them in a game design discourse, using Adobe Flash as the platform for developing these games.

The study examined the evolution of 15 students' computational thinking (specifically, the dimension of Decomposition) as a function of their changes in language use, design strategies, and game artifact production. Findings suggest that a scaffolded game design-based curriculum can provide an effective context for students to develop computational thinking and deep understandings and engagement with STEM subjects.

ACKNOWLEDGEMENTS

I would like to thank Alex Games, for his patience and guidance, which have been invaluable in this process. I would also like to acknowledge Carrie Heeter, whose encouragement and feedback has always been in service of producing a better product. And of course, Cliff Lampe, the third member of my committee, who was kind enough to give me the time and feedback required to complete this thesis.

I must also acknowledge Idit Caperton, who was kind enough to allow a Master's student to conduct the research that would ultimately become this thesis.

And of course, I must also thank Brittany Rugg, who always provided her love and support of me throughout this process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

This report concentrates on research the author conducted for Globaloria, a learning environment that integrates a platform of web 2.0 technologies such as wikis, discussion forums, and embedded rich media, with an educational game design curriculum invented by the World Wide Workshop Foundation (worldwideworkshop.org, 2011). Evidence that computer games and simulations can be used for learning in STEM (Science, Technology, Engineering, and Math) literacy has become increasingly available, including research focused on the learning sciences, educational psychology, and media studies (Games and Squire, 2011; National Academy of Sciences, 2010). In particular, computational thinking has emerged as a skill – or rather, a set of skills – that will be critical in regaining an economic advantage, in particular where STEM field jobs are concerned.

Pedagogies centered on the design of new technologies have also gained prominence in diverse areas of STEM learning, research, and practice over the last twenty years (Kolodner et al., 1998; Papert and Harel, 1991; Perkins, 1986). Design as a method of learning is strongly aligned with principles under the constructivist and socioconstructivist learning theories. Learning through design allows learners to be active participants responsible for their own knowledge construction process and places instructors in a support and guidance role that allows them to facilitate this process for learners favoring different strategic approaches to such construction (Kolodner et al., 1998).

GLOBALORIA

Established in 2006, Globaloria is a social learning network, within which is embedded a well structured, game-making curriculum that allows students to "create educational games and interactive simulations, for their own personal and professional development, and for the social and economic benefit of their communities" (Globaloria, 2010). At the core of the Globaloria curriculum is a community wiki and blog where the Globaloria staff post current assignments, lessons, and tutorials (see Figure 1).

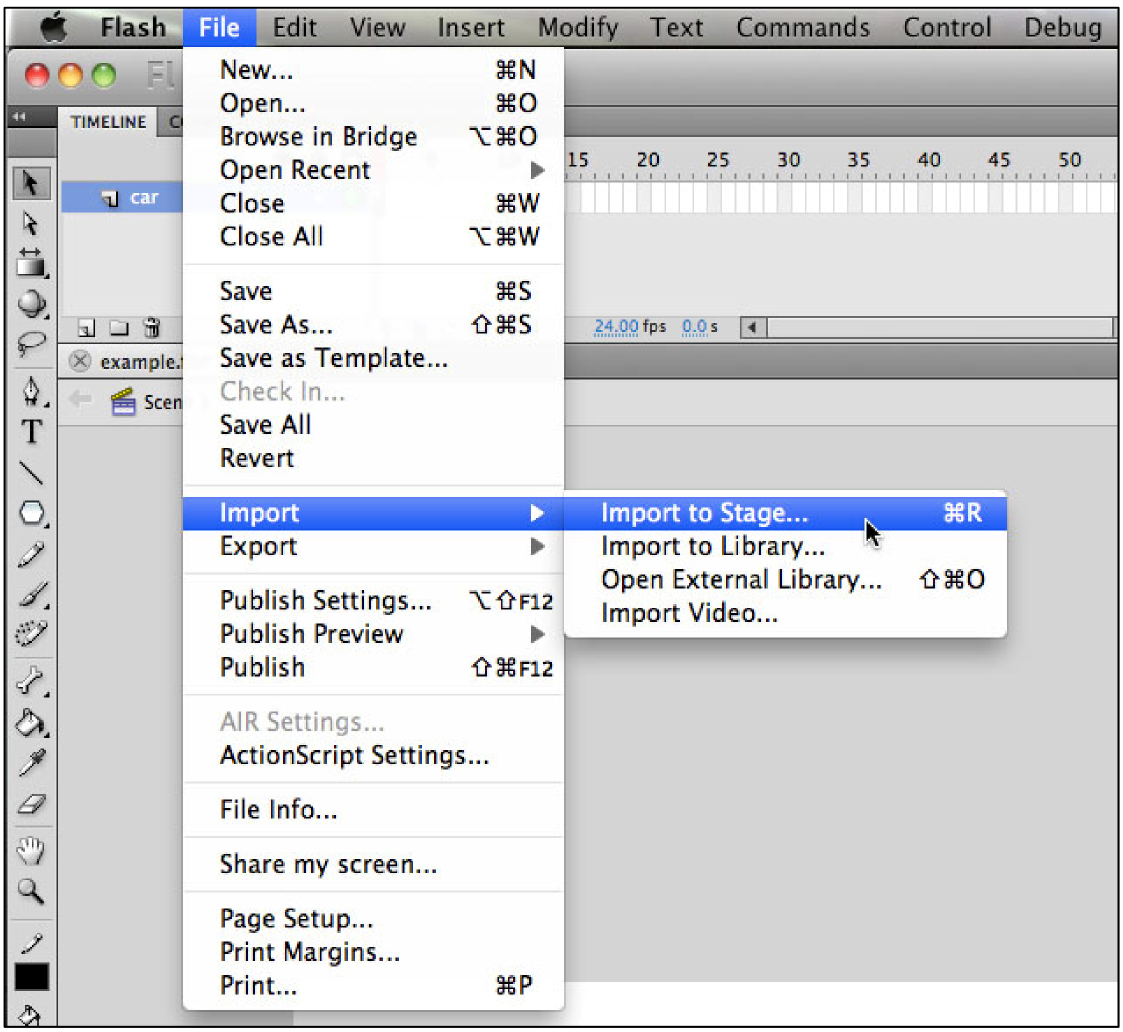


Figure 1: A screenshot from the Community wiki, showing students how to import objects to the stage in Flash. (For interpretation of the references to color in this and all other images, the reader is referred to the electronic version of this thesis).

Students also have their own personal wikis, to which they can post blogs, notes, assignments, favorite games and animations, and information about themselves (non-private information such as favorite movie, actor, video game, food, etc.) (See Figure 2).

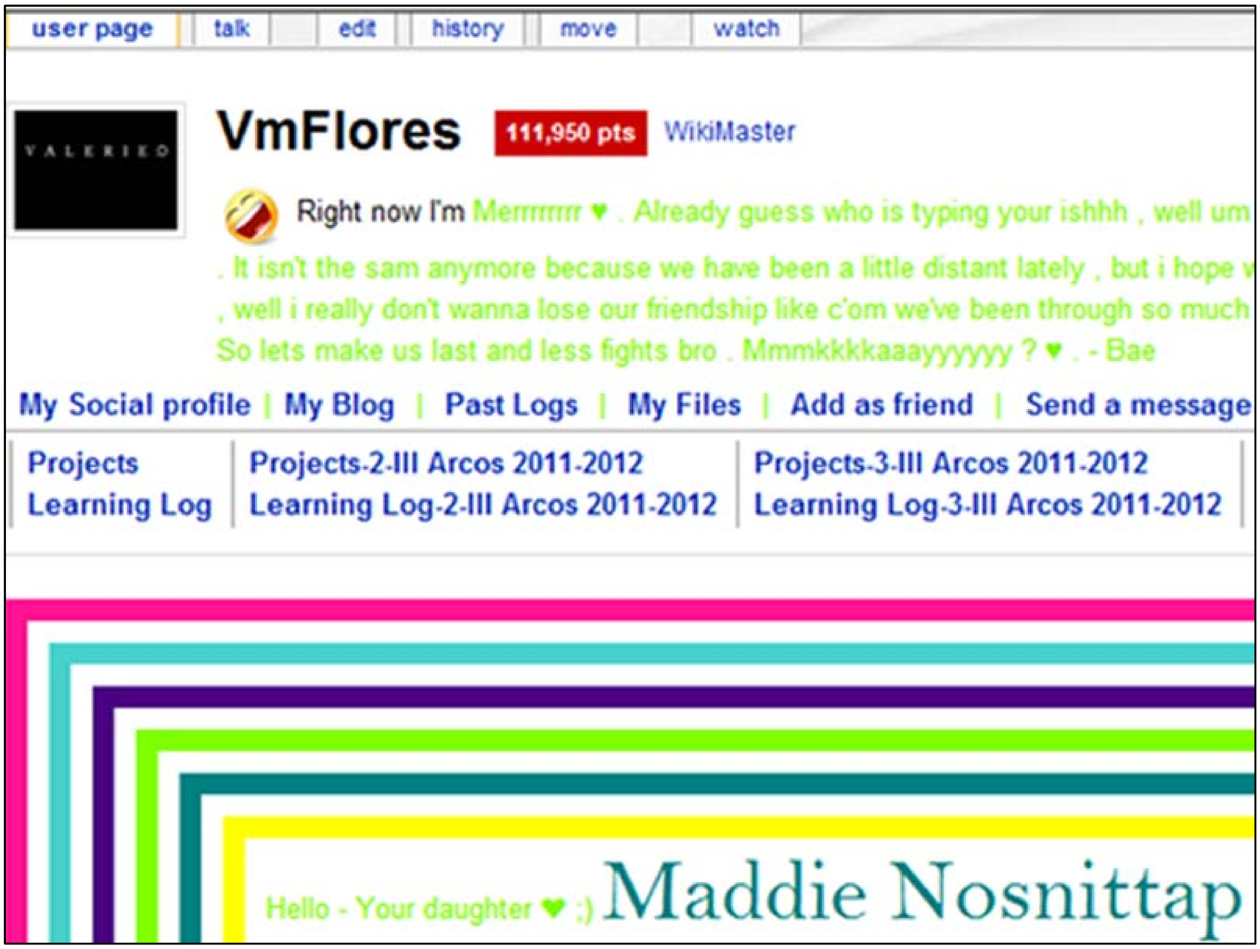


Figure 2: An example of a student's homepage, containing information about themselves.

The students' wikis are also connected to the community wiki so that students are able to find and access help from the lessons and tutorials, as well as communicate with each other. In addition to the community and personal wikis, students use Adobe Flash, a graphical and scripting software package used to create many of the browser-based games available today. As mentioned before, students in Globaloria learn to create educational games, simulations, and animations, all of which are created in Flash. Like many programming languages, Flash is a relatively complex language to learn, especially for middle school students. However, similar to libraries of codes that professional programmers may use, Globaloria provides libraries of code fragments commonly used in the game assignments on its community wiki, as well as tutorials and examples of how and when to use those scripts. For example, pieces of code that make objects move and follow other objects can be found on the community wiki, and the students simply have to find the appropriate code for the game or animation that they are working on, copy and paste that code into the appropriate location in their Actionscript, and update the instance names. Alternatively, students can find and download .fla files (Flash files use .fla as the three character document type) from the community wiki and the replace graphic assets with their own. They may also use the downloaded .fla file simply to see how the Globaloria staff created that particular animation or game so that they can replicate the game themselves (see Figure 3).

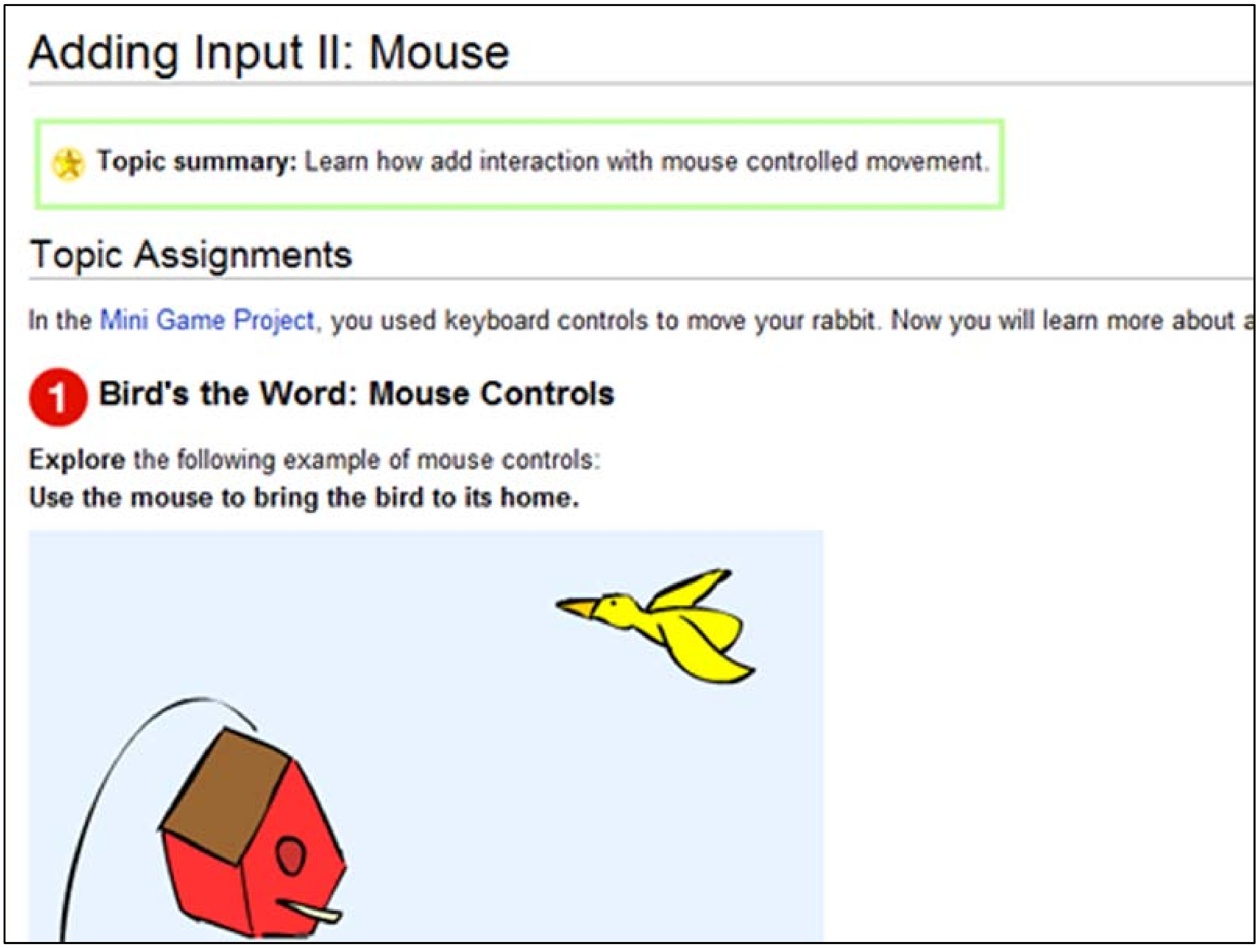


Figure 3: Tutorial on the community wiki that teaches students how to add mouse input to their Flash games. Underneath the picture of the birdhouse is a link with the source code that students can download and use as a template.

Structurally, Globaloria, functions like a full class. Students meet once a day and receive assignments and homework, just as if it were math or history. However, due to the more creative nature of Globaloria, classroom time is structured more like a design-based class. Students are given an assignment and allowed to use whatever tools they need to complete that assignment, and guidance and support when necessary. Unlike a traditional educational format of a teacher lecturing *at* students with little to no meaningful interactions, Globaloria's assignments often draw upon multiple disciplines and involve frequent, ongoing teacher-student interaction and feedback.

At the site that I visited in Austin, Texas, there were two Globaloria classrooms, each with a different teacher. All of the students enrolled for an entire year, and each classroom had 18-24 students. Each individual student attended class once per day, and the class periods were 45 minutes long.

Evidence that computer games and simulations can be used for learning problem solving has become increasingly available through research in the learning sciences, educational psychology, and media studies (Games and Squire, 2011; National Academy of Sciences, 2010). Modern videogames are complex sociotechnical systems where players engage in cycles of play interactions with rule-bound computer simulations built by teams of designers, computer programmers, and digital artists. To participate in the production of a videogame, producer-learners must be able to solve a series of complex problems using computer technologies, and to be able to think in terms of the computer tools they will use will enable them or limit them from doing so.

The body of knowledge and practice skills necessary to solve problems effectively using computing tools has been termed *Computational Thinking* (CT) (NRC, 2011), and as Wing (2010) has argued, these skills and knowledge are expected to be fundamental for countries hoping to maintain competitive workforces in the 21st century. Modern computational technologies (Computers and other ICT's) have fundamentally transformed the social, economic and even demographic mechanisms of a globalized world (Gee, Hull, and Lankshear, 1996). Today's workplaces are rapidly becoming sophisticated sociotechnical systems, where everything from serving a hamburger to producing the newest space exploration technology requires people capable of understanding the connections between software, hardware, information, and the way each of these mediates human activity. Social media and mobile technologies have made our social, entertainment, and civic lives exist in a sea of information, where effective and safe participation requires citizens capable of understanding and taking advantage of the powerful computational tools that now lie at their fingertips.

OPERATIONALIZING COMPUTATIONAL THINKING

Computational thinking (CT) is a way of thinking and solving problems effectively using the logical, mathematical, and representational tools that computers make available work our way through data, and transform it into actionable information. Over the years, multiple frameworks of computational thinking have been proposed by scholars, each emphasizing different aspects of the construct, from the more abstract and logical operations necessary to construct a software algorithm, to the more social aspects involved in solving a complex computational problem collaboratively (National Academy of Science, 2010). In order to operationalize the construct for this particular study, we turn to a framework recently proposed by Google (2010), which synthesizes most of these perspectives according to the actual practices of software professionals today. The framework characterizes CT according to five distinct dimensions that encompass habits of mind and practice germane to solving problems using computational tools. These are:

1. **Decomposition** is the breaking down, or ability to break down, a problem into its core components.
2. **Pattern Recognition** is the ability to see or identify recurring systemic connections between objects, as well as recurring subsystems in systems.
3. **Pattern Generalization and Abstraction** is the ability to recall previously encountered and use them to aid in the solving of applicable problems in different contexts; a form of near transfer that is characterized by students' ability to abstract.
4. **Algorithm Design** is the construction of a step-by-step process towards the solution of a design problem. The more sophisticated this skill, the more *efficient* the algorithm will be.

5. **Data Analysis, Modeling, and Visualization** refers to the students' ability to extract data from sources relevant to a phenomenon and to use it to construct a model that can be communicated to others. As an example, examining a table of daily ozone values in a metropolitan area, and using them to construct a histogram to share with others would be a form of modeling and visualization, but constructing a game with rules and mechanics based on the same values in a way that others could use them would be one as well.

These five dimensions offer key advantages for research and analysis. Specifically, 1) they are measurable, meaning that there are concrete and observable processes and products that can be captured by systematic research, either through observation or other means, and 2) they are applicable to a broad range of professions and activities involving the use of modern computing tools, which in our view is a necessary prerequisite to differentiate computational thinking from other forms of logical and mathematical abstract thought.

GAMES' THREE DIALOG FRAMEWORK OF GAME DESIGN AND CT

In a series of studies of children designing their own videogames, Games (2008; 2010) proposed a framework for analyzing and assessing learning in the context of game production as a function of the acquisition of key constructs central to the discourse of professional game designers. The framework examines increasing sophistication in this discourse as a function of the degree with which learners' decisions, language, and tool use reflect an increasing understanding of the nature of games as sociotechnical systems, through an awareness of a) the affordances of materials and tools available to them in creating games, b) the abstractions necessary for an idealized player to play a game with these materials (e.g. rules, mechanics, goals and so on), and c) the probable ways in which real players would interpret and understand these materials and abstractions during play. Rooted in Discourse theory (Gee, 2005; Games, 2008), the framework sees these dimensions as dialogic in nature, as a conversation between a designer and one or more players, mediated by the game. Their evidence is only visible in an analysis of the ways of talking and doing of learners during the cycles of iterative refinement and continuous formative feedback that are fundamental to the effective production of good games.

Table 1 describes the scope and nature of the evidence that would make each dialog overt:

Table 1: Games' Three Dialog Framework

Dialog	Description
Material Dialog Perspective	Refers to language and practices that reflect students' understanding and use of the techniques necessary to construct a game system, akin to DiSessa's (2002) notion of material intelligence. For example, a designer could not make a quality game using Flash, unless they understood both the programming principles of Actionscript and programming's connections to Flash's vector graphics system.
Ideal Player Dialog Perspective	Refers to language and practices that reflect an understanding of the abstractions that need to be encoded in the tools to transform a system of materials into a system of play, including the use of the specialist language that game designers use to describe the possible actions that these abstractions enable and limit for players (game rules, mechanics, etc.). For example, discussing the way that the rules in chess would define the possible ways in which a player could move a pawn.
Real Player Dialog Perspective	Refers to language and practices denoting an understanding of how to use the game system built from the materials to encode knowledge and metaphors that make it clear and overt to real players how the game is to be played. This dialog also involves an understanding of the encoding of knowledge extraneous to the game in order to support player decision making during the game. An example of a failure to understand this would be a learner designing a game for young children containing a lot of fast-scrolling text, with the expectation that they would need to read this to play.

As Games has shown, the Three Dialog framework is useful in characterizing the evolution of designers' forms of knowing and doing necessary to the construction of effective game play over time, as a function of changes in their discourse practices. If one works under the belief that the creation of a computer game is in essence the creation of a computer-based sociotechnical play system, and that, as the authors' experience with game design suggests, it requires the application of the five dimensions of computational thinking as key practices necessary to enact good game design, an analysis of learners' discourse over time using the three dialogs could be focused on those aspects of the learner’s discourse that reveal changes in the five dimensions during design activity.

RESEARCH QUESTIONS AND METHODS

The research described in this thesis applied the above theoretical framework, with the goal of documenting the evolution of children's grasp of Decomposition, one of the dimensions of computational thinking, in the context of their participation in the Globaloria curriculum. The following research question was addressed:

1. *How does the skill of Decomposition evolve as students' participate in Globaloria assignments related to that skill?*

The qualitative methodology of case studies (Stake, 1995) was used to address this question, grounded in the Three Dialog framework. Multimodal discourse analyses (Gee, 2005) was conducted of students' games, design decisions, and tool use over a period of four months of participation in Globaloria. The goal of the study was to produce a "thick description" (Geertz, 1973) of the Globaloria learning ecology and its impact on Decomposition computational thinking over time.

In line with Globaloria´s constructionist philosophy this study examined children's learning from a socioculturally situated perspective. In this perspective, language, action, and thought are seen as integrally interconnected (Gee, 1992; Vygotsky, 1978; Wertsch, 1998; Engstrom, 2005), and evidence of changes in thought emerges from triangulating evidence of changes in students' ways of enacting the solutions to design problems. Data were collected in the form of video and audio recordings, and coding those data provided evidence of design decisions, talk about those decisions, and the computational artifacts resulting from and supporting those decisions (game software, paper game designs stored in the Globaloria wiki) over time.

Data were collected from pre and post-assessment protocols that consisted of individual interactive interviews with Globaloria students, as well as from observations of their design activities during Globaloria class over a period of four months.

An interactive interview was conducted that consisted of asking students to conduct a problem decomposition task that required them to identify the core components of a simple Flash game and then try to recreate that system. Parallel video recordings of participants' and their computer screens were captured, with the goal of documenting participants' design decisions, choices for tool use, and language describing their thinking and practices during each stage of the protocol.

The assessment was not designed to measure declarative knowledge, but rather to assess how well students were able to recall, construct, and apply their own knowledge in the process of solving decomposition-related design problems, in line with the constructionist philosophy of the Globaloria curriculum. The decomposition dimension, displayed by students' through design decisions, tool choices, and verbal explanations of their solution strategies during the decomposition task, was the basis for coding observations for discourse analysis (Gee, 1999; 2005; 2007), as described in Figure 4:

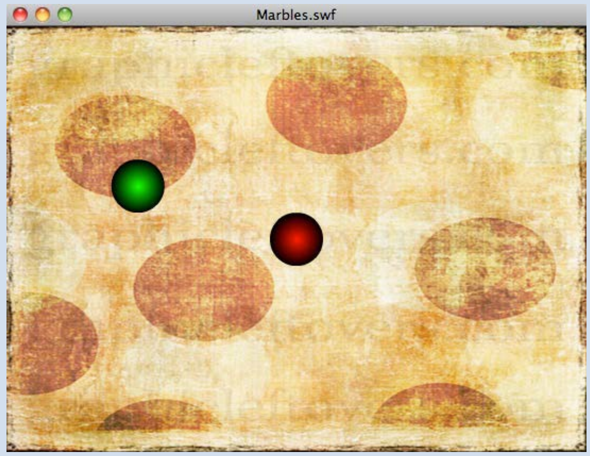
CT Dimension	Interview Task Sample
Decomposition: Students were shown a simple game made in Flash (see Figure a) and asked to recreate it. The goal of this exercise was to establish whether or not the students could look at a game in Flash and break it down into its core components. Successful completion of the task was not reliant upon the full reproduction of the marble game, but how deeply they could deconstruct the given system.	 *The Marble game*

Figure 4: The individual interview protocol.

Observations were coded according to categories generated from the intersection of the three levels of thought in the Three Dialog framework, and how decomposition was observed within this framework. At the Material level, students should be observed identifying the individual parts within Flash that would allow someone to reproduce the Marble game. The evidence of this is seen in the process through which they attempt to reproduce the Marble game, as they incorporate different elements of Flash into the game. At the Ideal level, students begin to think of the game in terms of its play mechanics, or the actions that are available to the player. Evidence of this is observed in their dialog as they talk about how a player would interact with the game. Through the Real Perspective, students will start to talk about the Marble game in their own terms, or in terms, which they understand. They may also design the game so that it is unlike the original game. For example, a student may decide to animate the marbles instead of using code to move them. In this case, they have interpreted the system as moving objects, whereas the core concept of the game is clicking on marbles to make them move. Removing the interactivity effectively transforms the game from an interactive system to a passively consumed animation. Table 2 summarizes these codes.

Table 2: Codes used for the analysis of decomposition in game design.

	Decomposition
Material Perspective	Breaking a problem into its software components
Ideal Perspective	Breaking a problem into its play abstractions
Real Perspective	Breaking a problem into its cultural interpretations

Data were collected from 15 students at the East Austin College Preparatory Academy (EACPA) in Austin, Texas at the beginning and end of four months of Globaloria participation. These 15 students ranged in age from 13 to 14 years old, and consisted of 7 females and 8 males. Participation in Globaloria was a result of enrollment in the school, as every student takes Globaloria as a class. Participation in the interviews was by stratified random selection, keeping gender as even as possible, given an odd numbered N. Interviews were conducted in the school library and during the school day; the schedule of students was coordinated through a school counselor. Over the course of two days, I conducted 15 total interviews, each lasting approximately 20 minutes.

Table 3: Sample and dates

Site	**Students**	**Pre**	**Post**
EACPA, Austin, TX	15	1/24/11 - 1/25/11	5/2/11 - 5/3/11

RESULTS

By and large, students displayed an increase in sophistication on decomposition strategies between pre and post-test. At pre-test, none of the students had been able to clearly identify in a discrete ways either the abstractions or the physical elements that they would use in the process of reproducing the clip. At post-test however, almost all of the students (14 out of 15) were able to identify marbles, the background, and code to move the marbles as necessary components of the game during the post-assessment, a clear improvement in their understanding of the materials necessary to begin reproducing the game.

Table 4: The basic components of the marble game that students identified (all names are pseudonyms).

	Background		Marbles		Code to Move Marbles	
	PRE	POST	PRE	POST	PRE	POST
Victor	-	-	+	+	+	+
Paul	-	-	-	-	+	-
Megan	+	+	-	+	-	+
Chloe	-	+	+	+	-	+
Katherine	+	+	-	+	-	+
Thomas	-	+	+	+	+	+
Cole	+	-	-	+	-	+
Bruce	-	-	+	+	+	+
John	-	-	-	+	-	+
Sarah	-	+	+	+	+	-
Elena	+	-	-	+	-	+
Amanda	-	+	+	+	-	-
Mario	-	+	-	+	-	+
Steven	-	+	-	+	+	+
Tori	-	+	-	+	-	-

Table 4 gives a breakdown of the basic elements of the Marble game that students identified as important for reconstructing it. A plus sign (+) indicates that the student identified that component, either verbally or through their design process, as an important part of reproducing the Marble game. A minus sign (-) indicates that they did not identify that component as important. Evidence for these plusses and minuses was observed through the students' design processes and in their think-aloud strategies (I will cover some brief examples here in the Results section, but for a more thorough description of a few selected, exemplar interviews, please refer to the Appendix).

As an example, Figure 4 shows the first steps that one student, Thomas, took to reproduce the game. The background and marbles are all present, and so this counted as evidence that he understood them as important parts of the system.

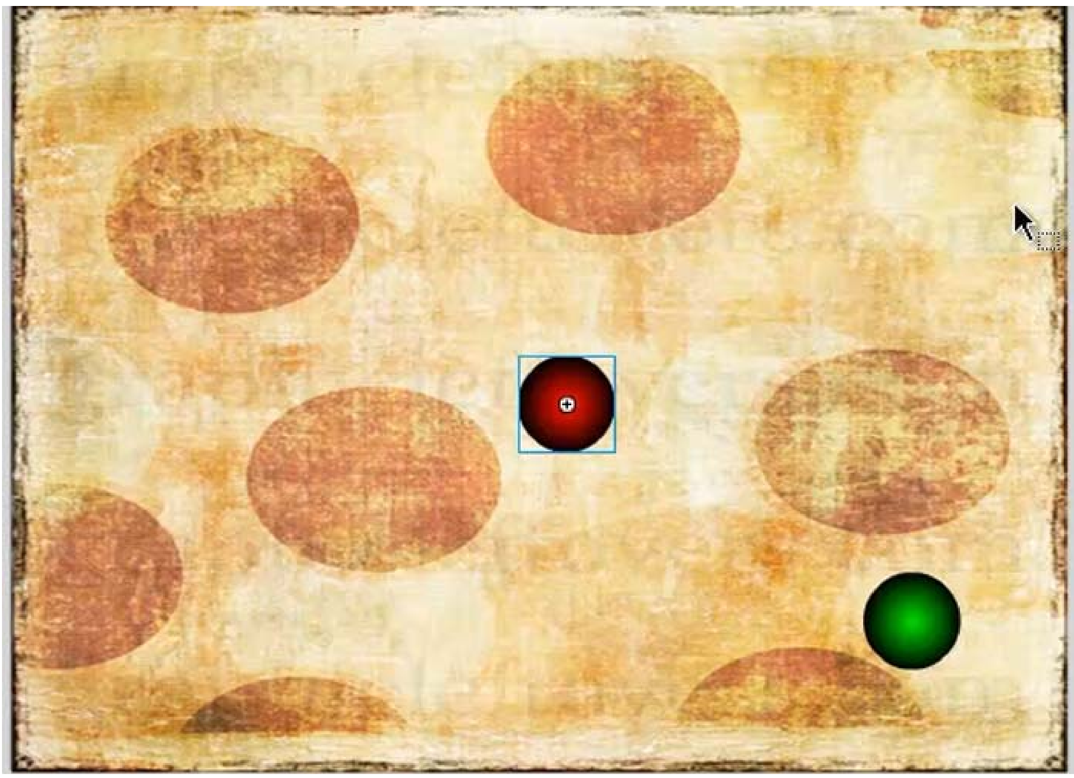


Figure 5: A student takes the first steps in reproducing the Marble game.

The background and the marbles are all assets that were provided to the students, and they provide the structural foundation for the game. A student could choose to create their own assets, but this was rare, and would be an indication that the student is driven more by the aesthetics of the game. "Code to Move the Marbles" is a common item on the students' list of what was important. As most of the students knew that there was "some code", but had a difficult time identifying the specific pieces of that code, I counted "some code" as its own category. I will differentiate *which* code further on in the results. However, the fact that students could identify "some code that makes the marbles move" shows that they understand the game as an interactive system, and not a passively consumable animation. As we will see later on, this distinction will have consequences on how the students approach their designs and how they interpret the Marble game in terms that they understand (i.e., I *understand* how to animate objects in Flash, but I *do not understand* how to make things move automatically when I click on them).

The three components presented in Table 5 (background, marbles, and some sort of code to make the marbles move) represent the most basic level of Decomposition for this game. In other words, these would be the minimum pieces necessary for the reconstruction of the Marble game, not including the title screen. As these are the most basic components, most of the students could identify all of the components, as can be seen in Table 5. Because of the students could so easily identify the most basic components of the Marble game, there was very little change observed from pre-test to post-test.

At the Ideal level, their discourse showed that a key change in their decomposition skill was driven by their appropriation of the specialist language of interactive media embedded in the Flash authoring system. An analysis of the discourse they used in their description of the process they would follow to recreate the clip at pre-assessment was represented by phrases such as *"I would have to put the marbles and make them move"*. In contrast, by post-assessment, the same statement took the form *"I need to create two marble movieclips on the stage, and then use the* *codes to program them to move"*.

This appropriation is important, for as Games (2010), and other scholars (Gee, 2007) have previously argued an expanded lexicon within a specialist discourse such as game design gives learners "tools to think with", that help them organize their mind and understand the abstractions and nuances used by experts in those fields, in addition to enabling them to communicate and learn from discourse with more experienced others. In this case, such language allowed learners to more deeply discuss and understand the abstract relationships between the visible elements of the game (the marbles on the stage), and the systemic interactions between them represented by the software.

Appropriation of decomposition coding language was particularly evident in students' design decisions and use of software tools at post-test, and most interestingly, also revealed two very different strategies. The first strategy (used by 4 students) was to rely on the use of direct timeline animation utilizing movieclips and motion blend techniques (tweens) to animate the marble collision animation directly on the timeline (this technique can be seen in Figure 6).

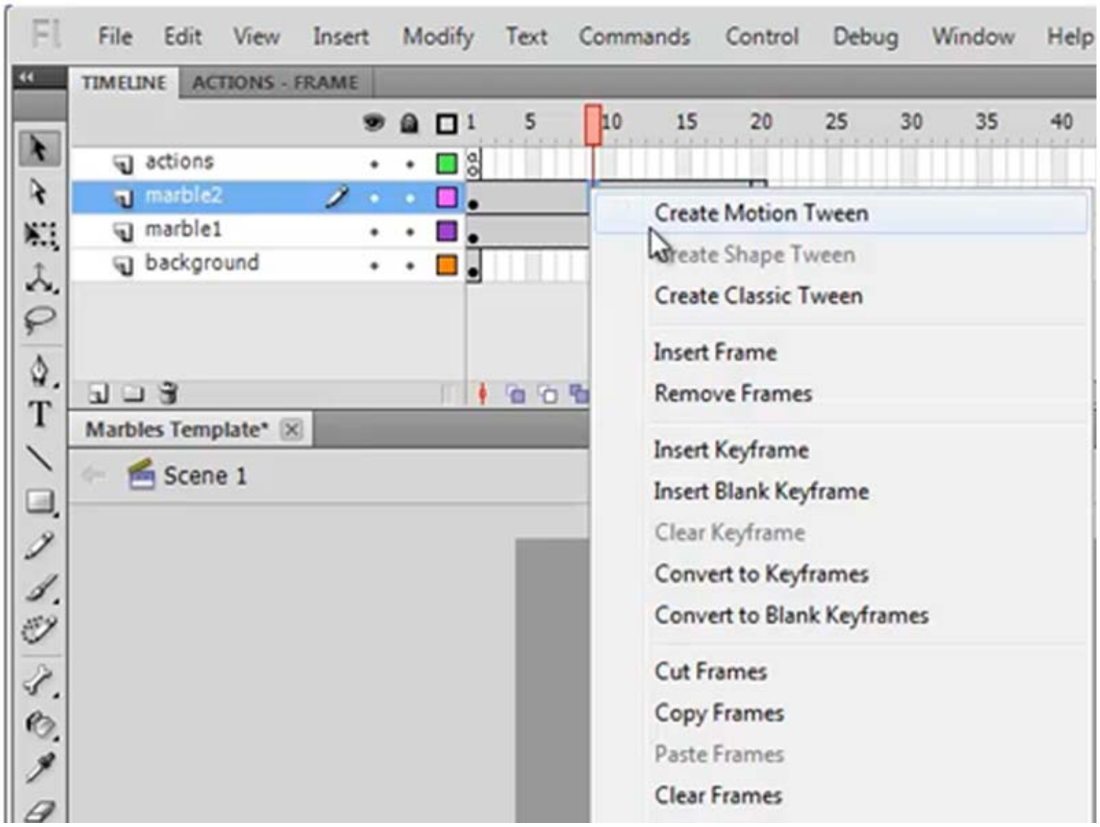


Figure 6: Keyframe animation of the Marble game.

This was, in general, a more sophisticated approach at decomposing the problem than what the same students generally showed at pre-test (e.g., searching for code on the wiki, but not knowing exactly what they are looking for), and demonstrated their understanding of some important abstractions, such as the use of vector graphics with interpolated offset positions using Flash tweens as a way to break down the problem (a similar technique to that used by animators with flipbooks in the days prior to computer graphics), but was more limited than the second approach in that it would limit their exposure to the interactive functionality in the Marble game. However, the animation technique is indicating that those students have abstracted the problem into terms with which they are familiar and can understand and solve. While they are giving up the interactivity in the system, they are abstracting a problem onto existing solutions, instead of abstracting solutions onto a problem. This is an important distinction, as the core idea behind computational thinking is that it is a way of solving problems, most commonly by abstracting known solutions onto new problems.

By contrast, other students, such as Chloe, approached decomposition of the problem by considering Actionscript programming as a key component of the problem-solving strategy. By thinking of the problems in terms of familiar elements, Chloe was able to search for a piece of code on the wiki that would implement collision detection into the game. Collision detection is essential for the red marble to detect "getting hit by" the green marble and starting its subsequent rolling. Figure 7 shows the code that Chloe found and inserted into her Actions layer.

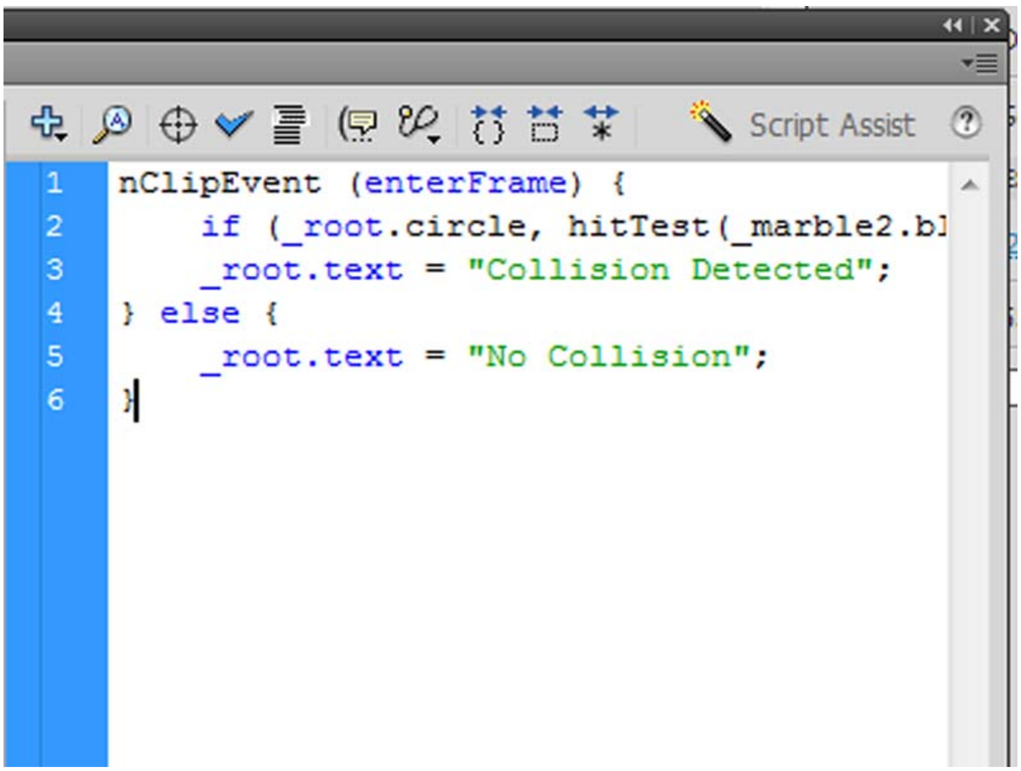


Figure 7: Incorrect collision detection code.

Although the code that Chloe found and used was not the correct code, it still provides evidence that she is thinking about the game more complexly, and in terms of the events that she has observed. Whereas in her pre-test she did not know that Collision Detection was a specific piece of code that she would need, she did identify it during her post-test.

All of the students turned to the wiki in order to find or search for code that they would need for the task. This process represents an understanding of the connection between the assets (the marbles, background, and buttons) and the code that drives the interactions between them. The most common approach to doing this was by navigating to a lesson in the wiki that was specifically focused on the intended task (i.e., animating objects within game control loop, setting up collision detection, etc.), and then copying and pasting the Actionscript code into their own code before modifying it to fit the instance names given to their own clips (see figure 8 below).

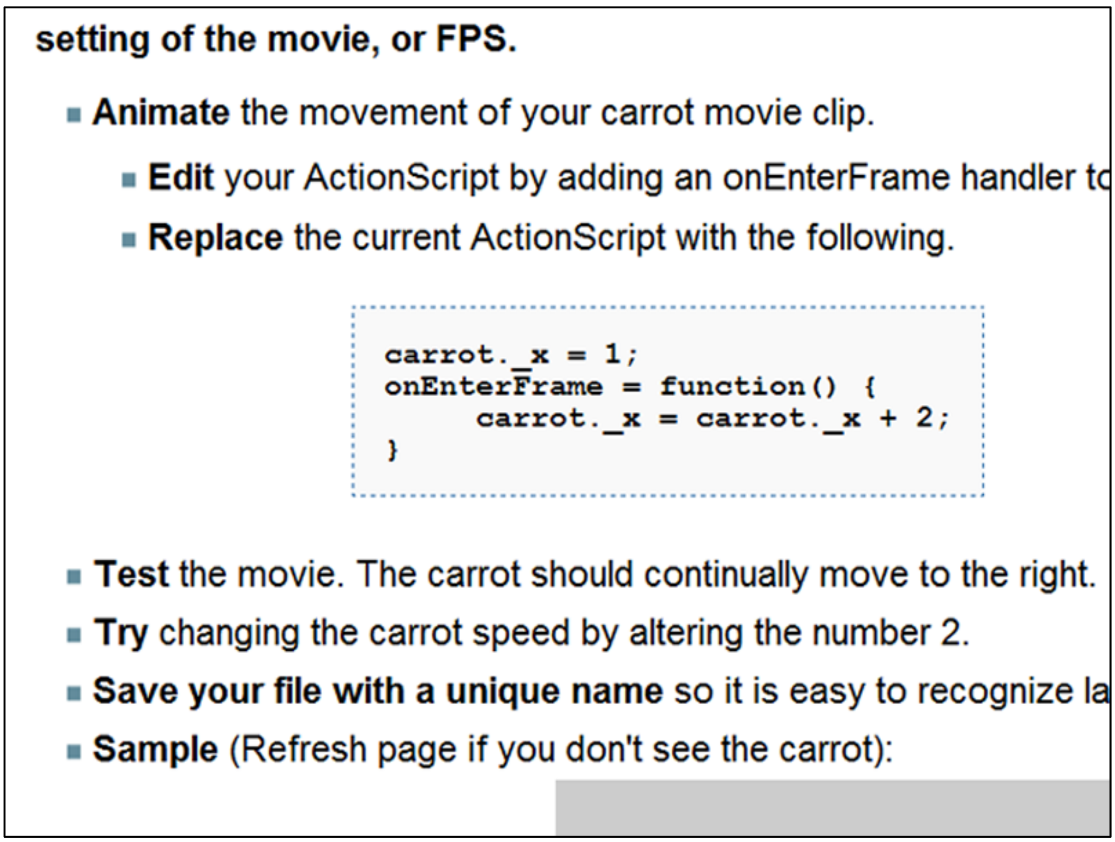


Figure 8: Copying code from a wiki lesson to the Flash Actionscript window

The above process highlights the critical role that students' understanding of the cause-and-effect relationship between their wiki lessons and their own projects plays in the development of their problem decomposition skills within Globaloria. In other words, students can see that the patterns and codes from their wiki lessons can be used in their own games, and this is important as it gives them a repository of code that they can lean on for assistance.

In both the post assessment interviews as well as the projects uploaded to the wiki, it was evident that students relied very strongly on wiki searches and a process of copying code, pasting it, and then observing the result by executing the movies they had designed. However, a clear limitation in their strategies was that the wiki seemed to produce a memory recall pattern that was more akin rote memorization that understanding. When asked to elaborate on why they though a specific code would do what it did, students were seldom able to explain its general function, and for code with more abstract systemic implications they usually had no idea of its effects. Clear examples of this were the functions *hitTest(object)*, which generates an event when an object has collided with another on the stage, and "gotoAndStop(*scene#*), which sends the movie to a specific scene, removing any elements existent on the stage and drawing new ones. In both of these cases, the functions were widely used in the students' final projects, however, during the interviews only about half of the students knew where the collision detection code would be necessary (8 out of 15), and only 4 students knew that the switch scene function would be necessary. A breakdown of these numbers can be seen in Table 6.

Table 5: Deeper level components of the Decomposition task.

	Collision Detection		Switch Scenes		Animation	
	PRE	POST	PRE	POST	PRE	POST
Victor	-	-	-	-	-	+
Paul	-	+	-	-	-	-
Megan	-	-	-	+	-	-
Chloe	-	+	-	-	-	-
Katherine	-	+	-	+	-	-
Thomas	-	+	-	+	-	+
Cole	-	+	-	-	-	+
Bruce	-	+	-	-	-	-
John	-	-	-	-	-	-
Sarah	-	-	-	-	-	-
Elena	-	+	-	-	-	-
Amanda	-	-	-	-	-	-
Mario	-	+	-	-	-	-
Steven	-	-	-	-	-	+
Tori	-	-	-	+	+	-

Table 5 follows the same structure as Table 4, meaning that the plus signs (+) indicate that the particular element (Collision Detection, Switching Scenes, and Animation) was observed, either through the students design process or through audio recording of their dialog as they worked on the game; minus signs (-) indicate that the particular element was not observed.

Table 6 shows that 8 students could identify a specific piece of code in their post-tests, beyond simply "some code", and only 3 students could identify scene switching as a necessary process in reproducing the Marble game. As can be seen, most of the students still could not identify the specific pieces of code to make the marbles move, nor could they find them on the wiki.

Interestingly, between pre-test and post-test, none of the students were able to identify player interactivity (a player clicking on a marble to start its motion) as a core piece of the

Marble game, let alone use Actionscript code to make this happen. This indicates the Globaloria curriculum was not as strong at connecting individual students to the notion of an audience (real players) for their designs as it could have been. This deficit may help to explain the strong similarities between most final projects students created in the wiki at the end of the course.

As a whole, these findings suggest a general trend toward an increasing sophistication in students' computational thinking, albeit one that is defined in terms of the students' games and the Globaloria wiki. However, the trend of memorizing small pieces of code, which are connected to specific game mechanics, definitely points to a learning mechanism at work in the curriculum that gives students an opportunity to appropriate new knowledge. It is also clear that the process of knowledge and skill acquisition may be limited in terms of students' understanding of the systemic nature of software and games. However, it is possible that if, within the wiki project, multiple exercises involving the use of the functions were mastered in different contexts (perhaps in a following academic year), such understanding would be more likely to emerge.

CONCLUSION

The findings of this study indicate that the Globaloria curriculum shows promise in positively impacting students' ability to decompose unfamiliar systems. However, like other constructionist, learner-centered learning environments, student self-reliance, initiative and curiosity play a fundamental role in driving learning. The findings from this study suggest that the program faces an important challenge in placing those expectations on students within schools that have, for years, organized students' learning lives around the instructionist (teacher-driven) models most often used by traditional instruction. A possible consequence of this conflict is that students may be unable to abstract the skills that they learn in Globaloria to unfamiliar problems, if those problems are not easily solved or broken down in terms of curricular material.

The conflict between students' lifelong experience in traditional classrooms with Globaloria's constructivist design likely has an effect on the activities and thinking of students in the program, and influenced their ability to decompose a system during the study observations.

In particular, this influence showed itself the strongest in the students' tendency to narrowly use what they had learned in previous lessons to attempt to tackle the design tasks rather than to take advantage of the other information available in the wiki to use that knowledge more effectively and creatively. It also showed in the students' tendency to stay within the information immediately available to them, rather than to seek new information, even when resources like the Internet were available to them during the task.

In terms of Games' Three Dialog Framework, analysis of the evidence collected from the individual protocols suggests that changes in the students' ability to decompose a Flash game was based heavily in the Material Dialog. In other words, the overall theme that emerged was that students relied on an increased proficiency with Flash to identify the key components and events within the Marble game, but showed relatively little interpretation of that system outside of terms of their curricular material.

This was evident in that the Marbles task required them to examine an interactive system with a pattern that did not directly map to a Globaloria lesson. What this means is that students prescribe to a very specific procedure for designing games in Globaloria. While this has advantages and disadvantages, just as any curriculum does, the Marbles task could not be completed by following that procedure. It required that students think procedurally, and beyond the lessons and material on the wiki, which mirror the rote memorization of traditional classrooms. Evidence of this insight is supported by the fact that the students who were able to progress beyond "some code" and identify scene switching or collision detection, only identify these features because they map directly to a pattern given to them in the wiki, and the copying and pasting of the analogous code served as a starting point for their own designs.

DISCUSSION AND FUTURE RESEARCH

This research points both to an opportunity and a challenge for Globaloria. On one hand, the Globaloria curriculum, in its current form, shows promise helping children develop the ability to learn and apply a base pattern of interactivity to a different context, as long as that context is closely similar (near transfer) to the original lessons. Succeeding at this task also requires the students to have learned that there is a connection between code and the visual representations it effects. However, the evidence also points out that students are learning a narrow and decontextualized set of patterns, which, if expanded, would give learners flexibility to think more deeply and creatively about the goals of interactivity, whether for games or other systems. These patterns could be introduced as other genres of games, or even other forms of interactive projects, provided as choices for the teams, thus allowing the students an increased degree of ownership of their projects overall. This would allow them to use a game format in its most effective form to learn knowledge relevant in other courses and contexts of interest to them, both of which are aspects that should positively impact their motivation throughout the Globaloria curriculum (Wigfield and Eccles, 2000).

From the perspective of the Three Dialogs, this continues to point at their increased mastery of a Material Dialog (learning to use the tools), but not of the Ideal or Real player Dialogs (for what kinds of purposes or people would the systems created with those tools be for). Developing this connection between tools, systems, and users is a necessary concept that students need to master to effectively use programming in the context of systems design later on. This author suggests that if game design is to be used as part of the curriculum (or the whole curriculum), then a systematic approach to introduce players to the patterns and principles involved in the creation of game systems is an important component to be added.

Finally, this study shows the promise for using the framework of computational thinking and the Three Dialog Framework for evaluating the evolution of learners' thinking and practice in 21st century pedagogies, such as the one advocated by Globaloria. While the scope of this study was limited by the resources and the number of participants, future research should:

1. Include more subjects, as well as a control group. A sample size of 15 is not ideal for this type of qualitative analysis, and a comparison group would help to strengthen the claim that a game design curriculum can support the development of computational thinking skills.
2. Include quantitative data and analysis, especially if those data can be linked to academic achievement measures. This can assist in the generalizability of the findings.
3. Examine additional computational thinking constructs, particularly where overlap of constructs and dimensions may occur. For example, I observed instances of pattern identification and abstraction, debugging, and algorithm building during the Marble game decomposition task. Creating measurement tools to highlight these constructs may be useful in designing a curriculum for their development.
4. Conduct comparative studies of individual versus group work among students, as well as to contrasts between students at different levels of achievement over time.

As previously stated, the purpose of these analyses was not so much to examine whether students learned how to use Flash, as much as it was to understand how students learn to think and act with the tools (Pea, 1996), a skill fundamental in a 21st century where computational technologies mediate almost every aspect of life.

APPENDIX

APPENDIX

This appendix contains selected examples of students that best represented the overall themes observed during the assessment interviews.

<u>*Thomas*</u>

Starting with Thomas' pre-assessment, I began the interview by showing him the Marble game and how it is played. I gave the student an opportunity to play the game for himself, which hardly took any time at all, due to the simplicity of the game, and once the student had a chance to see the Marble game in action, I asked Thomas how he would start making the game. In much the same way that Brown and Campione laid out their curriculum, Fostering Communities of Learners (1996), which built on Vygotsky's zone of proximal development (1962), I asked questions so as to elicit information that the students presumably knew, yet had a hard time articulating on their own.

1 *Luke: What do you think are the first steps that you would do to start creating this thing?*

2 *T: Make the background.*

3 *L: Ok, so you'd make the background. And then what would you do?*

4 *T: Draw the marbles.*

5 *L: Ok, so make the background and then the marbles; then what's the next step?*

6 *T: Add all the coding.*

7 *P: Add all the coding. Ok, so what is involved in adding all the coding?*

8 *L: Trying to figure out how to make the **green ball** hit the **red ball***

Note in line 8 Thomas uses the terms "green ball" and "red ball" instead of green and red "marble". Although easy to overlook, his use of differing terms here shows a lower level of connection to the cultural interpretation of the system that he sees (the Real Player Dialog). In terms of decomposition, Thomas is breaking down the game into simple shapes, not specifically marbles. However, later in the interview, he did start to use the term "marbles". Going past that, Thomas was not able to break the system down any further, and while searching the wiki, he was not able to find any code that he could apply to the game. To summarize Thomas' pre- assessment, he was able to decompose the system in both its first and second levels, which I have previously defined as the marbles and background, and some sort of code.

Moving forward to the post-assessment, Thomas ended up being one of four students who were able to identify scene-switching as a necessary component in the Marble game. In terms of decomposition, this shows us that Thomas has learned more about how Flash works. This makes sense, particularly in the context of the Globaloria classroom. The games that the students produce always contain a Title Screen, which including the name of the game and its authors. The title screen will also include buttons labeled *Instructions*, *Play*, and *About*, which will take the player to those scenes. The creation of a complete game in Globaloria necessarily means that students understand how to make scenes switch in Flash.

By further breaking down the system to deeper level, Thomas is showing that he knows that getting the game to switch from the introduction scene to the scene where the game actually takes place is a critical part of recreating the Marble game. In terms of the actual execution, Thomas was not actually able to make the game switch scenes; he ultimately gave up on that task and moved on to dealing with the problem of how to make the marbles move, which was the point at which he got stuck during his pre-assessment. One thing to note during this process, although it is not strictly related to the process of decomposition or computational thinking, is the fact that

Thomas organized the initial layers, which he used for the introduction screen and the Start button, into a folder on the timeline and then created new layers for the gameplay scene, which he then organized into its own folder. While this was entirely unnecessary, it shows that he has developed a habit that professional game designers and animators frequently exhibit to keep their work organized, which is a positive trend. Returning to the process of making the marbles move, Thomas was still unable to work that out. I eventually had to move on to the next activity in the interview after it became apparent that the process of making the marbles move was outside of his zone of proximal development, and Thomas would not be able to complete it, regardless of how much guidance he could be given.

Elena

Similarly to Thomas, Elena was asked if she thought that she could recreate the marble game on her own and identify what the main parts of the game were.

1 *L: What do you think are the main parts of this...if you had remake this?*

2 E: Um, the background colors.

3 L: The background colors...then what would you do?

4 E: Um, the shapes.

5 L: The shapes.

6 E: Um, the writing?

7 L: Mmhmm, the text...

8 E: The movements.

While the questions are essentially the same, the answers are quite different. Elena uses the term "background *colors*", the "shapes", and the "movements", whereas Thomas used the terms "background", "marbles", and "coding". Elena's answers show how focused she is on the visuals and the graphics; Thomas' answers indicate that he is thinking more about how the system actually works (I acknowledge that although I have pointed out that Thomas used incorrect terms (*ball* instead of *marble*), he was still using those terms in the context of trying to figure out how the system works). Elena's focus on aesthetics became apparent as she began to build the Marble game. Instead of using the given assets, Elena created her own assets, which consisted of a light-blue background, with purple squares and stars (Figure 9).

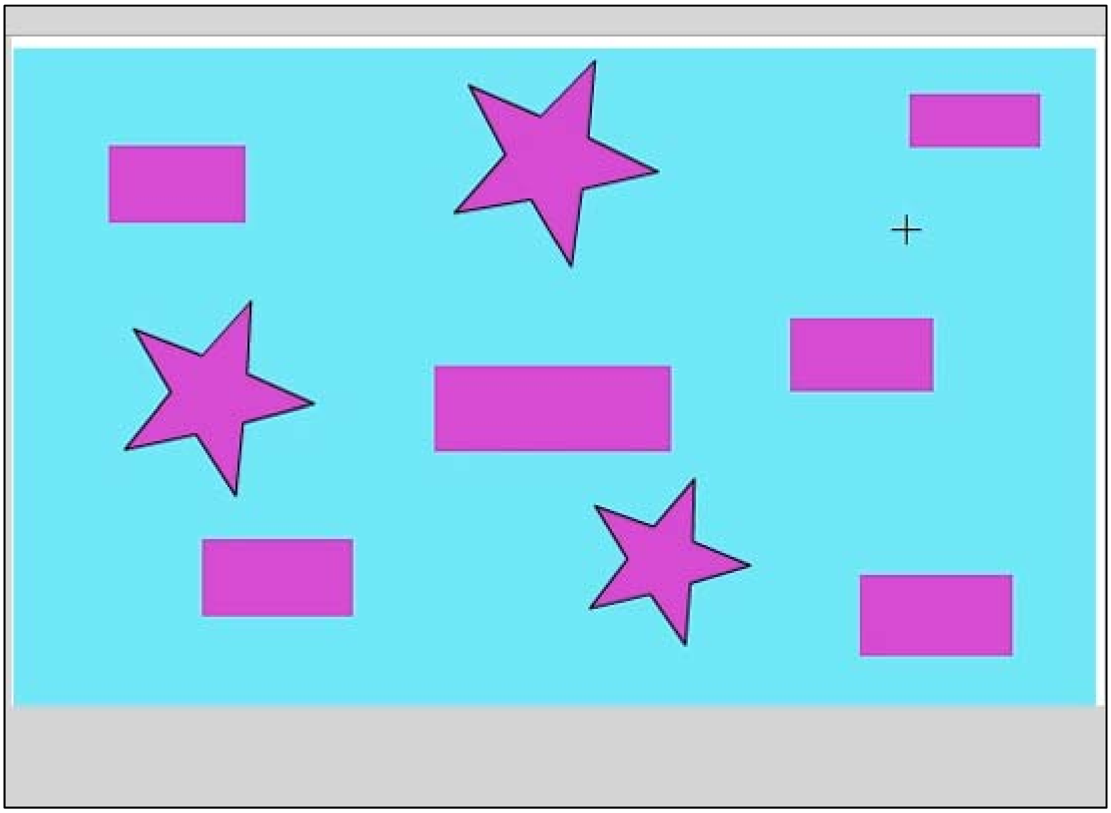


Figure 9: Elena's pre-test marble game.

This was a lengthy process, taking just over 5 minutes to complete. In terms of Games' Three Dialog framework, this aligns with the Real Player Perspective, and shows us that, similar to Thomas and the ball/marble error, she is displaying a lower cultural connection to the Marble game. As an analogy, this would be similar to tasking a student with drawing a dog and getting a picture of a cat because it also has four legs and a tail. This demonstrates that this student was particularly focused on the aesthetics of the game, which, in terms of decomposition, shows a different interpretation of that system.

Before moving on to discuss the post-assessment, it is worth noting that Elena also typed the word "instructions" on her background. While the specific term, "instructions", is not found anywhere in the Marble game, this indicates to us that she is applying a pattern to this game that she has seen in other games. The games that they produce in their Globaloria classes very often have a title screen with a button for instructions. In terms of the Real Player Dialog, she is thinking of the Marble game in terms that she is familiar with and understands. Beyond that, it is a good rule of thumb for a game designer to provide instructions to the player if the instructions are not built into the gameplay itself. In doing this, Elena is "behaving like a professional", similar to the way in which Thomas organized his layers into folders. These habits are habits of professionals and are most likely a product of the Globaloria curriculum. In support of this point, it makes sense that Elena would at least attempt to add instructions to her game, as all of the Globaloria students' final games had a button for instructions built into their games. During the post-assessment, little changed, in terms of her focus on the assets. She was again much more interested in making sure that she made good looking assets and less focused on getting the marbles to actually move. As an example of this, she highlighted the marbles in the library (which are already converted to symbols), and instead of dragging them onto the stage, she used them to figure out what color they should be. Using the pre-drawn and pre-converted assets as a template, she found the correct colors and used the oval/circle tool to create her own marbles, which ended up being ovals of various sizes. With regards to decomposition, she is aware that the marbles are a necessary component of the game, which covers the first level of decomposition outlined above.

REFERENCES

REFERENCES

1. Brown, A.L., & Campione, J.C. (1996). Innovations in learning: New environments for education. In L. Schauble, & R. Glaser (Editors), *Psychological Theory and the Design of Innovative Learning Environments: On Procedures, Principles, and Systems* (pp. 289-325). Hillsdale, N.J, England: Lawrence Erlbaum Associates.

2. Cottrell, N.B. (1968). Performance in the presence of other human beings: Mere presence, audience, and affiliation effects. In E.C. Simmel, R.A. Hoppe, & G.A. Milton (Eds.), *Social Facilitation and Imitative Behavior* (pp. 91-110). New York: Holt, Rinehart & Winston.

3. Engstrom, M.E., and Jewett, D. (2005). Collaborative Learning the Wiki Way. *TechTrends*, *49(6),* 12-15.

4. Games, I.A. (2008). Three Dialogs: a framework for the analysis and assessment of twenty- first-century literacy practices, and its use in the context of game design within *Gamestar Mechanic*. *E-Learning and Digital Media, 5(4)*, 396-417.

5. Games, I.A. (2010). Gamestar Mechanic: Learning a designer mindset through communicational competence with the language of games. *Learning, Media and Technology, 35(1)*, 31-52.

6. Games I.A. and Squire, K.D. (2011). Searching for the Fun in Learning: A Historical Perspective on the Evolution of Educational Video Games. In S. Tobias and J.D. Fletcher (Editors), *Computer Games and Instruction* (pp. 17-46). Charlotte, N.C.: Information Age Publishing.

7. Gee, J.P. (1992). The social mind: Ideology and social practice. New York, Bergin & Garvey.

8. Gee, J.P., Hull, G.A., Lankshear, C. (1996). *The New Work Order: Behind the language of new capitalism*. Sydney, Australia: Allen & Unwin.

9. Gee, J.P. (2005). *An Introduction to Discourse Analysis: Theory and Method*. New York, NY: Routledge.

10. Geertz, C. (1973). Thick description: Toward an interpretive theory of culture. In *The Interpretation of Cultures: Selected Essays* (pp. 3-30). New York, Basic Books.

11. Globaloria (2011). *What is Globaloria?* Retrieved from http://www.globaloria.org/intro#WhatIsGlobaloria.

12. Google. (2010). What is Computational Thinking? Retrieved from http://www.google.com/edu/computational-thinking/what-is-ct.html.

13. Hayes, E.R., & Games, I.A. (2008). Making Computer Games and Design Thinking. *Games and Culture, 3(3-4)*, 309-332.

14. National Academy of Sciences (2010). Report of a Workshop on the Scope and Nature of Computational Thinking, Washington, D.C.: National Academies Press.

15. Papert, S., & Harel, I. (1991). Situating Constructionism. In I. Harel and S. Papert (Eds.), *Constructionism* (pp. 1-12). Norwood, NJ: Ablex.

16. Pea, R. (1997) Practices of Distributed Intelligence and Designs for Education. In G. Salomon (Ed) Distributed Cognitions: Psychological and Educational Considerations. Cambridge, UK: Cambridge University Press.

17. Stake, R. (1995). The art of case research. Newbury Park, CA: Sage Publications.

18. Vygotsky, L.S. (1962). Thought and language. Cambridge, Mass: MIT Press.

19. Vygotsky, L.S. (1978). Interaction between learning and development. In *Mind and Society* (pp.79-91). Cambridge, MA: Harvard University Press.

20. Wertsch, J.V. (1998). Mind as Action. New York, NY: Oxford University Press.

21. White House. (2009). President Obama launches “Educate to Innovate” campaign for excellence in science, technology, engineering & math (STEM) education. Retrieved from the White House website: http://www.whitehouse.gov/the-press-office/president-obama-launches-educate-innovate-campaign-excellence-science-technology-en.

22. Wigfield, Allan and Jacquelynne S. Eccles. (2000). Expectancy-Value Theory of Achievement Motivation. *Contemporary Educational Psychology, 25(1)*, 68-81.

23. Wiggins, G. and McTighe, J. (2005). *Understanding by Design*. Washington DC: Association for Curriculum Development.

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

25. Wing, J.M. (2010). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society, 366*, 3717-3725.