Prospects of WebGL Technology Using in Modern Educational Process[*]

Yulia Zhuk
zhuk_yua@mail.ru

Evgenii Tetiuev
tetyuev.ev@gmail.com

Saint-Petersburg State Forestry University,
St. Petersburg, Russian Federation

Abstract

Sometimes we need to look at an object from a different angle to see its depth and complexity. This can be achieved with WebGL - a browser-based technology for visualization of 3D objects and scenes. Materials created with WebGL can be used as visual aids in a digital learning environment. This article is about the technology itself, its advantages and disadvantages, usage examples and opportunities of using it in the future.

Keywords: *digital learning environment,computer-aided design systems, modern educational process, 3D, WebGL, three.js*

1 Introduction

1.1 Technologies in modern life

It is difficult to imagine a modern society without the technologies that surround us. Having satisfied their basic needs, people showed a desire to communicate and receive information, which resulted in the dynamics of modern life itself, as well as the development of information and communication technologies. It is not surprising that the digital age has created a trend towards digitalization of the modern educational process [Omarova, 2018].

For a long time, the learning process was based on lectures and practical classes, and people used textbooks and notes, as well as visual materials and models to store the information. As our technology evolved the training materials were supplemented with photographs and educational films. The next important step was the invention of computers and the digitization of information. The following creation of the Internet and the digitization of printed publications took accessibility of information to a whole new level. Nowadays a teacher giving a lecture based on a presentation is just a familiar sight, and the presence of a projector in the audience is almost mandatory. Today, not only special materials and programs are created for the educational process, but even entire operating systems are created for educational purposes. The digital learning environment will be supplemented by new technologies, but this article is focused on a technology that has been around for almost 10 years, but remains little known.

1.2 Hidden features of the web browsers

We often use a program or tool without even knowing what features are hidden inside. Take an ordinary web browser, a program that people use every day. Most people understand that the browser is needed for displaying Internet pages, communicating and sharing files. Browsers are actively used in the educational process, mainly in distance education. We are used to using browsers and have a rough understanding of what they can do.

But there is something that not everyone knows. What happens if you type “do a barrel roll” in the Google search bar? If you didn’t do this before, the author of the article strongly recommends conducting this experiment, no matter if you’re using PC or mobile . But this Easter egg trick is nothing compared to the technology that browser developers have put in their programs. For example, modern browsers can display not only web pages and images, but also three-dimensional graphical objects.

The main advantage of using 3D graphics over images and films is that a 3D object can be rotated and viewed from a different angle. The idea of rendering 3D graphics in the browser is not new at all. In 2008, the author of the article took part in a project to digitize materials for teaching medical students [Skachkova, 2013]. The technology was based on creating 3D model of an object based on photos using the Strata Foto 3D application. Then the model was optimized (and adjusted if necessary) using Sculptris application, and the final step was creation of the Adobe Flash file using the Strata Live 3D. That Flash file could be later viewed using a web browser. At that time, using plug-ins was the only way to render 3D graphics in the browser, but even then this method had some advantages over using specialized programs. However technology is evolving, and in 2020 Google Chrome, the most popular browser[1], will stop supporting Flash technology. The fact is that the HTML[2] language itself has reworked and absorbed many of the plug-in’s features and made them impractical to use. Now, instead of using third-party plug-ins, we can use the built-in technologies of the browser engine. WebGL is one of such technologies.

2 WebGL

2.1 WebGL API[3] and its features

WebGL is an immediate mode 3D rendering API designed for the web. It is derived from OpenGL[4] ES 3.0, and provides similar rendering functionality, but in an HTML context [WebGL 2.0 Specification]. WebGL programs consist of control code written in JavaScript and shader code that is written in OpenGL ES Shading Language. Rephrasing, WebGL allows us to visualize 3D objects in an interactive environment inside of web browser without using any plug-ins. Interactivity is provided by using programming languages that allow us to perform actions on objects and models: load them from a file, create them from geometric primitives, change the scale, position, and rotation, set the texture, color, transparency, and so on. Using programming languages, we can create dependencies and scenarios for controlling the scene, camera, and objects, from simply rotating the camera around a static object to creating a full-fledged simulation of a CNC machine. For more information about what a WebGL-based project is, see one of the following chapters of this article.

[1] Google Chrome is the most popular browser in 2019 according to W3Counter stats [W3C, 2019]

[2] HTML – HyperText Markup Language - is the standard markup language for documents designed to be displayed in a web browser. Basically that’s a web page skeleton for the end-user.

[3] API – Application Programming Interface, a computing interface to a software component or a system, that defines how other components or systems can use it.

[4] OpenGL - Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics

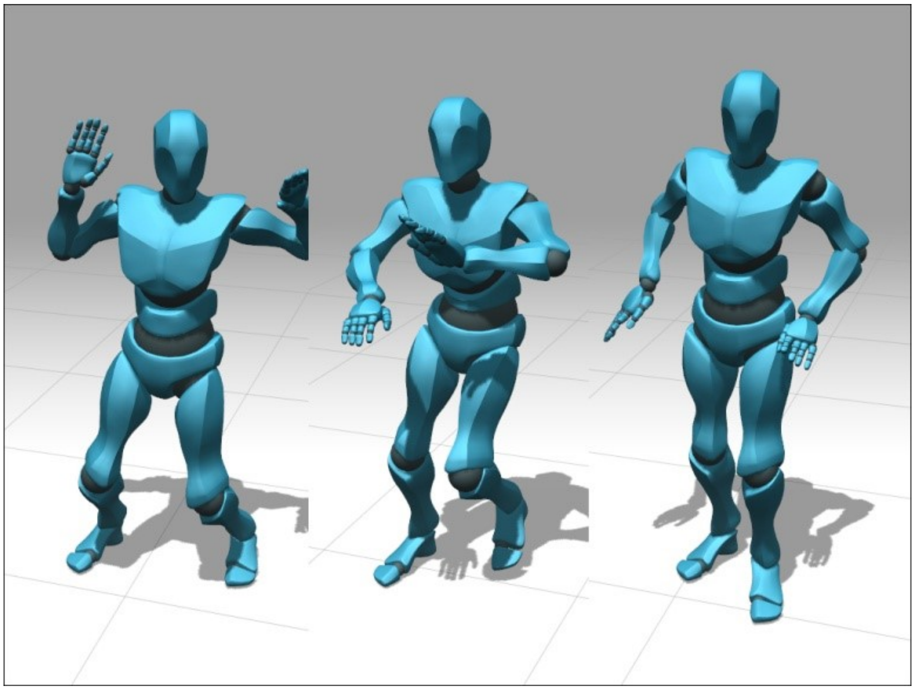


Figure 1: 3D dancing man, three.js official website's example

2.2 WebGL vs 3D graphics software, pros and cons

Strictly speaking, this comparison is not quite correct, since WebGL is a technology, and it doesn't make sense to compare it with programs. Therefore, we will compare working with 3D graphics in the browser using WebGL and using dedicated applications. We will divide programs into the following categories: editors, viewers, and interactive applications. Here we make a reservation: in this article, interactivity refers to the scripted reaction of objects, cameras, or scenes to user actions. We have also specifically moved editors to a separate category, since their interactive capabilities are far superior to those of all other programs. Computer-aided design systems (CAD) should be mentioned separately, but they will not be considered in this article, since their main purpose is development and design, not visualization. However, the categories are very vague, and some programs may fall into several sections at once.

Editors allow us to create objects, scenes, work with textures, effects, and animations, as well as save the scene to a file or render it. Currently, there are several WebGL-based editors, such as Autodesk Tinkercad, Clara.io, Vectary, but their capabilities are not even close to the level of popular programs such as SketchUP, Blender, 3DS Max, Maya and Rhinoceros 3D.

Basic viewers allow us to import a file and view its contents. There are many programs for viewing files in different formats, starting from FBX viewer by Autodesk to Open 3D Model Viewer with open source code. Even Microsoft has created its own program for viewing 3D objects. However, it is worth noting that in recent years, more and more WebGL based online services appear in search results list. Moreover, there are entire Internet libraries of 3D graphics with the ability to download and view files, for example, Sketchfab. Apart from specialized applications (e.g. Pepakura, is a program that splits a 3D model for subsequent printing and gluing), the capabilities of software and web browsers are almost identical.

Interactive applications include computer games, as well as applications for simulating the scene or a process, and simplified editors (from the kitchen builder to the LEGO constructor), and training programs. At the moment, the number of such WebGL-based applications is very small.

2.3 Advantages of WebGL

It would seem that WebGL loses to classic applications in terms of functionality, but this technology has its own advantages. Unlike classic programs there is no need to install a WebGL application. The technology itself is already embedded in the code of modern browsers, which means that the browser itself is enough to use it. Futhermore, the vast majority of operating systems, both desktop and mobile, have at least one pre-installed browser. This leads to the next advantage: WebGL implementation does not depend on the type of operating system or even platform, and graphics are displayed equally on PCs, laptops and smart phones. The third advantage comes from integrating WebGL into the HTML structure and linking with script languages, which provides huge opportunities for creating an interactive environment. Script execution also does not depend on the platform, but in this case, the task of developers is to provide the ability to work with the application using both mouse and touch input. Another advantage is accessibility. After being uploaded to a server, WebGL-based project can be accessed from anywhere in the world via the Internet. Also, using the server allows us to use special permissions [Ryabkov, 2018] to make the source code and models unavailable for download, contributing to the preservation of copyright on intellectual property.

2.4 A spoon of tar

Of course, there are some disadvantages. First of all, it's about performance, which is lower in comparison with classic applications. However, given the specs of modern computers and smart phones, this is not as much a critical disadvantage as a limitation on the complexity of materials and the size of the models used. The next flaw is the browser security policy. In particular, it prohibits downloading many types of files from the local disk [Chrome Policy]. However, this restriction can be circumvented by disabling it in the browser (a bad way, though it is possible in some browsers, e.g. Google Chrome) or using a simple virtual server such as EasyPHP. Also, for simple applications, you can avoid using 3D model files and use geometric primitives instead. However, the best solution is to use a physical server that can ensure full operation, availability, and security of the project. At the same time, this is an additional guarantee of security for the end user, since it is assumed that the files on the server are checked, optimized and cannot harm the system. This leads to the last drawback: using WebGL based applications requires downloading data over the network and, consequently, traffic consumption. However, in the format of an educational institution, this problem can be solved by using your own server and local network.

Summing up, WebGL is perfect for creating small and versatile cross-platform applications that are necessary for viewing 3D models, drawings, or diagrams. Whether it is a simple project with a minimal interface or a full-fledged interactive environment, the technology allows you to implement this task. At the same time, dedicated applications license can cost a lot, meanwhile WebGL technology is free, and all the costs of working with it are limited directly to the cost of developing the projects. As for fully working with 3D graphics, creating scenes and editing models, it is better to use specialized applications.

3 Using WebGL

3.1 Examples of using WebGL

Since its creation by the Khronos Group[5] in 2011 WebGL has been used primarily for making games, visualizing Internet libraries of 3D models, or used in specialized or indie projects. One of the best examples is the online resource SketchFab, which is a platform to publish, share, discover, buy and

[5] The Khronos Group, Inc. is non-profit member-funded industry consortium focused on the creation of open standard, royalty-free APIs for authoring and accelerated playback of dynamic media on a wide variety of platforms and devices

sell 3D and VR content. On the [Sky Live] project, you can look at the position of planets in the Solar system. And of course, it is worth mentioning Google Maps, where WebGL is used to display the spherical surface of the Earth. The technology is gradually developing with version 2.0 was released in 2017, and all the leading browser developers (Google, Mozilla, Apple) are members of Khronos group and include WebGL API in their browsers.

Despite this, it is surprising that the technology is still not widely used, especially in the digital learning environment. There is an extensive list of articles on the Internet on the topic of learning WebGL, but not learning with WebGL.

Meanwhile, many academic disciplines are closely related to each other: biology students study chemistry, technical students often study descriptive geometry, GIS students learn the basics of geodesy, etc. However, sometimes students have to learn a program that they will not use in the future to get the necessary knowledge of the subject. Furthermore, even in creative environment, students often spend more time studying the program than the subject itself, and at the same time use the minimum functionality of the program itself [Angel, 2017]. We are not talking about highly specialized applications that are constantly used in specific profession, but about general education disciplines. The solution to this problem is to get rid of such programs and unify the basement of materials, or create an application with simple and user-friendly interface. Taking into account the advantages mentioned earlier, the browser and WebGL have an excellent foundation for becoming the basis of such applications. Let's take a look at what has already been done.

The united team of Lviv Polytechnic National University and AGH University of Science and Technology in Krakow suggested using WebGL to visualize the volume structure of composite elements [Jaworski, 2016].

There are several examples for using the technology in displaying molecules and chemical compounds, ranging from a relatively small project that uses WebGL to visualize files made in Tinker (a specialized application to create models of molecules) [Vildanov, 2014] to an online platform that is able to display some of the chemicals, crystal lattice, and even biological compounds [Canvasmol]. A small detail here: both applications use geometric primitives, so they do not have restrictions on loading models mentioned before, so they can be run both online and locally.

The MAI team suggests using WebGL in the educational process for training specialists in the field of welding production [Vlasenko, 2016]. It is not surprising, because aerospace technologies are impossible without highly qualified specialists, who use the latest technologies in their training.

There are also articles on using WebGL to visualize mathematical modeling, data analysis, and even solving intelligent tasks [Chuvikov, 2015a]. The examples above show that WebGL is primarily used for visualization and creating interactive environments, which fully corresponds to the purpose of the technology.

We can also demonstrate some of the basic examples of using technology for regular and online courses. The first of them – a tiny virtual model of building a computer - was made as an experiment to learn the basic features of WebGL. The project is extremely simple and allows you to add components to the case by clicking on the appropriate buttons, as well as rotate the camera around the computer case to view it at the desired angle. The second test project is a "cheat sheet" for descriptive geometry classes, which loads a three-dimensional model of an object and its 3 projections, thereby helping students to unlock their potential of spatial thinking, so they could see and understand material better. The key part of this application is the ability to switch to orthographic camera, showing the difference between perspective and a projection. Both applications use Three.js library and have a weight range of 1-2 megabytes (including models and images) and work perfectly on a variety of devices. You can see images of these projects at figure 2. We are currently working on more serious projects.

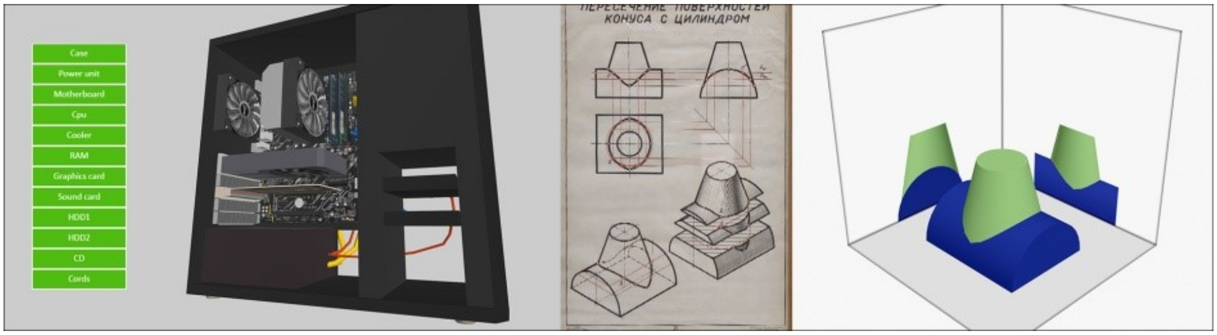


Figure 2: PC builder (left), paper scheme (middle) and 3D projections (right)

3.2 Possibilities of using WebGL in the future

Previously, we mentioned the use of WebGL to display the spatial relationships of molecules or the structure of materials, but at the moment these projects can only show static models. The logical development of these is to expand the list of displayed molecules and materials and to add the possibility to view the visualization of chemical reactions. However, why not take developments in this area to a qualitatively new level? The interactive capabilities of the technology are sufficient to create a virtual chemical laboratory that could be used to train students before accessing practical classes. The main advantage of such a project is safety: unlike real tools and reagents, virtual objects can show the consequences of incorrect actions without any harm to the laboratory and students. Speaking of virtual digital environments, it is worth mentioning the possibility of visualizing any device or mechanism. These can be projects of various complexities, ranging from the simplest model, the functionality of which is reduced to rotating the camera around the object and the appearance of pop-up hints on mouse on, to a full-fledged simulation of the process of, for example, a theodolite or a CNC metalworking machine.

Earlier in the article, we mentioned the project of digitizing materials for medical students. But why stop there? Modern technologies offer new tools for digitizing and creating models of physical objects. By combining them with visualization technology, you can create full-fledged virtual collections for a variety of directions. First of all, we are talking about the museums, which will allow people on the other side of the world to get access to the exhibits. Of course, many museums still offer virtual tours, but they are mostly limited to a set of photos or videos. Some have gone further and offer a virtual tour with a view of 360, and this is a significant progress already. Furthermore, there are examples of creating such scenes using different technologies [Borodkin, 2015], [Lavrov, 2019].

Some museums have already shown their interest in using WebGL, so far in a small amount [British Museum]. But we are talking about using the digitization technology to create accurate models of exhibits, including those that are usually not displayed to the public, and upload them to the Internet. There are examples (figure 3) but they are very few. The ability to digitize objects or entire scenes is a huge prospect for history and archaeology, because visualization can be used not only to display objects and antiquities, but also to draw a historical scene or a diagram of the location of artifacts within the cultural archaeological layer and any other things needed.

Developing the idea, we get to perhaps the most promising direction of using WebGL which is the distance learning. Distance learning programs and tools have been around for a long time; the idea is very tempting and encourages people to look for ways to create a digital learning environment, sometimes even using game engines [Chuvikov, 2015b]. However, today this topic has become more relevant than ever. The year 2020 greeted humanity with a new dangerous virus. Many industries were locked, and lots of educational institutions switched to distance learning. Even though distance learning allows you to conduct practical classes, students do not have the opportunity to visit laboratory classes, where they could view or touch educational materials. In this regard, there is a demand for additional educational materials that would be as simple, yet clear as possible. And WebGL fully meets these requirements. Be it an item, mechanism, machine or scene, all these objects can be recreated with the help of 3D graphics.

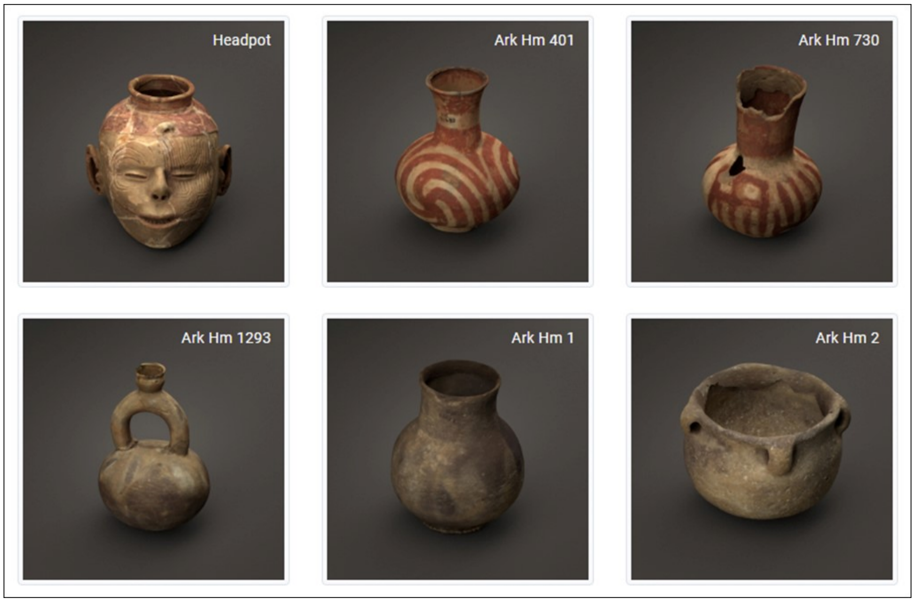


Figure 3: An exhibit at the [Virtual Hampson Museum]

4 Learning WebGL

4.1 Is it hard to learn WebGL?

WebGL runs on the GPU[6] on your computer. As such you need to provide the code that runs on that GPU. You provide that code in the form of pairs of functions. Those 2 functions are called a vertex shader and a fragment shader and they are each written in a very strictly typed C/C++ like language called GLSL. (GL Shader Language).Paired together they are called a program [WebGL Fundamentals].

Thus we will start by creating a function to compile and create the shaders, as well as buffers for data exchange. We will also need the canvas HTML element, where we will output the results of the shaders, and the source data of the objects that we are going to output. In fact, you can read the full text of the code at the link above. Basic WebGL has limited functionality, and extensions will require more code. Of course, it will need to be duplicated for each project, and the logical solution is to allocate the shared code to a separate library, thus no need to rewrite it every time. The next step is to create your own classes and libraries to simplify the actions that are being done all the time.

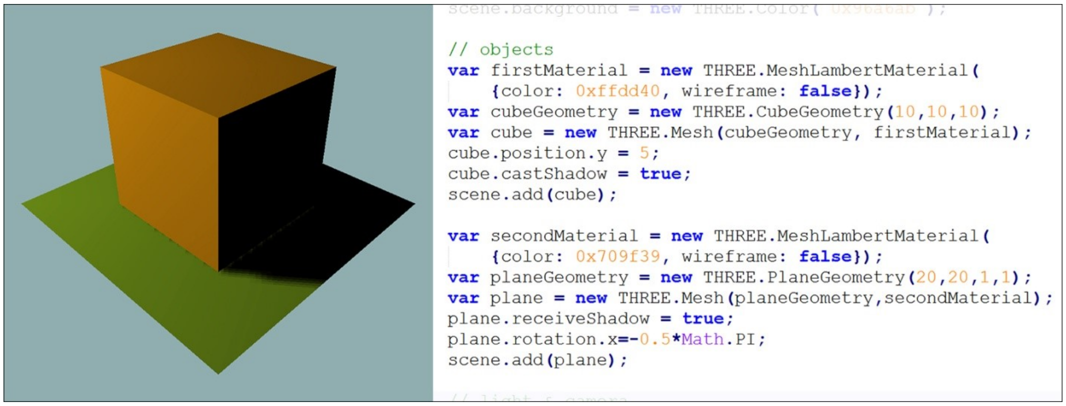
At this moment, studying WebGL looks like something that only an experienced programmer can do. However, this is wrong. Instead of copying large chunks of code every time or writing your own libraries, you can turn to solutions that are already available. There are several projects where the developer community creates libraries that make working with WebGL much easier. The main ones are: BabylonJS, PlayCanvas, OSG.JS, CopperLicht, A-Frame and Three.js. The last one is the most popular, and you can see this just by looking at the number of results in search engines. Also, resources like GitHub and StackOverflow have a huge number of requests related to this library. In

[6] GPU - graphics processing unit, a special stream processor used in computer graphics hardware

fact, the availability of a ready-made database of problems and their solutions also speaks in favor of choosing Three.js. It is highly likely that any question asked by a person working with Three.js has a ready answer somewhere on the Internet.

4.2 Tools for productive interaction with Example of using WebGL with a Three.js library

Here is a little example to get familiar with WebGL and Three.js. The project consists of only 3 files, and 2 of them are standard libraries (Jquery and Three.js core). The third one is an HTML file with 55 lines of text, of which 18 are HTML page markup, and the rest is JavaScript code, a rather short one. Initially, we planned to provide the full code and instructions on how to make your very first project, but then we decided to simplify the example even more for better understanding, so we hid the majority of the code. We hope that if you are interested in this topic, you can find more detailed tutorials on the Internet. Here is our example (figure 4):



```
scene.background = new THREE.Color( 0x96a6ab );

// objects
var firstMaterial = new THREE.MeshLambertMaterial(
    {color: 0xffdd40, wireframe: false});
var cubeGeometry = new THREE.CubeGeometry(10,10,10);
var cube = new THREE.Mesh(cubeGeometry, firstMaterial);
cube.position.y = 5;
cube.castShadow = true;
scene.add(cube);

var secondMaterial = new THREE.MeshLambertMaterial(
    {color: 0x709f39, wireframe: false});
var planeGeometry = new THREE.PlaneGeometry(20,20,1,1);
var plane = new THREE.Mesh(planeGeometry,secondMaterial);
plane.receiveShadow = true;
plane.rotation.x=-0.5*Math.PI;
scene.add(plane);

// light & camera
```

Figure 4: Cube and plane primitives inside the viewer (left), code for adding and configuring these objects in the scene (right)

As you can see on the code above, each object model is made of geometry and material, and its characteristics include a vector position, rotation and interaction with the surrounding scene. In this case, the three.js library performs all the necessary actions to create shaders and transmit information to the graphics core. Jquery is needed for Three.js to operate properly. Thus the developer's role here is to create a scene and fill it with objects. For this we used Three.js standard classes "scene", "cube", "plane", "camera" and some methods. However, this was the simplest example and only basic objects to visualize. More serious projects will require much more software code, and, perhaps, loading 3D models. Complex objects (such as models at the figure 1 or figure 2) will require the use of a loader, but importing them is also not a problem.

Summing up, is it difficult to learn using WebGL? No, it's not. Even if you are not a programmer, the code above does not look so complicated. However, you will need a basic knowledge of JavaScript, HTML and CSS[7], as well as understanding of the basics of 3D. And you will need a source for 3D models to use in your project that are either made by artists or by some sort of digitization. Anyway, every project is a lot of code, thus programmers and IT specialists in general have better chances of learning to use WebGL. However an author of this article used to be a 3D artist before getting into IT, so the other way is also possible. All you need is desire and effort.

[7] CSS - Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in HTML. It is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

Conclusion

While trends towards digitalization of the learning process continue [Altimentova, 2016], new technologies in education are only a matter of time. However, we should not underestimate the tools that are available to us today. The use of WebGL technology can serve to improve the quality of educational materials and programs, making them more understandable, visual and accessible. At the same time, colorfulness and interactivity should increase interest in the subject being studied. In addition, browser graphics have a huge opportunity for use in distance learning and creating virtual environments. Working with it is simple and clear, and creating project isn't that hard as well. Thus, WebGL is one of the most promising technologies for use in a digital learning environment.

References

[Angel, 2017] Angel E.(2017) The case for teaching computer graphics with WebGL: a 25-year perspective. IEEE Comput. Graph. App. 37(2),

[Altimentova, 2016] Altimentova D.Yu., Rozhko A.K. (2016) Information technologies in education // Scientific and methodological journal "Concept". - 2016. - Vol. 11. - P. 826-830 (In Rus.) == Al'timentova D. Yu., Rozhko K. A. Informacionny'e texnologii v obrazovanii // Nauchno-metodicheskij e'lektronny'j zhurnal «Koncept». – 2016. – T. 11. – S. 826–830. Available at http://e-koncept.ru/2016/86179.htm

[Borodkin, 2015] Borodkin L.I., Zherebyatev, D.I., Moor, V.V., Mironenko M.S., Ostapenko M.Yu. (2015) Virtual reconstruction of Moscow's Strastnaya square of the XVII-XIX centuries as an electronic digital archive // The Role of museums in the information support of historical science: collection of articles, 2015. (In Rus.) == Borodkin L.I., Zherebyat'ev D.I., Moor V.V., Mironenko M.S., Ostapenko M.Yu. Virtual'naya rekonstrukciya Strastnoj ploshhadi Moskvy' XVII-XIX vekov kak e'lektronny'j cifrovoj arxiv//Rol' muzeev v informacionnom obespechenii istoricheskoj nauki: sbornik statej, 2015.

[British Museum] Web resource, URL: https://britishmuseum.withgoogle.com/

[Canvasmol] Web resource, URL:https://alteredqualia.com/canvasmol/

[Chuvikov, 2015a] Chuvikov D.A., Yurchik, P.F., Moskalev, A.G., Golubkova, V.B. (2015) Assessment of the applicability of three-dimensional game cores for creating distance learning systems // Industrial ACS and controllers. 2015. No 10. (In Rus.) == Chuvikov D.A., Yurchik P.F., Moskalyov A.G., Golubkova V.B. Ocenka primenimosti trexmerny'x igrovy'x yader dlya sozdaniya sistem distancionnogo obucheniya // Promy'shlenny'e ASU i kontrollery'. 2015. No 10.

[Chuvikov, 2015b] Chuvikov D.A.(2015) Application of 3D technologies in the Web for solving intellectual problems // Automation and control in technical systems. - 2015. No 1(13), eISSN: 2306-1561 (In Rus.) = Chuvikov D.A. Primenenie 3D texnologij v Web pri reshenii intellektual'ny'x zadach // Avtomatizaciya i upravlenie v texnicheskix sistemax. - 2015. No 1(13)

[Chrome Policy] eb resource, URL: https://developer.chrome.com/extensions/contentSecurityPolicy

[Jaworski, 2016] Jaworski N., Lobur, M., Iwaniec, M. (2016) Composite Materials Microlevel Structure Models Visualization Distributed Subsystem based on WebGL // 2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH).-2016.

[Lavrov, 2019] Lavrov A. V., Spiridonov, A. M., Lavrova, A.V.(2019) The Interface displacement between the Museum spaces in virtual reality systems // Almanac of scientific works of young scientists of ITMO University. Volume 1: proceedings of the XLVIII scientific and methodical Conf., Feb. 2019, ITMO University, Saint Petersburg. (In Rus.) == Lavrova A.V., Spiridonova A.M., Lavrov A.V. Interfejs peremeshheniya mezhdu muzejny'mi prostranstvami v sistemax virtual'noj real'nosti // Al'manax nauchny'x rabot molody'x ucheny'x Universiteta ITMO. Tom 1: materialy' XLVIII nauch. i uch.-metod. konf., 2019 g. SPb.: Universitet ITMO, 2019.

[Omarova, 2018] Omarova S.K. (2018) Modern trends in education in the era of digitalization // Pedagogy. Questions of theory and practice. No 1 (9) 2018 (In Rus.) == Omarova S.K. Sovremennye tendencii obrazovaniya v epohu cifrovizacii // Pedagogika. Voprosy teorii i praktiki. No 1 (9) 2018, Available at https://cyberleninka.ru/article/n/sovremennye-tendentsii-obrazovaniya-v-epohu-tsifrovizatsii/viewer

[Skachkova, 2013] Skachkova T.E., Popova, N.Y., Antokhin, A.I. (2013) Digital sculpture as a method of visualization in the teaching of biology // Mathematical and computer modeling in biology and chemistry. Prospects of development. II international scientific Internet conference. - Kazan, September 2013. - Volume II. - Pp. 22-26. (In Rus.) == Skachkova T.E., Popova N.Y., Antokhin A.I. Cifrovaya skulptura kak metod vizualizacii v obuchenii biologii // Matematicheskoe i komp'yuternoe modelirovanie v biologii i himii. Perspektivy razvitiya. II Mezhdunarodnaya nauchnaya Internet-konferenciya. – Kazan', sentyabr' 2013. – Tom II. - S. 22–26.

[Sky Live] Web resource, URL: https://theskylive.com/

[Ryabkov, 2018] Ryabkov A.V. (2018) Principles of protection of educational content from unauthorized changes / / Alley of science. 2018. No 4 (20). (In Rus.) == Ryabkov A. V. Principy' zashhity' obrazovatel'nogo kontenta ot nesankcionirovannogo izmeneniya // Alleya nauki. 2018. No 4 (20).

[Vildanov, 2014] Vildanov A. N., Shafeeva E.P.(2014) Three-Dimensional browser visualization of molecules using WEBGL technology // Modern problems of science and education. 2014. No 6. (In Rus.) == Vil'danov A.N., Shafeeva E.P. Trexmernaya brauzernaya vizualizaciya molekul s pomoshh'yu texnologii WEBGL // Sovremenny'e problemy' nauki i obrazovaniya. 2014. No 6.

[Virtual Hampson Museum] Web resource, URL: https://hampson.cast.uark.edu/

[Vlasenko, 2016] Vlasenko A.N.(2016) Use of WebGL in the educational process when preparing specialists in the field of welding production / A. N. Vlasenko, V. V. Stepanov / / Remote and virtual training. - 2016. No 12. (In Rus.) == Vlasenko, A.N. Ispol'zovanie WebGL v obrazovatel'nom processe pri podgotovke specialistov v oblasti svarochnogo proizvodstva / A.N. Vlasenko, V.V. Stepanov // Distancionnoe i virtual'noe obuchenie. - 2016. No 12.

[WebGL Fundamentals] Web resource, URL: https://webglfundamentals.org/

[W3C, 2019] Browser & Platform Market share, web resource, URL:https://www.w3counter.com/globalstats.php

[WebGL 2.0 Specification] Web resource, URL: https://www.khronos.org/registry/webgl/specs/latest/2.0/1