Camp CyberGirls: Using a Virtual World to Introduce Computing Concepts to Middle School Girls

Caitlin Hulsey
School of Computing
Clemson University
Clemson, SC 29634-0974
cbhulse@clemson.edu

Toni B. Pence
School of Computing
Clemson University
Clemson, SC 29634-0974
tbloodw@clemson.edu

Larry F. Hodges
School of Computing
Clemson University
Clemson, SC 29634-0974
lfh@clemson.edu

ABSTRACT

In this paper we report on the design and results of a one-week, residential summer camp experience that introduced computing concepts to middle school girls in the context of an online, multiplayer, virtual world known as the Curiosity Grid. In contrast to programming environments designed specifically as teaching tools to introduce children to programming, virtual world programming exposes novice learners to a more representative computer science experience. Students write real code and get real syntax errors when their code is not correct. They also design objects in a three-dimensional world where knowledge of mathematical concepts such as 3-D global and local coordinate systems, and 3-D transformations are important to the creation of objects and behaviors. Programming artifacts from the camp and feedback from the camp participants provide a strong argument that middle school girls can handle the challenge of this approach and even be enthusiastic about it.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education - *computer science education*, *literacy*.

General Terms

Design, Human Factors, Languages

Keywords

Outreach, Middle School Girls, Virtual Worlds, Educational Programming Environments

1. MOTIVATION

A recent report sponsored by the Association for Computing Machinery and the Computer Science Teachers Association makes it clear that we are failing to prepare students with the computing skills that are needed, not only for jobs in computer science and information technology, but for almost every career in the information age [24]. The number of high school students who take AP Computer Science is a tenth or less of those taking AP Calculus or AP Biology [16]. For women the statistics are even worse. In 2012 less than 19% of students taking AP Computer Science were female [6]. This pipeline continues to shrink in college. Data collected from 152 US computer science departments indicate that less than 13% of bachelor's degrees in computer science awarded in 2012 were to women [25]. One of the problems with increasing student interest in computing as a major or career is that they have almost no experience with computer science as a discipline while in middle school and high school. They are required to take courses in subjects such as mathematics, literature, history, and science. However, computer science is not considered to be a core discipline that high school students must take, and no US state requires a computer science course as a condition of a student's graduation [24].

In response to the need to provide middle and high school students with knowledge of and experience in computing, many colleges and universities offer weekend workshops or one-week summer camps in computer science with goals of both teaching computing concepts to students and motivating them to consider majoring in a computing discipline in college [2,4,5,7,11]. These experiences are usually designed around programming environments that have been created specifically as teaching tools. Popular examples include Alice and its successors such as Storytelling Alice and Looking Glass [7,8,9,11,13], Scratch [2,4], and Lego Robots [5,14]. Common characteristics of these environments are that they are designed to be visual environments that can be used for development of simple games or animated stories, they employ a drag-and-drop metaphor, and they require little mathematical sophistication. Recently there has also been a trend to build companion websites to encourage and support long-term engagement of students in programming [9].

In this paper we report on a one-week, residential summer camp experience that introduced computing concepts to middle school girls in the context of a virtual world. Virtual worlds are online, multiplayer, user customizable via geometric modeling and programming, three-dimensional, social spaces where players are represented by avatars. The largest and best known virtual world is Second Life [22], but many other choices are available that range from publicly accessible worlds such as those hosted on OSGrid [19] or Third Rock Grid [1] in addition to worlds that are privately run by business, the military, or educational initiatives [10]. Many are used for educational purposes for a varied list of subjects. Examples include programs in virtual worlds designed to inform students about geosciences [20], forensic science [17], diversity [15], and international culture and politics [23].

SIGCSE'14, March 5–8, 2014, Atlanta, Georgia, USA.

ACM 978-1-4503-2605-6/14/03…$15.00.
http://dx.doi.org/10.1145/2538862.2538881

However, we could find no reviewed literature on using virtual worlds to introduce middle school students to computing! This is surprising since virtual worlds already include many of the characteristics that are being designed into programming environments for young students. Virtual worlds are visual environments with highly developed scripting (programming) and modeling tools. Since all interactions take place within an online, multiplayer virtual space, they are also social environments by default. Issues that may have discouraged their use for teaching computing concepts could be related to the relative sophistication required for scripting and modeling objects in virtual worlds. Programming in a virtual world is not a drag-and-drop exercise. Students write real code and get real syntax errors when their code is not correct. They also design objects in a three-dimensional world where knowledge of mathematical concepts such as 3-D global and local coordinate systems, and 3-D transformations (rotations, translations, and scales) are important to the creation of objects and behaviors. There are also issues of online safety for younger students if the environment allows interaction with unknown or unauthorized players.

2. CAMP CYBERGIRLS DESIGN

Camp CyberGirls was a one-week residential camp for middle school girls with the objectives of introducing them to computing concepts and encouraging their interest in computing with respect to their future college and professional aspirations. Students arrived on a Sunday afternoon, and the program ended the following Friday morning with a presentation to family and friends. Lab-based, hands-on sessions were held for two hours each morning and for three hours each afternoon. Two instructors were present for each lab session. The camp was offered through funding support provided by the Charles H. Houston Center at Clemson University [3] and the Clemson School of Computing [21]. The students lived in college dorms for the week, ate in the student cafeteria, and took part in organized activities outside of the computing lab sessions that included a welcome cookout on Sunday night, bowling night, game night and movie night. Staff from the Charles H. Houston Center recruited the students, provided supervision of the students outside of class, and organized the social events during the week.

2.1 Participants

Sixteen girls took part in the camp. Five of the participants were rising eight graders, nine were rising ninth graders, one was a rising seventh grader, and one was entering tenth grade. Eight were Caucasian and eight were African-American. All but one student were from schools in the tri-county area around Clemson University. IRB approved consent letters asking permission to collect data from pre- and post-questionnaires were sent to parents before the camp and provided to students on the first day of the camp.

2.2 Curiosity Grid

All computing activities for Camp CyberGirls took place in the context of the Curiosity Grid (CG) online virtual world using the Imprudence viewer [12]. Curiosity Grid is owned and operated by 3rd Rock Grid [1] and was created to support education in the virtual world. CG differs from many similar platforms in that it is a limited access grid with strict procedures in place to make it a safe environment for minor students. Citizens (players under 18) were provided accounts only after being recommended through the Virtual Environments Group in the School of Computing at Clemson University. Mentors (18 and over) must go through a background check before interacting with Citizens on the Grid. Avatars representing citizens all have a common last name that differs from the last name of avatars that represent mentors so that players know whether they are interacting with another student or an adult.

A virtual piece of real estate in CG is called an island and visually appears as an island in a large ocean (Figure 1). Locations on an island are specified in (x,y,z) coordinates where z represents elevation, x is an East/West aligned axis, and y is a North/South aligned axis as measured in virtual meters from the southwest corner of square regions of size 256 by 256 meters. Camp CyberGirls was held on an island made up of four of these regions.

Figure 1. Virtual island on Curiosity Grid

CG's capabilities and interface are based on software originally developed by Linden Labs that is now distributed as an open source platform. Programming is done in the Linden Scripting Language (LSL). LSL has syntax similar to C/Java. LSL is event driven, and includes states, 3D variable types, and many built-in functions [18]. Modeling objects is done by modifying and connecting basic geometric shapes known as prims [18]. LSL programs are always connected to an object in the virtual world and control that object's behavior.

Figure 2 illustrates the menu-based modeling format that allows a user to select and modify a prim. The user can control position, size and rotation of the prim with respect to a three-dimensional coordinate space, texture and color of the prim, and modify the shape of the prim in a number of other ways with menu-based tools for hollowing, tapering, cutting, slicing, and twisting. Individual prims can be linked to other prims to create more complex models. Figure 3 shows a mailbox that was modeled by a student from seven prims. Associated with the mailbox is a script written in LSL that displays a message above the mailbox, allows a player to leave items in the mailbox, and allows the mailbox owner to open the mailbox and see items left for them (Figure 4).

2.3 Implementation

At the start of the Monday morning session of the camp, participants took a survey that addressed their interest in majoring in a computing discipline and their attitudes to computing. Next, participants received computer accounts to log into the lab computers and Curiosity Grid accounts that included being assigned one of three "standard" avatar designs.

During their first session, the instructors took the girls through all the options that were available to them to customize the avatar that would represent them in Curiosity Grid. CG offers a large set of menu-based controls that allow users to personalize their avatar's appearance and clothing. Each girl started with one of three "standard" avatar designs: *Ella*, *Bonnie* and *Maria*. The tools provided allowed the girls to end with a unique avatar to represent them in the virtual world and express their creativity. Figure 5 shows final avatars created by three students who started out as the standard *Ella* avatar. This exercise was followed by instruction on how to control their avatar and their environment in the virtual world via keyboard and menu commands. The day ended with a scavenger hunt to find and record the locations of ten objects on the island. This task allowed the girls to get used to controlling their avatars, further explore the virtual island, and become familiar with the coordinate system used to specify positions and elevations on the island.

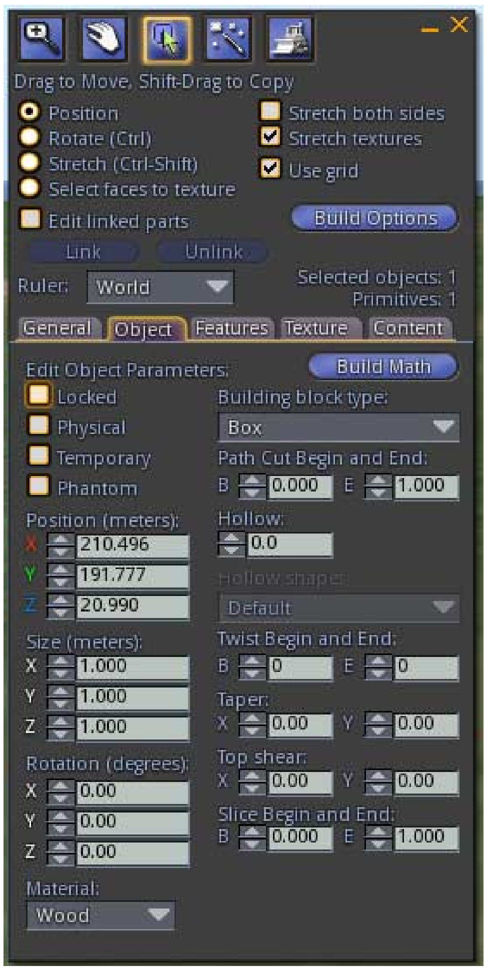

Figure 2. Modeling Interface

The remainder of the week's lab schedule was based around a series of programming and modeling challenges to be done in the virtual world. These challenges were used to teach concepts of computing including: 3D modeling, texturing, scripting structure and syntax, animation, events and event handlers, and coordinate systems, while also building up each girl's presence on the virtual island via adding new and interesting items to their claimed space. We also adopted the gaming idea of player expertise levels as an extra motivation to complete challenges. Students who completed the first ten challenges were promoted from the rank of *Novice* to the rank of *Apprentice* in our virtual world. At the end of the camp students were provided with instructions on how to access Curiosity Grid from their home computers and had the opportunity to continue with a second set of challenges in order to advance from the rank of *Apprentice* to *Journeyman*.

Figure 3. Mailbox modeled by a student.

```
default
{
   state_entry()
   {
      llAllowInventoryDrop(TRUE);
      llSetClickAction(CLICK_ACTION_OPEN);
      llSetText("Rylee Palmetto's Mailbox.",<1.0, 1.0, 1.0>, 1);
   }
}
```

Figure 4. LSL code to control mailbox.

In the first challenge, the girls used menu-based programming to create gestures. Gestures are short animations used for communication and entertainment on CG. The second challenge started off with each girl locating an "empty lot" on the virtual island to claim as their avatar's space and then building a sign to display that claim. The challenges became more difficult as the participants progressed and built upon the skills of previous challenges. The first day had a structured schedule for what tasks to do, but as the week progressed, the girls could complete challenges at their own pace. The goal was to complete ten challenges and be promoted from a *Novice* to an *Apprentice CyberGirl*. Each challenge consisted of a written set of instructions to create and/or program an object. The initial challenges had very detailed write-ups with step-by-step instructions for the tasks. Later challenges were less detailed and assumed that the previous challenges had been completed and understood. Instructors were always in the computer lab with the students going desk-to-desk answering questions.

Figure 5. The avatar on the top left is the original Ella avatar. The other three avatars are modifications of the Ella model created by three different girls.

The first ten challenges were:

1. Programming gestures and triggering them with a keyboard command.
2. Modeling and texturing a sign to mark out their area of the island.
3. Writing a script so that their sign displayed a message when an avatar approached it.
4. Modeling and texturing a mailbox.
5. Scripting their mailbox to receive messages.
6. Modeling a bench and scripting it so that that an avatar would sit on it correctly.
7. Building and scripting a fountain with a moving water texture.
8. Modeling and scripting a sliding door.
9. Modeling and scripting a complete house.
10. Modeling and scripting a working lamp for their house.

All the girls completed at least ten challenges by Friday morning, with several girls completing twelve challenges. Figure 6 shows the front of one of the houses that was created with working doors, a fountain with flowing water, furniture scripted so that an avatar could sit correctly on it, a scripted mailbox and working lamps.

3. FINDINGS AND DISCUSSION

Pre-camp and Post-camp surveys were given to camp participants to measure student intentions toward majoring in a computing discipline, attitudes toward computing, and overall satisfaction with the camp. All sixteen students participated in both the pre and post surveys. The results from the camp satisfaction survey (given on the last morning of the camp) contained eight questions that were scored on a Likert scale of 1 to 5 where 1 = highly dissatisfied, 2 = somewhat dissatisfied, 3 = neutral, 4 = somewhat satisfied, and 5 = highly satisfied (Figure 7). Mean scores on all questions indicated a high degree of satisfaction with all aspects of the camp.

Figure 6. House created and scripted by a student.

Survey questions that were asked both pre-camp and post-camp whose responses were rated as significantly different ($p < 0.05$) are shown in Table 1. These four questions were scored on a Likert scale of 1 to 5 where 1 = strongly disagree and 5 = strongly agree.

There was a significant positive increase in student attitudes toward computing, i.e., at the end of the camp they felt that applying to college in a computing discipline was more valuable and they were more confident that they could achieve good grades in computing courses. At the same time they were less confident that they coud learn to understand computing concepts and liking to use computer science to solve problems decreased.

How do we reconcile the mixed results in Table 1, particularly in light of the high overall scores the camp was given in Figure 7 indicating a high degree of satisfaction with the camp experience?

The modeling and scripting environment in a virtual world is more challenging than the typical tools used to introduce middle school students to computing. Most, such as Alice or Scratch, use a drag-and-drop metaphore for programming that isolates students from the need to pay attention to syntax. Indeed, avoiding syntax is often listed as a positive in programs desigend to motivate student interest in programming [8,13]. Alice and Scratch are also primarily 2-D environments that require little mathematical sophistication with respect to coordinate systems, rotations, translations or scales. Students can be successful in these environments, come away with a new interest in majoring in a computing discipline while also still remaining generally ignorant of many of the skills they must develop to be successful in a computing major. We could view this approach as equivalent to encouraging them to become a writer without exposing them to grammar.

Alternately, virtual world modeling and programming incorporate a wide variety of learning experiences that other approaches leave out. Curiosity Grid is a three-dimensional world. Modeling an object exposes students to both global and local 3D coordinate systems, scaling, translation, rotation, texture-mapping and RGB color coordinates.

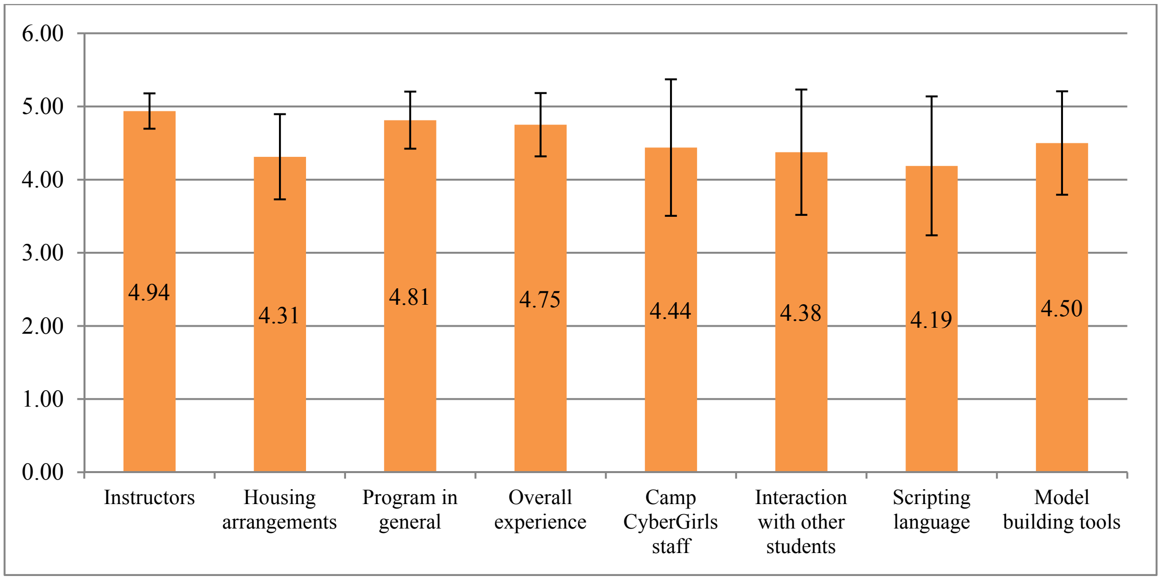

Figure 7. Camp Satisfaction Survey. Questions that were scored on a Likert scale of 1 to 5 where 1 = highly dissatisfied, 2 = somewhat dissatisfied, 3 = neutral, 4 = somewhat satisfied, and 5 = highly satisfied.

Table 1. Survey questions whose responses were rated significantly different ($p < 0.05$) post-camp as compared to pre-camp. Questions were scored on a 5-point Likert scale with 5 = strongly agree and 1 = strongly disgree.

Statement	Pre-Camp Mean/SD	Post-Camp Mean/SD
For me to apply to college in a computing discipline is (extremely valuable/ worthless)	3.812/0.8342	4.250/0.9309
I do think that I can learn to understand computing concepts.	4.813/0.4031	4.438/0.9639
I can achieve good grades (B or better) in computing courses.	4.375/0.7188	4.688/0.4787
I do like to use computer science to solve problems.	4.500/0.7303	3.938/1.123

With the Linden Scripting Language (LSL) students type in code and get syntax errors when the code is incorrect. Students experience that designing, building and scripting are challenging tasks that require sustained effort and attention. One student lost almost a full day's work when the girl sitting next to her tried to help her by pressing the wrong key on her neighbor's keyboard! The student who lost the work was very frustrated! However, she immediately began work on reproducing the models and scripts and ended up completing as many challenges as the other campers. She also learned a valuable lesson about saving her code!

Although challenging, learning programming in the context of a virtual world provides a great deal of motivation to make things work. The modeling and scripting languages are powerful. Every object can be coded so that it interacts with users and the world. Fountains can be modeled and then textured and programmed so that water flows. Lamps can be designed that actually light up. Objects (we had dogs, pandas, and sharks) can be modeled and then scripted to respond to basic questions from a user. The camp satisfaction means for both the programming language and the modeling tools indicate that our students were more than somewhat satisfied with the tools they learned to use. The fact that all sixteen students completed at least ten modeling and programming challenges indicates that, in the context of a week-long camp, middle school students can learn and use real computing and mathematical concepts that have not been watered down. Students who signed up for Camp CyberGirls were all confident or highly confident on the pre-camp survey that they could learn computing concepts. On the post-camp survey, 13 of the 16 remained confident or highly confident, even with a realistic introduction to what it takes to succeed.

In addition to our surveys we also asked for subjective feedback from the girls in a group feedback session held on the morning of the last day of the camp. When asked for suggestions on how we could improve the camp if it was offered next summer, there were no negative comments on the programming or modeling tools. They suggested that ending the lab sessions at 4:00 each day left them with an extra hour before dinner that they could have used to program. The class should have gone to 5:00. One girl commented that she would like to have had harder programming challenges. They also liked the events held outside of the lab times, particularly the welcome cookout that gave them the chance to get to know each other on the first night.

4. FUTURE WORK

Based on the success of our first summer camp using a virtual world to introduce students to computing, we are planning to expand our program to offer multiple camps next summer. For future camp sessions, we plan to gather more data from participants and monitor their participation after the camp by tracking if and when the participants log into the virtual world from home. We are also planning a weekend Camp CyberGirls reunion meeting this year for the participants. In conjunction with the Curiosity Grid, we will be posting our challenges and other materials online for other groups who might be interested in using them for their own programs.

5. ACKNOWLEDGMENTS

Our thanks to Ms. Starlett Craig and the Charles H. Houston Center for logistical and financial support for Camp CyberGirls. Thanks also to Jim Saint Clair (a.k.a. Laz) and the staff at Third Rock Grid who have worked closely with us since the beginning of this project to create Curiosity Grid and to support our activities. Partial support for this research was also provided by an NSF REU Site Grant CNS-0850695, the Palmetto Island Fund, and the C. Tycho Howle Endowment.

6. REFERENCES

[1] 3rd Rock Grid, Retrieved July 30, 2013 from http://www.3rdrockgrid.com/.

[2] Burke Q. and Kafai, Y.B., The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, SIGCSE '12. ACM, 2012, 433-438.

[3] Charles H. Houston Center | Clemson University, South Carolina 2013. Retrieved July 31, 2013 from http://www.clemson.edu/centers-institutes/houston/.

[4] Craig, M. and Horton, D. Gr8 designs for gr8 girls: a middle-school program and its evaluation. In Proceedings of the 40th ACM technical symposium on Computer science education, SIGCSE '09, pages 221-225, New York, NY, USA, 2009. ACM, 221-225.

[5] Doerschuk, P., Jiangjiang, L., Mann, J., 2007. Pilot Summer Camps in Computing for Middle School Girls: From Organization Through Assessment, *ITiCSE'07*, June 23–27, 2007, Dundee, Scotland, United Kingdom, 4-8.

[6] ECS. Exploring Computer Science. CS Education Statistics, 2013, Retrieved July 31, 2013 from http://www.exploringcs.org/resources/cs-statistics.

[7] Gross, P. A., Herstand, M. S., Hodges, J. W., and Kelleher, C. L. 2010. A code reuse interface for non-programmer middle school students. In *Proceedings of the 15th international conference on Intelligent user interfaces* (IUI '10). ACM, New York, NY, USA, 219-228. DOI=10.1145/1719970.1720001 http://doi.acm.org/10.1145/1719970.1720001

[8] Hardnett, C., Gaming for Middle School Students: Building Virtual Worlds, GDCSE '08, Miami, FL, USA, February 2008, 21-25.

[9] Harms, K. J., Kerr, J. H., Ichinco, M., Santolucito, M., Chuck, A., Koscik, T., Chou, M., and Kelleher, C. L. 2012. Designing a community to support long-term interest in programming for middle school children. In *Proceedings of the 11th International Conference on Interaction Design and Children* (IDC '12). ACM, New York, NY, USA, 304-307. DOI=10.1145/2307096.2307152 http://doi.acm.org/10.1145/2307096.2307152

[10] Hill, V. and Meister, M. 2013. Virtual worlds and libraries: Gridhopping to new worlds. College & Research Libraries News 74, 43-47.

[11] Hu, H. H. 2008. A summer programming workshop for middle school girls. *J. Comput. Sci. Coll.* 23, 6 (June 2008), 194-202.

[12] Imprudence:Downloads - Kokua Wiki, Retrieved July 30, 2013 from http://wiki.kokuaviewer.org/wiki/Imprudence:Downloads.

[13] Kelleher, C., Pausch, R., and Kiesler, S. Storytelling alice motivates middle school girls to learn computer programming. In Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, 1455-1464.

[14] Larkins, D. B., J. Moore, J. C., Rubbo, L. J., and Covington, L. R. 2013. Application of the cognitive apprenticeship framework to a middle school robotics camp. In *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). ACM, New York, NY, USA, 89-94. DOI=10.1145/2445196.2445226 http://doi.acm.org/10.1145/2445196.2445226

[15] Lee, J. J., & Hoadley, C. M. Ugly in a world where you can choose to be beautiful: teaching and learning about diversity via virtual worlds. In Proceedings of ICLS 2006, 383-389.

[16] Libeskind-Hadas, R. The Computing Community Consortium Blog, A Service of the Computing Research Community. Towards a New AP Course in Computer Science, 2010, Retrieved July 31, 2013 from http://www.cccblog.org/2010/07/19/towards-a-new-ap-course-in-computer-science/.

[17] Monroy, C., Klisch, Y. and Miller, L. M. Emerging contexts for science education: embedding a forensic science game in a virtual world. In Proceedings of the 2011 iConference (Seattle, Washington, 2011), 622-629.

[18] Moore, D., Thome, M., and Haigh, K. Z. *Scripting your world, the official guide to second life scripting*. Wiley Publishing, Inc., Indianapolis, Indiana, 2008.

[19] OSgrid - OpenSimulator Metaverse, Retrieved July 30, 2013 from http://www.osgrid.org/.

[20] Russell, D., Davies, M., and Totten, I., "GEOWORLDS: Utilizing Second Life to Develop Advanced Geosciences Knowledge, *2008 Second IEEE International Conference on Digital Games and Intelligent Toys Based Education.* Nov. 2008, 93-97 doi: 10.1109/DIGITEL.2008.50

[21] School of Computing | Clemson University, SC 2013. Retrieved July 31, 2013 from http://www.cs.clemson.edu/.

[22] Second Life, Retrieved August 1, 2013 from http://www.secondlife.com/.

[23] Steinkuehler, C. and Alagoz, E. 2010. Out-of-school virtual worlds based programs: a cross-case analysis. In *Proceedings of the 9th International Conference of the Learning Sciences - Volume 2* (ICLS '10), Kimberly Gomez, Leilah Lyons, and Joshua Radinsky (Eds.), Vol. 2. International Society of the Learning Sciences, 304-305.

[24] Wilson, C., Sudol, L., Stephenson, C., and Stehlik, M., 2010. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. ACM Available as: http://www.acm.org/runningonempty/fullreport.pdf.

[25] Zweben, S. The Computing Research Association. *CRA Taulbee Survey* (2011-2012), 2013.