Experiential Learning in VR

Tanvir Talukder

CIS 4914 Senior Project
Department of CISE
University of Florida

Advisor: Dr. Angelos Barmpoutis
Email: angelbar@ufl.edu

Date of Talk: April 21, 2016

Abstract

The problem under investigation in this paper is the diminishing level of competency in the United States (U.S.) in mathematics and science education. The University of Florida (UF) is currently investigating the use of culturally responsive teaching (CRT) and experiential learning methodologies to combat this problem. In order to perform user tests, a set of software applications need to be developed to gather more qualitative and quantitative information.

The solution presented in this paper is an application program interface (API) which utilizes Visineat, an open 3D data toolbox created by Dr. Angelos Barmpoutis, and multiple other software frameworks to create a comprehensive system for rapid creation of experiential learning applications in virtual reality (VR). The culmination of the software development efforts by my colleague Adam Fortier and I have allowed us to create an example application that showcases the versatility of our API. The example application is a simple quiz game that incorporates hand tracking, VR headsets, and client-server communication to provide an immersive learning experience. We learned about key advantages and disadvantages of virtual reality in education and identified positive characteristics and open problems in our current software solution.

Introduction

The problem of interest is that the capacity of the United States to compete and innovate in STEM areas at the international level is diminishing [1]. The University of Florida is currently investigating the use of virtual reality as a means of experiential learning in order to help solve this problem. This investigation requires the development of more interactive games and simulations for testing and deployment in schools. Our work seeks to address this problem by directly contributing to the software resources being utilized for research at the University of Florida.

Many individuals have developed experiential learning frameworks to address the problems in education. Gloria Jean Merriex developed an award-winning curriculum to teach mathematical concepts to elementary students using music and body gestures [1]. Researchers at the University of Florida developed a kinesthetic gaming framework for enhancing elementary math education [1]. Details regarding these relevant works and other studies are provided in the Supporting Literature section of this paper.

Our solution to the presented problem is the development of a web-based virtual reality development API geared towards experiential learning. This API provides the means for future developers to create applications for students to learn in interactive, simulated environments.

Dr. Angelos Barmpoutis, an associate professor at UF, created a powerful toolset, Visineat, to allow rapid development of web-based, 3D applications. Our project aims to extend this toolset by encapsulating it into an API that brings multiple software frameworks together to create a comprehensive interface for developing applications particularly geared towards experiential learning in virtual reality. An example application has been developed and our solution will be delivered to Dr. Barmpoutis to support research in this area.

Problem Domain

The problem of the diminishing fluency and interest in STEM education in the U.S. has a promising solution in the use of experiential learning. While experiential learning is more commonly performed in a physical environment, there are a larger range of possibilities when the use of simulated environments taken into consideration. This reasoning brings us to placing the domain of this problem in the area of virtual reality in computer science and engineering. Our solution is directly embedded in this problem domain and intersects with the area of education.

Supporting Literature

A primary contributor to our problem is rapidly increasing cultural diversity. Schools with higher cultural diversity have reported lower scores in standardized assessments in mathematics and sciences than schools that are predominately populated with European American students [1]. Without a change in teaching curriculum, linguistic and cultural barriers may continue to negatively impact the United States with regards to education.

Gloria Jean Merriex is an individual who tackled the problems in education by providing a unique and effective educational experience. Her award-winning CRT curriculum utilized music and body gestures to teach mathematical concepts to fifth grade elementary students [1]. This curriculum was implemented in the high-poverty and predominately minority school, Duval Elementary School. In the four years before the implementation of her curriculum, the school's rating fluctuated within the range "C- F" on a scale of "A-F". This rating was measured by the State of Florida Accountability System based on FCAT scores. Gloria managed to take a school with an “F” rating and bring it to an “A” within the first year of the implementation of her curriculum. In the subsequent five years, the school rating fluctuated within the range “A-B”. The year immediately following the discontinuation of this program, the school’s rating dropped from an “A” to an “F”. This strongly indicates a positive impact of culturally responsive teaching techniques and experiential learning on education.

Theories behind experiential learned have been well-defined by John Dewey, an American educator. He developed an experiential learning model that has far-reaching implications for American education. Dewey’s model was constructed around three paradigms: the unity of abstract knowledge and doing in the real world, the unity of action and reflection, and the unity of the individual within the community [2]. The need to unite action and reflection in order to enhance its educational value is perhaps the most firmly grounded assertion that can be made about experiential learning [3].

While these concepts are well-understood, the ability to create learning activities that encompass all of these concepts in an effective manner can be difficult to accomplish. With regard to STEM areas, we are further encumbered by cultural boundaries and limitations in resources.

In this paper, we investigate the use of virtual reality applications to provide an enhanced learning experience while transcending these boundaries and limitations. A well-designed game or simulation can provide adequate visual, tactile, and auditory feedback that provides a much larger scope of learning experiences while allowing the individual to remain in a controlled environment.

This larger scope includes a range of perspectives and enhanced visualizations of complex information that cannot be achieved utilizing traditional means of teaching [4]. Christine Youngblut is a researcher who was sponsored by the Defense and Research Projects Agency to generate a comprehensive report on the educational uses of VR technology in practical and developmental environments. A key finding in her report indicated that students enjoy working with virtual worlds and that their experience can be highly motivating [5].

Two of the applications reviewed in her report were NewtonWorld and MaxwellWorld. These are VR applications that teach students about scientific concepts through a simulated environment where they can change scientific parameters and observe their effects. Specifically, NewtonWorld taught students about kinematics and dynamics of motion and MaxwellWorld taught students about electric fields. Studies conducted with these applications had major findings indicating that participants were very enthusiastic about the 3D nature of the learning environment and the ability to observe phenomena from different viewpoints [5]. Participants reported that observing multiple viewpoints was both motivating and crucial to understanding.

Although virtual reality is readily thought of as an opportunity to freely explore another world, the power of virtual reality in education lies in well-defined structure and control. Individuals can be placed in a problem-solving space and be granted immediate, data-driven feedback. This optimizes both the knowledge-action and action-reflection paradigms constructed by Dewey.

Technical Solution

The use of virtual reality in education is still in the early stages of being brought into the public education system in the United States. In order to increase the scope of this solution, more interactive games and simulations need to be developed for testing and deployment in schools.

Our proposed solution is the creation of a web-based virtual reality application development API. This API will allow rapid prototyping of experiential learning applications in virtual reality and contribute to research in this area being conducted at the University of Florida. It utilizes Visineat, the Open 3D Data Toolbox created by Dr. Angelos Barmpoutis, and supports plugability of a range of input devices as a means to create a holistic system that is optimized for experiential learning application development.

Our API was designed with future developers in mind. We utilize object-oriented design principles to ensure portability, reusability, and maintainability. Through the application of design patterns and dependency inversion, we were able to modularize the various subsystems in our API. By programming to an interface instead of an implementation, we can reduce object coupling and dependencies, and allow developers to easy extend and customize our API to suit their needs. We developed an example application utilizing our API to demonstrate its functionalities and practicality of use.

Our solution can be divided into five components: graphics rendering, collision detection, event handling, input device management, and multiplayer support.

Graphics Rendering

Rendering a virtual environment requires careful consideration into how objects are oriented around the individual and his or her representation in the virtual space. Visineat provided us the resources we needed to allow efficient rendering of complex objects. Through Visineat, we can pass the instruction of rendering to the GPU, and thus allow far less impact on the frame-rate of our application.

Developers have the option of customizing his or her own objects or choosing from a default set of objects. For more complex objects, such as humans or cars, we allow options such as OBJ file loading along with MTL file support.

By using Visineat, we can also provide a great distribution of support for translating the virtual space into different environments. These environments include: virtual reality headsets, 3D TV's, and red-cyan 3D glasses.

Collision Detection

Collision detection lies at the core of providing a user the ability to interact with the virtual space. This detection is performed utilizing axis-aligned bounding boxes and an octree data structure.

Axis-aligned bounding boxes (AABBs) are rectangular prisms that encompass the area of an object and do not change their orientation with respect to the x, y, and z axes. Instead, when an object changes its orientation, the bounding box reshapes itself in order to continue to fully encompass the object, while maintaining its own orientation. The use of AABBs allow quick collision detection between objects, because it greatly reduces the number of comparisons needed. Only three comparisons are needed to determine if two AABBs are intersecting and only six comparisons are needed to determine if one AABB is contained within another.

An octree is a space partitioning tree with eight children per node. The octree assumes a finite world space and subdivides it into octants. Like the objects in the world space, the octants in the octree are represented by axis-aligned bounding boxes. This is to allow quick detection of whether an object fits within an octant. The operation to insert an object into the octree involves finding the smallest octant that can fit the entirety of the bounding box of the object. Each octant has a maximum capacity of objects, and when that capacity is exceeded, the octant subdivides its own space into more octants and disperses its current contents among the new octants.

When an object moves, the octree is traversed to find candidates for collision as opposed to simply checking against every object in the world space. Average time complexity for collision detection is reduced from $O(n^2)$ to $O(n*\log_3(n))$. The use of AABBs and an octree data structure allows for efficient lookup of object collisions per frame by discarding large quantities of objects from consideration.

However, some problems include: reduced accuracy of object collision detection, and reduced efficiency in the case of frequent object insertions and removals in the world space. While our collision detection subsystem works well for the example application developed using our API, future developers may want to utilize a different system to accommodate different needs. By making our collision detection system modular, we allow developers to swap our system with their own and continue to utilize the other subsystems in our API.

Event Handling

Our solution provides an event handling architecture to dispatch primitive and custom events. Primitive events are defined as events at the highest level of abstraction. One supported primitive event is a collision between one object and another. Upon collision, all of the objects involved are notified of the event via an event dispatcher.

Custom events are defined as more specific forms of primitive event. For example, imagine you have a driving simulation application where you want different actions to be performed when collisions occur between the car and other objects. When a car hits another car, you may want to modify the acceleration and direction of movement of both vehicles and apply a form of vehicle damage. However, when a car hits a large tree, you may only want to apply those changes to the vehicle and not the tree. Our versatile event handling subsystem elegantly handles these events to promoting easy development of highly interactive applications.

Input Device Support

In order to gear our solution towards experiential learning, we need to be able to support a wide variety of input devices. Input devices provide real-time information regarding actions taken by users and the types of devices used are directly related to the user’s experience in the virtual space.

Motion tracking devices are particularly useful for experiential learning applications because they can provide real-time data of the user's motion, which can be used for visualization and data collection. In our example application, we utilize hand tracking provided by the Leap Motion Controller and VR headset capabilities provided by the Oculus Rift to provide an immersive experience for the user. Other existing motion tracking services that may be used for experiential learning include: object tracking, full-body tracking, face tracking, and eye tracking.

The input device subsystem in our API was made modular in order to accommodate for the large variety of input devices we have currently have and any future input devices that will be created.

Multiplayer Support

Multiplayer support is important because in order to provide ideal conditions for experiential learning, there needs to be a sense of community in the environment [2]. The core of multiplayer support is the exchange of data streams between clients participating in the same session. This allows users to participate in both collaborative and competitive problem-solving games and simulations. In a recently developed framework for kinesthetic gaming using culturally responsive teaching, it was demonstrated that this form of teaching promoted learning as a collaborative process rather than a competitive one [1].

Combining multiplayer support with motion-tracking information can provide a much larger range of applications and an enhanced experience. Users can more naturally interact with the environment and the representations of other users in the same environment.

Our example application utilizes a specific client-server communications protocol provided by Visineat, however we clearly do not want all users of our API to be restricted to this particular system. Therefore, we have designed our multiplayer subsystem so that developers can plug in their own client-server implementations into our API.

Results

A result of our software development efforts is the ability to create experiential learning applications in a well-designed and modular development platform. This is concretely demonstrated in our example experiential learning application that utilizes the Leap Motion Controller for hand tracking and the Oculus Rift as a VR headset.

Our example application is a simple quiz game where participants are presented with a question and given the task of moving answer cubes to a designated area to indicate his or her answers. A given question will have multiple answers and participants have a limited amount of time to perform the task.

The participants' hands are visualized in the 3D space in real-time through the manipulation of hand tracking data streams provided by the Leap Motion Controller. Participants can use their hands to select objects and move them around within the bounds of the world space. Selection is performed during a pinching or grabbing gesture on an object and deselection is performed upon the stop of the gesture. After the task has been completed or the maximum time interval has been reached, participants are provided feedback regarding the correctness of their choices.

When more than one player reside within the same session, color-based feedback is provided to indicate that another player is performing an operation in the same world space. Specifically, the answer cube selected by another player has a blue color mask applied to it. Color masking is also used to indicate right or wrong answers. When a user places an answer cube in the designated area, the cube will have a green or red color mask applied to it depending on the whether the answer was correct or incorrect. A limited time interval prevents players from simply checking every answer that is provided. It also encourages collaboration because the task may be performed most optimally through teamwork.

This application expresses the versatility of our API and alludes to future applications in education. It demonstrates that a developer can rapidly create a learning experience that encompasses the necessary components of a well-designed experiential learning application.

Conclusions

The culmination of our work is the creation of an API that will support and promote research in experiential learning. Utilizing Visineat and multiple other software toolsets, we were able to create a comprehensive interface for rapid application development in the virtual reality subspace of computer science.

After cross-referencing multiple scholarly resources and examining relevant works, we have determined that this is a viable solution provided that the games and simulations created by the API adhere to proper design characteristics. These characteristics include: well-ordered problem-solving, meaningful goals, possibility spaces, feedback and information, and empowerment and reward [2].

Some disadvantages and open problems we have in our API is the lack of a physics engine and support for a wider array of graphic file formats. These features are important components of developing virtual spaces with more realistic structures, animations, and object interactions. Despite lacking these features, an inherent advantage of our API is the ability to extend it in order to add new functionalities such as a physics engine and object model import.

Future Work

With assistance from Dr. Barmpoutis, we will be conducting user tests outside the scope of this project and writing a research publication to be submitted to a conference. As we progress in our research we will be making modifications to our API codebase to make it a more powerful platform for our own research and other developers.

Acknowledgements

I would like to thank my advisor, Dr. Angelos Barmpoutis, for his guidance, technical support, and motivation towards completing a successful project. Angelos has been an amazing mentor to me for nearly 2 years, and my passion for computer science and its applications was largely impacted by my work with him in research at the University of Florida.

I would also like to thank my colleague, Adam Fortier, for his contributions to our project, and for his motivation in continuing work in this area as we progress in our respective careers.

References

[1] A. Barmpoutis, Q. Ding, L. Anthony, W. Eugene, M. Suvajdzic, “Exploration of Kinesthetic Gaming for Enhancing Elementary Math Education using Culturally Responsive Teaching Methodologies,” in *Proceedings of VR16 Workshops: IEEE Virtual Reality 2016 Workshop on K-12 Embodied Learning through Virtual & Augmented Reality*, 2016, pp. 1-4.

[2] A. Barmpoutis, B. DeVane, and J. C. Oliverio, “Applications of Virtual Environments in Experiential, STEM, and Health Science Education Chapter 41,” in *Handbook of Virtual*

Environments: Design, Implementation, and Applications, Second Edition, K. Hale and K. Stanney, CRC press, Taylor & Francis Group, 2014, pp. 1055-1071.

[3] S. F. Hamilton, “Experiential Learning Programs for Youth,” *American Journal of Education*, University of Chicago Press, 1980, pp. 179–215.

[4] J. N. Bailenson, N. Yee, J. Blascovich, A. C. Beall, N. Lundblad, and M. Jin, “The Use of Immersive Virtual Reality in the Learning Sciences: Digital Transformations of Teachers, Students, and Social Context,” *The Journal of the Learning Sciences*, Taylor & Francis, Ltd, 2008, pp. 102–141.

[5] C. Youngblut, “Educational Uses of Virtual Reality Technology,” *Institute for Defense Analysis*, 1998, pp. 1-90.

Biography

Tanvir Talukder is an undergraduate computer science student at the University of Florida graduating in May 2016. He is currently working in two research assistant positions at the University of Florida and has worked in two prior research positions at UF in the past. The areas of research he supported included: neuroscience, physical tele-therapy, and neuromuscular physiology. He also worked as an application developer at J.P. Morgan Chase, where he worked on customer-facing production code with a back-end agile team for an upcoming web application.

Tanvir has worked in a wide array of software development environments and considers himself proficient in Java, C++, and JavaScript. He acquired an application developer position with AT&T and will begin working at their current headquarters in June 2016.

With the experience he has obtained over his college career, Tanvir has acquired a passion for technologies involving gaming and simulation. Tanvir plans on pursuing a graduate degree in computer science after working some time in industry. He has a strong interest in computer science research and its applications and plans on continuing research in the applications of gaming and simulation technologies. To support his interest, Tanvir took formal coursework in entrepreneurship at UF.