Collaborative Education in a Virtual Learning Environment

Hani Bani-Salameh

Department of Software Engineering,
The Hashemite University,
Zarqa 13115, Jordan
E-mail: hani@hu.edu.jo

Jafar Al-Gharaibeh

Architecture Technology Corporation,
Eden Prairie, MN 55344, United States
E-mail: jafar@atcorp.com

Clinton Jeffery

Department of Computer Science,
University of Idaho,
Moscow, ID 83843, USA
E-mail: jeffery@uidaho.edu

Ziad A. Al-Sharif

Department of Software Engineering,
Jordan University of Sci. & Tech, Irbid, Jordan
E-mail: zasharif@just.edu.jo

Abstract:

For the last few decades, there was tremendous interest in harnessing 3D virtual environments for education and training. Multi-user game-like environments make use of non-player characters and quest activities in tutoring and training. This approach incorporates successful active learning and incremental progress. This article presents issues encountered while adapting the multi-user online game genre for educational virtual environments. In this endeavour, non-player characters play a central role in organizing and delivering educational content. Educational virtual environments call for additional kinds of extension beyond generic user content creation as delivered in some virtual worlds like SecondLife. Support for these environments will emphasize the creation of new activities. A set of libraries and techniques named Portable Non-player character tutors and Quests reduces the effort needed to develop and integrate educational non-player

This work was partially supported by the National Science Foundation; grant agreement DUE-0402572.
This work was supported in part by the Deanship of Academic Research, The Hashemite University, Jordan.

characters. These non-player characters are introduced in CVE, a platform for rapidly developing educational virtual environments. The Portable Non-player character tutors and Quests framework enables formatted web-based exercises, quizzes, and educational content to be incorporated into virtual worlds, reducing the effort needed to create new content. The framework presented in this article introduces initial non-player characters that supports educational quest activities to the 3D environments users.

Keywords: Virtual World; Collaborative Virtual Environment (CVE); Non-Player Characters(NPCs); Educational Quests; MMO; PNQ(Portable Non-player Character Tutors and Quests).

Biographical notes: Hani Bani-Salameh is an assistant professor in the Department of Software Engineering at The Hashemite University, Jordan. He holds a Ph.D. in Computer Science from the University of Idaho. His research interests include software engineering, computer supported cooperative work (CSCW), software development environments, collaborative software development in virtual environments, and social networking and social media. He studies social interactions in social networks and online environments.

Jafar Al-Gharaibeh holds B.S and M.S. degrees in computer science from Yarmouk University in Jordan, and Ph.D. degree from the University of Idaho. His doctoral research was about programming language support for virtual environments. The test vehicle for the research was the Unicon programming language (unicon.org) where the language was extended with several features including concurrent threads, expanding the 3D facilities, and dynamic textures. He is presently an engineer in the R&D department at Architecture Technology Corporation in Minneapolis, Minnesota. His current research interests include, Computer Networking, Wearables and Sensor Technology, Graphics and Virtual Environments, Programming Languages, and Concurrent Programming.

Clinton Jeffery holds a B.S. from the University of Washington, and M.S. and Ph.D. degrees from the University of Arizona. His doctoral research developed a framework for program execution monitoring and visualization. He developed the Unicon programming language (unicon.org), and has done work in automatic debugging and collaborative virtual environments. He is presently an associate professor in the department of Computer Science at the University of Idaho.

Ziad Al-Sharif is an Assistant Professor at Jordan University of Sci. & Tech, Irbid, Jordan. He joined the Department of Software Engineering on February of 2010. Dr. Al-Sharif received his Ph.D. degree in Computer Science in December of 2009 from the University of Idaho, USA. He received his MS. degree in Computer Science in August of 2005 from New Mexico State University, USA. His research interests are in programming languages, software engineering, digital forensics, and collaborative virtual environments.

1 Introduction

The noticeable and spectacular success of MMO games brought with it a great interest for CVEs in education. Virtual worlds are effective at bringing people together and supporting

the primary requirements for a successful social interaction such as chatting, group meetings, and discussion sessions. Many MMO users “work" closely with people they have never met. This article addresses two main obstacles to adapting the MMO genre to education: the high cost of developing new virtual environments, and the need for domain-specific activities inside educational virtual worlds.

As yet, no educational MMO has a significant following, possibly because the education software industry has been unable to attempt a large-scale educational MMO due to the high costs involved. MMOs take millions of dollars to develop. Some researchers have hired game companies to write educational virtual environments, or have a customized commercial off the shelf environment such as ActiveWorld (activeworlds.com), but most researchers and specialists, who are interested in working with such environment, choose to use a general-purpose environment such as SecondLife.

This article begins in Section 2 with a description of a virtual environment called CVE that reproduces a computer science education environment. The first version of CVE was built and placed on Source Forge for public access; it lacks the aspects that make MMOs special such as: discovery, quest activities, and experience levels and advancement (1). Related work in the literature appears in Section 3. A set of requirements for an MMO-style educational virtual environment discussed in Section 4. The central building blocks for the solution presented in this article are the NPCs, who serve as tutors and record users' accomplishments. NPCs are presented in Section 5. An evaluation study of the use of CVE and NPCs in educational environments is presented in Section 6. Finally, this article sums up the current state of this work and discusses future directions in Section 7.

2 Overview of CVE

This article introduces a collaborative virtual environment called CVE (http://cve.sourceforge.net/) that serves as a platform for developing virtual environments (2) that addresses domain support tools, particularly those tools that help in CS education. CVE is a multi-platform educational environment that was built to support two primary use case scenarios: (a) distance learning by college computer science students, and (b) software development and group collaboration.

CVE is an environment where users can interact with each other within a 3D world. It provides developers with an awareness of other users' activities and what they are doing. CVE allows them to chat with other team members and with developers from other teams in real time. It also allows users to invite developers, software engineers, instructors and students to collaborate in code editing, compilation, and debugging.

CVE is built using a very high level language called Unicon (unicon.org). The Unicon language is supported with the necessities to build CVE; these include: high level 3D interfaces to graphics, networking, and audio support (3). Writing CVE in Unicon reduced its associated development and maintenance costs compared with systems programming languages. This section focuses on the main components of CVE.

Avatars

CVE's avatars are simple, featuring the ability to point, an identifying label, and a visual indicator when chatting. The default CVE avatars are hardwired humanoid. Other 3D models used for avatars in CVE were produced in Microsoft .x format using tools such as 3D Studio Max.

Collaborative IDE

CVE provides a synchronous real-time editor that allows users to edit their code and documents privately. Also, it allows them to invite others into a collaborative editing session on the fly when consultation is needed. CVE provides a collaborative shell that allows developers to compile, run, and debug their programs synchronously; it allows them to watch the compilation messages and to share the inputs and outputs of the running program (4)(5)(6).

The shared editor was implemented using an approach similar to that of Group Homework Tool, which supports synchronous, collaborative coding among novice programmers (5). Changes and edits to the shared artefact are done locally on the client, and then delivered to the server to update the original version where clients can apply their modifications to the text.

Social Network Features

CVE allows developers to find other developers and project partners, and ask for help. They can check the other users' activity, and get help in solving a particular problem or issue. CVE supports features that provide the developer with a better awareness and appropriate online presence.

Chat Systems: Text and VoIP

In CVE voice is used alongside other tasks. Location-based voice supports distance- and room-based connections. A virtual cell phone allows private connections, uninterrupted by local conversations. The Unicon VOIP facilities run on Linux; they have been ported to Windows but are not yet reliable there.

3 Related Work

This work is related to many collaborative virtual environments. NPCs, educational quests, and CVE owe much to chat, MUDs, and MOOs (7). Much work has been done that is related to 3D research virtual environments, such as DIVE (8), and NICE (9). Many systems take a virtual-centric view where everything is done in the 3D environment, or view the virtual environment as merely providing awareness of other users (10), while in CVE tools and NPC quests are integrated into the environment where users can easily switch between the tools without effort.

Some virtual environments' source code are publicly available, such as University of Manchester's Maverik (11) and Alice (12). As an education project, JABRWOC uses Alice to support 3D virtual reality as an educational instructional domain, instead of the delivery medium. A related education project mentioned earlier is Viras (13), an educational virtual environment built using Active Worlds. Active Worlds is not customizable to domain-specific education tasks, but it is a suitable domain for generic virtual environment construction (2).

CVE's collaborative IDE is a computer supported cooperative work (CSCW) tool. Some related work is a tool called CROCODILE that provides a virtual educational environment (14). The CSCL community spent time trying to explore using 3D worlds as educational

environments (18). To the best of our knowledge there is currently no tool which integrates a collaborative development environment into a 3D virtual environment as is the case for CVE.

Wong et al. (15) conducted a study to evaluate the effectiveness of using a virtual school environment in order to treat preadolescent children who suffer from social anxiety disorder. The result showed that the virtual environments are viewed as acceptable and effective by both the children and their parents. The findings showed that virtual environments are easy to implement by children and played a useful role in the treatment of childhood social anxiety disorder.

Karsakov et al. (16) proposed 3DVLE (3D Virtual Learning Environment) to support collaborative learning. They conducted a study on 3DVLE in the educational process in master's courses. The findings showed that 3DVLE supports knowledge sharging and training, and enhances communicational and interpersonal skills.

Loureiroa and Bettencourt (17) conducted a study in Second Life with the support of web 2.0 tools to replicate and complement the classroom. The study focused on finding issues that might affect and encourage knowledge sharing in learning contexts using virtual learning environments; the aim was to enhance the educational experience using the online tools. It was found that learners in such environments feel more confident and have the willingness to learn and be creative. Virtual environments provide the ability to share information that can be accessed by a larger number of learners.

As shown in the literature SecondLife is one of the most studied and investigated virtual environments as an online teaching tool. Studies have been designed to investigate SecondLife in traditional courses (face to face) (19)(20)(21)(22), for use in hybrid courses (both traditional and online) (19), and for use in synchronous online courses (23).

4 Requirements

This work aims to study the educational virtual environments, the goal is to get a broad understanding of such systems, and to find out about the needed functionality (requirements) for their users. This work studied a set of virtual environments (1) that have been used as an educational environments and studied as a representative virtual environment (e.g. SecondLife (SL)), and (2) that have been used to teach about virtual environment(e.g. World of Warcraft (WoW)). From the literature it has been found that the following requirements support the goal of creating an MMO-style educational virtual environment. Where necessary, the requirements are illustrated with examples from Computer Science education.

Requirement 1: Virtual Environment Basics

A virtual environment is an immersive 3D virtual space with shared access to persistent state. The objective in this paper is an educational program that feels like a 3D videogame, where users move around, go to an instructor's office, or conduct virtual lab work.

Reported experience indicates that a "3D space provides better social awareness than 2D places of comparable complexity. 3D virtual spaces borrow successfully from users' expertise at navigating real spaces to provide superior ease of use" (13)(4)(2). Voice communication is as important as the other collaborative tools in a virtual environment. If 3D graphics make a virtual environment feel like a place, voice communications make

that place feel more real. In return, a virtual environment provides a context in which the management and costs of voice can be improved.

Requirement 2: Platforms and Scalability

An educational virtual environment must scale to the tens or hundreds of students in courses that it supports; this is a lower target than an MMO. Many games require a discrete graphics processor, or run on only one operating system. An educational virtual environment must run acceptably on “common denominator" hardware, including slow network connections. Running offline with reduced functionality is desirable when no connection is available. Virtual world servers should not demand special machines; an ordinary computer should be enough to run a virtual world server.

Requirement 3: Cost Minimization

The cost of a virtual environment is high, so a major requirement for education is to minimize both construction and use costs. Many educators have adequate computer equipment but can incur little or no on-going monthly costs for software.

The design and artwork for a large custom space imposes prohibitive construction costs, so this paper considers only the construction of a modest virtual space such as a model of a real-world academic department, where textures might be derived from camera images instead of extensive art. To introduce a working and usable 3D environment, even a single floor of a building, requires many models and textures. For models, the problem is manageable and largely solved. Creating many quality textures from photos should be semi-automatable, but most camera based textures at present are built at high manual cost using a tool such as PhotoShop or the Gimp to modify their size and make them tileable.

Requirement 4: Collaborative Domain Support

Many online education systems such as WebCT lack domain-specific support. Some academic subjects are taught with simple text and images, but many require specialty tools. Consider Computer Science as an example subject. Generic collaboration via e-mail, chat or a shared text editor are inadequate. Integrated Development Environments (IDEs) represent one of the most important tools in developers’ everyday activities; domain support in computer science entails a collaborative IDE for virtual pair programming, debugging, and distributed code reviews.

Requirement 5: Instructor Support

A virtual environment provides a view of what others are doing and simplifies discovering who is available, or queuing for the attention of an expert, an instructor, or team leaders. These users receive a large proportion of collaboration/assistance requests, motivating special attention in the user interface that will be lacking in generic virtual environments (5). The “killer application" for an educational virtual environment is to improve the efficiency with which users see who is available online to help, and then obtain that assistance. Maximizing the convenience of obtaining help is critical. A solution replaces sequences of e-mails and phone calls with a brief, direct online interaction; load balancing and scalability of instructional or services are also improved.

Requirement 6: User-created Activities

Success of an educational virtual world depends on the range of things to see and do. In an ideal, open virtual world educators can add new content; not just physical objects, but activities that engage the users. The kind of activities undertaken in MMOs include: quests (completing some task for a reward), combat (battling a human or computer-controlled enemy), socializing (text or voice chat), crafting (creation or gathering of virtual objects), travel (walking or riding to a known destination), and exploration (visiting a possibly hostile, unknown territory).

Requirement 7: User Experience and Skill Advancement

An educational virtual environment must model its users. Although social virtual environments usually overemphasize avatars, MMOs focus more on modelling user experience and skills development. MMO avatar appearance generally reflects in-game accomplishments and provides motivation for particularly challenging endeavours. MMOs' focus on experience and skills development is popular with users and perfectly suited to education. Fundamental to a focus on user experience and skill advancement are a data representation for knowledge in different subjects, and a means of acknowledging and rewarding accomplishment.

5 Non-Player Characters (NPCs)

The NPCs framework presented in this article is called *Portable Non-player character tutors and Quests* (PNQ, pronounced “pink"). The design was made to be independent of any specific virtual world, which makes it easy to share NPCs and quest educational contents among different virtual worlds.

Each PNQ NPC has a profile that consists of several sections such as *id, avatar, behaviour, dialogue, and knowledge*. The profile is read at start up via HTTP by an NPC agent, a program that “plays" the NPC as a client on the virtual world server. NPC profiles are created and maintained as web pages in basic HTML or XML format, a task familiar to many instructors. A friendly wizard for creating profiles is available. Creating a NPC is performed by specifying its *(a) knowledge, (b) behaviour, and (c) dialogue models*. Detailed information about the three profile models can be found at (1).

5.1 NPC Agents

As mentioned in (1), a PNQ NPC client can connect to the server like any avatar in the environment; there's an indicator flag for each NPC avatar. This way helps the environment by: *(1) freeing the server from NPC management, (2) allowing NPCs to run from any machine, and (3) allowing a human to “play" an NPC*. Moving the NPC to a different machine entails moving the NPC's dynamic knowledge about users and their quests. This problem is partially solved by logging dynamic data on the server, if the server supports that, which updates the NPC upon its launch. If the NPC is connected to a server that does not have a user model, quests completed in that virtual world are logged to the CVE server.

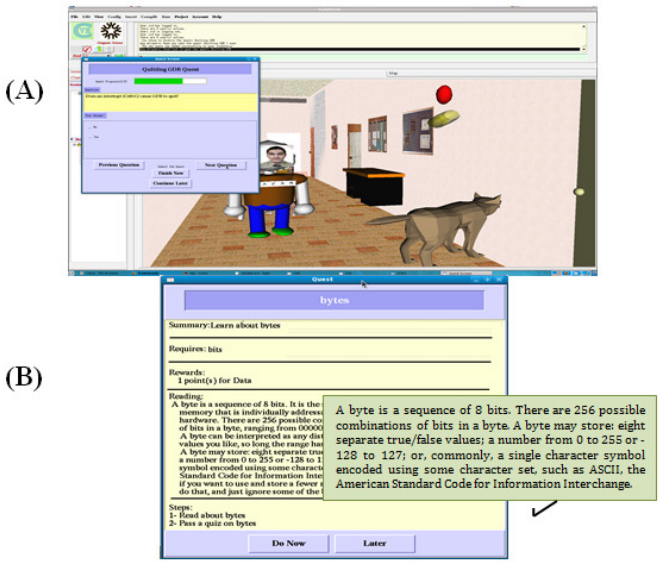

Figure 1 Shows an invitation dialog for a quest.

5.2 Tutors and Quests in the Virtual World

NPCs can pose in many shapes and images such as pets, monsters, and followers, but their primary job is a quest giver who delivers educational content. Figure 1 (A) shows a quest giver NPC in CVE. A red quest sphere marks NPCs with quests available. The dialog in Figure 1 (B) opens when a quest sphere is clicked.

When the user clicks the red quest sphere, a message called “Quest GiveMe" is sent to the server, and then forwarded to the NPC agent. The NPC checks the available quests for that user in its knowledge base, and sends back a list of quest titles and URLs. The server checks if the user has completed or is currently on these quests. The NPC maintains quest knowledge for every user, but if the NPC goes offline, it may need an update. A message will be sent to the server to show that the quest is finished, and a record of the finished quest will be added to the NPC database. The NPC checks for other available quests, and sends them to the server that checks whether the user has completed them. The quest will be approved and forwarded to the client. Once the user downloads and accepts the new quest, the message “Quest Accept Title" will be sent to the server, and a new quest will be added to the user’s quest list. Figure 2 shows how the user starts a new quest.

6 Comparison Based on the Requirements

As part of the PNQ project, concurrent with their implementation in CVE, an experiment built a virtual space with NPCs and quests in SecondLife. SecondLife focuses on user-created content, including the buildings, virtual objects, and avatars’ appearance. Educational institutions use it for marketing, on-line lectures and discussions. A scripting language in SecondLife called LindenScript allows virtual objects to incorporate complex behaviour and access external content via HTTP (1). SecondLife does not support MMO-style quest activities, and has no model of experience and skills development which are major

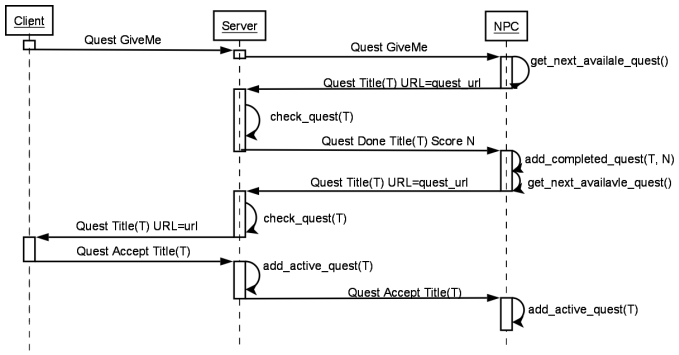

Figure 2 New quest scenario (showing messages exchanged between the NPC, client, and server).

motivators for MMO users. Without support for modelling user activities and experience, the cost of developing substantial educational applications in SecondLife is high. An exception would be visual arts and architecture where SecondLife's construction tools directly support the design and discussion activities of many courses (24).

The experiment was made generalize the PNQ NPCs described in this article above. Educators are so eager to use SecondLife that it has its own forum (sleducation.wikispaces.com) and dedicated staff at Linden Labs. The experiment's main goal was to imitate the CS department environment at the University of Idaho.

SecondLife (SL) meets Requirement 1 (basics) and 2 (platform) trivially, although it is more demanding of hardware (3D graphics) and network bandwidth than desired. Its demands are much higher than World of Warcraft. It has no standalone mode. For Requirement 3, SecondLife affords low construction costs, but maintenance costs paid to Linden Labs can be substantial. Creating the UI's campus with land purchased from SL cost around US $60, with a cheap monthly fee of US$18. This met the low-cost requirement for us but might still be unacceptable to many educators. Using the land purchased in SecondLife, students built a virtual UI CS department, with unsatisfactory results. The students constructed a facade with a few floors but were unable to build an interior given the 347 graphics primitives that Linden allowed on the plot. SecondLife offers no support for specific domains such as education, so for experimental purposes to address Requirement 4, an external IDE was used. Virtual objects were used to deliver computer programming homeworks and assignments. The student clicks on the object and chooses the file from the available list. Then, the file was sent to the server in order to compile, run, debug, and compare it with the right answer (1). Instructors are regular users, so Requirement 5 is addressed via SecondLife's support for users to manage frequent interactions with other users. SecondLife supports separate tabs for independent discussion threads with different users. It provides no queuing, nor an easy way to compare relative availability of instructors. It is possible to improve on what is built-in within SecondLife by constructing appropriate virtual objects. For Requirements 6 and 7, SecondLife non-player characters were implemented by a C# program using libopenmetaverse (www.openmetaverse.org) to login to a regular user account. Scripted virtual clothing enabled the NPC to be informed when a user clicked on it; in other respects the NPC interacted with the user primarily via chat commands. The SecondLife implementation of PNQ NPCs is usable, but at present educational quests involving programming tasks are more cumbersome there than in CVE. Table 1 summarizes the comparisons between the two environments.

Table 1 A Summary of the Comparisons between SecondLife and the CVE Environment

	Second Life (SL)	CVE
REQ #1 (Virtual Environment Basics)	- Demands for high hardware(3D graphics) - Voice Support	- 3D and Voice Support
REQ #2 (Platform and Scalability)	- Demands for higher network bandwidth than desired. - No standalone mood.	- Runs on "common denominators" hardware - Runs on slow network connection - Uses an ordinary PC server (no demand for special machine)
REQ #3 (Cost Minimization)	- Low construction cost - Monthly cost for software	- Minimized construction cost - No use costs
REQ #4 (Collaborative Domain)	- External IDE	- IDE is Integrated within the environment
REQ #5 (Awareness & Presence Support)	- Support for interactions management - No queuing - No availability	- Managing Notifications - Collaborative sessions
REQ #6 (User-created activities)	- NPCs implemented by a C# program - Difficult quest activities	- Implementing NPCs is easier and more usable than in SL.
REQ #7 (User Experience and Skill)		

7 Evaluation

The evaluation process resulted in both of qualitative reports from developing the CVE system with its various components, and quantitative survey results from users' experience in CVE and other virtual worlds. The following sections describe the evaluation results.

7.1 Development Observation

CVE must process multiple event streams: a TCP server connection and separate window system input streams for the 2D tools and the 3D subwindow. These connections must be checked continuously. On UNIX the function select() handles this, while it handles only network connections on Microsoft Windows.

For the CVE environment, performance has been an issue, especially running on low-end hardware and in scaling to larger numbers of users. Virtual environments are heavily I/O bound. The client and the server had their event-driven I/O multiplexor rewritten to process all input prior to sending output for each step. This change alone scaled CVE from handling 4 users to handling 26 users on midrange Pentium 4 hardware (2).

The study expected that the server will be the bottleneck, but it has been found that the server load has seldom exceeded 10%. CVE remains I/O bound, but scalability is limited by clients' graphics rendering, dominated by OpenGL code written in C. The entire suite of textures for an environment had to be replaced at 1/4 size, and the avatar graphics reduced to a very low polygon count in order to run CVE on a classroom full of laptops with integrated graphics. Scaling to hundreds of users would entail reducing the graphics further by abstracting groups of far-away users. Wireless internet performs substantially worse than wired networks for collaborative tools; latency variability between different packets is a problem on wireless networks.

Our experience and knowledge of integrating domain-specific tools is limited. The CVE collaborative IDE's use of command-line tools such as compilers and debuggers employs pseudo-terminals running inside editor widgets. Graphical domain tools require

Table 2 The list of questions from the summer camp survey..

1.	I had experience working with Virtual Environments prior to the Summer Camp.
2.	Education in a virtual environment is more fun than being in a classroom.
3.	Learning is faster (in areas of study like programming for example) in a virtual environment. a. compared to being in a classroom. b. compared to an web-based online class.
4.	If we had this summer camp in one place where we all get together in a computer lab to do the same things we did during the camp, would you still be interested to do it ?
5.	In a virtual environment, you can see other people's avatar moving around, and you can see what they are doing. Do you think that helps you to interact and be more social than if it would be chat-only or web-based environment?
6.	Would you recommend this style of camp be offered in future summers?
7.	Experience working with Virtual Environments prior to the Summer Camp.

reimplementation or extension to work with our framework (25)(26). Future efforts will result in additional domain-specific tools and identification of common principles.

7.2 User Experience

To evaluate CVE relative to other virtual worlds, the research group ran computer science summer camps. In the latest camp, 11 participants were introduced to four different software environments: CVE, SecondLife, World of Warcraft, and Alice. The camp targeted students and faculty in New Mexico and Idaho (8 male, 3 female; 7 students, 4 faculty members). The camp lasted five weeks with a one-hour group session and one hour of suggested independent activities each day. Instructors and campers met solely by means of the internet.

At the end of the camp, a survey was conducted to get feedback on users' experiences. The survey consisted of a set of 15 statements asking the participants to give a rating using a scale of 1-5, where one indicated complete disagreement and five was complete agreement. The survey was emailed to the participants. The small number of participants causes us to consider the results preliminary in nature. Table 2 shows the list of the question in the survey. Table 3 shows the users' responses to the questions.

Users' experience in virtual environments ranged from beginner to expert. Most users found education in a virtual environment more fun than being in a classroom. They think they can learn more and faster in subjects such as programming in a virtual environment compared with a classroom and especially compared with a web-based online class. Users found virtual environment avatar interaction to be more social than if they were in a chat-only or web-based environment.

Users who responded to the survey found that wandering around in the 3D world interacting with NPCs and asking for quests, which normally end with a set of questions, is more entertaining than taking online (web-based) questions. Also, they feel that the NPC quest activities induce them to start new quests, and that the whole experience enhances the educational process and helps them learn more.

The users show a great interest in communicating with the NPCs to start the quests. They feel that NPCs are an important part of any 3D virtual environment as long as they can offer quests and allow interesting chat sessions.

Table 3 Users responses to the camp survey.

Q#	1 (Disagree)	2	3	4	5 (Agree)	Mean	Median
1	1	2	1	1	1	2.83	2.5
2				5	1	4.17	4
3a			2	2	2	4.00	4
3b		1		2	3	4.17	4.5
4		2	2	1	1	3.17	3
5				2	4	4.67	5
6				1	5	4.83	5
7				1	5	4.83	5

8 Conclusions and Future Work

This article presented experiments in adapting World of Warcraft-style non-player characters and quests in an educational virtual environment. The SecondLife non-player characters and quests worked but were not very satisfactory; if Linden Labs elected to support non-player characters and quests, SecondLife would become dramatically more useful for education. The SecondLife experiment was still beneficial in that it resulted in a basic framework for developing non-player characters and educational quests that would run in multiple virtual worlds, including the CVE system and SecondLife. The CVE platform proved suitable for prototyping virtual environments that work well for the small to medium class sizes that it was designed for.

The framework presented in this article introduces NPCs that support educational quest activities to the 3D environments' users. Tools make the process of creating new NPCs and adding new educational content easy, and it is straightforward to make them independent from any particular virtual world. Many of the quests that are the most fun will involve interactions that are specific to a particular virtual environment. However, judging from World of Warcraft's quest mix, many quests can be world-independent. For tutorial NPCs they will depend on a learning domain which may or may not have a visible representation in a virtual world. Without a direct virtual world embodiment of the material being taught, the success of tutoring is defined mainly by the NPC's ability to convey the material via its text chats or scripted behavior, and its ability to sustain student interest.

Given the availability of interesting NPCs that live in multiple worlds, significant challenges remain, such as providing rewards that increase character capabilities in multiple worlds. However, most users will interact with NPCs through a single virtual world interface. The primary function of tutorial NPCs is to enable content to be used in multiple environments; the rewards can be world-specific.

Future directions for this research include: (a) improving the environments scalability by dividing the work and network load on the server across multiple servers/machines, and collaboration support and group activities for different groups and domains such as distributed development and academic disciplines.

Acknowledgment

This work was partially supported by the National Science Foundation; grant agreement DUE-0402572. This work was supported in part by the Deanship of Academic Research, The Hashemite University, Jordan.

References

[1] J. Al-Gharaibeh and C. Jeffery, PNQ: Portable Non-Player Characters with Quests, in *Proc. 2010 International Conference on Cyberworlds* (*CW 2010*), Singapore, October 20-22, 2010: pp. 294-301.

[2] C. Jeffery, A. Dabholkar, K. Tachtevrenidis, and Y. Kim, A Framework for Prototyping Collaborative Virtual Environments. Groupware: Design, Implementation, and Use, *Proc. of the 11th Collaboration Researchers International Workshop on Groupware* (*CRIWG 2005*), Porto de Galinhas, Brazil, September 25-29, 2005: pp. 17-32.

[3] C. Jeffery, O. El-khatib, Z. Al-sharif, and N. Martinez, Programming Language Support for Collaborative Virtual Environments, in *Proc. of 18th International Conference on Computer Animation and Social Agents* (*CASA 2005*), Hong Kong, China, 2005.

[4] H. Bani-Salameh, A Social Collaborative Distributed Software Development Environment, PhD Dissertation, University of Idaho, Department of Computer Science, August 16, 2011.

[5] H. Bani-Salameh, C. Jeffery, Z. Al-Sharif, I. Abu Doush, Integrating Collaborative Program Development and Debugging within a Virtual Environment. Groupware: Design, Implementation, and Use, in *Proc. of the 14th Collaboration Researchers International Workshop on Groupware* (*CRIWG 2008*), Omaha, Nebraska, September 14-18, 2008: pp. 107-120.

[6] H. Bani-Salameh and C. Jeffery, Collaborative and Social Development Environments - a Literature Review, *International Journal of Computer Applications in Technology*, 2014 Vol.49, No.2, pp. 89-103.

[7] J. Smith, Basic Information about MUDs and MUDding (MUD FAQ). Available at http://www.lysator.liu.se/mud/faq/faq1.html. Accessed July 27, 2015.

[8] E. Frécon, Dive: Communication Architecture and Programming Model, *IEEE Communications Magazine*, 2004, 42(4): pp. 34-40.

[9] M. Liebregt, Collaborative Virtual Environments in Education, in *The 2nd Twente Student Conference on IT*, January 21, 2005.

[10] M. Robinson, S. Pekkola, J. Korhonen, S. Hujala, T. Toivonen, and M.-J. O. Saarinen, Extending the Limits of Collaborative Virtual Environments. In Churchill, Snowdon and Munro (eds.): Collaborative Virtual Environments. Springer-Verlag, (2011): pp. 21-42.

[11] R. J. Hubbold, X. Dongbo,and S. Gibson, Maverik-The Manchester Virtual Environment Interface Kernel, in *Proc. of the 3rd Eurographics Workshop on Virtual Environments and Scientific Visualization*, Monte Carlo, Monaco, (February, 1996): pp. 11-20.

[12] R. Pausch, T. Burnette, A.C. Capehart, M. Conway, D. Cosgrove, R. DeLine et al, Alice: A Rapid Prototyping System for 3D Graphics, *IEEE Computer Graphics and Applications*, (1995), 15(3): pp. 8-11.

[13] E. Prasolova-Førland and M. Divitini, Collaborative Virtual Environments for Supporting Learning Communities: an Experience of Use, in *Proc. of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, (2003): pp. 58-67.

[14] Y. Miao, Design and Implementation of a Collaborative Virtual Problem-Based Learning Environment. M.S. Thesis, Technical University of Darmstadt (Germany).

[15] N. Wong, D. Beidel, J. Spitalnick, The Feasibility and Acceptability of Virtual Environments in the Treatment of Childhood Social Anxiety Disorder. Journal of Clinical Child & Adolescent Psychology, 43(1): pp. 63-73.

[16] A. Karsakov, A. Bilyatdinova, A. Hoekstra, 3D Virtual Environment for Project-Based Learning, in *Proc. of the 8th IEEE International Conference on Application of Information and Communication Technologies,* (*AICT 2014*), pp. 468-472.

[17] A. Loureiro, T. Bettencourt, The Use of Virtual Environments as an Extended Classroom – A Case Study with Adult Learners in Tertiary Education. Procedia Technology (2014). Vol.13, pp. 97-106.

[18] R. Hämäläinen, P. Häkkinen, S. Järvelä, and T. Manninen, Computer-Supported Collaboration in a Scripted 3-D Game Environment, in *Proc. of the Conference on Computer Support for Collaborative Learning* (*CSCL 2005*), Taipei, Taiwan, (2005): pp. 504-508.

[19] S. Hornik and S. Thornburg, Really Engaging Accounting: Second Life as a Learning Platform. Issues in Accounting Education, 25(3), pp. 361-378.

[20] M. Mayrath, T. Traphagan, E. Heikes, and A. Trivedi, Instructional Design Best Practices for Second Life: a Case Study from a College-level English Course, *Interactive Learning Environments*, 19(2), (2011), pp. 125-142.

[21] A. deNoyelles and K. Seo, Inspiring Equal Contribution and Opportunity in a 3d Multiuser Virtual Environment: Bringing Together Men Gamers and Women Non-gamers in Second Life, *Computers and Education*, 58, (2012), pp. 21-29.

[22] L. Sierra, C. Gutierrez, C. and Garzon-Castro, Second Life as Support Element for Learning Electronic Related Subjects: A Real Case, *Computers and Education*, 58, (2012), pp. 291-302.

[23] G. Steel and S. Jones, Using Virtual Environments for Synchronous Online Courses, *Journal of Teaching and Learning with Technology*, 2(1), (2013), pp. 56-61.

[24] C. Jeffery, Using Non-Player Characters as Tutors in Virtual Environments, in *Proc. of the Researching Learning in Virtual Environments* (*ReLive08*).

[25] S. P. Reiss, Connecting Tools Using Message Passing in the Field Environment, *IEEE Software*, 7(4), (1990): pp. 57-66.

[26] D. Garlan and F. Ilias, Low-cost, Adaptable Tool Integration Policies for Integrated Environments, in *Proc. of the Fourth ACM SIGSOFT Symposium on Software Development Environments*, December 03-05, Irvine, California, USA, (1990): pp. 1-10.