Virtually There: Emerging Designs for STEM Teaching and Learning in Immersive Online 3D Microworlds

Sneha Veeragoudar Harrell (Chair, Organizer), Dor Abrahamson, Graduate School of Education,
University of California-Berkeley, Berkeley, CA 94720-1670
Email: om_sneha@berkeley.edu, dor@berkeley.edu
Leonel Morgado, GECAD, Micaela Esteves, Dep. Engineering,
Universidade de Tras-os-Montes e Alto Douro, Portugal
Email: leonelm@utad.pt, micaela@estg.ipleiria.pt
Martin Valcke, Hendrik Vansteenbrugge, Department of Education,
Ghent University, H. Dunantlaan 2, B9000 Gent (Belgium)
Email: Martin.Valcke@UGent.be, Hendrik.VanSteenbrugge@UGent.be
Eric Rosenbaum, Media Lab, MIT, Cambridge, MA 02139 eric_r@mit.edu
Sasha Barab (Discussant), Indiana University, sbarab@indiana.edu

Abstract: Four research projects used Second Life™, a 3D virtual-world platform, to investigate aspects of technology-enhanced STEM education. These European and USA studies, which differ in their pedagogical-philosophy commitments, theoretical frameworks, methodologies, and target content, critically examine a range of cognitive, affective, technical, and social factors pertaining to the prospects of students' and teachers' successful engagement with immersive microworlds. Specifically, each project describes students' successes and challenges in creating complex virtual artifacts and collaborating in real time with peers and the broader community. The design-based research studies of mathematical and computational literacy present sample student artifacts and discuss the learning they evidence. Collectively, we posit that overcoming the following obstacles could make virtual worlds both effective and exciting learning environments: professional development (technical skill, affective disposition), collaboration with school systems (logistics of access, allocation of resources), alignment with targeted content (harnessing students' creative divergence), and initial learning curves (issues of teacher-to-student ratio).

Overview of Symposium Panel

Recent technological developments have created opportunities for education researchers to evaluate the prospects of virtual worlds as arenas for facilitating STEM designs. Motivating factors for utilizing this new medium include its plasticity—it could potentially augment traditional learning environments by immersing learners in accessible reconstructions of real, confabulated, or hybrid loci that are remote in place, time, and/or scale, such as foreign lands complete with their geographical features, ancient archeological sites that come to life, sophisticated laboratories for safe handling of any contraptions and chemicals, microscopic organisms enlarged a million-fold, or galaxies reduced to neighborhoods. Yet as with any excitement created by "cool" technology, comes the sobering research-based acknowledgment that education practice must adapt prudently so as to assimilate the media in ways that best serve students. This symposium presents findings from a total of four European/USA research projects that study cognitive and affective factors contributing to students' and teachers' development of mathematical and computational literacy. Varying across participants, pedagogical commitments, theoretical frameworks, and methodology, the projects have in common a utilization and critical examination of the 3D virtual world as a medium for implementing designs for teaching and learning STEM content. Also, all of the learning environments were developed within the virtual worlds (*Teen*) *Second Life* (SL/TSL) (Linden Research, 2007). Finally, all projects seek to provide contexts that leverage students' natural social inclinations, cater to a broad spectrum of initial capacity and idiosyncratic interests, position mathematics and/or computer-science content as conducive to the solution of suggested or emergent authentic problems, and provide formative-assessment infrastructure (cf. Barab, et al., 2007). Briefly, each project is described below:

- Morgado and Esteves explore SL as a platform for teaching and learning introductory computer programming in Computer Science (CS) undergraduate courses. The main focus was transposing programming concepts and evaluating teacher and student needs by using an iterative action-research process. Morgado and Esteves describe how observations of unanticipated events led to parallel small-scale inquiry-based research efforts.

- Veeragoudar Harrell and Abrahamson are conducting design-based research on critical-pedagogy frameworks, computational literacy, intellectual identity, and mathematical agency. Their project, *Fractal Village*, is an experimental unit in which at-risk student participants learn core computer science concepts and mathematical reasoning through collaboratively constructing a community within a virtual world.
- Valcke, Vansteenbrugge, and Veeragoudar Harrell describe the *Second Life Impact on Beliefs and Anxiety* study, which utilizes a richly social and visually interactive graphical environment to support and investigate student–teacher development of conceptual/procedural knowledge and affective disposition toward mathematical content.
- Rosenbaum reports on *Scratch for Second Life* (S4SL), an innovative interface that lowers the barrier for creating interactive, dynamic content in *Second Life* by enabling users to program SL objects using a graphical building-block language (Maloney et al., 2004).

Our collective preliminary findings in these ongoing projects attest to the potential of these immersive learning tools to create opportunities for students to engage creatively with STEM content. We have witnessed multiple entry points to collaborative learning with students sharing resources, communicating, and even selling objects they created. However, we all still face serious implementation challenges due to issues of coordinating platforms, access to technology, professional development, and teacher-to-student ratio demands.

The symposium will begin with a 5-minute introduction by the chair. Each of the four ensuing 14-minute presentations will report major findings, demonstrate the content learned, and respond to audience questions. Thereafter, we will have a 14-minute participatory component in which attendees will be invited to personally interact with any of the technologies. The symposium will close with a 15-minute critical summation by our discussant, Dr. Sasha Barab (IU), a leading design-based researcher with particular expertise in online environments.

Abstracts of Panel Participants
Second Life for Teaching and Learning Introductory Computer Programming

Leonel Morgado and Micaela Esteves

A substantial amount of research has been conducted on teaching & learning introductory programming. A recent worldwide analysis (Schulte & Bennedsen, 2006) focused on themes and challenges of learning CS content, in particular as perceived by teachers: programming is a difficult subject to learn, typically taught using traditional programming paradigms; major hurdles include basic topics such as parameters, references, and pointers; typical teaching focuses on language specifics and coding, albeit design topics are also expected to be learned; teachers are ambitious, expecting students to master at a high level about 80% of the topics; and students are not equipped for tackling problems demanding high-level abstraction. Other approaches focus on issues such as motivational elements (*e.g.*, Leutenegger & Edgington, 2007) or the theoretical perspective of CS instruction (East, 2006). These challenges team to hinder the professional and academic progress of CS students, as repeatedly reflected in reports of CS students cognitive struggle with programming content, negative affective disposition toward it, and consequential systematic avoidance of programming projects or career paths involving programming (Miliszewska & Tan, 2007). Research on teaching approaches to overcome these issues has provided recommendations, *e.g.*: provide students well-written code examples; analogies work well to illustrate unknown concepts (*ibid*).

The study presented here aims to improve programming education, by attempting to combine various prior recommendations, namely to provide a graphically appealing environment that expresses students' code without increasing its complexity, thus enabling the development of programming abstractions within a richer context, increasing the availability and participation of faculty in students' projects, and enhancing real-time student cooperation. We report on unexpected yet significant observations on how SL's culture and society impact students' programming experience, motivation, and learning.

Second Life as a programming environment

SL programming is performed in Linden Scripting Language (LSL), which has C-style syntax and keywords (AA.VV., n.d.). 3D objects created in SL can receive scripts, executed concurrently. Each script has its own state-machine: programming is done by triggering events or responding to them (by environment interactions or programmatic components). The programmer defines the states of each state machine and how/when to change over. The language's programming libraries include communication with external servers: e-mail, XML remote procedure calls, and HTTP support, which is being used extensively by many developers to supplement SL's programmability with Web services hosted externally, and to use SL as an alternative interface to external services. Figure 1 presents a sample programming session: the avatar in the center has just created a sphere. On the left, the grayed section shows the sphere's contents (currently, just one script, a "Hello, World" example). By double-clicking, that script was opened as another window (top right). The dotted lines emanating from the hands of the two avatars indicate that both are editing the same object--and indeed both can be programming it, even sharing the same code.

Second Life to learn introductory computer programming

One can consider two approaches to learning introductory computer programming using SL: (1) replace traditional languages and environments altogether by the SL environment and LSL code; and (2) use SL as a source of additional context for projects based on a traditional programming environments/languages. This report focuses on approach #1 (see below), but we have also pursued #2 on a small scale. Namely, we taught students participating in a course focusing on Windows application development in C# how to program SL objects to send e-mails. E-mails were sent in two situations: when an avatar touched them, and when an avatar was detected within 2 meters. This SL program can be created in approximately 5 lines of code. For the full semester, the students developed a Windows-based marketing management application on "interested customers" and "potential customers," but instead of using abstract made-up text files of data, they employed SL to generate e-mails with "real" data: whenever an avatar would touch an object, the action would be reported as an "interested customer"; when an avatar was detected within 2 meters, the action would be reported as a "potential customer". At the end of the semester, students responded to a feedback questionnaire regarding their experience.

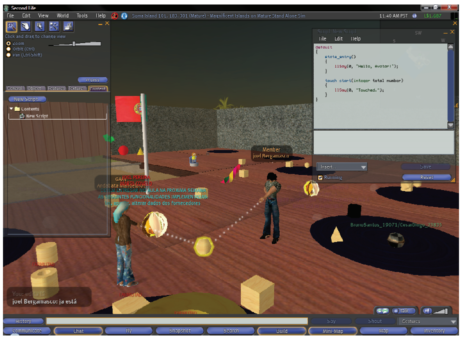

Figure 1. Collaborative SL programming.

Preliminary findings

For our research on teaching entirely within SL, we are following an action-research methodology. Two initial settings early in 2007 lasted one semester: Setting *A* included 5 beginners (1^{st}-year students), and Setting *B* included 4 somewhat more experienced students (2^{nd}-year), who albeit familiar with programming concepts had no experience developing a semester-long project on their own (Esteves et al., 2006). Later in 2007, settings *C* and *D* further pursued this approach, but now participants filled-in a questionnaire on their perspective on programming before being given the option of electing SL as a development platform. The same questionnaire was filled-in by all other non-participant students of the same course, as a comparison measure between study and control groups.

Preliminary results from these efforts indicate that students do not necessarily welcome the graphics richness of SL: most students do have a positive or neutral attitude towards it, but a minority dismisses it as "unserious," "awkward," or "complex." Students learning how to program by programming physical interactions in SL (e.g., making a dog follow you and obey your voice commands) are typically motivated; and students who focused primarily on non-visible techniques such as data structures and string processing, and who benefited from the environment just for enhanced context and not as a source of feedback for programming behavior, did not seem to display any motivational advantage over students employing a traditional console-oriented (text-only) approach. Two unexpected events have impacted students' engagement in setting *B*, as we now describe. A student received a proposal by an avatar in SL to buy his programming assignment, not as a violation of student conduct but as a sanctioned exchange of virtual commodities; another student received a professional proposal to provide SL programming services for a company. Scripting, we learned, is a marketable skill in SL even at an introductory level. By deploying our students in a publicly accessible virtual world, thus, we inadvertently exposed them to the powers of an authentic micro-economy, and these social forces serendipitously contributed to our students' "on-the-job" engagement in developing the core skills targeted by our experimental unit.

These unanticipated anecdotal experiences suggest a curious paradox. SL, a cutting-edge technologically advanced 3D immersive environment, may harbor opportunities for powerful "back to basics" learning of introductory programming, in terms both of content and context: *content—SL* coding employs simple devices and practices, without complex development environments or complex compiler errors and with immediate perceptual feedback; and *context*—beginners' applications are quite similar in look and feel to experts', as was the case in the days before graphical user interfaces, and programming is a marketable skill greatly valued by the SL community. At the same time, this virtual simplicity includes advanced elements such as concurrent execution, encapsulation, and non-programming interface elements (3D modeling), which still patently mark expert–novice differences. Whether, as a community of researchers, we will be able to build on the environment's pedagogical opportunities and successfully address its inherent challenges for CS learners, is a question demanding further research.

It Takes a Virtual Village: Transforming Urban-Youth Intellectual Agency Through Critical Computational Literacy

Sneha Veeragoudar Harrell and Dor Abrahamson (http://edrl.berkeley.edu/projects/fv/)

Gordon and Bridglall (2006) have articulated 'affirmative *development'* as a framework for educational activism. Fostering student agency, they emphasize, is core to this effort, particularly with respect to STEM content. Inspired by this call, we designed and implemented a critical/constructionist-pedagogy learning environment, *Fractal Village* (Veeragoudar Harrell & Abrahamson, 2007) that constitutes both an empirical environment for research on an emergent model of *mathematical agency* (Veeragoudar Harrell, 2007) and a potential means of fostering such agency. In Fractal Village, students interface through programming procedures to engage collaboratively in TSL-based imaginative construction activities. The purpose of the materials selected/built for this study was to create context for activities eliciting students' *generative themes* (Freire, 1968) that the designers-as-teachers could then reflexively and strategically match with mathematical concepts (e.g., variables, functions) and computer-science concepts (e.g., recursion, looping), such that students appropriate the STEM content *apropos* of tackling emergent construction problems identified and articulated by the students. Key research objectives are to: (1) study relations amongst cognitive, affective, material, technological, and social factors apparently contributing to mathematical agency (e.g., content knowledge, procedural/media fluency, discourse practices, and self-image); (2) delineate design principles for fostering mathematical agency; (3) implement within a school a sustainable critical-pedagogy program in collaboration with school leadership and personnel; and (4) investigate benefits and limitations of an exciting new technology. We are particularly interested in students' navigation of the multiple identities that this medium evokes (Gee, 2003)—in the physical world, in the virtual world, and, crucially, in their liminal intersecting spaces (Zuiker, 2007)—and how these may be leveraged so as to support students who have not developed academic practices to find a new voice and, so doing, develop STEM fluencies in voicing their *e-merging* identities. We have found that 'liminality'—a construct first coined by the anthropologist Victor Turner (1967) to depict psycho–social in-between spaces that cultures create for rites of passage (e.g., manhood, bereavement)—captures virtual worlds' mixture of outlandish, programmatic, and transformative qualities that *displace* (Blikstein, in press) students, inviting them into cocoons that foster their change into empowered agents of their own prospects.

Methodology

The participants in this study were a class of thirteen 15 – 22 years old students from an urban California alternative high school. All students at this school enroll because they have been expelled from the local mainstream schools. Students at this school are and mostly African–American and Latino–American, and all qualify for federal free lunch programs. Over half of the students in the class we worked with have Individual Education Plans (IEP) due to being categorized as Special Education students. We worked with the class over fourteen 110-minute sessions. The last four sessions were spent with students learning html in order to build a customized web page that presented the work they had accomplished in the intervention. Additionally, the students took a field trip to Linden Research Laboratory and made a final presentation to a research group at our university. Raw data collected in this study consist of: digital movies of collaborative work, screen-capture movies that archive every keystroke/mouse-click produced by each student, students' journal logs, movies of day-by-day individual interviews with a set of focal students, rich field notes, and participant-generated mixed-media artifacts, i.e., worksheets, modeling constructions, and computer screenshots (see Figure 2, below). Also, we videotaped the design-team's debrief/plan sessions. We performed microgenetic, qualitative analysis so as to reconstruct episodes particularly informing of ways in which mathematical-agency factors interacted in and between the real and virtual worlds. Further, using grounded theory (Glaser & Strauss, 1967), we searched for patterns that developed into new theoretical constructs.

Findings

Engagement and collaboration. By and large, students persevered in complex construction involving programming and mathematics content. Of note, students who previously had been diagnosed as unable to focus on any single activity for more than 20 minutes, exhibited persistent engagement with their artifacts for up to 90 minutes. For example, a student who had created a large glass building (see Figure 2, second from left) wanted to build stairs between the floors. He explored multiple solution paths, one of which included implementing recursion—an important computer science concept—to build a procedure that generates stairs of customized length. He then shared his code, and his cohort adapted it, each to their respective needs within the context of their projects, in "open source" professional spirit and practice. Indeed, whereas in the real world the students preferred to work independently, in the virtual world they shared objects, scripts, and skills they had developed. Such inter-space differences in collaboration practices, we believe, are complexly related to students' *e-merging* identities.

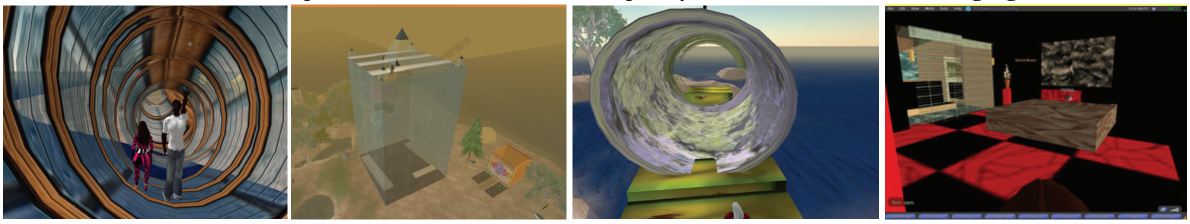

Figure 2. Student-constructed objects: tunnel, building, bridge, home.

Identity clashes, onset of agency. Reputation as a successful student in traditional curricular activities may inversely predict success in alternative activities. Further, we have implicated students' engagement of imagination as causal, or at least indicative, of successful transition. Namely, students across all skill levels who "let go" and designed fanciful constructions were more inclined to experiment with the technology, take on demanding tasks, and develop positive dispositions toward the content and procedures.

Within- and between-cohort liminality and the co-construction of mathematical agency. Designing for liminal experiences—particularly in the context of implementing critical/constructionist pedagogy design in complex learning environments—requires substantial, genuine one-to-one teacher–student relationships. Indeed, students and teachers cited the researchers' trust and caring as key to student engagement. Yet, we discovered, effective critical design begins at home… Namely, in working to create liminal experiences for students, as it turned out, the five researchers—who also come from diverse backgrounds—necessarily created *for themselves* liminal experiences that mirrored and anticipated students' experiences. That is, while the student cohort dynamically cohered as a community of real/self-projected personae in a safe space enabling pretense play, reflection, therapy-like conversations, and learning, a similar space for collaborative introspection was instantiated in the researchers' meetings back at the laboratory. Through these transformative meetings, some of the researchers were first able to deeply sympathize with the students, probably because their own corresponding issues were tapped.

Toward sustainable design. Implementing Freirean–Papertian pedagogy with integrity, we learned, is highly demanding: in order to identify each student's idiosyncratic wells of expressivity and match target content seamlessly to those kernels, it truly takes a village of design-researchers collaborating with a principal and teachers. We are currently examining frameworks for consolidating our curriculum within the school, e.g., students become mentors for the next implementation, while we port the curriculum to other schools and scale it beyond.

The Impact of Second Life Experiences on Teachers Mathematical Beliefs and Anxiety

Martin Valcke, Hendrik Vansteenbrugge, and Sneha Veeragoudar Harrell

Mathematics education for pre-service teachers must address teacher beliefs and affect in addition to their acquisition of knowledge and skills (Schoenfeld, 1985). We examine the potential of a constructionist learning environment, *Fractal Village* (Veeragoudar Harrell & Abrahamson, 2007), to support teachers' alignment of their beliefs with those conducive to effective practice (e.g., Langrall, Thornton, & Malone, 1996). The study's rationale is to pretest teachers' beliefs, have them engage in SL activities, then posttest them and evaluate for any change.

The interaction between virtual learning environments, cognitive processes, and affective variables can be described by building on the expectancy-value theory of Wigfield and Eccles (2000). Their model implies that beliefs consist of affective components, goal orientations, competency judgments and perceptions about the tasks to be carried out. These beliefs are influenced by cognitive processes about attributions and earlier experiences. This complex set of beliefs, affect, and cognitive processes is influenced by experiences within the learning environments and the wider instructional context (appraisal, success experiences, values of peers, teachers, parents). These beliefs, in turn, impact teachers' choices, persistence, and engagement in specific teaching contexts. The model also delineates reciprocal dynamical relations between external and internal processes.

Theoretical framework: Toward impacting teachers' anxieties toward mathematics

A key component of teachers' identities are their beliefs (Beijaard, Meijer, & Verloop, 2004). Ernest (1989) reviewed studies focusing on teachers' beliefs about mathematics and concluded that three belief components have a critical influence on mathematics-related instructional practices: the nature of mathematics, the nature of mathematics teaching, and the process of learning mathematics. Askew, et al., (1997), found that highly effective primary school teachers embodied constructivist beliefs that are in sharp contrast to 'transmission' beliefs of less successful teachers. Quillen (2004) discussed contrasts between *relational beliefs* versus *instrumental beliefs*. There is also a growing interest in the theoretical role of affective variables within mathematics, such as math anxiety. Research demonstrates how mathematics anxiety inhibits cognitive processing (Ashcraft & Kirk, 2001). With an eye on remedial intervention, Bragg (2006) advocates the relevance of virtual- and game-based environments to influence mathematics learning.

Design of the study

Fifty undergraduate students enrolled in a mathematics-education course are involved in the study. During a five-week period, the students, after being introduced to SL technology, begin activity in Fractal Village by discussing jointly the future of the village and what roles individuals can take on and what objects individuals can contribute to the village. Students explore a variety of seed fractal structures in *the virtual environment* that allow them to generate fractal objects through an easy-to-use graphical interface. Participants are scaffolded towards constructing their own fractal seeds that they can share with others and may complete these tasks independently or collaboratively, e.g.,. through sharing code or objects with each other to facilitate building progressively more complex objects. Participants also engage in evaluating whether objects created by their peers are in fact fractals. At every juncture students are engaging in collaborative learning, discussing and sharing, while being accountable for contributing to the village. With the help of a facilitator, mathematical language for describing properties of fractals is integrated into normal discourse.

Participants respond to 3 pre/post activity instruments, and focal students are interviewed on a weekly basis throughout the entire 5 weeks. The three pre/post activity instruments are: (1) a scale for determining participants' level of experience with computers, the Internet, and virtual Internet-based environments; (2) a mathematics anxiety scale (the Revised Mathematics Anxiety Rating Scale, MARS, Plake & Parker, 1982) and (3) an instrument for determining participants' beliefs about mathematics (the Mathematics Beliefs Questionnaire, developed by Lazim, Osman, and Salihin, 2004). Participant demographics are collected; the project runs Jan. – April, 2008.

Analysis and results

Analysis will involve quantitative as well as qualitative data. Quantitative analysis will be based on path analysis techniques (SEM) to measure the direct impact of initial math beliefs and the math anxiety level and the mediating impact of student characteristics on the dependent variables. Analysis of the qualitative data will centre on the semi-structured clinical interviews with focal students and examination of artifacts produced by participants. Specifically, we will look for data that grounds our understanding of how students mathematical anxieties and beliefs change over the course of the five weeks.

Scratch for Second Life

Eric Rosenbaum

Virtual worlds let people make things together in new ways: they provide new social and technological affordances for collaborative construction of complex artifacts in a shared space. Programming is one way to make these artifacts more meaningful, by adding behaviors and interactivity which bring them to life. Unfortunately, the barrier to entry for programming has typically been high. Scratch for Second Life (S4SL) extends the graphical building-block programming language developed in the Scratch project (Maloney et al, 2004) to make programming easier in SL. Here I report on initial field tests of S4SL. I suggest ways that programming can enhance constructive learning in virtual worlds, and ways that the social context provided by virtual worlds can enhance the learning of programming.

Scratch is a programming language designed to allow young students to make their own animations, games, and interactive art and share them on the web. Programs authored in *Scratch* consist of stacks of puzzle piece-like blocks, each representing a command. The blocks are divided into colored categories such as “motion,” “sound,” “sensing,” and “variables.” Users create programs by snapping the blocks together in a sequence.
S4SL is a modified version of *Scratch* that uses the same underlying software, but with a different set of blocks that are specific to actions in *Second Life*. The blocks are translated into their equivalent in Linden Scripting Language (LSL), which can be attached to a *Second Life* object. Figure 3, for example, shows a set of S4SL blocks representing behaviors for a bunny. The code and bunny were created by a participant in the first field test of S4SL. The first stack of blocks causes the bunny, when someone clicks on it (“touching” is clicking in *Second Life*), to hop toward its owner, and say the word “hop” as it does so.

Figure 3. *Scratch for Second Life* blocks (left) defining behaviors for a bunny rabbit in *Second Life.*

The next stack is a “forever” stack, so it runs continuously. It is always checking if the bunny is more than five virtual meters from its owner. If it is, it hops toward the owner. The final stack makes the bunny apologize when it bumps into someone. The equivalent LSL code for these behaviors is several dozen lines of text, compared to just sixteen blocks in S4SL. In addition to being shorter and appearing less complicated than LSL, the S4SL code is designed to be easier to construct, understand and modify without errors.

How can virtual worlds help people learn to program?

Programming is typically a solitary activity. By contrast, the rich social context of a virtual world provides many benefits to a person learning how to program, from communication with people who can provide both emotional and technical support, to a vast landscape of situated examples and an audience to admire and critique one’s work (Bruckman, 1998). All these factors have proven important to the growing community that is forming on the *Scratch* website (Monroy-Hernandez, 2007), where people can interact with each others’ creations, download and modify each others’ programs, and communicate through comments and
forums. We hope that S4SL will open up new possibilities for collaborative learning. We imagine groups of learners collaborating to build and program together, manipulating and sharing objects and code seamlessly and synchronously in a shared virtual space.

Pilot Study

A pilot study was performed on S4SL, with a group of 10 students in an adult education course on virtual worlds. Most had no programming experience. All had used *Second Life* to participate in class, and completed an assignment using *Scratch* during the previous week. They were given a one-week assignment to build an object in *Second Life* and assign to it a behavior using S4SL. They were shown a brief demonstration of S4SL and offered assistance during office hours in *Second Life*.

The goals of this early test included evaluating the software for basic functionality, finding bugs, and ensuring that the interface and connection to the *Second Life* client (via copy-and-paste) were usable. We also hoped to see what people are able to create, find out from them how the experience felt and what they would add or change about the software. The group was chosen partly out of convenience (students under 18 are not allowed on the main *Second Life* grid) and partly because the actual audience for the application will end up being both children and adults.

Overall, the students succeeded in building objects such as creatures and buildings and creating scripts for them (e.g. the bunny in the figure). The scripts were mostly very simple. Most of the objects moved, made sounds, and reacted to touches or chat commands. Four of ten students attached scripts to more than one object, and two of those used the "broadcast" block to make the objects communicate with each other. Most students reported satisfaction and pleasure at the ease of creating their scripted objects. Some students were frustrated, aside from technical glitches, by a sense that expected functionality was missing from S4SL. Some students reported a significant sense of empowerment from being able to create their own scripted objects.

Future Directions

We are currently researching ways for people to share S4SL code within *Second Life* in order to enable fluid collaborations between people sharing S4SL programs, objects and ideas. Future studies will focus on fostering these collaborations and evaluating them. We hope to do additional studies with adults who are novices, adults who are already experienced *Second Life* builders but do not have programming experience, and with children.

References

AA. VV. (no date). *LSL Portal*. http://wiki.secondlife.com/wiki/LSL_Portal [November 17th, 2007].

Ashcraft, M. & Kirk P. (2001). The relationships among working memory, math anxiety, and performance. *Journal of Experimental Psychology*, *130*(2), 224-237.

Askew, M., Brown, M., Rhodes, V., Wiliam, D., & Johnson, D. (1997). *Effective teachers of numeracy: Report of a study carried out for the Teacher Training Agency.* London: King's College, University of London.

Barab, S., Zuiker, S., Warren, S., Hickey, D., Ingram-Goble, A., Kwon, E.-J., Kouper, I., & Herring, S. C. (2007). Situationally embodied curriculum: Relating formalisms and contexts. S*cience Education, 91*, 750-782.

Beijaard, D., Meijer, P.C., & Verloop, N. (2004). Reconsidering research on teachers' professional identity. *Teaching and Teacher Education*, *20*, 107–128.

Blikstein, P. (in press). Travels in Troy with Freire: Technology as an agent for emancipation. In P. Noguera & C. A. Torres (Eds.), *Paulo Freire: The possible dream*. Rotterdam, Netherlands: Sense.

Bragg, L.A. (2006). Students' impressions of the value of games for the learning of mathematics. In Novotná, J., Moraová, H., Krátká, M. & Stehlíková, N. (Eds.). *Proceedings of PME 30* (V. 2, pp. 217-224). Prague: Charles University.

Bruckman, A. (1998). Community support for constructionist learning. *Computer Supported Cooperative Work*, *7*, 47-86.

Ernest, P. (1989). The knowledge, beliefs and attitudes of the mathematics teacher: a model. *Journal of Education for Teaching, 15* (1), 113 - 33.

East, J. (2006). On models of and for teaching. In R. Anderson, S. Fincher, & M. Guzdial, (Eds.), *Proceedings of the 2006 International Workshop on Computing Education Research* (pp. 41-50). New York: ACM.

Esteves, M., Morgado, L., Martins, P., & Fonseca, B. (2006). The use of collaborative virtual environments to provide students contextualisation in programming. In A. Méndez-Vilas, A. Solano Martín, & J.A. Mesa González,. (Eds.), Current developments in technology-assisted education (Vol. 2, pp. 1496-1500). Badajoz, Spain: Formatex.

Freire, P. (1973). *Pedagogy of the oppressed* (M. B. Ramos, Trans.). New York: Seabury. (Originally 1968)

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

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory*. Chicago: Aldine Publishing Company.

Gordon, E. W., & Bridglall, B. L. (Eds.). (2006). *Affirmative development*. New York: Rowman & Littlefield.

Langrall, C., Thornton, C., & Malone, J. (1996). Enhanced pedagogical knowledge and reflective analysis in elementary mathematics teacher education. *Journal of Teacher Education, 47*, 271-282.

Lazim, M., Osman, A., a Salihin, W. (2004). The statistical evidence in describing the students' beliefs about mathematics. *International Journal for Mathematics Teaching and Learning*, October Issue.

Leutenegger, S. & Edgington, J. (2007). A games first approach to teaching introductory programming. *ACM SIGCSE Bulletin, 39*(1), 115-118.

Linden_Research (2007). *What is Second Life?*, http://secondlife.com/whatis/ [November 18th, 2007].
Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A sneak preview. In Y. Kambayashi , K. Tanaka, & K. Rose (Eds.), *Proceedings of the Second International Conference on Creating, Connecting, and Collaborating Through Computing* (pp. 104-109). Kyoto: Kyoto University.
Meijer, P.C., & Verloop, N. (2004). Reconsidering research on teachers' professional identity. *Teaching and Teacher Education*, *20*, 107–128.
Miliszewska, I., & Tan, Grace (2007). Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming*. Journal of Issues in Informing Science & Information Technology, 4*, 277-289.
Monroy-Hernandez, A. (2007, June). ScratchR: Sharing user-generated programmable media. Poster presented at the annual meeting of *Interaction Design for Children*, Aalborg, Denmark, June 6-8.
Plake, B.S., & Parker, C.S. (1982). The development and validation of a revised version of the mathematics anxiety *rating scale. Educational and Psychological Measurement, 42*(5), 551-557.
Quillen, M.A. (2004). *Relationships among prospective elementary teachers' beliefs about mathematics, mathematics content knowledge, and previous mathematics course experiences.* Unpublished doctoral dissertation. Blackburg, VA: Virginia Polytechnic Institute and State University.
Schoenfeld, A. (1985). *Mathematical problem solving*. New York: Academic Press.
Schulte, C., & Bennedsen, J. (2006). What do teachers teach in introductory programming? In R. Anderson, S. Fincher, & M. Guzdial, (Eds.), *Proceedings of the 2006 International Workshop on Computing Education Research* (pp. 17-28). New York: ACM.
Turner, V. (1967). *The Forest of Symbols: Aspects of Ndembu Ritual*. Ithaca: Cornell University Press.
Wigfield, A., & Eccles, J. S. (2000). Expectancy value theory of achievement motivation. *Contemporary Educational Psychology, 25*, 68-81.
Zuiker, S. (2007). *Transforming practice: Designing for liminal transitions along trajectories of participation.* Unpublished doctoral dissertation. Indiana University, Indiana.