# WIP: Teach Fast Fourier Transform Through Web Simulation and Visualization

**Dr. Chao Wang, Arizona State University**

Chao Wang received her Ph.D. in Electrical Engineering from University of Wisconsin, Madison. She is currently a Teaching Professor in Ira. A Fulton Schools of Engineering at Arizona State University.

**Kyle Brim, Arizona State University, Polytechnic Campus**

Kyle Brim received his BA in Mechanical Engineering Systems with a Secondary Focus in Robotics and Electrical Engineering from Arizona State University. He designs educational web applications using Unity and AR/VR applications using Unreal Engine.

# WIP: Teach Fast Fourier Transform Through a Web-Based Interactive Visualization Application


## Abstract

This Work-In-Progress paper presents the design and implementation of a web-based interactive visualization tool to help students understand and implement in software a Decimation-In-Time (DIT) Fast Fourier Transform (FFT) algorithm in a Real-Time Digital Signal Processing (DSP) Systems course. The paper will also discuss the assessment plan to evaluate the effectiveness of the application.


## Introduction

Fast Fourier Transform (FFT) is one of the Top 10 Algorithms of 20th Century by the IEEE magazine Computing in Science & Engineering [1]. It is an efficient way to compute Discrete Fourier Transform, which is widely used in many applications to perform Fourier analysis in the frequency domain.

A Decimation-In-Time (DIT) version of the Fast Fourier Transform (FFT) algorithm [2] is taught in a Real-Time DSP Systems course, a 15-week 4-credit hour senior level course in the electrical engineering curriculum. The objectives of the course are to provide students with knowledge and hands-on experience in translating DSP and communications concepts into real-time software for embedded systems using a microcontroller board with DSP capabilities.

To teach the FFT algorithm, the necessary mathematical theory and derivations are first introduced. Students are also given simple examples to calculate 2-point and 4-point FFT by hand. They then work on similar homework problems to reinforce the learning. A lab exercise is assigned to translate the theory into practice by implementing the FFT algorithm on a microcontroller. Students later apply the FFT algorithm and see it in action in an application-oriented project involving speech analysis and a real time spectrum analyzer. Despite the usefulness of the FFT algorithm in spectral analysis, students have trouble understanding the butterfly structure in the algorithm and they find it especially difficult to translate and implement the algorithm in software in the lab exercise.

Taking advantage of the advances in computer and web technologies, an interactive visualization application of a C Implementation of Decimation-In-Time Fast Fourier Transform algorithm is built using Unity game engine and is rendered in a web browser using WebGL. Students can choose to compute 2-point, 4-point or 8-point FFT from the web application. The skeleton of the C code to implement the FFT butterfly algorithm is provided just like in the lab exercise. Students need to fill in parts of the code by choosing options from drop-down menus. Students can then run the code and visualize how values are calculated on the butterfly diagram and propagated from left to right as each line of code is executed.

This paper will discuss in detail how the web-based interactive visualization tool is designed and implemented. In addition, plan on how to assess the effectiveness of the tool in helping student understand and implement the FFT algorithm will also be discussed.

The rest of the paper is organized as follows. First, existing work is briefly reviewed in the background section. The implementation is described in detail next, followed by assessment plan. The paper ends with conclusion and future work.

## Background

Enabling technological advances in artificial intelligence, computing power and mass storage, computer-aided design, software analysis and simulation techniques, multimedia communications, virtual reality, etc. have been identified as the most promising methods to improve the development and delivery of electrical and computer engineering curriculum [3].

It has now become a common practice to use technologies to enhance learning. For example, computer animations make teaching complex topics more effective and more efficient [4]. Computer simulation tools such as PSpice and MATLAB are widely used in electrical engineering curriculum. In the specific domain of teaching digital signal processing, it has been shown that emphasizing applications, programming, visualization and intuitive understanding increases the effectiveness of DSP education [5].

In addition to the improvement of student learning outcomes, technologies can further help students develop deeper understanding of fundamental principles and become self-directed learners. They also allow anytime/anywhere delivery, help reduce direct faculty involvement in course delivery and thus ease the burden of class size scaling [3].

## Implementation

Unity [6], a game creation engine was used to create the application and export it to a WebGL 2.0 [7] format. The application itself has been uploaded to an S3 AWS bucket [8]. This allows it to be directly embedded into any webpage including a Learning Management System (LMS) such as Canvas [9] using the web address. This also makes it very easy to update the application since the embedded web address remains the same when the contents of the S3 bucket are updated. The application can be run using various web browsers such as Chrome, Firefox, Safari and Edge [10].

As shown in Fig. 1, the user can choose whether they want to calculate a 2-point, 4-point, or 8-point fast Fourier transform through a drop-down menu. They can then input any set of real integer numbers that they wish to use for the calculation, or they can generate a set of random numbers by clicking the “Generate Random Values” button.

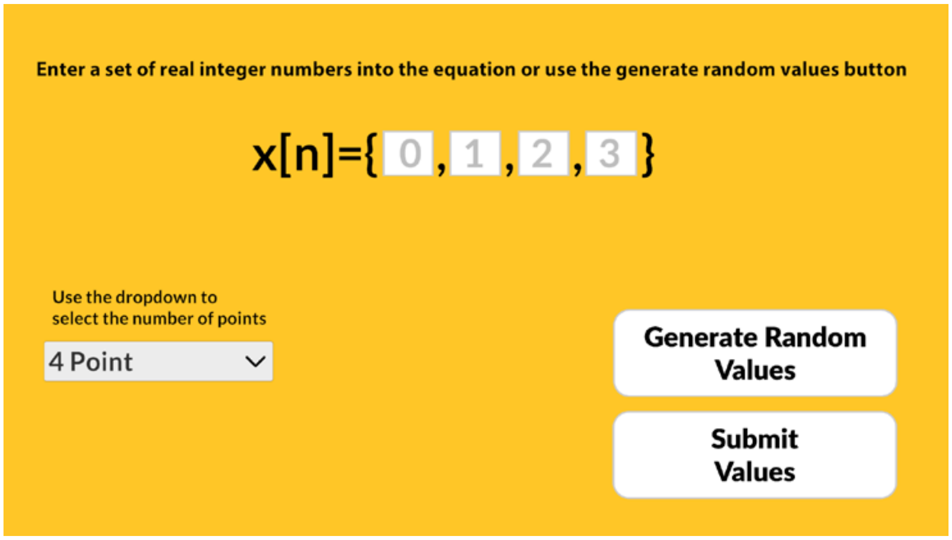


Fig. 1 Start page of Fast Fourier Transform (FFT) interactive web application where users can enter numbers for FFT calculation.

After submitting the desired numbers by clicking the “Submit Values” button, they are shown how to calculate the bit reversal form of the chosen array through a simple animation. The end of the animation is shown in Fig. 2.

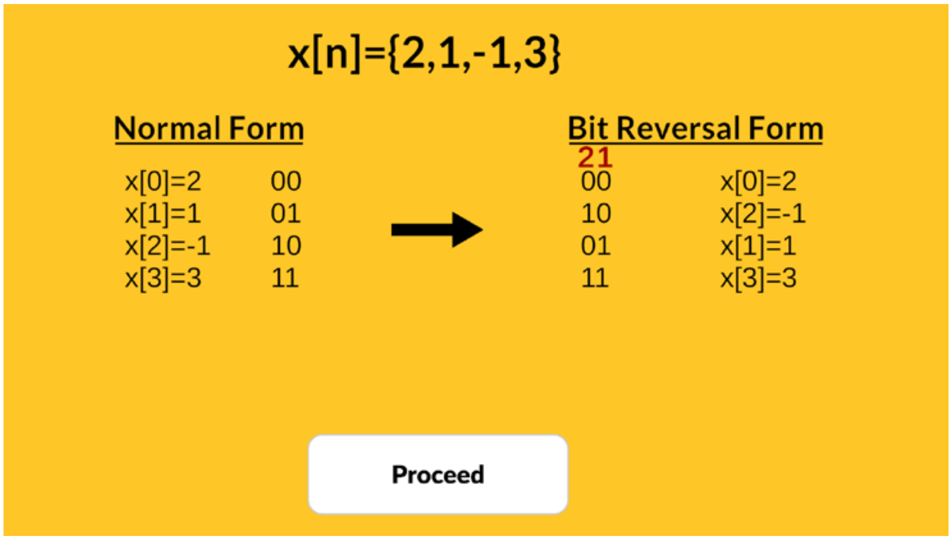


Fig 2. The input vector (on the left) is rearranged into bit reversal form (on the right).

They then enter the final part of the application containing a visual representation of how FFT is computed with the butterfly structure on the left and the pseudo code of a C implementation on the right as shown in Fig. 3. The user can toggle between “Variable” view and “Number” view through the slider button at the top left of the application window. Note that the DIT FFT algorithm implemented in the application uses “in-place” calculation, i.e., the same array X[n] used to store input (on the left side of the butterfly diagram in Fig. 3) will store the FFT output after the algorithm has been executed (on the right side of the butterfly diagram in Fig. 3).

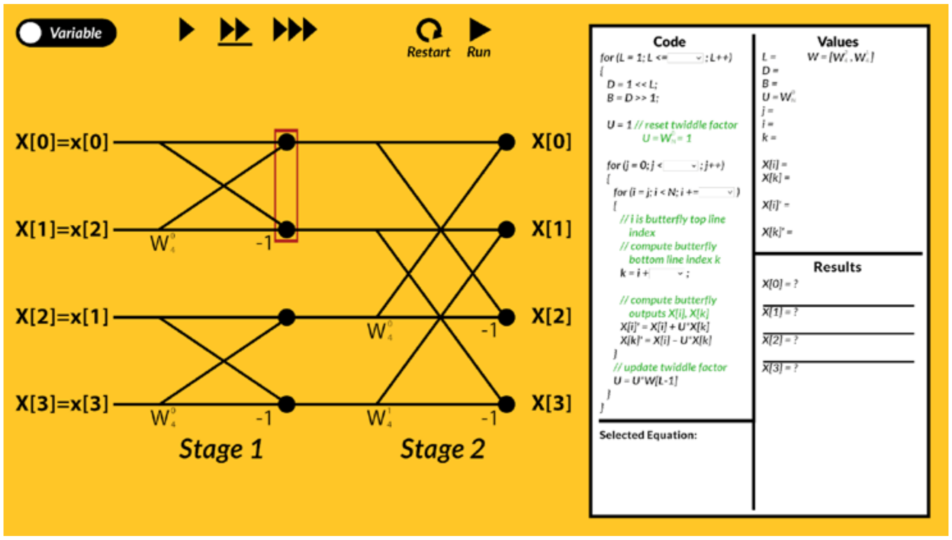


Fig. 3 (a) Variable view of the FFT butterfly structure and C implementation pseudo code.

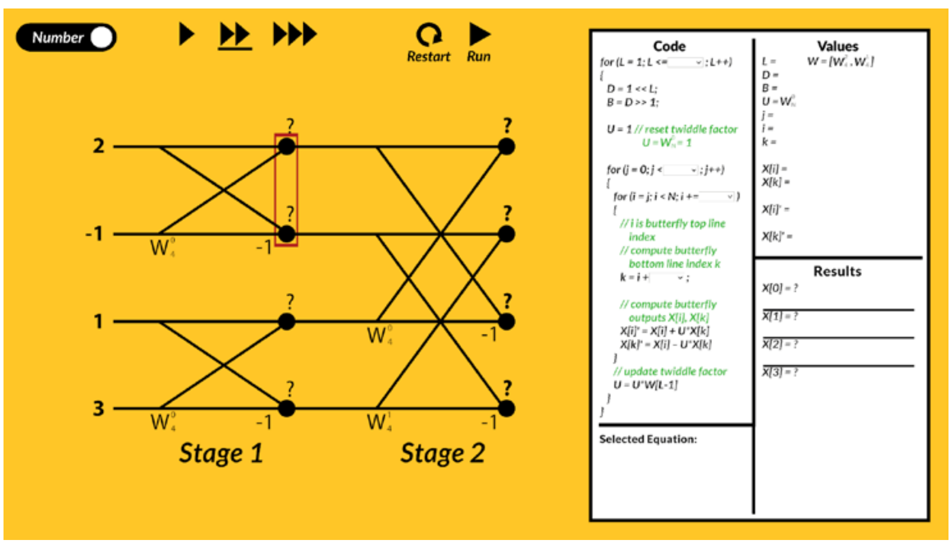


Fig. 3 (b) Number view of the FFT butterfly structure and C implementation pseudo code.

The user starts by looking at the block of the code on the right side of the screen under “Code” in Fig. 3, they will then fill out the empty section in the code using a dropdown menu. After this they can select “Run” to begin running the code. There are also three other buttons at the top of the screen that are used to control the speed at which the playback will occur so they can comfortably view how the code runs. If they have selected the incorrect option from one of the dropdowns, they will be given an error message specific to which mistake they made. Then the program will pause and the dropdowns they selected incorrectly will unlock allowing them to select the correct answer based off the error message shown. After “Run” has been clicked, the program will begin to increment through the code on the right, one line at a time, updating as it goes the value of the variables and the butterfly structure as shown in Fig. 4.

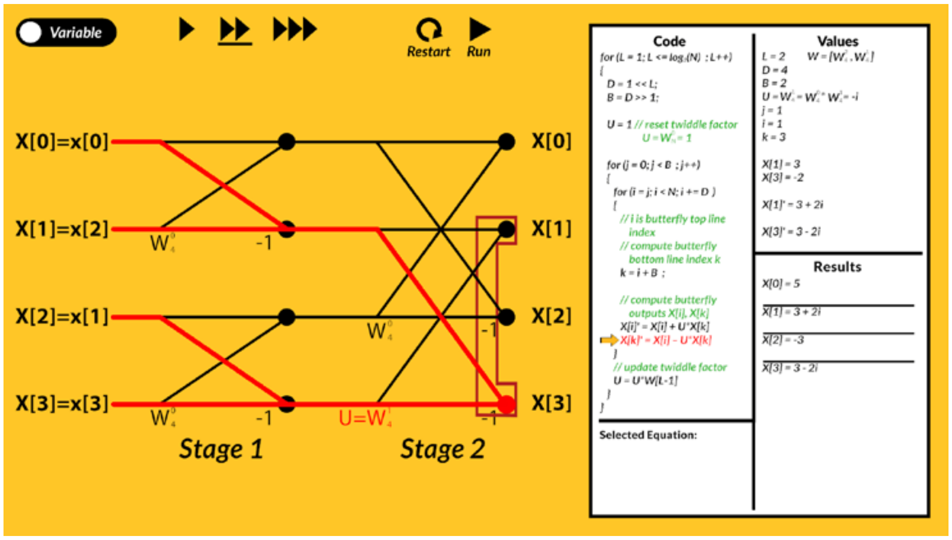


Fig. 4 (a) Variable view of the FFT butterfly structure as the code is executed line by line.

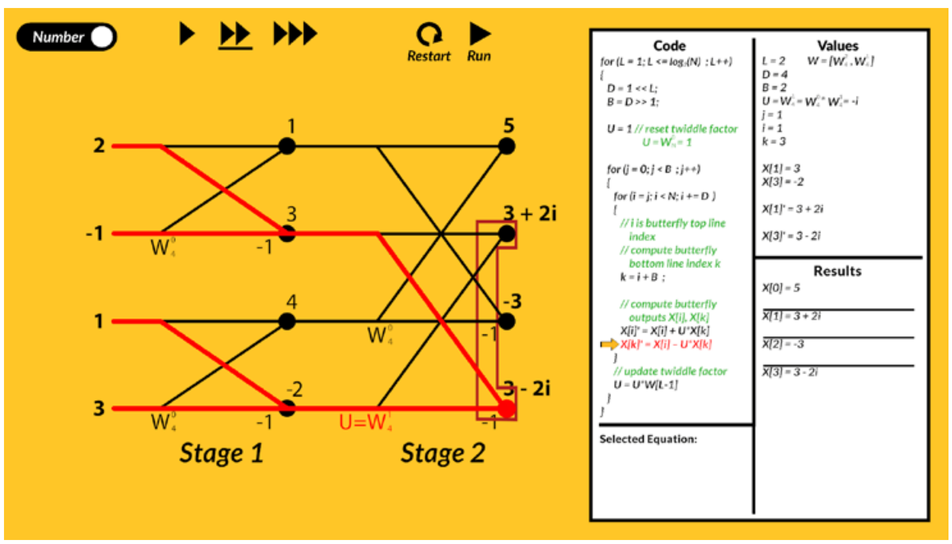


Fig. 4 (b) Number view of the FFT butterfly structure as the code is executed line by line.

In Fig. 4, the line of code just got executed is highlighted in red with an arrow in front. On the butterfly structure, the paths of values that contribute to the calculation of the current FFT number are also highlighted in red. The current values of all the variables in the code are updated on the right side under “Values”. As highlighted in red in Fig. 4 under “Code”, while the butterfly outputs get updated, to avoid confusion, under “Values” the variable X[n] represents the old value before the update, and the variable X’[n] with the apostrophe shows the updated value. The final FFT output are displayed under “Results” as they become available.

After the code has finished running and the final FFT values have been calculated and added to the “Results” on the right as shown in Fig. 5, the user can hover over any dot on the graph to get additional information. When the user hovers over a dot, red lines are drawn to show how that value is calculated. The relevant equation and calculation are also shown under "Selected Equation".

The "Restart" button at the top of the application can be pressed at any time to allow the user to input a new set of numbers to calculate or enter the same set of values to repeat the calculation.

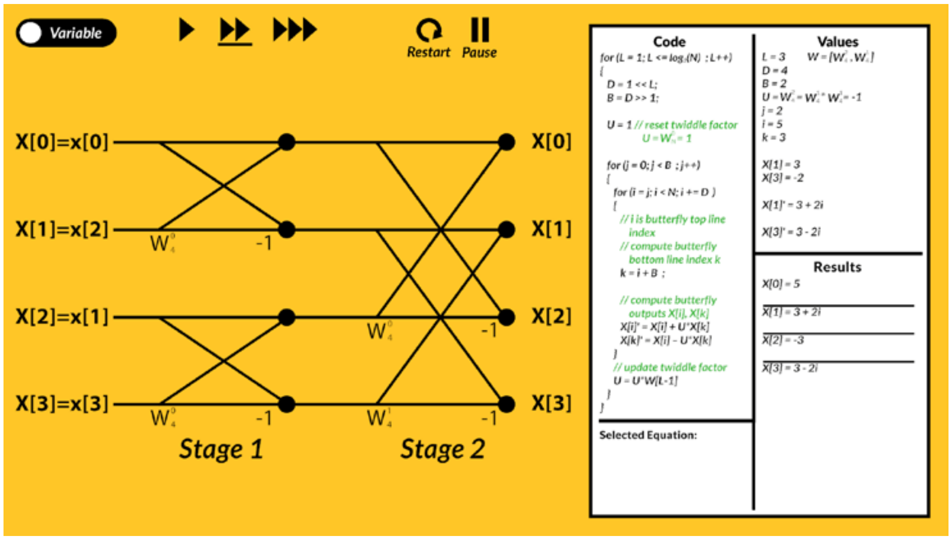


Fig. 5 (a) Variable view of the FFT butterfly structure after all code has been executed.

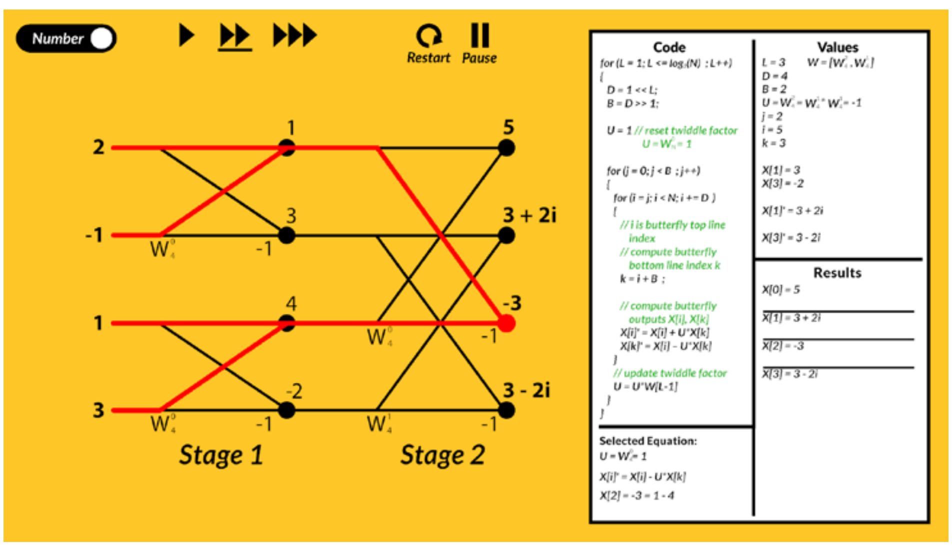


Fig. 5 (b) Number view of the FFT butterfly structure as the cursor is hoovered over the dot under FFT result "-3".

The current implementation of the application limits the input array to real integers with a maximum number of three digits. Non-integer real numbers and complex numbers are not supported. In addition, the application is designed around desktop browsers. The text could be too small to be viewed on mobile browsers. However, given the main purpose of the interactive application is to help students understand the FFT butterfly structure and software implementation, these limitations shouldn't pose serious issues.

**Assessment Plan**

The web-based Fast Fourier Transform interactive application will be piloted in a Real-Time DSP Systems course in Spring 2025.

The FFT homework problems will be modified to include the use of the online application. First, students will be asked to run the application with the same set of numbers as in the homework and compare the result with their hand calculation. This is to get students familiar with the butterfly structure and how values are calculated and propagated.

The current homework set has only 4-point FFT calculations. A problem on 8-point FFT calculation will be added to the homework using the test vector from the lab exercise. Students will be asked to show how an FFT number is calculated. They also need to show how the intermediate results along the paths are calculated and show the variable values in the code when these results are calculated. This exercise not only helps students understand a more complicated 8-point butterfly structure, it also tests students' comprehension of the code and potentially helps them with their software implementation of the algorithm.

To test the effectiveness of the interactive application, a pre- and post-quiz will be administered. Students will be asked to take the pre-quiz before they do the homework problem utilizing the application and take the post-quiz after they finish the homework problem. The pre- and post-quiz has the same set of questions including multiple choice and fill in the blank questions. The questions are like the homework questions except that they are based on a 16-point FFT implementation. Students won't be able to get the answers from the interactive application, but if they truly understand how it is implemented, they should be able to find answers for any $2^n$-point FFT implementation.

To collect feedback on students' perception of the FFT interactive application, three Likert-type scale questions will be included in the End-of-Semester Survey as shown in Table 1.

Table 1. Survey questions for the web-based FFT interactive application.

| Symbol | Question |
|---|---|
| Q1 | The interactive FFT application is easy to use. |
| Q2 | The interactive FFT application helps me understand the butterfly structure and how the FFT values are calculated. |
| Q3 | The interactive FFT application helps me understand how the FFT algorithm is implemented in software. |

One free response question such as "Is there any part of your experiences with the FFT interactive application that sticks out in your mind? Any suggestions for improvement?" will also be added in the End-of-Semester Survey to collect any additional feedback students might have regarding the application.

## Conclusion and Future Work

A web-based interactive application to visualize the FFT algorithm and its software implementation has been designed and implemented. The goal is to use the available technology to teach the complex FFT algorithm more efficiently and more effectively. An assessment plan was also created.

Future work includes collecting and analyzing survey and pre-/post-quiz results, improving upon the application based on student feedback. Positive feedback from students would warrant the development of additional interactive applications such as comparisons between linear and circular convolution, fixed-point vs. floating point implementation, etc.

## References

[1] J. Dongarra and F. Sullivan, “The Top 10 Algorithms,” in Computing in Science and Engineering, Volume 2, Issue 1, pp. 22 – 23, Jan. 2000.

[2] A. V. Oppenheim and R. W. Schafer, “Chapter 9: Computation of the Discrete Fourier Transform,” in *Discrete-Time Signal Processing*, 3$^{rd}$ Edition, Prentice Hall Signal Processing Series, 2009.

[3] F. C. Berry, P. S. DiPiazza, and S. L. Sauer. "The future of electrical and computer engineering education," *IEEE Trans. on Education*, vol. 46, no. 4, pp. 467-476, Nov. 2003.

[4] S. Kuyath, "How Computer Animations Make Teaching Complex Topics More Effective And More Efficient, " in Proc. 2002 ASEE Annual Conference, Montreal, Canada, June, 2002.

[5] K. K. Parhi, "Teaching Digital Signal Processing by Partial Flipping, Active Learning, and Visualization: Keeping Students Engaged With Blended Teaching," in IEEE Signal Processing Magazine, vol. 38, no. 3, pp. 20-29, May 2021.

[6] “Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine,” unity.com. https://unity.com/ (accessed Jan. 14, 2025).

[7] “WebGL - Low-Level 3D Graphics API Based on OpenGL ES,” khronos.org. https://www.khronos.org/webgl/ (accessed Jan. 14, 2025).

[8] “Amazon S3 - Cloud Object Storage - AWS,” aws.amazon.com. https://aws.amazon.com/s3/ (accessed Jan. 14, 2025).

[9] “Canvas LMS solutions for school and beyond. - Instructure,” instructure.com. https://www.instructure.com/canvas (accessed Jan. 14, 2025).

[10] “WebGL Browser Compatibility,” unity.com. https://docs.unity3d.com/2022.3/Documentation/Manual/webgl-browsercompatibility.html (accessed Jan. 14, 2025)