Online Lab Design and Development

Dustin Christianson
University of Regina - Computer Science
email: dc.christianson@gmail.com

***Abstract*— In the wake of the COVID-19 pandemic, our array of online tools have become increasingly useful. Yet, there remains a real need for adaptable online solutions to help assure overall institutional success in academia, and industry. Online tooling (like Zoom, for instance) have a number of benefits to this end. However, what I call the online-lab-gap has become increasingly apparent. The regular delivery of course material in the natural sciences often requires the completion of a lab component–a component which can prove more difficult to deliver virtually than other material, for instance lecture content. To this end, I introduce LabCraft as a research study exploring the utility of video game engines as a framework for delivering a virtual alternative, and helping fill the online-lab-gap.**

I. INTRODUCTION

It is perhaps particularly true now, in the wake of the COVID-19 pandemic, that our array of online tools have become increasingly useful. Especially in an academic setting, which traditionally relies on large gatherings of individuals in close proximity, our methods must be adaptable.

We have seen over recent history a real need for adaptable, online solutions to aid in the delivery of course material and the assurance of student, faculty, and overall institutional success. Likewise, we have seen emerging technologies start to fill these gaps.

A number of tools are already available to this end: Zoom, Teams, and other online meeting software has become commonplace in academia and industry alike. While a student receiving a lecture over zoom may not have precisely the same experience as a they might in a lecture hall full of their peers, at least the lecture can still be delivered. Learning management systems (URCourses, for instance) help provide the infrastructure required to deliver a variety of course content in a unified and systematic way. Project management tools like JIRA and Trello help organize teams and tasks, online calendars can synchronize events across time-zones and continents,

There are other benefits, too: online learning allows individuals to attend from anywhere they have a viable internet connection and a computer (or even a phone). Further, online lectures can often be pre-recorded and delivered entirely asynchronously, allowing students and faculty the liberty of studying on their own schedule. In industry a lot of the same benefits arise; zoom meetings can facilitate interfacing between clients, suppliers, shareholders and any one else across different time zones and huge distances.

However, no matter the number of advantages, there seems to be a bit of lag specific to the realm of scientific study–in particular for those courses requiring a lab component. Traditionally, post-secondary students of subjects such as astronomy, chemistry, physics, and even computer science, are required to complete mandatory laboratory studies which (with the exception of computer science, perhaps) typically demand infrastructure and tooling which cannot be so easily rendered in an online format. I call this the *online-lab-gap*.

Throughout this semester we researched useful methods for delivering lab-learning content in a virtual way. Using the Ursina Engine (a video game framework written in Python), we began to develop a physics lab simulation aimed at helping deliver educational content which traditionally required the use of physical laboratory tools and environments. Further, we explored questions around the motivation of learners and decided against simulating a photo-realistic 3D laboratory setting in favour of a blocky, voxel world (ie: Minecraft).

A. *MOTIVATION*

In broad, the motivation for this work stems from three main areas:

- To diversify and adapt our online tools to be more comprehensive (the online-lab-gap)
- To explore the viability of a video game engine as an online learning tool, rather than just only a source of entertainment
- To understand better what motivates individuals to pursue engagement with learning material

I believe it is critically important to motivate learners, and in order to do so effectively requires understanding their motivation–at least to some degree. Video games, like it or not, tend to motivate their users and I think it is worth taking some time to explore the potential for using video game engines as a framework for building out tooling that will contribute to closing the online-lab-gap.

B. *BACKGROUND*

Long before any global COVID pandemic, since at least 2008, folks have been exploring the subject. Colin Price has researched at length the potential for using video game engines (in particular the UT2004 engine) to develop learning materials specifically for physics. It is interesting to note that, according to Price, ”both teachers and students” alike have the opportunity to use this tooling to develop concrete educational substance.[1] [1] In 2019, Price introduced a solution with more than thirty physics simulations, covering a wide array of natural phenomena, developed in conjunction with learners and educators focused on ”advanced level physics”. While it appears to be a robust offering, Price seems to believe it is critically important to offer ”visually realistic experimental apparatus” and the ambience of a ”typical school laboratory setting”.[2]

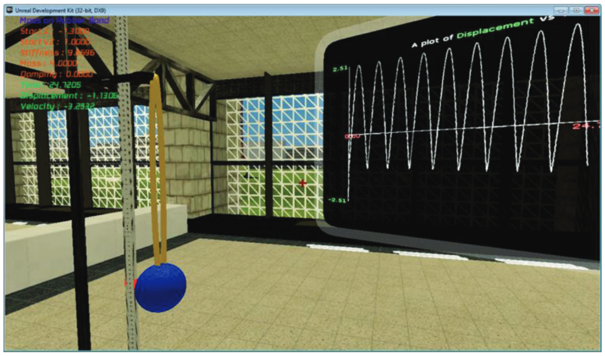

Fig. 1. The PhysLab Environment

Although most likely agree that becoming familiar with traditional instrumentation and laboratory environments should be a priority for students, I and others question whether this method of delivery is most vital. Using a video game engine as a framework to simulate the cold and sterile world of a physics laboratory might be what some call ”chocolate covered broccoli”.[3] Perhaps a radically different type of simulation environment would encourage more motivated engagement with the material in question?

Enter LabCraft.

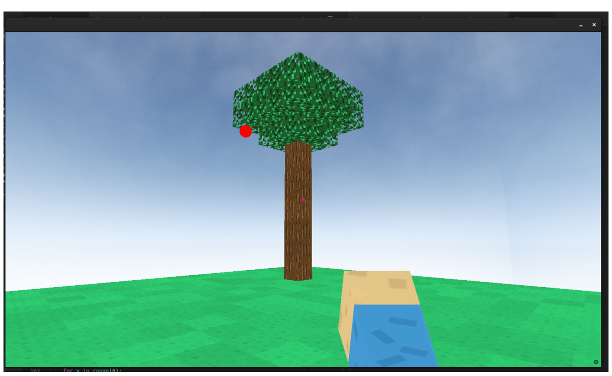

Fig. 2. The LabCraft Environment

II. LabCraft

LabCraft started as a Minecraft-clone built in Python using the Ursina Engine[4] as a framework.[2] Minecraft is a hugely popular video game among children, teenagers, and adults today–it is truly an all ages game. It has taken the world by storm and if you haven’t heard of it, or don’t recognize the ”blocky, voxel” world, you’ve got some catching up to do. Voxels are, quite simply, cubes (a bit like regular toy building blocks in how they fit together). They are like a 3D version of a pixel.

One of the main appeals of starting with a Minecraft-clone, and hence the ”blocky, voxel” world of LabCraft, was to create an environment inviting and exciting to users. To provide them with a game-based framework for learning and exploring (physics, in this case) that played off their familiarity with Minecraft in order to encourage and motivate them toward exploring the opportunity for online laboratory simulations.

A difficulty, however, presented itself insofar as the Ursina Engine contains very little in the way of built-in and pre-packaged physics. To the best of my knowledge, in fact, it contains almost none. Although having a robust system of entities and a fairly comprehensive three-dimensional matrix to contain them, Ursina at bottom offers little more than a few functions for changing their position and detecting some collisions. There are no built-in methods to account for forces like gravity or friction, nor even velocity or acceleration.

Having this in mind, LabCraft might seem to start off a bit behind with its beginnings with the Ursina Engine. Another video game engine as a framework, such as the Unreal Engine, contains at least some bare-bones tools for dealing with physics up front. Though, after reckoning with this material first-hand, I have come to see this as less of a drawback and more of an opportunity. As much as this may slow down LabCraft users in terms of churning out laboratory-grade physics simulations, it is also a concrete opportunity for students, teachers, and researchers alike to delve a little bit deeper into just what exactly we think about what online labs ought to be, and how we come to reckon with their design and development.

I think this ”problem” actually multiplies the utility of LabCraft beyond that of being merely an online, virtual tool for recreating rudimentary experiments common to the physics laboratory. Using tooling like the Ursina Engine, built out of the Python programming language, offers learners and educators the opportunity to discover for themselves exactly what is most important in developing virtual laboratory simulations. It encourages and motivates users to ask questions about what facets of physics simulations are really the most important, and how they should be implemented. If a student wants to model, for instance, the affect of gravity on an apple falling from a tree, they are certainly free to do

[1] I think a good solution to the online-lab-gap problem should incorporate a ’two-way street’ dynamic, rather than being solely ’handed down from above’–as much as students learn from teachers, the opposite should also be true.

[2] Using a Python-based framework, as opposed to something like the Unreal Engine (C++ framework) is an advantage because the language syntax is intuitive and easy to learn, making it perfect for users of all levels of expertise, from beginner to advanced. I made a good degree of progress myself, and had almost zero experience with Python when I began.

so–however, in such an undertaking, they will be required to develop their own simulation from almost the bottom up.

LabCraft will provide the user with a few voxels out of which the user can build their experiment, along with a few basic simulations as examples, but beyond that it is almost entirely up to the user to decide what length to go to in terms of actual simulation. Thinking again of simulating the affect of gravity on an apple falling from a tree, the user is faced with other uncertainties: should the tree be made out of dirt, or should it in fact look like a tree? Need the apple look like an apple? See Fig. 3.

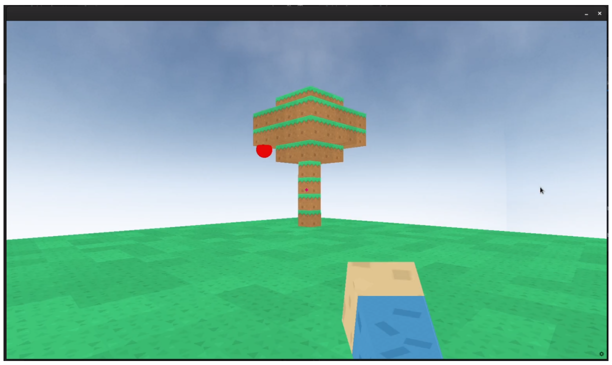

Fig. 3. An early iteration of the LabCraft tree.

At what rate should the apple actually fall? Need it fall at the same rate it would here in a laboratory on Earth? Or what else might be discovered if we adjust the rate of gravity accordingly? Further, what on earth should a researcher do with the resulting data?[3]

As a user, I also wanted to see some level of interactivity with the simulation parameters. Where a given experiment implementation allows for it, the user might like to change values associated with the running simulation–in real time. To this end I proposed and implemented, relatively easily, a makeshift set of sliders with which the user can interact to adjust the values of variables in the program controlling the simulation in order to see live results on the actual demonstration before them.

I think it is important that the user need not constantly close and open the program in order to make changes to the simulation and see their results; that sort of interaction rapidly destroys the immersion and becomes a nuisance to the overall user experience. With an interface of sliders readily accessible in the simulation itself, the user can make convenient changes on the fly without having to close and open the program or change the code directly. In addition, due in part to the friendliness of the LabCraft framework, the slider interface can also be generalized with relative ease, and can be readily adapted to other simulations as they are added.

[3]At present, a LabCraft user can choose to output their experiment data in a CSV or TSV file format, for later use. It is intuitive enough to integrate with other Python modules to present rudimentary graphs as seen in APPENDIX B - Fig. 5, but the implementation details are currently left up to the user.

Admittedly, the falling-apple experiment is only the most basic of physics exercises to simply illustrate the effect of gravity on an object in free-fall. With the push of a button, now, LabCraft will plainly and simply offer up this simulation to the user. However, the sorts of questions alluded to above are important to ask, and will not always come up in an environment where all the instrumentation and experimentation is tailor-made to the subject matter. I argue that, in its working toward creating an online lab environment, LabCraft offers and encourages a wider array of inquiry and so, not only works toward filling the online-lab-gap, but also shows us clues toward other gaps in traditional experimental methodology. LabCraft motivates users to explore beyond the basic demonstrations of traditional physics, into deeper questions about how we ought to frame our experiments and simulations.

While it is quite clear that LabCraft fails to accurately simulate a traditional physics laboratory, especially in comparison to PhysLab, it is also worth acknowledging the broader scope of a project like LabCraft and what it can offer to students, teachers, and researchers. LabCraft is fully capable of accurately simulating realistic physics, but it is up to the user to figure out how, and that takes LabCraft beyond the realm of merely offering an online lab for recreating physics experiments. LabCraft as an online learning tool, although still in its infancy, has grown to become an adaptable framework that encourages the exploration of its own implementation.

This facet is not to be understated. In the unity of game engines and online tools for education, there are plenty of opportunities to diversify our present toolkit and make it more comprehensive. Video games engines are very clearly viable frameworks for creating online learning tools, as evidenced by PhysLab, LabCraft, and others. The simple fact that LabCraft encourages and motivates users to ask questions should be reason enough to believe that further research in this field is necessary.

III. RESULTS

In brief, throughout the course of this research I have found that LabCraft fulfills and sustains the three main purposes set out in the MOTIVATION section above. There seems to be a lot of potential for such a tool to diversify our common methods of online learning and help close the online-lab-gap. Further, it has been made increasingly clear through research in the field that video game engines can serve quite well as virtual learning tools. Finally, although the questions cannot be completely answered in this scope, it is clear that a tool like LabCraft provides motivation and encouragement to engage with the subject matter on a deeper level.

IV. DISCUSSION

A good chunk of the development of LabCraft thus far has been working to overcome the some of the immediate ”drawbacks” of the Ursina Engine as a framework for online lab development. Whereas other frameworks, for instance the

Unreal Engine, are packing built-in physics, Ursina is clearly lacking in this regard as alluded to above. Again, however, it is worth noting that this sort of drawback presents an equal opportunity for delving deeper into the subject matter. It was a key factor in motivating me to uncover the capabilities of the framework and to come to understand how they can be implemented in a productive way.

Of course it is all well and good to view physics simulations in a photo-realistic three-dimensional environment that simulates a real life physics lab, but it is something else entirely to reckon with building out that functionality from the ground up. This is where I believe the utility of LabCraft can really shine.

At present, LabCraft does not offer the user (teacher, student, researcher, or whomever) an array of canned experiments and simulations for sitting back and viewing at the click of a button–and I believe that if it did, it would miss the point. Instead, LabCraft encourages the user to build out their own, from the ground up, using only some very basic entities and functions to get started. I alluded earlier to the building block-esque nature of the ”blocky, voxel” world and I think it is fair to take that notion a little bit farther.

What LabCraft really offers the user is a set of building blocks for working out their own virtual lab environment, tailored not necessarily specifically to physics experimentation per se, but rather tailored to a broader and more generalized scientific questioning ultimately rooted in computer science and that is not simply by accident.

There is more opportunity in broad to pursue research into the development of LabCraft, I think not only as a utility for offering virtual physics experiments, but also as a method of researching computer science. It could be a perfect tool for students, teachers, and researchers alike to explore different subject matter in this field in particular. Topics like basic programming syntax and concepts like variables, functions, data-types, and arrays, interfacing with the operating and file systems, three dimensional modeling, computer graphics, and even networking (just to name a few) are well within the capabilities of LabCraft. I would encourage the reader, and anyone else interested, to give it some further thought.

In the end, while an assortment of more than thirty built-in simulations is certainly impressive and tremendously useful in filling the online-lab-gap, PhysLab might be lacking in precisely what LabCraft seems to offer. What that is, exactly, is a good question. I believe in part it is a fascinating world that encourages questions about how and why simulations are important to us, and what details about them are most crucial. In addition, it offers up a framework for finding the answers.

V. CONCLUSION

In brief, I believe there is a very strong case to be made for LabCraft. It can be a useful to help diversify our online toolset in the increasingly virtual world. It is a great example of a game-engine framework adaptable to the needs of online learners. Finally, it works to motivate the user toward asking questions and finding answers. Of course in its current state it is not a rival to something like PhysLab, but no longer does it need to be. Ultimately, I think LabCraft is a very promising piece of work alone, and it has played a key role in motivating questions of my own.

APPENDIX A - ADDITIONAL IMAGES

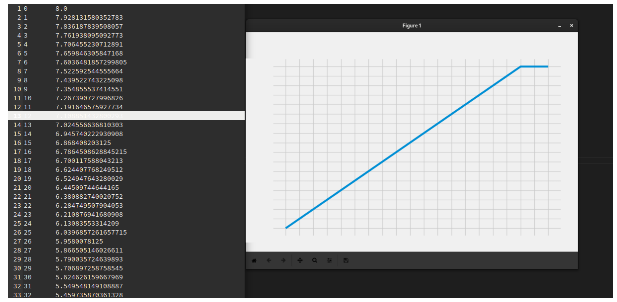

Fig. 4. A Rudimentary Data Set and Graph

ACKNOWLEDGMENTS

Kind thanks to Dr. Trevor Tomesh for facilitating this research project, and the University of Regina for making it possible.

REFERENCES

[1] C. B. Price, “Learning physics with the unreal tournament engine,” Physics Education, vol. 43, no. 3, pp. 291–296, 2008.
[2] C. B. Price and R. Price-Mohr, “PhysLab: A 3D virtual physics laboratory of simulated experiments for Advanced Physics Learning,” Physics Education, vol. 54, no. 3, p. 035006, 2019.
[3] M. P. Habgood and S. E. Ainsworth, “Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games,” Journal of the Learning Sciences, vol. 20, no. 2, pp. 169–206, 2011.
[4] P. Amland, “Ursina Engine,” ursina engine, 2022. [Online]. Available: https://www.ursinaengine.org/. [Accessed: 10-Apr-2022].