Using Virtual Worlds to Engage Typical and Special Needs Students in Kinesthetic Computer Science Activities: A Computer Science Unplugged Case Study

Daniela Marghitu
Department of Computer Science and Software Engineering
Auburn University
United States
daniela@eng.auburn.edu

Tim Bell
Ben Kearns
Brett Ward
Kim Stephen-Pope
University of Canterbury
New Zealand
tim.bell@canterbury.ac.nz
{bmk33, bdw49, kms32}@student.canterbury.ac.nz

Abstract: Students with disabilities are often overlooked when it comes to majoring or having a career in science, technology, engineering, and mathematics (STEM) disciplines. It is well known that innovative hands-on science and math activities increase the content knowledge and confidence of students with disabilities thereby increasing their likelihood of majoring in STEM disciplines. In this case study we investigate the use of Virtual Worlds to provide “hands-on” activities in Computer Science, without requiring students to learn to program before they can participate. Two different kinds of virtual worlds are explored: Second Life (where users interact through a personal avatar in real time), and the Alice programming language (where users control characters in a virtual world through program instructions).

Introduction

Survey evidence indicates that more than 50% of students with disabilities who go on to attend college actually do so at a public 2-year institution (U.S. Department of Education, 1999). Moreover, of those students that enroll at 4-year institutions, students with disabilities are less likely to complete a baccalaureate degree than those without disabilities (U.S. Department of Education, 2008). The obvious impact of this is that there are very few students with disabilities who pursue graduate degrees. This problem is especially exaggerated for STEM disciplines. Only 9% of students with disabilities major in STEM fields in graduate school compared to 13% of students without disabilities (U.S. Department of Education, 2008) and only 1.1% of all STEM doctorate recipients are persons with disabilities (National Science Foundation, 2005).

It is well known that innovative hands-on science and math activities increase the content knowledge and confidence of students with disabilities thereby increasing their likelihood of majoring in STEM disciplines (Lam, 2008). Kinesthetic activities are iconic mediators (Bianco 2006, Bianco 2008) that enable students to experience an idea through physical activity. In this study we use material from a collection of kinesthetic activities called Computer Science (CS) Unplugged (csunplugged.org). The purpose of the CS Unplugged project is to expose students to the great ideas of Computer Science without them having to learn to program first – in fact, without using a computer at all. For example, Figure 1 shows an activity from CS Unplugged, where students follow a parallel sorting network marked on the pavement with chalk to sort 6 numbers into order.

![Figure 1: CS Unplugged sorting Network Activity](image)
The sorting network activity involves comparing numbers with people who are encountered in the process of moving through the network marked on the ground. The students choose their path based on a simple rule, and come out the other end with their numbers sorted into order. In the process they have encountered concepts like comparison, ordering, and especially parallel processing. This activity has been used in many contexts, and is invariably engaging for students and gives them insight into the kind of computational issues that computer scientists work on.

However, not all students are able to participate in physically active challenges (Marghitu, 2008); in some cases they may have mobility impairment, others may have personality problems that make it difficult for them to interact with others, while others simply may not be able to find enough peers to make up a team and try the activity. We are exploring the potential of providing pseudo kinesthetic activities for such students by integrating the CS Unplugged approach with virtual worlds. These include Second Life (SL), a virtual environment in which individuals interact on-line through avatars in a virtual 3D Space, and Carnegie Mellon University’s Alice, a programming environment for novice programmers that allows students to create interactive 3D virtual worlds including movies and games.

SL is a virtual world that users move around in using only the arrow keys on a computer. For this reason it is easy to access for those with mobility disabilities (although, for example, it presents serious problems for those with visibility problems). Users can walk, fly, and even dance in this world, and can interact with other users by typing (chat) or speech.

Alice was designed to make the process of learning to program easier and less frustrating for beginning programmers by addressing two of the common difficulties beginning programmers encounter: syntax errors and invisible state. In Alice, students construct programs by dragging and dropping code elements, which removes the possibility of making syntax errors. Alice is currently being used in CS1 courses in more than 10% of USA colleges and universities, as well as many successful K12 outreach programs for typically developing and special needs children (Marghitu, 2008). The imitation of movement, although, in an artificial realm, provides a knowledge transfer from the avatar character to the physical student who is directing the avatar – the student is “experiencing” what the avatar experiences. By imitating a real-world activity in a virtual world, the student is engaging in a novel form of imitation, which is a crucial part of developing as social beings (Gaussier, 1998), (Kuniyoshi, 1990). Thus virtual worlds provide exciting new possibilities for learning, which we explore in this paper. As well as providing accessibility to those with mobility or social difficulties, such environments may also be used to capture those who are interested in digital technology, but haven’t explored the concepts that drive it.

The Study

A key attraction of teaching kinesthetic activities in a virtual world is that students need only be able to interact with a pointing or typing device to “move” around in the virtual environment, making the activities accessible to those with limited mobility. For the virtual world of Alice, students also need to write small amounts of program code, but most of this is done by “drag and drop” editing with a pointing device.

Porting CS Unplugged activities to a Virtual World may seem to be the antithesis of what the off-line CS Unplugged project is about, but we are simply acknowledging that for some students a virtual world is a form of reality that works better for them, and for some it can offer the ability to participate in activities that otherwise would be impossible for them. Such environments, while obviously beneficial to mobility impaired students, may also be suitable for students who are isolated for social or geographical reasons. Furthermore, it removes some barriers that exist in the real world for everyone, allowing the development of imaginative environments, large spaces, the use of virtual actors from a fictional story, and the ability to create visual and aural fantasies that will engage children’s imagination. Students themselves can contribute creative ideas. For example, in our virtual sorting network, one group of students decided to ride through the network on cubes, even though this wasn’t an intention of the designers of the system (Figure 2).

With an environment such as SL, we primarily expect to build a world where students can play the games from CS Unplugged, as it can be used to create detailed and immersive virtual environments that could be very beneficial to learning. It may even be beneficial to have students construct the world themselves, although the learning curve associated with building the content for the virtual environments is rather steep, as it incorporates different types of skills, primarily modeling and scripting. These are skills that relatively few people possess, which unfortunately rules out the possibility of getting students to develop their own activities in SL, unless the activities were very simple or we provided large amounts of scaffolding.
Fortunately, once a user has these skills, the activities themselves need not be difficult to create, although can be a little time consuming. However, as with the physical CS Unplugged activities, we plan to provide the environment to the students so that they can experiment with the ideas of CS without having to learn substantial programming skills first.

Because the Alice environment is designed to help beginner students to create their own programs, the CS Unplugged material can be used with Alice by having the students implement those activities for themselves. Alice is naturally based around human (or animal) characters moving and interacting, therefore the program students would work with in Alice can be a very realistic model of the activity, as it would be done in CS Unplugged. The students could implement the activity from a description or video of the activity. Alternatively, if feasible, they could initially participate in the activity physically, giving them more insight into how it works. Alice provides a number of simple tools for creating, rather simply, a wide range of worlds, and these worlds can then be easily modified to expand or alter the world. An example of part of an Alice program is shown in Figure 3, showing how the “actors” that are viewed when the program is running are really just objects with easily understood methods.

The use of objects such as people, animals, and everyday items makes it a fun tool to use. A weakness may be that students become distracted by details of the characters and lose track of the main purpose. Each object has, as shown in Figure 3, a number of default methods, making it easy to create small scenes without designing significant functionality. If necessary, students could be provided with objects and methods that would make useful components of the activity they are trying to implement, rather than having to build all of it from scratch.
As an example, Figure 4(a) shows the key component of a sorting network in Alice – a simple comparator – with two characters (a cat and a rabbit) going through it. Students could be given this simple version, and be asked to generalize it to work for three inputs (a solution is shown in Figure 4(b)).

![Figure 4: Alice worlds generated for (a) a two-way sorting network, and (b) a three-way sorting network.](image)

The level of challenge in a virtual world can be adapted to the students’ ability to problem solve as they proceed, providing “flow” in the game (Csikszentmihályi, 1990), that is, making it difficult enough to be challenging, but not so difficult to be frustrating. For SL, this could mean having different levels of challenges for students that automatically change to match a student’s achievements; in Alice it would mean providing levels of difficulty ranging from watching an implementation (easy), to building one from scratch (hardest), with most students doing something in-between these extremes.

Hickey-Moody and Wood (2008) discuss the use of virtual worlds for students with disabilities, and point at that while they may be excellent for some people with disabilities, they can be partially or fully inaccessible to other people with disabilities. For example, SL and Alice are very visual, and is potentially less accessible to visually impaired or color blind students than the real world. However, this issue is currently being addressed (White, 2009), and virtual worlds do have more potential for alternative representations than the real world. Presentation should take into account a range of disabilities; for example, alternative text and text caption will be useful for students with hearing and some cognitive disabilities. These issues are addressed in the Web Content Accessibility Guidelines (WCAG) 2.0 (http://www.w3.org/TR/WCAG20/).

**Method and Findings**

A sorting network was implemented in SL (Bell, 2009). In Figure 5, six individuals are using this sorting network. They start on the green squares in a random order. If they have followed the instructions correctly, at the end of the network they come out on the red squares (in the distance) with their numbers in sorted order. Initial trials with the system were done with two main groups of people. The people shown in Figure 5 are in two New Zealand cities, and one is in China. A further trial was done with a group of Japanese students (average age about 25 years old); one student was deaf, and the others had upper and lower limb mobility problems, including 4 who used a wheelchair.

The sorting network components were also implemented in Alice (see Figure 4), and this experience was used to judge the level of skill needed for students to work in such an environment. The process of implementing the system in Alice involved designing the layout (nodes and links), and methods for the “actors” to interact as they pass through the network.

![Figure 5: Six avatars in SL in the Sorting Network activity, being controlled by six people in real life](image)
From the initial experience we have had with SL and Alice we have noted the following issues:

- The main area of SL is restricted to people 18 and over, and there is another area for teenagers (13 to 17 years old). This makes it impossible for younger students to access the main area, and there are severe restrictions on teachers accessing the teen area. For proper evaluations with students we will use a local system such as OpenSimulator. An important issue we encountered with the OpenSimulator is that because it is run "locally", the performance may not be as good as it should be, leading to sluggish responses for avatars in the virtual world. This would be frustrating for students, so such an environment needs to be configured to be fast and reliable before students use it.
- Some had difficulties learning to use SL, and another noted that the primary language of the environment was English, which created a new barrier for second-language students.
- Most of the students enjoyed the experience and would like to use it more. Although one student preferred the "real world" and found the virtual environment "lonely", others enjoyed the interaction in the virtual environment, and one preferred interactions in SL to the real world. One mobility-impaired student appreciated that it used less space. Another found that they preferred the SL environment "far and away" compared with the physical world, although also noted that while the environment was fun, personal interactions in the real world are more meaningful. The deaf student enjoyed using the text chat.
- For completely new members of SL, there would have been benefit in having some very simple exercises to build confidence in simple matters like talking to each other, and moving around, before engaging in a challenging CS based activity. On the other hand, it may turn out that such activities provide a good medium for developing this confidence.
- In the initial design, participants acquire a numbered t-shirt, but to put it on they must first take off their top, which is culturally unusual and also requires in-world skills not directly related to doing the activity. In future versions we would use a simpler approach, such as wearing a vest or hat.
- For the Alice implementation of the sorting network, we found that the drag and drop interface was easy to use, although some functionality would be difficult to find for new users, as many key methods are hidden away in odd places. Sometimes, when worlds start getting more complex, it can be difficult to keep track of every object in a given world (which, to some extent, helps enforce the use of good variable and object names, as well as enforcing pre-planning of the world). There are also some problems with objects interacting, especially with objects moving and colliding within the virtual world. Although these can be worked around, they could prove problematic for an inexperienced user.
- The Alice system is undergoing changes, with a new version incorporating Sims characters due for release in mid 2009. Once the new system becomes stable it will be a highly motivating environment where students can create worlds rather than just use them.
- Alice creates a world where students have an avatar that they dress and control. Ownership is key to engaging the student, and being able to dress your character and personalize the object connects them into learning. The new version of Alice will have a new range of Sims outfits; in an activity like the sorting network the characters need to be individuated for the sorting process, and so the act of “dressing” the characters is an important part of the activity.

Conclusions and Future Work

The two environments that we have explored offer rich and accessible worlds for students to “play with” ideas from CS, without having to be unduly focused on learning detailed programming skills. SL provides a more immediate kinesthetic experience in a virtual world where students can interact with others and given objects, whereas Alice encourages students to create imitations of their real-world experiences, potentially enabling them to understand those experiences better because of the higher level cognitive involved in the process.

We plan to further explore the benefits of these two approaches, and particularly how the SL and Alice approaches can benefit students with disabilities. Future work will include more evaluation of the systems described here, and implementation of other kinesthetic activities from the CS Unplugged collections.

For example, the “Treasure Hunt” activity involves the students moving around in a Finite State Automaton, following transitions between “islands” (states) to try to find their way to “Treasure Island” (the accepting state). In a virtual world the transitions could be implemented using vehicles (such as ships or trains), or a maze that restricts the paths. Other activities involve objects like cards that can be flipped over, and these could be simulated in a virtual world with large objects such as mats or tiles on the ground, creating a more exciting experience than working with normal playing cards. In some cases restrictions need to be placed on students, such as
a sorting activity where only two objects can be compared at a time, to simulate the restrictions that conventional computing devices have. Again, there is considerable potential for imposing such restrictions in an artificial environment.

The use of a virtual world opens up rich possibilities for creating exciting environments that wouldn’t be practical in a conventional teaching environment, making them more accessible, and also making it possible to remove or add restrictions on students that couldn’t be achieved so easily in the real world. We look forward to gaining more experience by exploring these opportunities, as well as finding ways to overcome new challenges that virtual environments bring.

Acknowledgements
Some of the experience reported here was gained with the assistance of Hiroki Manabe, Giovanni Bianco and Mick Grimley.

References


National Science Foundation, STEM Doctorate Recipients, (2005), Survey of Earned Doctorates 2005


