

# Web Browsers as a 3D Visualization Environment

*Literatür Makalesi/Review Article*

 Feyza Nur KOÇER ÖZGÜN

Department of Architectural Design Computing, Istanbul Technical University, Istanbul, Turkey  
[kocerf@itu.edu.tr](mailto:kocerf@itu.edu.tr)

(Geliş/Received:14.02.2022; Kabul/Accepted:25.05.2022)

DOI: 10.17671/gazibtd.1072993

**Abstract**— The challenge of modeling and rendering required for high-quality 3D visuals leads to the use of new environments. Working with web browsers and allowing real-time interaction, 3D graphic libraries have become platforms where design and software can be executed together. The aim of the study is to explore the usage ways of the web environment, and their potentials in 3D visualization. After examination of declarative and imperative classification in 3D graphics, the focus is on the WebGL standard, which is imperative and has recently come to prominence in web application and content production. In this context, discussed approaches towards the usage of interactive 3D web graphics in the representation process of designs are: 1. We examined web graphics libraries that allow real-time interaction and 3D modeling through scripting. 2. We focused on tools that are developed as plug-ins to 3D modeling programs on desktops and provide web integration. 3. We discussed the library-based editors which both offer models in a web browser and allow coding simultaneously. The general framework prepared in the study presents the use of these approaches in modeling, visualization, and presentation in the context of web graphics libraries and discusses their advantages and disadvantages. This discussion is a preliminary study of more comprehensive research that can be prepared for the development of next-generation desktop and web-based modeling approaches in the context of using web browsers as visualization media.

**Keywords**— 3D visualization, 3D graphics, web browsers, real-time interaction

## 3B Görselleştirme Ortamı Olarak Web Tarayıcıları

**Özet**—Yüksek kaliteli 3B imajların elde edilebildiği modelleme ve bilgisayar programlarında karşılaşılan işlem zorluğu, yeni platformların geliştirilmesini gerekli kılan önemli bir etkidir. Web tarayıcıları ile çalışan ve gerçek zamanlı etkileşime izin veren 3B grafik kitaplıkları, tasarım ve yazılımın birlikte yürütülebildiği platformlar haline gelmektedir. Bu çalışmanın amacı, web ortamının kullanım şekillerini ve 3B görselleştirmedeki potansiyellerini keşfetmektir. 3B grafiklerde bildirimsel ve zorunlu sınıflandırmanın incelenmesinden sonra, web uygulaması ve içerik üretiminde zorunlu olan ve son zamanlarda ön plana çıkan WebGL standardına odaklanılmaktadır. Bu bağlamda, tasarımların temsil sürecinde etkileşimli 3B web grafiklerinin kullanımına yönelik tartışılan yaklaşımlar şunlardır: 1. Gerçek zamanlı etkileşime ve komut dosyası aracılığıyla 3B modellemeye izin veren web grafiği kitaplıkları. 2. Web tarayıcılarından yararlanmanın ikinci yolu olarak, masaüstü bilgisayarlarda 3D modelleme programlarına eklenti olarak geliştirilen ve web entegrasyonu sağlayan araçlar. 3. Hem bir web tarayıcısında modeller sunan hem de eş zamanlı kodlamaya izin veren kütüphane tabanlı editörler incelenmiştir. Çalışmada hazırlanan genel çerçeve, bu yaklaşımların modelleme ve görselleştirme odaklı kullanımını web grafik kütüphaneleri bağlamında sunarak; avantaj ve dezavantajlarını ele almaktadır.

**Anahtar Kelimeler**— 3B görselleştirme, 3B grafikler, web tarayıcıları, gerçek zamanlı etkileşim

### 1. INTRODUCTION

The change in the graphic language in the web environment over time is one of the most important

factors that enable the developments in internet technology to be followed. With the increasing complexity of programs and applications, the forms of representation in digital media also change. With the developments in computer technologies, the use of digital

media in the visualization process of a design is also supported by different creation, representation, and analysis methods [1]. Visual works that require constant software and hardware updates, pave the way for innovations in the context of web-based graphics. Almost all modern versions of browsers provide access to special graphic contents without plug-ins, and this results in the expansion of usage areas of 3D graphics constantly [2]. With the developments in the graphics technologies of computers, it has become easier to obtain high-quality products in 3D visualizations in a short time. Unlike the display of web content, there are different potentials in interacting with 3D graphics such as applying transform and translate functions to the 3D model or providing detailed renderings from different angles simultaneously.

This study discusses the use of the Web environment in a visualization context through interactive 3D graphics. In order to benefit more from the expanding usage network of the internet environment, many content and studies are integrating with the Web. Representing a common point of developers, designers, and users in different profiles, the Web has the potential to play an important role in the creation, editing, and representation of 3D models. In the second part, we examined the studies using the Web environment for 3D visualization. The third part focused on Web-based 3D graphics as declarative and imperative. In the fourth part, we researched the ways of integrating the Web into the visualization process and made a classification of these integration approaches. These ways discussed include the use of the Web environment in 3D visualization both as a design tool and as a presentation medium. Each has different advantages and disadvantages, but these approaches using interactive 3D graphics contribute to the modeling and visualization process through Web-based content and applications.

## 2. 3D VISUALIZATION IN WEB ENVIRONMENT

The 3D visualization within the scope of the study refers to the modeling process for the structure or object in an environment where features such as light, shadow, and material are affected within a scene element. These wholistic visual representation opportunities pave the way for to be created virtual models for different purposes. In this context, the possibilities and qualities offered by the scene set in the virtual environment also differ according to the specifics of the disciplines where they are used. One of the areas that are familiar with processes such as spatial navigation and object-based transforms that exist in 3D modeling programs is architecture. The use of the web environment directly in the preparation and presentation of spatial designs has advantages such as real-time interaction, fast animating, or availability from different platforms.

The development of the Web also affects the game industry which continues to grow very rapidly. The fact that there are different parameters in the game besides the 3D visualization causes the Web to compete with different platforms in this area [3]. However, good results

can be obtained in the current situation in space-oriented games that require a fast display. Web-based game engines, specially developed with Javascript infrastructure, provided a suitable environment for achieving performance, ease of development, and abstraction goals [4]. Open-source geospatial maps and settlement data in Web can be integrated into multi-platform user interfaces in games [5] and 3D renders of objects in the game are ready for user interaction in the Web environment area [3,6]. The accessibility of the necessary hardware has led to an increase in AR / VR integrated Web content [7]. 3D models of existing structures or newly designed structures that can be integrated with WebVR become more accessible with VR cardboards easily.

With the development of the technological possibilities required in the digitalization of 3D buildings and objects, the field of virtual heritage has also developed significantly [2]. As a virtual heritage archive, the Web is providing an interactive representation environment for architectural cultural heritage to reach a large number of users. In the spatial visualization context, one of the most used areas of the Web environment is 3D virtual tours. Virtual tours that enable photorealistic 3D models to be accessed via Web browsers without the need to download any add-ons or programs, can also develop compatibility with AR / VR environments. The development of 3D Web libraries such as Cesium JS, WebGL Earth JS, wrld3d has also made the use of 3D maps widespread. The ability to use dynamic 3D visualization of geospatial data paved the way for creating 3D and 4D virtual city tours with real data [8]. The 3D virtual tour, which was created for the University of Mauritius to be published on the web, was prepared with a 3D modeling program in the background, and the WebGL framework was used for representation and visualization [9]. An important part of the virtual tours that allow real-time rendering to be experienced by the user is interior spaces and objects. With its interactive use in the Web environment, it is possible to display the artworks in the museum as high-quality 3D models and to filter them according to different criteria [10,11]. While the 3D graphics are displayed, the ability to simultaneously display the text or additional data of this model is a connection that must be provided in terms of the strength of the representation environment [11]. In the visualization process, benefiting from specialized services using Javascript frameworks developed for the Web positively affects the performance of the 3D model [12]. In this context, the ability of the users to interact with the virtual place or object without the need to download any program or file is supported by frameworks developed based on the Web.

## 3. WEB-BASED 3D GRAPHICS

The Web, which offers advanced content over pictures, audio, and video, is a wide-ranging access platform for sharing information in digital media. Evolving from a text-based system to an enhanced and interactive environment, the Web contains new application and usage

potentials with the experience opportunities it offers to users [13]. Since traditional 2D graphics do not meet the needs of the web environment [14], Graphics Processing Unit (GPU) has been developed for the integration of 3D visuals. As a class of hardware, GPUs reduced the time spent on rendering 3D graphics and enabled the use of real-time, photorealistic visualization techniques [2]. WebGL, in particular, has been an innovative step in making plug-in-free hardware and interactive 3D graphics popular in web browsers [13]. It accelerates the imaging process of 3D graphics by enabling web browsers to access the GPU [15]. The increasing integration of digital content in 3D over time requires the development of interactive 3D content that can provide high performance even on mobile devices [16].

The web integration process contains various problems related to the visual quality, digital opportunities of the related platform, or performance, and these parameters require different solutions. These solutions developed for the representation of advanced 3D graphics in the web environment were evaluated in two main groups [13]. Evans et al. explain the prominent features in the declarative and imperative approaches as in Figure 1 [2]. When the two approaches, declarative and imperative are considered within the scope of programming, the most important differences are the focal points in computer interaction. The imperative approach is mainly concerned with the question of how. Declarative, on the other hand, is based on programming to answer the question of what, in other words, it gives direct orders. For 3D graphics, it is difficult to make such a sharp distinction, because the number of techniques to create 3D graphics in web browsers is high, and they can benefit from different features. Even if a 3D graphic framework is under a class, there may be sub-frameworks with features that also benefit from the other approach [2].

3D Graphics	
x3dom	<u>Declarative</u> Scene graph Part of HTML Document DOM Integration CSS/Events
xml3d	
WebGL	<u>Imperative</u> Procedural API Drawing Context Flexible

Figure 1. Declarative and imperative approaches of 3D graphics in web environment (adapted from [2]).

3D content based on scene graphics creates a declarative solution [10]. X3DOM [13] and XML3D [16] are two different solutions developed declaratively. In the imperative approach that includes WebGL, the program state is changed using a series of statements [10]. The difference between these two techniques that allow the creation, modification, sharing, and interactive experience of 3D graphics on the web is their basic approach and target users [17].

### 3.1. Declarative Approach

As seen in Figure 1, it is possible to define a 3D graphic more declaratively than imperative in the use of scene graphics, DOM integration, and being a part of an HTML document. In the Declarative approach, access to 3D objects is provided by HTML DOM. An inclusive declarative approach that integrates existing Web technologies developed for different purposes such as JSON and XML in stage construction, DOM for manipulations, and CSS for visual design is important in terms of the potential use of interactive 3D graphics [13]. The declarative approach is defined only the function to be performed.

The X3D (Extensible 3D) standard, which is an important step for 3D imaging in the web environment, uses XML infrastructure and enables hierarchical modeling [18]. However, to work with X3D, a plug-in is required for Web browsers. In this sense, X3DOM has been developed that can work without a plug-in and as part of an HTML document.

X3DOM is an open-source Javascript framework, based on WebGL, developed for declarative 3D graphics in HTML5 [13], and has a wide usage support by Web browsers. It follows the method of placing the graphics in the DOM inside the web page and displays 3D content without the need for plugins [16]. It is based on a standard and XML-based file format for 3D computer graphics (X3D) [19]. Complex models and scenes can be easily integrated into the web environment with X3DOM.

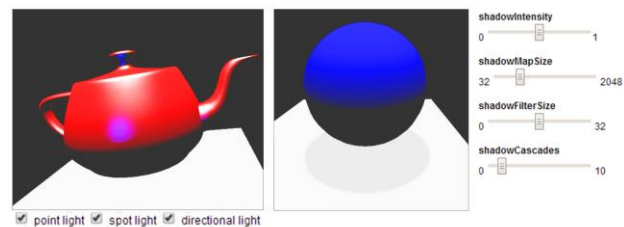


Figure 2. Lighting and shadow effect in X3DOM [20].

Another declarative item, XML3D takes a declarative approach by using different platforms in the representation of 3D data and leverages existing web technologies such as Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), the Document Object Model (DOM) and Asynchronous JavaScript And XML (AJAX) [16]. Render quality is at the forefront in XML3D, therefore instead of creating standart libraries with only available material options, it offers users the option to create custom materials [16]. Creating a lightweight scene with meshes, shaders, light sources, and viewpoints through a hierarchical way is possible with XML3D [21]. It provides a unique abstraction to algorithms to create a general 3D scene beyond the web context and is one of the important steps taken in Web-based 3D graphics.

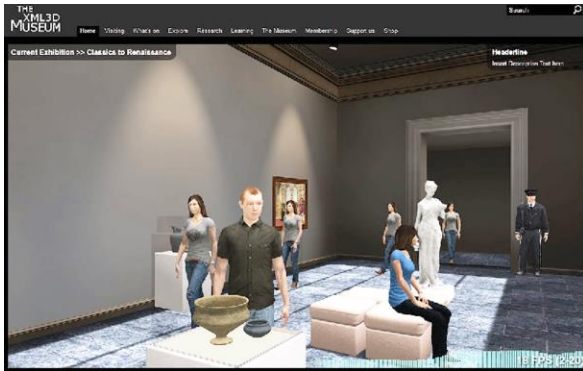


Figure 3. A virtual museum scene in XML3D [21].

### 3.2. Imperative Approach

In the imperative approach, scripting languages are used. The things to be done during the coding process should be included in the step-by-step coding. It covers an explanatory sequence of actions. WebGL (Web Graphics Library) offers a new standard for interactive 3D graphics. It was launched in 2011 and can be controlled entirely by Web browsers. WebGL uses the HTML5 <canvas> element and the DOM interface and makes it possible to draw 3D graphics on the Canvas element of HTML5 without requiring any plug-in to be installed in a compatible web browser [22]. Displaying 3D graphics with various plug-ins such as Flash, or Silverlight [19] has now been replaced by HTML5 and WebGL [23]. HTML5 is a new set of standards formed by the combination of HTML, CSS, and Javascript [24]. WebGL is a Javascript library that enables interactive 2D and 3D renderings. Through WebGL, applications that used 3D graphics have become available with normal HTML elements and a low-level DOM interface. In addition, high-end computer games can be developed with WebGL, as well as 3D web applications [25]. It is possible to use most functions effectively with the development of open source libraries in WebGL, which has a complex structure since there are many sub-libraries in its content. The handicap of this situation, which contributes positively to increase the quality of the created product and reducing the background load, is that the technical information required is too much [10].



Figure 4. Various navigation tools in 3D virtual tours example [26].

Representing a very large standard in terms of its general structure, WebGL can be used in the development of various Web applications or to generate content that can be presented via Web browsers. Since it contains so many

features, the design process in the content to be generated depends entirely on the experience of the developer. However, with Javascript libraries such as Babylon.js and Three.js, 3D graphics can be abstracted from this wide WebGL environment and applications can be developed quickly and easily. The use of WebGL in 3D design visualization has an important potential in terms of both the development of creative tools and collaborative work. Beyond being a presentation environment, the Web, which now acts as a design environment, can offer almost all of the functions that designers use in desktop modeling programs. Recently developed 3D Web applications have similar qualities both in terms of interaction and as blueprint techniques in their interfaces. In this sense, Web interfaces that can appeal to the habits of designers coming from modeling programs make it easier as well as developers to use these Web platforms.

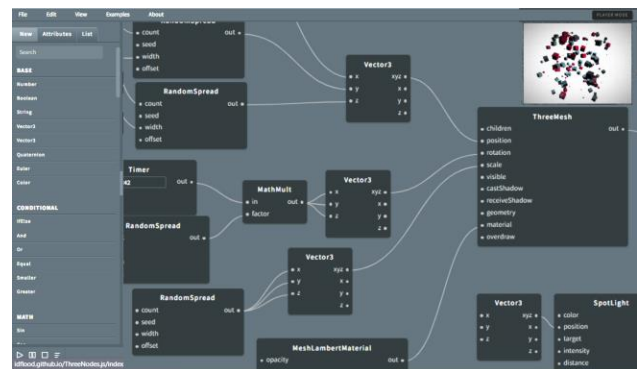


Figure 5. Web application sample that allows designing with Blueprint [27].

## 4. INTEGRATION OF 3D GRAPHIC LIBRARIES

Each of the interactive 3D graphics has its pros and cons. However, in the end, they all provide visual communication at the point of meeting the Web environment with the user. Software changes over time require updating these 3D graphics and even disable some sub-frameworks. In addition, the programs that were initially developed as desktop applications are made Web-integrated, and now 3D models can be exported in formats suitable for the Web with various plug-ins in most modeling programs. 3D model and 3D graphics integration can be achieved with many different methods, but the most decisive point here is the scripting aspect of this integration. Although coding has become quite common with the expansion of interdisciplinary working areas, it is not one of the specialties of designers. This is a barrier to the use of the potential of the Web in terms of 3D modeling and visualization. In this context, the ways of using interactive 3D graphics are gathered under three main headings as scripting-based, plugin-based, and editor-based in the study.

Today, there are different methods followed in Web integration as application and content. The methods discussed in this study are for 3D visualization, to show a 3D scene interactively in the Web environment. Frameworks handled in declarative and imperative

approaches offer a suitable infrastructure for the interactive representation of 3D graphics in the Web environment. However, their distinctive features offer different solutions for how these frameworks can be used in a 3D visualization process. WebGL, which overshadows the declarative approaches, has reached a very wide use in a short time. The point that highlights WebGL within the scope of the study is that it offers structures that can work as plug-ins especially in 3D modeling programs and that it can provide a hybrid modeling environment with editors working directly in Web browsers.

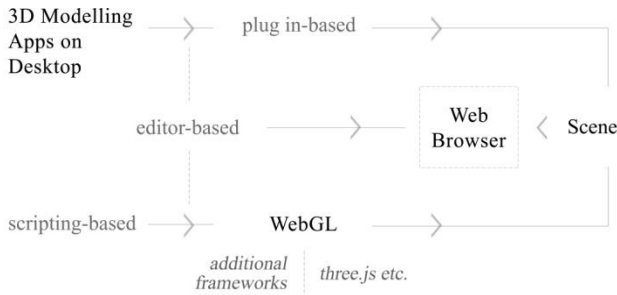


Figure 6. Scripting- based, editor- based and plugin-based 3D scene integration to the Web browser.

#### 4.1. Scripting-based Integration

Web integration in 3D graphics is based on scripting. Web content with interactive 3D graphics can use editors such as VSCode or an IDE (Integrated Development Environment) such as Visual Studio. In the creation of content using interactive 3D graphics by scripting, debug and compilation may be required in rendered graphics. Technologies developed in this sense handle this processing differently. One of the most important differences of the WebGL environment is that rendered graphics do not require compiling before view and debugging. When comparing X3DOM and JS libraries works on 3D, X3DOM, which provides fast and good results in the integration of the model to the HTML page, can be preferred for publishing in the web environment. On the other hand, JS frameworks like Three.js that can directly manipulate the 3D scene are a better option for viewing more complex applications [28]. To be able to draw on the HTML5 canvas using these libraries, generally, a Javascript code must be used and they are not dependent on a 3D standard such as X3DOM. To create a 3D scene, the scene, camera, the renderer must be defined, so it requires more code [28]. WebGL frameworks such as Three.js or Babylon.js can be used with HTML5 in a development environment or JavaScript editors. They can load the model files at once [29]. Three.js is a Javascript library that allows 3D design on web browsers by benefit from the content of WebGL developed continuously. Using this library, 3D models, animations, web applications, games, and many visual applications can be developed easily by writing code at a minimum level [25]. With the contributions of people working in different disciplines, open-source frameworks,

such as Babylon.js, Three.js libraries, continue to be developed.

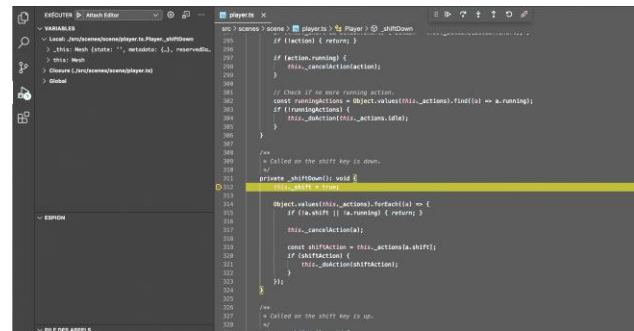


Figure 7. Scripting in VSCode, Babylon.js [30].

#### 4.2. Plugin-based Integration

Accessing popular 3D modeling environments such as Sketchup and Blender via plugins, WebGL is an efficient way for designers who want to use the Web environment as a presentation framework. These plugins, which both enable the use of modeling tools that designers are used to and enable the use of the Web environment through plug-ins. On the other hand, specialized export formats can be used through the application to produce web-integrated content as X3DOM and XML3D. Plugins have been developed to integrate XML3D with various modeling programs, but most of them do not have up-to-date versions as more modern approaches are available [31].

Sub-frameworks and mid-level JavaScript libraries that facilitate work in the world of WebGL, which has a huge infrastructure in terms of content, also establish a bridge between WebGL and various popular 3D applications [17]. InstantReality framework [32], which provides AR/VR integration in accordance with different industry standards such as X3D, OpenGL 2.0, Java, offers a compatible integration with the Web environment.



Figure 8. Siena Cathedral visualization with X3DOM [32].

#### 4.3. Web editor-based Integration

The use of modeling programs that allow 3D design in the digital environment is mostly in the form of desktop applications. The most used tools by designers are shown on the interface with icons in the form of shortcuts. It

allows modeling in a 3d virtual environment on x, y, and z axes. In these applications, modeling can be done without the need for any scripting. Settings such as light, texture, material, shadow can be adjusted. Some programs such as Sketchup and Autodesk Fusion 360 have also developed editors for the Web environment that allows these programs to be used without downloading so that they can be accessed from any computer. However, these programs do not have a scripting option that the user can use to add the tools they want to the interface. These Web-based editors, including tools for which scripting is still possible, provide a drag-and-drop interface with pre-defined tools for user interaction with scenes and objects for visual programming [17].

The Web editor-based platforms discussed in the study allow 3D models to be produced directly using a modeling program interface, and there is a scripting option based on the additional frameworks they are developed on the Web. The X3DOM framework aims to meet the existing HTML5 integration of 3D content and develops X3D elements as part of the HTML5 DOM. It allows AR/VR integration over the Web, but it does not have a specialized network as WebGL, so its possibilities in terms of Web editor are limited.

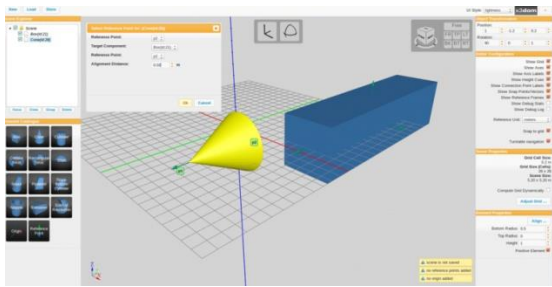


Figure 9. X3DOM-based Web editor [33].

WebGL, which offers the most advanced environment of modern Web technology, has a large number of Web content developed by both individual developers and companies with a large user base. Not only the pages produced with WebGL but also the editors that offer to model, editing, and rendering in the browser have very advanced interfaces. WebGL-based frameworks developed over JS, such as Three.js, Babylon.js, allow you to experience the modeling tools they offer directly on a Web editor. Frameworks such as A-Frame and Playcanvas also function as game engines, allowing the creation of AR / VR environments through a web browser. Harp.gl, which enables the production of 3D maps on an urban scale in terms of graphic production, or WebXR editors that enable the integration of 3D scenes into the virtual environment are more specialized WebGL-based frameworks.

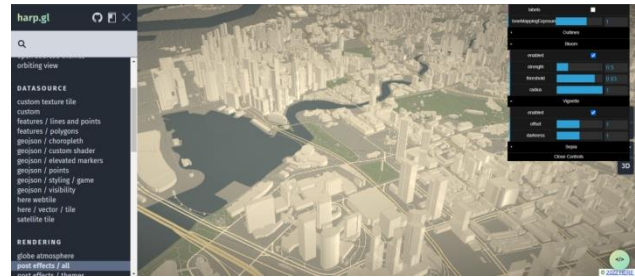


Figure 10. The Web-based editors of Harp.gl [34]

Desktop applications of web-based editors, on the other hand, have all the visual qualities of a modern 3D modeling program. It enables the production of visuals and animations suitable for the Web environment directly with its export options. Blueprint and scripting-based editors are included in the program. For an instance, the editor of Babylon.js is designed to work with VSCode and it can remotely debug scripts of objects in the scene easily. In addition to visualization in a highly advanced framework such as Three.js, it is also possible to use more specialized sub-libraries and frameworks for adding physics, particle systems, kinematic, or textual formats.

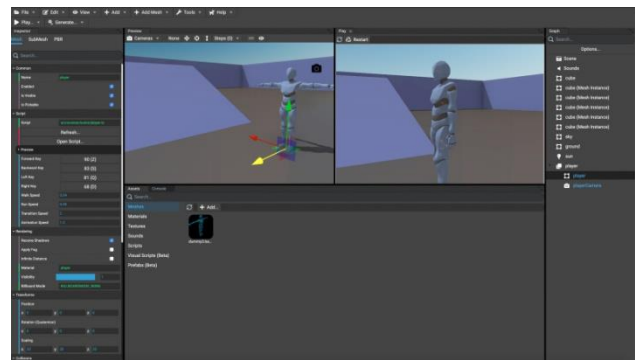


Figure 11. Desktop application of Babylon.js [30].

## 5. DISCUSSION AND CONCLUSION

Within the scope of the study, we focused on the question of how we can integrate Web-based 3D graphics into the 3D visualization process as a designer. After examining the interactive 3D graphics system as declarative and imperative, we prepared a general framework for the methods followed in integrating these graphics with the Web. Web content, which has become a huge part of daily life, is still a field dominated by developers and programmers in terms of visual design. Especially in fields such as architecture and industrial design that are open to collaboration and interdisciplinary work, benefiting from the Web environment has important potential. In the light of the information compiled as a result of the research in the study, the prominent points are:

Table 1. Declarative and imperative approaches in terms of integration methods of 3D graphics libraries.

	Declarative		Imperative
	X3DOM	XML3D	WebGL
Scripting-based	<ul style="list-style-type: none"> <li>▪ A framework of HTML5, based on X3D standard and XML-based file format</li> <li>▪ Adding a JavaScript file and codes to the HTML DOM</li> <li>▪ Composing scene from the nodes</li> <li>▪ CSS file via the script, link tags</li> <li>▪ Need an &lt;x3d&gt; element, which describes the X3DOM context in which the scene will be displayed</li> <li>▪ The whole 3D content of the X3DOM context is described as a scene</li> <li>▪ Fast and good results in the integration of the model to the HTML page</li> </ul>	<ul style="list-style-type: none"> <li>▪ An extension to HTML5, integration with CSS and DOM libraries</li> <li>▪ Composing scene from the nodes</li> <li>▪ Defined inside the &lt;xml3d&gt; element</li> <li>▪ Composing scene from nodes</li> <li>▪ To add some 3D geometry to our scene, it uses the &lt;mesh&gt; element.</li> <li>▪ CSS file via the script and link tags</li> <li>▪ Externally defined assets: External JSON or XML file just as also done by a &lt;mesh&gt; node</li> </ul>	<ul style="list-style-type: none"> <li>▪ 3D API, JS library designed to simplify 3D rendering in the browser using WebGL</li> <li>▪ JavaScript code is drawn in an HTML5 canvas, therefore more code is required to create a 3D scene</li> <li>▪ Renders scenes to a single &lt;canvas&gt; DOM element that contains the entire visualization on the page</li> <li>▪ Plugin-free rendering in Web browsers; cross-browser and cross-platform compatibility</li> <li>▪ Better option for viewing more complex applications</li> </ul>
Plugin-based	<ul style="list-style-type: none"> <li>▪ Modelling-based: Exporting 3D model as X3D</li> <li>▪ Browser-based: The current implementation of the X3DOM fallback model needs an InstantReality plugin, a Flash11 plugin or a WebGL-enabled browser.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Modelling-based: Small-scale, individually developed plugins for 3D exporting compatible with XML3D</li> <li>▪ Browser-based: Firebug plugin which has tools for HTML, CSS, DOM and JavaScriptfor, in Firefox browsers</li> </ul>	<ul style="list-style-type: none"> <li>▪ WebGL compatible export options supported by common modeling programs and game engines, and plugins developed for specialized frameworks</li> <li>▪ No need for plugins for Web-browsers</li> </ul>
Editor-based	<ul style="list-style-type: none"> <li>▪ Various Web-based editor trials that are close to modern interfaces but need updating [https://examples.x3dom.org/editor/component_editor/]</li> </ul>	<ul style="list-style-type: none"> <li>▪ Does not have a direct editor-based developed application.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Editors with a wide framework network, modern interfaces, cross-platform support, based on customized solutions: Three.js, Babylon.js, Playcanvas, Harp.gl, A-Frame...</li> </ul>

- Considering the Web environment in a design process has a very strong potential in terms of the opportunities it offers to the designer. Throughout the designing, editing, and rendering process in the standard environment offered by desktop-based 3D modeling programs, the designer uses the existing tools of the program. By adding scripts to these programs, it is possible to develop the tools in their content, but the related languages allowed by each program are different. A standard scripting language is not used. In this context, the WebGL environment offers designers a vast framework of standards in itself to create their own 3D modeling world.
- Architecture is a field that appeals to different human profiles from designer to user. It is very important that access to 3D graphics and displays is fast and high quality, and that they can interact with the computer or mobile devices. Therefore, Web browsers, which can be accessed

on desktop computers, tablets, or smartphones, are a very good option. Providing AR / VR integration at the same time, especially in the presentation of interior architecture applications on web-based platforms, provides a much more effective presentation than 2D rendering images. In this way, 3D models can be used directly in the web environment via the link without downloading different model programs or plugins.

- In today's technology, where interdisciplinary communication and interaction have become increasingly important, the web environment provides the infrastructure where design can meet with different disciplines. Unlike the back-end software that is more dominated by developers, the Web, which has the opportunity to work with a visual result, can be easily used in the design and presentation of 3D models with Javascript, WebGL through script-based, editor-based, or plugin-based approaches. In addition, it

is possible to add new tools that support direct interaction with the model to the Web interface with scripting.

- Another point that highlights the web environment is that it offers a presentation environment that can be updated at any time through its accessibility. It may be difficult to make changes in models prepared in 3D modeling programs according to the formats in which the model is prepared or exported. However, the fact that 3D graphics are available for common use in modeling carried out with frameworks in the web environment allows a sub-framework prepared by a developer to be integrated into other works. In this sense, we can think of the Web as a multi-purpose, sharing environment. This solution may include not only creating a 3D model or rendered image but developing a design tool that can be used directly for a specific purpose in the modeling process.
- One of the most difficult aspects of coding is to make sense of the relationship between scripting and its result. Therefore, observing the result of scripting on 3D graphics in addition to modeling with the interface, especially in Web editors, is an easier way than directly creating content with scripting.

This study investigates the Web environment in the context of 3D visualization, especially in terms of designers, unlike technical studies for 3D graphics in the Web environment. By expanding the scope of this work in the future, ways of integration can be tested directly via Web tools. In addition, different case studies can be made on which tools or which methods can be used more easily by designers in the classification of 3D graphics for Web integration.

## REFERENCES

- [1] A. Koutamanis, "Digital architectural visualization", *Automation in construction*, 9(4), 347-360, 2000.
- [2] A. Evans, M. Romeo, A. Bahremand, J. Agenjo, J. Blat, "3D graphics on the web: A survey", *Computers & Graphics*, 41, 43-61, 2014.
- [3] K. Curran, C. George, "The future of web and mobile game development", *International Journal of Cloud Computing and Services Science*, 1(1), 25, 2012.
- [4] M. Nobel-Jørgensen, **WebGL based 3D Game Engine**, Master's Thesis, IT University of Copenhagen, 2012.
- [5] N. Polys, J. Hotter, M. Lanier, L. Purcell, J. Wolf, W. C. Hession, ... , J. D. Ivory, "Finding frogs: Using game-based learning to increase environmental awareness", **In Proceedings of the 22nd International Conference on 3D Web Technology**, 1-8, 2017.
- [6] R. Van der Spuy, **Advanced Game Design with HTML5 and JavaScript**. Apress, 2015.
- [7] S. Neelakantam, T. Pant, **Learning Web-based Virtual Reality: Build and Deploy Web-based Virtual Reality Technology**, Apress, 2017.
- [8] M. O. Mete, D. Guler, T. Yomralioglu, "Development of 3D web GIS application with open source library", *Selçuk Üniversitesi Mühendislik, Bilim Ve Teknoloji Dergisi*, 6, 818-824, 2018.
- [9] R. K. Moloo, S. Pudaruth, M. Ramodhin, R. B. Rozbully, "A 3D Virtual Tour of the University of Mauritius using WebGL", **In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)**, 2891-2894, 2016.
- [10] M. Potenziani, M. Corsini, M. Callieri, M. Di Benedetto, F. Ponchio, M. Dellepiane, R. Scopigno, "An advanced solution for publishing 3D content on the web", **In Museums and the Web**, 2013.
- [11] M. Callieri, C. Leoni, M. Dellepiane, R. Scopigno, "Artworks narrating a story: a modular framework for the integrated presentation of three-dimensional and textual contents", **In Proceedings of the 18th International Conference on 3D Web Technology**, 167-175, 2013.
- [12] T. Alatalo, T. Koskela, M. Pouke, P. Alavesä, T. Ojala, "VirtualOulu: collaborative, immersive and extensible 3D city model on the web", **In Proceedings of the 21st International Conference on Web3D Technology**, 95-103, 2016.
- [13] J. Jankowski, S. Ressler, K. Sons, Y. Jung, J. Behr, P. Slusallek, "Declarative integration of interactive 3d graphics into the world-wide web: Principles, current approaches, and research agenda", **In Proceedings of the 18th International Conference on 3D Web Technology**, 39-45, 2013.
- [14] X. Wei, W. Sun, X. Wan, "Architecture and Implementation of 3D Engine Based on WebGL", *Applied Mathematics*, 7(07), 701, 2016.
- [15] Internet: [https://www.tutorialspoint.com/webgl/webgl\\_tutorial.pdf](https://www.tutorialspoint.com/webgl/webgl_tutorial.pdf). Retrieved Date: 20.05.2022.
- [16] K. Sons, F. Klein, D. Rubinstein, S. Byelozyorov, P. Slusallek, "XML3D: interactive 3D graphics for the web", **In Proceedings of the 15th International Conference on Web 3D Technology**, 175-184, 2010.
- [17] M. Potenziani, M. Callieri, M. Dellepiane, R. Scopigno, **Publishing and consuming 3D content on the web: A survey**, Now Publishers, 2018.
- [18] T. E. Kalaycı, Y. D. Uğur, "X3D ile İnternet Üzerinde Üç Boyut", *Akademik Bilişim*, 02-04, 2005.
- [19] J. Behr, P. Eschler, Y. Jung, M. Zöllner, "X3DOM: a DOM-based HTML5/X3D integration model", **In Proceedings of the 14th international conference on 3D web technology**, 127-135, 2009.
- [20] Internet: <https://doc.x3dom.org/tutorials/index.html>. Retrieved Date: 20.05.2022.
- [21] F. Klein, K. Sons, D. Rubinstein, P. Slusallek, "XML3D and Xflow: Combining declarative 3D for the Web with generic data flows", *IEEE computer graphics and applications*, 33(5), 38-47, 2013.
- [22] T. Parisi, **WebGL: up and running**. O'Reilly Media, Inc., 2012.



- [23] L. Herman, T. Reznik, "3D web visualization of environmental information-integration of heterogeneous data sources when providing navigation and interaction", *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3), 479, 2015.
- [24] P. Lubbers, B. Albers, F. Salim, T. Pye, **Pro HTML5 Programming**, New York, NY, USA. Apress, 2011.
- [25] M. Karlı, **Yapay Zekânın Tasarımcıyla İşbirliği Ve Tasarım Olan Ekisi**, Master's Thesis, Istanbul Technical University, 2019.
- [26] Internet: <https://www.middlesex.mu/about-mdx-mauritius/campus-virtual-tour>. Retrieved Date: 22.05.2022.
- [27] Internet: [http://idflood.github.io/ThreeNodes.js/index\\_optimized.html#example/spreads1.json](http://idflood.github.io/ThreeNodes.js/index_optimized.html#example/spreads1.json). Retrieved Date: 22.05.2022.
- [28] Internet: <https://viscircle.de/x3dom-vs-threejs-why-the-concrete-application-case-decides/?lang=en>. Retrieved Date: 23.05.2022.
- [29] L. Li, X. Qiao, Q. Lu, P. Ren, R. Lin, "Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading", *IEEE Access*, 8, 88474-88486, 2020.
- [30] Internet: <http://editor.babylonjs.com/>. Retrieved Date: 23.05.2022.
- [31] Internet: <https://github.com/ksons/xml3d-blender-exporter>. Retrieved Date: 23.05.2022.
- [32] Internet: <https://www.x3dom.org/showcase-siena-cathedral-in-x3dom/>. Retrieved Date: 20.05.2022.
- [33] Internet: [https://examples.x3dom.org/editor/component\\_editor/](https://examples.x3dom.org/editor/component_editor/). Retrieved Date: 20.05.2022.
- [34] Internet: <https://www.harp.gl/docs/master/example/>. Retrieved Date: 20.05.2022.