Use of Unity in Scientific Simulation and Modeling for Research and Education

Nathan Hutchins, Dr. Loyd Hook, Willam Friedel, Zack Kirkendoll
Department of Electrical and Computer Engineering
The University of Tulsa
Tulsa, Oklahoma, USA
{nathan-hutchins, wgf603}@utulsa.edu

***Abstract*—Simulation and Modeling in scientific research and education is very important when it comes to displaying results to the scientific community as well as the public. This can be very time consuming using the standard scientific tools with very minimal results. The University of Tulsa's Vehicle Intelligence and Autonomy Lab (VAIL) has begun using the C# based game engine Unity for the development of simulations in research and classroom settings. This paper compares the use of MATLAB and Unity for the simulation, modeling, and informational display portion of an ideal Newton's Cradle project. Using the ideal Newton's Cradle as the example allows us to verify and validate the accuracy and usefulness of Unity as the modeling and simulation of systems under development by VAIL. Unity has been verified in two ways, first, MATLAB was used to develop the behavioral model of the ideal Newton's Cradle then the output was imported into Unity for verification and validation. Second, the ideal Newton's Cradle model was developed entirely in Unity, using C#, and then the result was compared to MATLAB, again, for verification and validation. The results of VAIL testing on Unity are very positive. Using Unity, VAIL is able to more quickly develop Simulations and prototypes for the current research.**

Keywords—Simulation, Unity, Modeling, Education

I. Introduction

Current simulation and modeling in education and research is almost exclusively limited to basic programming languages, such as Java, 'C', and Matrix Laboratory, MATLAB. These tools are established in the research and education worlds as an acceptable standard of scientific research for calculation and modeling[1].

When working on simulation and Modeling projects for research or classroom work in science and engineering it is important to be able to visualize your information and allow others to visualize your data in a way that can be more easily understood. The background of this paper comes from the idea that not everyone that reads these papers are scientists and engineers, or like to see plain graphs and tables of numbers, then told to infer their conclusion or read the needed information from a table. This type of scientific and engineering communication perpetuates the idea that science and engineering are difficult which is not as true as it seems. Using programs and engineering tools to help alleviate this issue is the best way to educate people on difficult topics in a way they can understand.

It was in a case like this that VAIL began investigating the use of Unity as another tool in our research toolbox for communication and visualization purposes. After working with Unity for several weeks, it became clear that, with its inclusion of native graphics, we were able to create simulations that clearly represented our points with greater accuracy and detail than before while at the same time increasing the audience that could understand our topics with the visualizations in outside demonstrations.

The main goal of this paper is to investigate the use of Unity in education and research, comparing its scientific simulation results to the more expensive and more widely used MATLAB for verification and validation of Unity for scientific use.

II. Example

The research started by implementing, in MATLAB, an ideal Newton's Cradle, also called a Newtonian demonstrator, similar to the one pictured in figure 1, to gather a research standard baseline[2][3].

Figure 1: Sample Newton's Cradle

Our ideal Newton's Cradle is a five-mass device that has perfect energy transfer during collision, perfectly elastic collisions. The plot in figure 2 is the plot of the angle, in radians, of the masses. As you can see in figure 2 the middle 3 masses do not move and the edge masses are oscillating to a max value of radians. Note that in figure 2, the five lines representing the five masses have been shifted to show all lines clearly. Mass 1 is shifted by 0.2 rad, Mass 2 is shifted by 0.1 rad, Mass 3 is not shifted, Mass 4 is shifted by -0.1 rad, and finally Mass 5 is shifted by -0.2 rad.

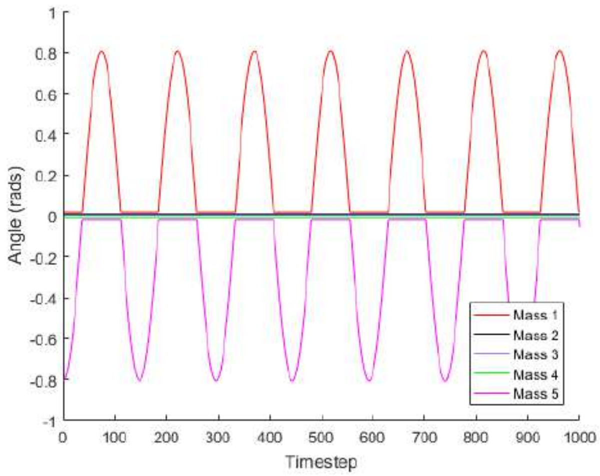

Figure 2: Angle Plot of Ideal Newton's Cradle in MATLAB

After successfully modeling an ideal Newton's Cradle in MATLAB we moved to model the same object in Unity displayed in figure 3. This was accomplished by starting a Unity project and importing five identical balls on the end of a solid lever arm. Mass transfer was handled by Unity's collision detection and this program ran at Unity's native clock rate.

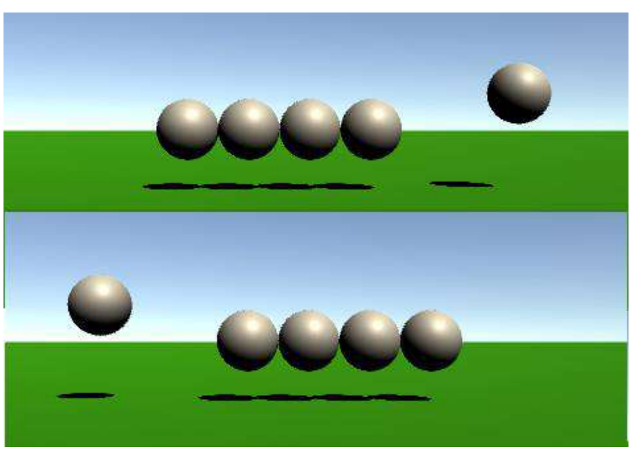
Figure 3: Newton's Cradle Simulation in Unity

Using a C# script, information about the position and timing of the Newton's Cradle was recorded for further analysis and comparison with the MATLAB data. The data recorded in unity was imported to MATLAB and plotted on the same angular plot, with a modified time scale, as the previous plot which is displayed in figure 4. Again, for clarity, the five lines representing the five masses have been shifted to show all lines clearly. Mass 1 is shifted by 0.2 rad, Mass 2 is shifted by 0.1 rad, Mass 3 is not shifted, Mass 4 is shifted by -0.1 rad, and finally Mass 5 is shifted by -0.2 rad.

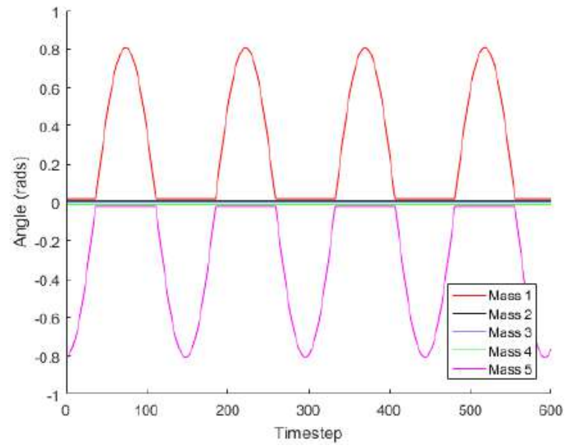

Figure 4: Angle Plot of Ideal Newton's Cradle in Unity

As the figure shows, the graphs are nearly identical. This reinforces the ability to use Unity for research simulation and modeling as well as giving the benefit of giving a starting project for teaching Unity to students. In figure 5, the plot shows the plot that includes Unity mass 1 plotted on MATLAB mass 1, as well as Unity mass 5 plotted on MATLAB mass 5, which shows that the two plots are very close with any errors being well within the acceptable deviation of error.

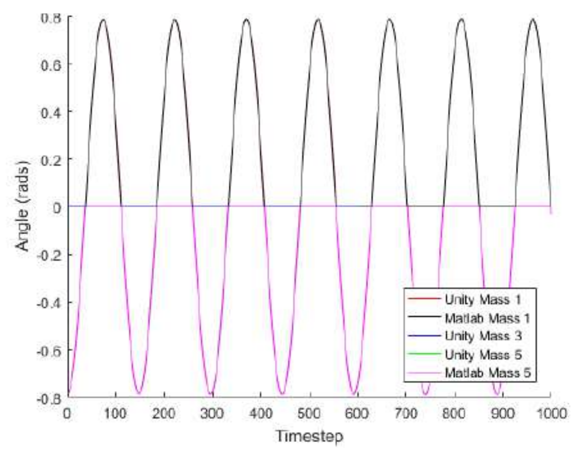

Figure 5: Angle Plot of Comparison of Unity and MATLAB for Ideal Newton's Cradle

III. Examples of Increased Complexity

The simple Ideal Newton's Cradle was just the first of many complex simulations and models that are being developed for teaching Unity to the research group and students. The complexity of future systems are exponentially increasing to include flight systems and autonomous vehicles for the development of these systems. A few other examples of simulations in unity are the inverted pendulum problem and the 2D and Unmanned Arial Vehicle (UAV) flight simulator, figure 6.

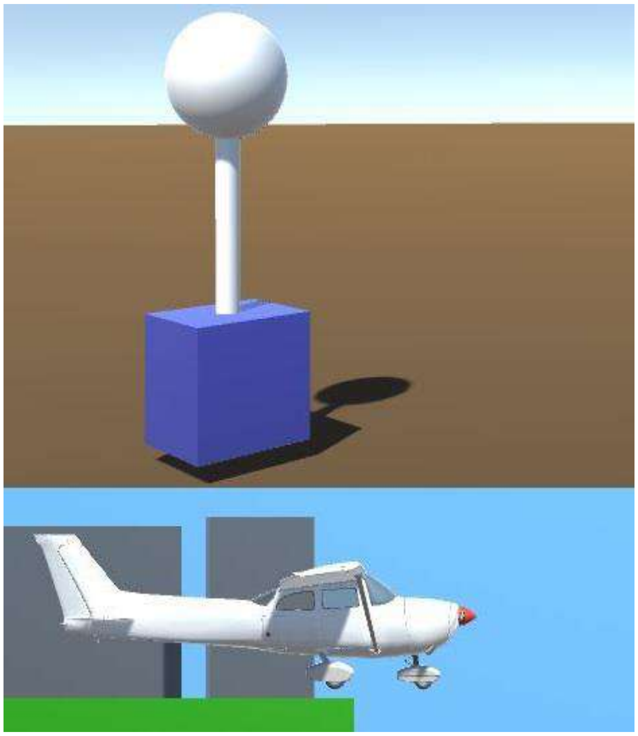

Figure 6: Top: Inverted Pendulum in Unity. Bottom: 2D UAV Flight Simulator.

Theses more complex examples are used to enforce different simulation and control ideas such as PID control (inverted pendulum) and Flight Dynamics (2D UAV Flight Simulator) [4], [5].

IV. Benefits of Unity

Unity is better than basic programming languages for making images and models for simulation visualization purposes since the graphics are the main focus of the program. Using Unity, VAIL is able to use pre-made models in simulation spending much less time on programming line for line graphics as would be needed in other languages, such as C's 'graphics.h' or native Java graphics programming. Using 3D models allow for a more detailed display of information with minimal graphics programming overhead[1].

From the example of the Newton's Cradle, we have shown that the physics, which were programmed in C# and MATLAB were basically identical with the margin of error being well below even our measuring tolerances. The biggest obstacle for working with Unity was, and still is, timing. Since unity compiles and runs for each machine, we get drastic changes in the speed of the program from platform to platform, therefore, for this comparison all programs, Unity and MATLAB, were run on the same research computer to keep the timing as consistent as possible.

There have been several 'non-video game' related projects produced in Unity, which include many training and emergency simulation programs, most of these have been used primarily for isolation of information. Recently there have been a few scientific papers published on using unity for the visualization of scientific data in the realm of autonomous systems, specifically autonomous vehicles. VAIL is currently developing an autonomous systems acceptance simulator, the Electronic Car Learning and Intelligence Program Simulator (ECLIPS), for research in human factors, acceptance, and ethics of autonomous systems [6][7].

The size of the community for Unity is one of the best parts of the adoption program into the VAIL development chain. Because of the intended nature of Unity, a video creation engine, the amount of user support available from Unity and the involved community vastly overshadows that of even the biggest scientific programming languages, including MATLAB. Over the past year of using Unity, VAIL has not run into a programming based problem that has not been solved with a short internet search.

The cost of Unity, being free for students and education research use makes it almost unbeatable. Other mentioned software, including MATLAB, can cost hundreds, if not thousands of dollars. That being said, Unity is not a full replacement for MATLAB in the VAIL research space, it is being used to develop simulations for programs and support MATLAB based research by simplifying the visualization of MATLAB data.

V. Conclusion

The use of Unity in the VAIL research group has proven a very lucrative endeavor, as well as introducing Unity to our students at the University of Tulsa, which allows them to expand their knowledge of simulation and modeling beyond basic MATLAB. Unity is not only flexible enough to display complex information in an easily legible way, but is also accurate enough for displaying this complex information correctly. Unity, moving forward, will be used not only for research modeling and simulation but also as a way for students learn simulation and modeling in classes at The University of Tulsa.

With the validation of comparing Unity to the research standard MATLAB, The University of Tulsa has verified that Unity can be used as an education and research simulation tool and is pushing forward the use of Unity in several of our simulation and research design courses.

VI. Future Work

The use of Unity in VAIL will continue with the development of the Electronic Car Learning and Intelligence Program Simulator (ECLIPS) for use in developing algorithms of for use in autonomous systems, including and autonomous cars and planes, as well as helping investigate human factors and acceptance of these systems for future deployment.

Acknowledgment

The work in this paper was supported by The University of Tulsa's Electrical and Computer Engineering Department Aircraft Simulation and Control class.

REFERENCES

[1] E. S. Roberts, E. S., Roberts, and E. S., “A C-based graphics library for CS1,” *ACM SIGCSE Bull.*, vol. 27, no. 1, pp. 163–167, Mar. 1995.

[2] J. D. Gavenda and J. R. Edgington, “Newton’s cradle and scientific explanation,” *Phys. Teach.*, vol. 35, no. 7, pp. 411–417, Oct. 1997.

[3] J. G. Winans, “General Statements of Newton’s Laws,” *Am. J. Phys.*, vol. 23, no. 5, p. 300, May 1955.

[4] R. F. Stengel, *Flight dynamics*. Princeton University Press, 2004.

[5] F. G. Irving, *An Introduction to the Longitudinal Static Stability of Low-Speed Aircraft.* Elsevier Science, 2014.

[6] T. L. Dunn and A. Wardhani, “A 3D robot simulation for education,” *Proc. 1st Int. Conf. Comput. Graph. Interact. Tech. Austalasia South East Asia - Graph. ’03*, vol. 1, no. 212, p. 277, 2003.

[7] U. Manawadu, M. Ishikawa, M. Kamezaki, and S. Sugano, “Analysis of individual driving experience in autonomous and human-driven vehicles using a driving simulator,” *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM*, vol. 2015–Augus, no. July, pp. 299–304, 2015.