

COVID-19 Education in Immersive Virtual Reality Environment in Lifebrush

Michael Wahba

University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4
mwahba@ucalgary.ca

ABSTRACT

Lifebrush is an open-source virtual reality (VR) environment which uses a multi-agent system to model complex biological systems. There is a need to demonstrate Lifebrush's applicability to modeling novel biological systems and develop tools to help educators use Lifebrush. This project created an interactive VR learning environment where users can visualize the molecular structure of the COVID-19 virus and interact with a simulation of viral community transmission. A "Snapshot Slideshow" tool was developed to easily let users create slideshows with their Lifebrush creations and a "PDB Importer" tool was developed to create a pipeline from PDB files from the Protein Database to OBJ files for use within Unreal Engine, however the "PDB Importer" tool is still not functional.

INTRODUCTION

Lifebrush is a virtual reality tool, created by Dr. Timothy Davison at the University of Calgary[1]. Lifebrush was created in the Unreal Engine [2] and allows the creation of dynamic molecular illustrations using a multi-agent system based upon a customized Entity-Component-System. Lifebrush was designed to be an educational tool for use in classrooms, as such there is a need to demonstrate Lifebrush's versatility and applicability as an educational tool.

To demonstrate Lifebrush's usefulness, Dr. Davison created Lifebrush scenarios demonstrating the mitochondrial energy complex, and the neural synapse. For the experienced user, the Unreal Engine is an extremely powerful and customizable game engine, however for the inexperienced user the Unreal Engine can be daunting and inaccessible.

The motivation for this project is to create a new educational scenario for Lifebrush while also creating tools to make it easier for educators to create new scenarios for topical systems. The model system for this project will be the COVID-19 virus. COVID-19 was chosen because it is both very topical and provides an opportunity to show how Lifebrush can model a virus on different scales. COVID-19 can be modeled on the microscopic scale, in how it interacts with the human lung cell as well as the macroscopic scale in how it spreads via community transmission.

The community transmission simulation is particularly important because at the time of writing (Dec. 2020) vaccines for COVID-19 are about to be rolled out across North America. However, in a survey of 1000 people only 59% of respondents said they would get the vaccine and only 53% would give it to their children[3]. It is essential right now to have an educational tool which can demonstrate the impact vaccinations can have on the outcome of a population.

In the upcoming years, there will be much research into understanding how COVID-19 was able to become a worldwide pandemic. This project is a demonstration of how Lifebrush can be used to both teach foundational information about the virus and elucidate new information as well.

Educational scenarios in Lifebrush consist of a set of molecules the user can paint with as well as a set of meshes the user can paint on. Behind the scenes of each scenario is the logic which dictates the interactions of the molecules. Lifebrush uses an agent-based modelling system to simulate these dense systems (Fig. 1).

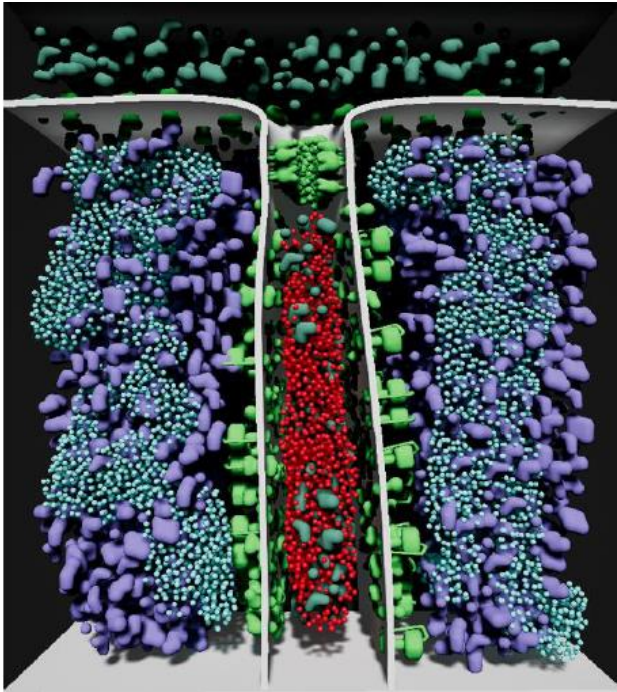


Figure 1 Molecular simulation of a mitochondria created in Lifebrush.

Another important aspect of this project is the learning environment itself. Until now, Lifebrush scenarios take place against an unmemorable, sanitary backdrop. The COVID-19 Lifebrush scenario created for this project will take place in an environment that feels more real and immersive. Virtual reality has a unique opportunity to provide an unparalleled sense of presence and immersion in the user, heightening the experience. This project also provides an opportunity to evaluate Lifebrush's effectiveness as a learning tool. Demonstrating Lifebrush's ability to quickly create models based on emerging models is imperative to Lifebrush's utility as a learning tool.

The work conducted in this project was intended to set a solid foundation for future researchers to build on. By showing what is possible with Lifebrush's current implementation, future researchers will have inspiration to expand Lifebrush into a diverse library of interactive educational experiences.

EXPLANATION OF THE PROBLEM

This project seeks to demonstrate Lifebrush's potential as an educational tool by creating a scenario about the COVID-19 virus. This project also includes creating tools aimed at educators for creating their own scenarios for any subject.

The research question of interest is: How can Lifebrush be used within classrooms as an immersive, interactive learning tool?

This problem is of particular interest because as technology becomes more prevalent in the classroom, it is necessary to

alter design patterns to accommodate the end users, that is, not only the students who will eventually be using the technology but also the teachers who will be creating digital lessons who may not have a technical background. This is especially important in a virtual reality (VR) environment where the user may not be familiar with UI/UX design standards for a VR application. In a sense, there are two end-users to consider: the student and the teacher. This line of inquiry will be of interest to HCI researchers interested in how to design VR apps for inexperienced VR users.

RELATED WORK

Previous work related to this project can be broken up into the following sections: 1) The previous work done on Lifebrush by Dr. Timothy Davison. 2) Previous work done on interactive virtual reality learning environments. And 3) previous work done modelling COVID-19.

Lifebrush

Lifebrush is a project created by Dr. Timothy Davison at the University of Calgary under the supervision of Dr. Christian Jacob [1]. Lifebrush is a virtual reality environment created in the Unreal Engine which allows users to create dynamic molecular simulations.

Lifebrush uses discrete element texture synthesis to generate agents along a path painted by the user. Lifebrush abstracts away a complex multi-agent system that handles the simulation of generated agents. Lifebrush uses a modified version of an Entity-Component-System called a Graph-Entity-Component-System which applies a graph topology to agents in a scene to allow for quick traversal of agents in the simulation [1].

Interactive Virtual Reality Learning Environments

Virtual Reality (VR) is "widely used in the fields of education and training due to its potentials in stimulating interactivity and motivation" [4]. One of the motivations for taking advantage of a VR environment is that it allows students to access otherwise inaccessible spaces such as the inside of a cell. This project emphasizes this finding by taking the user both down to the scale of a virus and up to the scale of an entire community.

VR learning environments have been shown to benefit students' spatial knowledge, experiential learning, motivation/engagement and contextualization [5].

Shim et al. evaluated the efficacy of VR technology for teaching topics in biology. They conclude that VR biology simulations can help "explain abstract science concepts and events by visualization in reality and immersion" [6]. This conclusion is also reflected in a study done in 2013 by Tan et al. where virtual reality exercises "helped clarify understanding while increasing interest and engagement" [7]. Virtual reality learning environments prove to be very effective in helping students engage with the educational material.

Interactivity is an important aspect of immersion and telepresence in a VR environment[8]. One of the conditions which create a sense of immersion in a virtual reality environment is creating a space where “the user’s actions must have a non-trivial impact on the environment” [9].

However, Rousseau et al. demonstrated that while a fully interactive VR environment aided in student’s problem-solving abilities it did not provide conceptual change [10]. In that study, it was the passive virtual environment which supported student’s ability to reflect and recall new information.

Taking both those findings into consideration, this project provided both a passive learning environment as well as an active learning environment. The passive environment is exemplified in the COVID-19 Gallery immersive learning environment. The active component of this project is exemplified in the Community Transmission Simulation. Both of which are detailed in the “Developed Solutions to Problem” section.

Modelling COVID-19 Community Transmission

For modelling the spread of a virus through a population, a multi-agent model was adopted. Doneic et al. applied a multi-agent system to simulate road traffic because a multi-agent system allows simulation of complex phenomena with many actors [11]. The motivation to apply a multi-agent system to traffic also applies to applying this system to a community of agents in a population. The multi-agent system allows an abstraction of the complicated behaviours of individuals.

Wasik et al. evaluated the effectiveness of multi-agent systems to model Hepatitis-C infection by comparing it to the current method based on differential equations [12]. Researchers concluded that the multi-agent approach has many advantages to other model types in the parameters of flexibility, re-usability and the potential to apply the same model to other viruses.

Swarup et al. highlight the many challenges in the field of computational epidemiology and suggest multi-agent systems can contribute much to the field of epidemiology [13]. Large-scale multi-agent simulations offer the opportunity to “rigorously investigate the effects of intervention strategies under varying initial conditions, varying levels of vaccine efficacy” [13]. In designing the Community Transmission Simulation for this project, it was important to let the simulation parameters be adjustable in real-time so users can see the impact of altering the virus’ rate of transmission, death rate and the percentage of immunization.

DEVELOPED SOLUTIONS TO PROBLEM

COVID-19 Gallery Immersive Learning Environment

Previous scenarios created for Lifebrush take place in a sanitary, artificial environment. While sufficient for testing

and tweaking Lifebrush’s features, it was necessary in this project, to show how Lifebrush can be integrated into a natural, realistic environment to enhance immersion. This liminal space that previous Lifebrush experiences took place in does not resemble any real-world space and as a result can detract from the user’s sense of immersion.

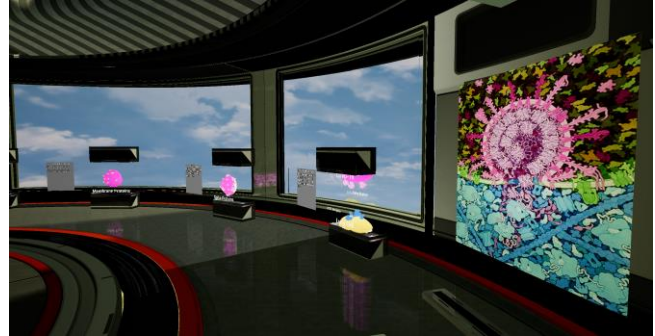


Figure 2 Screenshot of the COVID-19 Gallery Immersive Learning Environment. Each station contains a 3D model created in Lifebrush along with a text description.

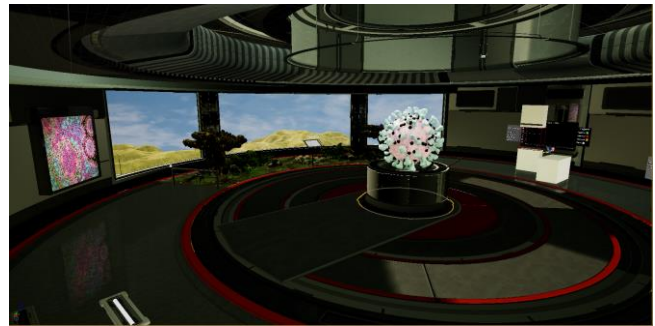


Figure 3 Screenshot of COVID-19 Gallery showing the indoor garden as well as the COVID-19 model as a centerpiece.

The inspiration for this project’s environment was to create a virtual reality gallery where Lifebrush models are on display. COVID-19 models created in Lifebrush line one half of the circular room, accompanied by informational text panels akin to a real museum gallery. The gallery is also populated by stylized illustrations of COVID-19 by David Goodsell, an indoor garden as well as a 3D model of the COVID-19 (created by protuberanka and downloaded from TurboSquid.com) virus as a centerpiece. Each of these elements are in service of creating immersion through matching the user’s expectation of the game environment with their expectations of the real-world environment [9]. The real-world environment in question is that of a museum where one would expect art pieces of various media (i.e. painting, sculpture, mixed-media, etc.)



Figure 4 Closer look at the gallery stations present in the COVID-19 Gallery.

PDB Importer

One of the goals of this project was to create tools for educators who wish to use Lifebrush. The Unreal Engine is a very versatile game engine which has an extensive API and documentation solely for creating new tools integrated directly into the editor. These tools can be created as “Plugins” which can then be packaged and moved between projects .

A PDB (Protein Data Base) file is commonly used in bioinformatics and biological modeling. PDB files detail the coordinates of molecules in 3D space, and with a PDB file one can use a program like PyMol to create 3D models of molecules for use in Lifebrush.

A PDB Importer plugin was created and integrated into Lifebrush. When a user clicks the “PDB Importer” button or selects the option from a menu, a separate editor window pops up where the user can enter in a list of PDB Identification Numbers for import. Since PyMol is used for fetching, converting and exporting the PDB files into OBJs, the user is required to provide the path to their installation of PyMol. Having access to the PyMol installation path is necessary because each installation of PyMol comes with all the files and scripts needed to activate the virtual environment to easily access all of PyMol’s scripting features. The goal is to make sure the end user does not have to deal with python installations or setting up virtual environments, all they need is the PDB Importer plugin and an installation of PyMol.

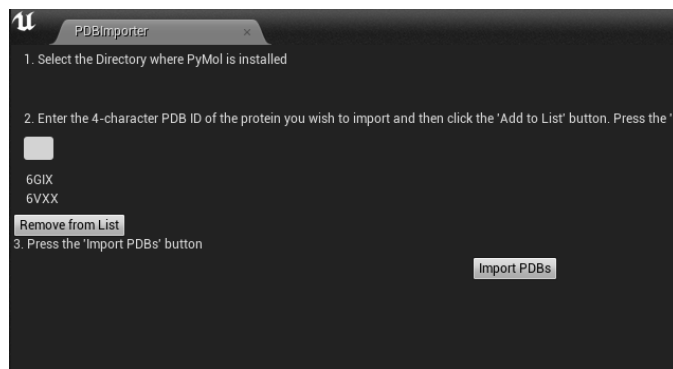


Figure 5 Screenshot of PDB Importer in the Unreal Engine Editor.

The PDB Importer UI in the Unreal Engine editor is connected to a bash script which will execute a separate PyMol script. That PyMol script will fetch the desired PDB file from the Protein Database and export it as an OBJ file inside the Unreal Engine project directory.

As of now, the UI for the PDB Importer is complete as well as the PyMol script. Both are functional in their respective domains but within the time frame allocated in the schedule, I have been unable to connect the UI with the PyMol script. A future direction for this project would be to complete the PDB Importer and put it on the Unreal Engine Marketplace for others to use.

Snapshot Slideshow

In Lifebrush, users “paint” dynamic molecular “paintings” that are actually 3D models. These models can then be saved as “snapshots”, allowing these 3D models to be manipulated within the scene.

The idea of the “Snapshot Slideshow” tool is to create an Unreal Blueprint which will allow the user to easily create an interactive slideshow with all their snapshots. Unreal Blueprints [14] are similar to prefabs in the Unity game engine and are reusable classes of Unreal Actors which can be instantiated within a scene (Fig. 6).

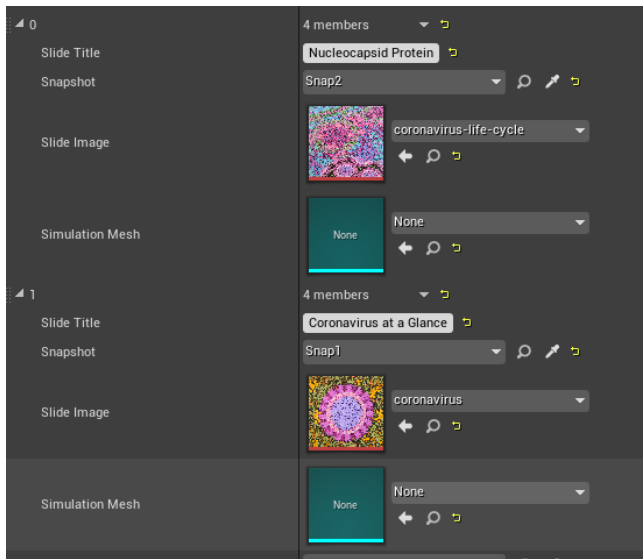


Figure 6 Snapshot Slideshow interface in the Unreal Engine Editor. Users can create in-game slideshows by dragging and dropping assets and filling in information fields.

The Snapshot Slideshow Blueprint contains a list of slides that can be populated by the user. Each slide consists of a Lifebrush snapshot as well as description text to accompany the snapshot. Populating the slideshow with the slides is as simple as drag-and-drop and fill-in-the-blanks. At runtime, that public-facing information is used to create the slideshow which the user can navigate through with the onscreen UI.

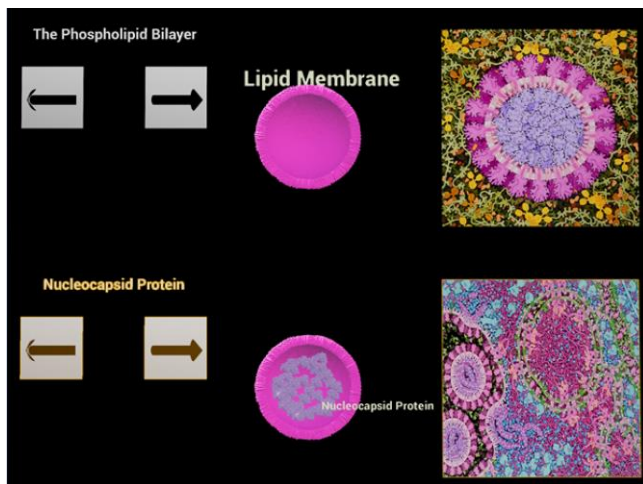


Figure 7 Snapshot Slideshow Blueprint lets users flip through a preset slideshow of images and 3D models created in Lifebrush.

This feature was successfully implemented within the timeframe allocated in the project proposal. A future direction for this tool would be to enable the snapshots to be dynamic simulations rather than static snapshots. After learning more about the Lifebrush API's implementation of the NVIDIA Flex particle system and Realtime Mesh Component Plugin, I believe it is possible however it would

require more time than what was originally allocated. The Flex system object within a scene contains a pointer to the active snapshot at a given time. It is a matter of changing that pointer to point to the snapshot that is currently on screen.

Community Transmission Simulation

Another goal of this project was to show the versatility of Lifebrush as an educational tool. Lifebrush has extensively been used to model systems at the microscopic scale such as mitochondrial energy production and synaptic transmission. However, Lifebrush's abstraction of swarm intelligence makes it an ideal tool to model topics in public health.

The Community Transmission Simulation uses Lifebrush's API to simulate the transmission of COVID-19 through a community. Agents that are healthy, immune or infected with COVID-19 can be placed within the simulation and the resulting simulation will show how herd immunity affects populations.

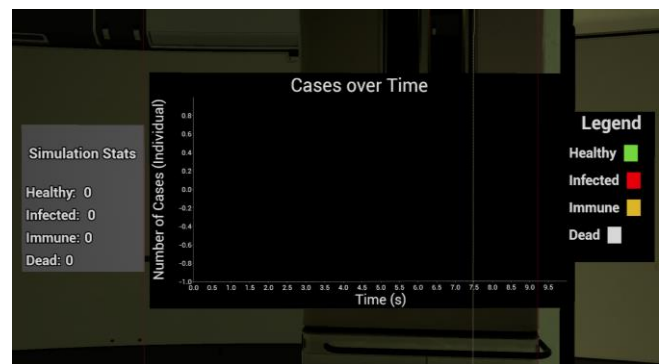


Figure 8 In-game graph provides real-time data on the number of healthy, infected, immune and deceased agents in the Community Transmission Simulation

All agents move randomly within the confines of the simulation via a random walk provided in the Lifebrush API. If a healthy agent comes into contact with an infected agent then there is a chance that the healthy agent will undergo state change from healthy to infected (Fig. 9). Successful state transition from healthy to infected depends on the r_0 value of the simulation and the probability of infection. Both of these values can be altered during runtime via an in-game UI.

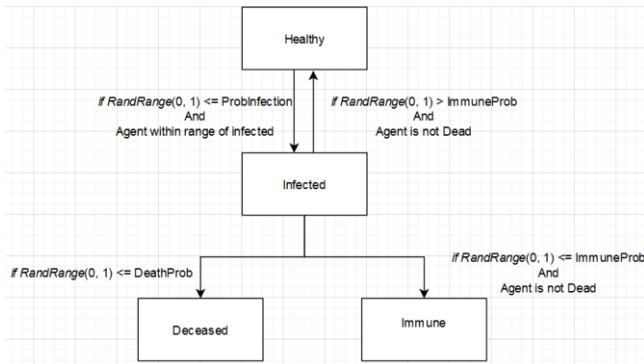


Figure 9 Graph showing the possible state transitions for agents in the COVID-19 Community Transmission Simulation.

After a user-defined time interval, infected agents can either transition back to being healthy, transition into an immune agent, or become deceased where it will be removed from the simulation. The probability of each of these transition events can be altered via the in-game UI (Fig.10).

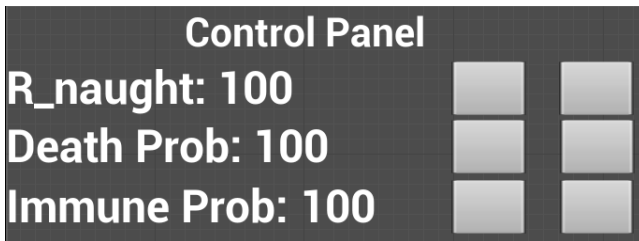


Figure 10 In-Game UI panel which allows control of COVID-19 Community Transmission Simulation.

During each simulation interval, for each infected agent a random number is generated. If that random number is less than or equal to the probability of death, then that particular agent “dies” and is removed from the simulation. If the agent does not die then there is a chance it will become immune. Immune agents can no longer become infected and cannot infect other agents (Fig. 11).

```

if(timeInterval == 0)
    foreach(InfectedAgent in Simulation)
        RandFloat = RandRange(0,1)
        if(RandFloat <= DeathProb)
            infectedAgent.Dies()
        else if (RandFloat <= ImmuneProb)
            infectedAgent.becomesImmune()
    
```

Figure 11 Pseudocode for algorithm determining whether an infected agent transitions to become an immune agent or dies.

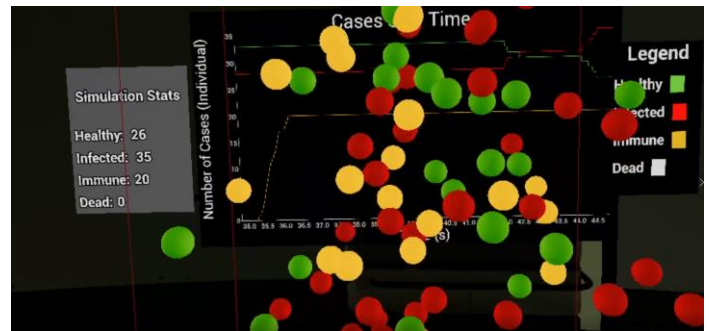


Figure 12 Agents in the Community Transmission Simulation interact with each other while real-time stats are being tracked in the in-game UI.

When a given infected agent comes into contact with any healthy agent, the probability of infection is calculated as such.

$$Prob_{Infection} = 1 - (numInfected/r_0) + RandRange(0,0.5)$$

Figure 13 Equation to calculate probability of transmitting virus from infected agent to healthy agent where *numInfected* represents total number of healthy agents already infected by infected agent.

The number of agents that the infected value has already infected is divided by the r_0 value of the simulation (which can be altered at runtime). That value is subtracted from 1 to give a percentage, then a random number between 0 and 0.5 is added to that value to introduce an element of randomness to each potential infection event. A random number generator then chooses a float between 0.0 and 1.0 and if that random float is less than or equal to the previously calculated probability value then the healthy agent becomes infected. The reason for this calculation is to approximate the effect of a virus’ r_0 value which quantifies a virus’ infectivity. The probability of an infected agent infecting a healthy agent decreases as the number of infected agents approaches r_0 (Fig. 14).

```

if(healthyAgent.inProximity(InfectedAgent))
    probInfection = 1 - (InfectedAgent->numInfected / r0) +
    RandRange(0, 0.5)
    RandFloat = RandRange(0,1)
    if(RandFloat <= probInfection)
        healthyAgent.becomeInfected()
    
```

Figure 14 Pseudocode of algorithm determining whether a healthy agent transitions to an infected agent.

The final component of the Community Transmission Simulation is a line graph showing the number of healthy, immune, infected and deceased agents over the simulation time interval. The user can adjust infection parameters and visually observe how those parameters affect various rates of transmission and death (Fig. 12).

Future directions for this tool would be to implement recording of the data generated from this simulation. In its current implementation, the simulation discards data as soon as the application closes. It would be useful for researchers using this tool if the data were recorded and exported in CSV format.

EVALUATION OF DEVELOPED SOLUTION

The developed solution was evaluated by getting subjects to play through the developed Lifebrush scenario and having the subjects fill out a questionnaire.

The questionnaire was heavily based on the work done by Tcha-Tokey et al. in their findings in *A Questionnaire to Measure the User Experience in Immersive Virtual Environments* [15]. Tcha-Tokey et al. categorized the questions present in Witmer's Presence Questionnaire [16] into their respective contexts. The context of importance for this study was specifically immersion and presence of the user in the virtual environment. Additional questions were added to evaluate the benefits of virtual learning environments relative to traditional lecture-based methods.

Ideally, evaluation would entail inviting subjects to the lab and letting users experience the virtual reality experience in a controlled environment and then conducting the survey. Unfortunately, due to COVID-19 safety measures this was not possible in the time frame of the study. Instead, the Lifebrush scenario and questionnaire were sent out to colleagues to elicit their feedback. To try and gain as much data as possible, a special non-VR version of the Lifebrush scenario was created so subjects who do not possess VR gear can still participate. Since the non-VR users will not be able to experience the interactive elements of the Lifebrush simulation, their questionnaire had to be altered to focus on immersion and presence rather than the interactive elements.

The questionnaire is detailed in Table 1.

| |
|---|
| 1. How much did the visual aspects of the environment involve you? |
| 2. How much did the auditory aspects of the environment involve you? |
| 3. How involved were you in the virtual environment experience as a whole?* |
| 4. How natural was the mechanism which controlled movement through the environment?* |
| 5. Did this experience teach you new information about COVID-19? If yes, what did you learn? |
| 6. Was this experience an effective way to teach topics in Biology? Why or why not? |

7. Would you prefer this experience over a traditional slide-based lecture? Why or why not?

Table 1 Questionnaire administered to subjects to evaluate immersion, presence and learning outcomes of virtual learning environment. Asterisk (*) denotes questions administered only to VR-users.

At the time of writing this report, there have been no responses to the questionnaire, thus there is no data to be discussed.

CONCLUSION

I successfully designed a VR learning environment containing both passive and active elements for teaching users about the COVID-19 virus. The passive elements consist of a gallery where users can view exhibits created in the Lifebrush environment which show the molecular make-up of COVID-19. The active element shows how community transmission of a virus can affect a population at the community scale.

The "Slideshow Snapshot" tool was successfully implemented as a re-usable Unreal Blueprint. While I was able to successfully configure a working UI for the "PDB Importer" tool and an automated PyMol script for converting PDB files to OBJ files, the "PDB Importer" is not functioning as a complete pipeline. At the time of writing this report I have not received critical feedback on the state of the developed solution and cannot conclude on its efficacy as a VR learning environment.

Future directions for this project include completing implementation of the "PDB Importer" tool, playtesting the COVID-19 VR learning environment and eliciting feedback and adding more interactive elements into the gallery including letting users paint their own COVID-19 molecular models with Lifebrush.

REFERENCES

- [1] T. Davison, F. Samavati, and C. Jacob, "LifeBrush: Painting, simulating, and visualizing dense biomolecular environments," *Computers and Graphics (Pergamon)*, vol. 82, pp. 232–242, Aug. 2019, doi: 10.1016/j.cag.2019.05.006.
- [2] "The most powerful real-time 3D creation platform -UnrealEngine." <https://www.unrealengine.com/en-US/> (accessed Dec. 13, 2020).
- [3] K. Megget, "Even covid-19 can't kill the anti-vaccination movement," *The BMJ*, vol. 369. BMJ Publishing Group, Jun. 04, 2020, doi: 10.1136/bmj.m2184.
- [4] L. Freina and M. Ott, "A Literature Review on Immersive Virtual Reality in Education: State Of The Art and Perspectives." Accessed: Dec. 09,

2020. [Online]. Available: <http://www.google.com/patents/US3050870>.
- [5] B. Dalgarno and M. J. W. Lee, "What are the learning affordances of 3-D virtual environments?," *British Journal of Educational Technology*, vol. 41, no. 1, pp. 10–32, Jan. 2010, doi: 10.1111/j.1467-8535.2009.01038.x.
- [6] K.-C. Shim, J.-S. Park, H.-S. Kim, J.-H. Kim, Y.-C. Park, and H.-I. Ryu, "Application of virtual reality technology in biology education," *Journal of Biological Education*, vol. 37, no. 2, pp. 71–74, 2003, doi: 10.1080/00219266.2003.9655854.
- [7] S. Tan and R. Waugh, "Use of virtual-reality in teaching and learning molecular biology," in *3D Immersive and Interactive Learning*, Springer Singapore, 2014, pp. 17–43.
- [8] J. Mütterlein, *The Three Pillars of Virtual Reality? Investigating the Roles of Immersion, Presence, and Interactivity*. .
- [9] "Immersion, Engagement, and Presence A Method for Analyzing 3-D Video Games," 2003. Accessed: Dec. 07, 2020. [Online].
- [10] M. Roussou, M. Oliver, and M. Slater, "The virtual playground: An educational virtual reality environment for evaluating interactivity and conceptual learning," *Virtual Reality*, 2006, doi: 10.1007/s10055-006-0035-5.
- [11] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espi c, "A behavioral multi-agent model for road traffic simulation," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1443–1454, Dec. 2008, doi: 10.1016/j.engappai.2008.04.002.
- [12] S. Wasik, P. Jackowiak, M. Figlerowicz, and J. Blazewicz, "Multi-agent model of hepatitis C virus infection," *Artificial Intelligence in Medicine*, vol. 60, pp. 123–131, 2014, doi: 10.1016/j.artmed.2013.11.001.
- [13] S. Swarup, S. G. Eubank, and M. v Marathe, "Computational Epidemiology as a Challenge Domain for Multiagent Systems." Accessed: Dec. 09, 2020. [Online]. Available: www.ifaamas.org.
- [14] "Blueprints Visual Scripting | Unreal Engine Documentation." <https://docs.unrealengine.com/en-US/ProgrammingAndScripting/Blueprints/index.html> (accessed Dec. 13, 2020).
- [15] K. Tcha-Tokey, E. Loup-Escande, O. Christmann, and S. Richir, "A questionnaire to measure the user eXperience in immersive virtual environments," Mar. 2016, doi: 10.1145/2927929.2927955.
- [16] B. G. Witmer and M. J. Singer, "Measuring Presence in Virtual Environments: A Presence Questionnaire," 1998. Accessed: Nov. 27, 2020. [Online].