

WebAR development tools: An overview

Dmytro S. Shepiliev^a, Yevhenii O. Modlo^d, Yuliia V. Yechkalo^b, Viktoriia V. Tkachuk^b, Mykhailo M. Mintii^a, Iryna S. Mintii^{a,c}, Oksana M. Markova^b, Tetiana V. Selivanova^a, Olena M. Drashko^a, Olga O. Kalinichenko^a, Tetiana A. Vakaliuk^{b,e}, Viacheslav V. Osadchyi^f and Serhiy O. Semerikov^{a,b,c}

^aKryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

^bKryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine

^cInstitute of Information Technologies and Learning Tools of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

^dState University of Economics and Technology, 5 Stephana Tilhy Str., Kryvyi Rih, 50006, Ukraine

^eZhytomyr Polytechnic State University, 103 Chudnivska Str., Zhytomyr, 10005, Ukraine

^fBogdan Khmelnytsky Melitopol State Pedagogical University, 20 Hetmanska Str., Melitopol, 72300, Ukraine

Abstract

Web augmented reality (WebAR) development tools aimed at improving the visual aspects of learning are far from being visual and available themselves. This causing problems of selecting and testing WebAR development tools for CS undergraduates mastering in web-design basics. The research is aimed at conducting comparative analysis of WebAR tools to select those appropriated for beginners.

Keywords

augmented reality software tools, WebAR, A-Frame, AR.js, Three.js, JSARToolKit

CS&SE@SW 2020: 3rd Workshop for Young Scientists in Computer Science & Software Engineering, November 27, 2020, Kryvyi Rih, Ukraine

✉ sepilevdmirij@gmail.com (D.S. Shepiliev); eugenemodlo@gmail.com (Y.O. Modlo); uliaechk@gmail.com (Y.V. Yechkalo); viktoriya.tkachuk@gmail.com (V.V. Tkachuk); mykhailo.mintii@gmail.com (M.M. Mintii); irina.mintiy@kdpu.edu.ua (I.S. Mintii); markova@mathinfo.ccjournals.eu (O.M. Markova); vitro090@gmail.com (T.V. Selivanova); lenamd888@gmail.com (O.M. Drashko); olgakalinichenko6@gmail.com (O.O. Kalinichenko); tetianavakaliuk@gmail.com (T.A. Vakaliuk); poliform55@gmail.com (V.V. Osadchyi); semerikov@gmail.com (S.O. Semerikov)

🌐 <https://kdpu.edu.ua/personal/ismintii.html> (I.S. Mintii); <https://kdpu.edu.ua/personal/vstania.html> (T.V. Selivanova); <https://kdpu.edu.ua/personal/omdrashko.html> (O.M. Drashko);

<https://kdpu.edu.ua/personal/ookalinichenko.html> (O.O. Kalinichenko);

<https://sites.google.com/view/neota/profile-vakaliuk-t> (T.A. Vakaliuk);

<https://mdpu.org.ua/new/uk/kafedra-nformatiki-kbnetiki/2990.html> (V.V. Osadchyi);

<https://kdpu.edu.ua/semerikov> (S.O. Semerikov)

🆔 0000-0001-6913-8073 (D.S. Shepiliev); 0000-0003-2037-1557 (Y.O. Modlo); 0000-0002-0164-8365 (Y.V. Yechkalo); 0000-0002-5879-5147 (V.V. Tkachuk); 0000-0002-0488-5569 (M.M. Mintii); 0000-0003-3586-4311 (I.S. Mintii); 0000-0002-5236-6640 (O.M. Markova); 0000-0003-2635-1055 (T.V. Selivanova); 0000-0002-9823-5101 (O.M. Drashko); 0000-0002-7057-2675 (O.O. Kalinichenko); 0000-0001-6825-4697 (T.A. Vakaliuk); 0000-0001-5659-4774 (V.V. Osadchyi); 0000-0003-0789-0272 (S.O. Semerikov)



© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

1. Introduction

Nowadays, augmented, or mixed, reality is a popular technology which is gaining in popularity among educators as well. The basic application of augmented reality (AR) to education is provision of visualization by visualizing computer models of systems and objects using general (a computer equipped with an integrated or external web-camera) and specific (AR goggles, VR helmets) tools, and adapted devices (mobile phones).

The latter are the most widespread devices among participants of the academic process. As of May 2020, there are 8.3 mln users of mobile games, 26.2% of which are 18–24-year olds [1]. According to 2019 data, Ukrainian mobile users at the age of 16-25 are the most frequent users of social networks (92%), message exchange tools (66%), video applications (64%) and mobile games (50%) [2]. Comparison of the two latter sources enables drawing a conclusion that there are from 16 to 16.6 mobile users in Ukraine who are senior school pupils, students of all educational levels or young people. According to the Government Statistics of Ukraine, at the beginning of the 2019–2020 academic year, there are 1.440 mln students of colleges, secondary technical schools, universities, academies and institutes [3], 4.163 mln of which are 16–25-year-olds [3, p. 26].

Comparison of the given statistics indicates that each student has two mobile numbers on average attached to one of the popular stores of mobile applications. It can be either a single mobile phone with two SIM-cards or two mobile phones. As of December 2020, 79.8% of mobile devices in Ukraine had the operating system Android installed [4], while the leading web browser was Chrome (67.8%) [5]. Considering that the third most popular browser, Opera, is built on the Chrome core [6], Chrome accounts for 80%.

In spite of some specific clients for social networking, message exchange, video watching and mobile games, a browser is a universal tool for performing relevant tasks and developing new software in the web-environment quickly. The main advantage of web-clients is their universality as there are Chrome versions for all mobile platforms. Low operating speed of programmes has been the basic disadvantage until recently. Appearance of WebAssembly [7] in March 2017 enabled acceleration of web-programmes and transition of software designed for other platforms into the web-environment.

Access to high-quality libraries of computer vision, image recognition and monitoring (AR-ToolKit and its web-version JSARToolKit5 [8]) is the key to designing WebAR software. At the same time, joint application of JSARToolKit5 and WebGL reveals that:

- markers applied by JSARToolKit5 are a two-dimensional type of a barcode of low visualization,
- visualization of objects by means of WebGL requires in-depth understanding of principles of 3D graphics and greater amounts of a software code.

Thus, the mentioned WebAR development tools are aimed at enhancing visualization of the teaching process, yet they lack visualization and accessibility themselves, this fact indicating the research problem of selecting and testing software tools of WebAR development designed for beginners mastering the basics of web-design - students of lyceums and junior Informatics students.

The following article section are aimed at solving the following task: performing a comparative analysis of WebAR designing tools in order to select those suitable for beginners.

2. Review of WebAR designing tools

2.1. Augmented reality: its notion and hardware for Web

For the recent 60 years, it has been innovative (trendy, interesting, useful, etc.) to program AR, while its application to the web-environment is conditioned by current development of technology. To start this process, one needs an AR-compatible browser like Firefox or Chrome and a text editor (like mcedit or Sublime). Beginners need basic knowledge of web-design (HTML, CSS and JavaScript), while experience of working with web API and GitHub is also applicable.

Hereafter, by AR, we mean an ability of a device (a mobile device or a web browser) to monitor an image or represent a 3D object over the image. The main aim of AR is to show a computer model online in the real-mode space to establish interaction of a user in the real-mode space and a 3D model in the virtual one.

AR can be both marker and markerless. In case of the marker AR, a device tracks a 2D marker: when it is found, a 3D object is visualized. In the markerless variant, a device searches for a flat surface (a table, a floor, etc.) and place a 3D object there.

Using a camera, AR enables visualization of computer-generated objects in games, marketing and other software as in case of arranging furniture in a living room or trying on new clothes before buying. It is a great opportunity for business as it demonstrates a product before a customer really buys it [9].

There are specific devices designed for AR like helmets and headsets enabling a user to plunge into a modelled environment.

AR supplies a real-life world with 3D models controlled by mobile devices anyplace. Virtual reality (VR) makes a user get involved into a modelled world by means of head mounted devices (HMD).

Interactivity in AR and VR software is provided in a similar way. For example, VR actually uses controllers and occasionally hand-tracking enabling a user to interact with 3D objects inside a scene they are situated in.

Major hazards of using HMD in VR include:

- eye strain,
- dizziness and headaches after using HMD.

Unlike VR, AR does not possess any of these hazards for health. Yet, the ability of users to remain concentrated on what they are doing is disturbing, in particular, while using AR, for safety reasons.

Smartphones with iOS (11 and higher under iPhone and iPad control) and Android (7.0 and higher) are the most widespread type of devices for AR.

For web browsers, like Google Chrome and Mozilla Firefox, AR is applicable if they support WebRTC [10] and WebGL [11].

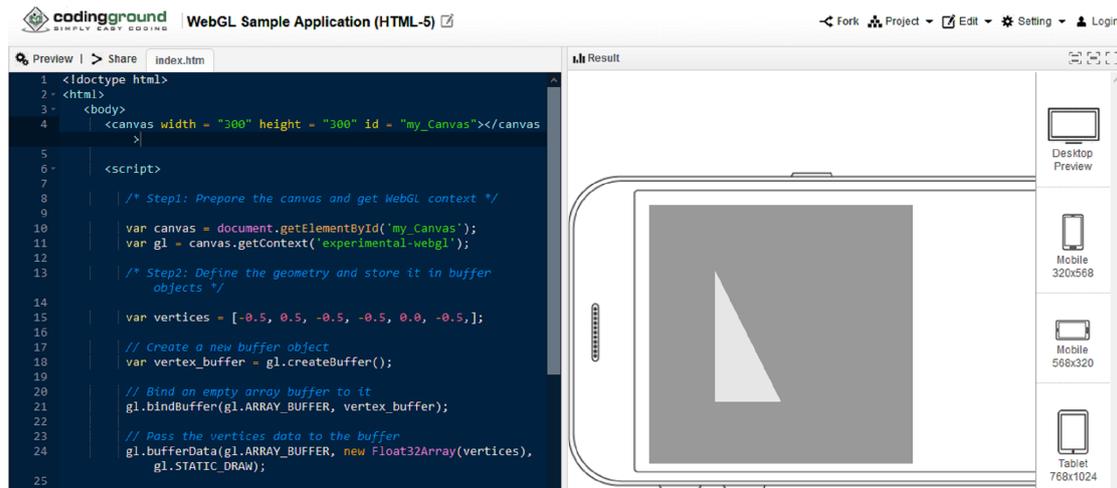


Figure 1: The source code and the result of the simple WebGL programme running.

Microsoft HoloLens is an HMD-like AR headset which is designed currently [12]. Like Google Glass [13], it is designated for corporate use, yet, unlike Glass, it is compatible with Windows 10, not with Android.

2.2. Tools of visualizing computer models in the Web

WebGL (OpenGL ES for the Web) is API for 3D graphics in the web browser designed by the Khronos Group Inc [11]. WebGL uses a shader programming language GLSL (OpenGL Shader Language) and is part of an document object model (DOM API) of the browser. All the leading browser designers – Apple (Safari), Google (Chrome), Microsoft (Edge), and Mozilla (Firefox) are members of WebGL Working Group. The current version WebGL 2.0 corresponds to the standard OpenGL ES 3.0 API.

The programme described by WebGL contains both a JavaScript code and a C-like code GLSL. The example of simple software for building a triangle (figure 1) from [14] contains 105 rows of the code comprising five blocks:

1. Preparing a canvas (HTML object canvas) and obtaining a rendering context of WebGL.
2. Defining geometry attributes such as vertexes, indices, etc., and storing them in buffer objects.
3. Designing and compiling software for vertex and fragment shaders.
4. Connecting shader software and buffer objects.
5. Rendering a required image.

Application of WebGL for visualizing computer models in the Web is not the easiest way to do it: the code will account for thousands of rows and require high-level skills in designing 3D graphics by means of OpenGL. Considering the fact that the first research task involves selection of tools for beginners, it is reasonable to use libraries that make application of WebGL simpler.

Table 1
Assessment of efficiency of using libraries for WebGL

| Title | Animation | Native audio | Network access | Considering laws of physics | WebGL version | WebXR support | Import | Export | License | General assessment |
|-------------|-----------|--------------|----------------|-----------------------------|---------------|---------------|------------------|--------------------|------------|--------------------|
| A-Frame | + | + | - | - | 1.0 | + | Many formats (3) | HTML, three.js (2) | Free | 10 |
| CopperLicht | + | + | - | + | 1.0 | - | - | - | Free | 5 |
| OSG.JS | + | + | - | - | 1.0 | + | - | - | Free | 5 |
| Three.js | + | + | - | - | 2.0 | + | Many formats (9) | Many formats (4) | Free | 19 |
| Verge3D | + | + | - | + | 1.0 | + | Many formats (3) | glTF (1) | Commercial | 9 |
| Clara.io | + | - | - | + | 1.0 | + | Many formats (5) | Many formats (6) | Commercial | 15 |
| Babylon.js | + | + | - | + | 2.0 | + | Many formats (5) | Many formats (9) | Free | 21 |

Table 1 contains results of assessing efficiency of using JavaScript libraries for WebGL objects.

Considering the fact that the conducted analysis is designated for teaching beginners, there are the best-rated non-commercial libraries determined for working with WebGL – Babylon.js, Three.js and A Frame.

Babylon.js [15] enables designing complex 2D-objects by using an essentially smaller code size than that for WebGL: the code size shown in figure 2 is 5-fold smaller than that of the WebGL-code for a simpler scene in figure 1. The designed scene is controlled by simple controlling elements. Unlike other rating participants, Babylon.js is a physical engine widely applied to designing computer games for the Web.

Three.js is a free 3D graphics library of general application written in JavaScript. On <https://threejs.org/>, one can find many high-quality demonstrations designed by using it [16]. Its author, Ricardo Miguel Cabello, known also as Mr.doob, is one of the WebGL pioneers. For this reason, this library is often used for designing other libraries.

Object designing in Three.js involves three steps:

1. determining object geometry – vectors, colours, etc,
2. determining materials – a rendering method for an object,
3. composing geometry and materials.

The code size required for designing the 3D scene by Three.js (figure 3) is a bit larger than that of Babylon.js. It is explained by combining in Babylon.js the operations of designing an object and those of adding it to the scene into a single call of a constructor of a relevant class.

A-Frame is actually a tool of rapid prototyping: a greater part of its application is a HTML-like code. A-Frame commands are described by tags similar to HTML ones, yet, unlike the

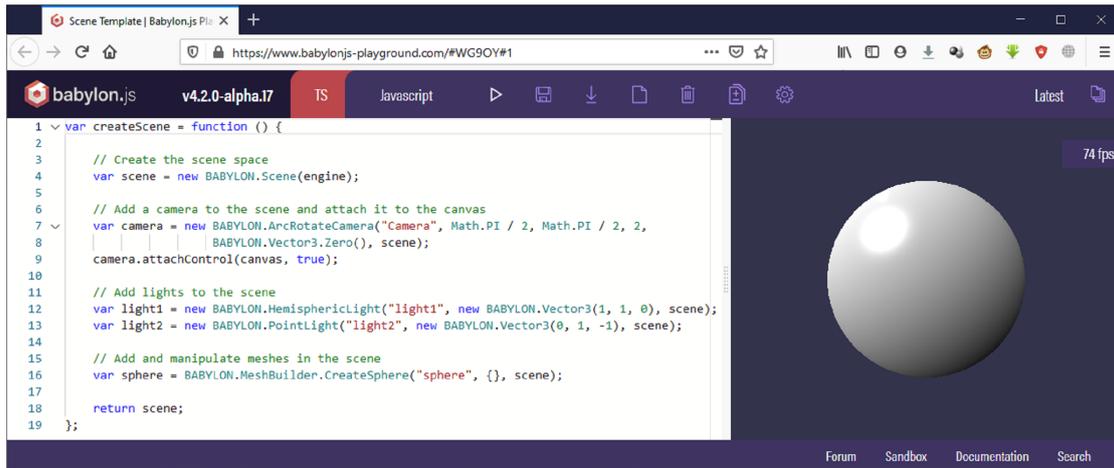


Figure 2: The source code and the result of the simple programme running in Babylon.js.

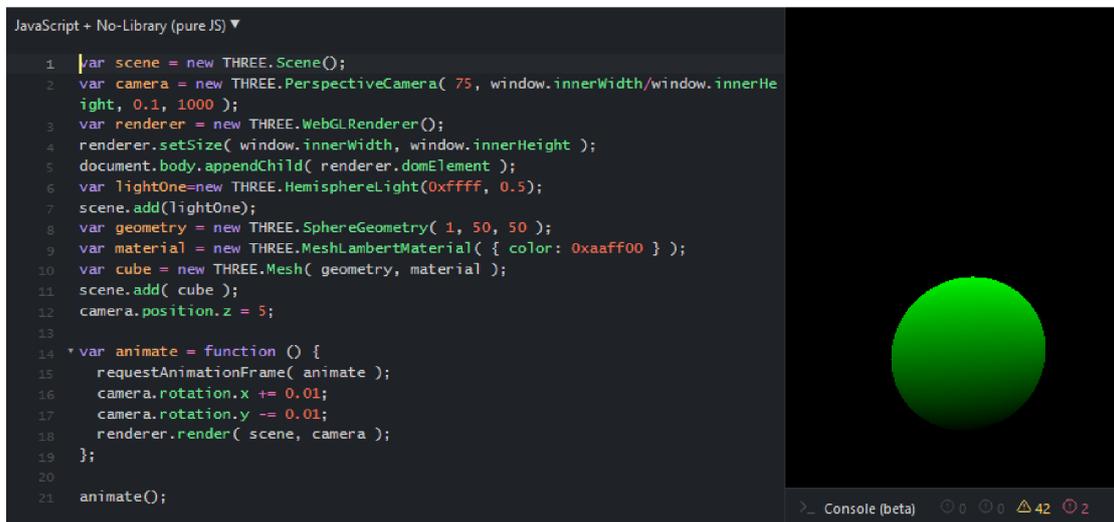


Figure 3: The source code and the result of a simple programme running in Three.js.

latter, are interpreted not in a client-side browser, but are a means of accessing JavaScript on the server side [17].

Figure 4 is significant as the A-Frame code size required for designing the same sphere is a three-fold less than that of Babylon.js/Three.js.

Thus, in spite of the third place in the rating of functional opportunities, A-Frame is the leader in terms of visualization and accessibility among the considered libraries for 3D graphics on the Web. Considering the fact that A-Frame is a superstructure over Three.js, it is reasonable to master them in a loop mode starting from A-Frame.



Figure 4: The output source code and the result of the simple programme running in A-Frame.

2.3. Tools for tracking real-life objects

As [18] contains comparative analysis of the most widespread AR SDK, let us focus on those applicable to designing on the Web – WebAR SDK. Unfortunately, as of January 2021, there are only three free tools, each of which is associated with considered tools of building computer models.

In August 2019, Babylon.js announced **Babylon AR** [19, 20], the project integrating Babylon.js and the computer vision library OpenCV. At present, there is implemented tracking of standard markers similar to QR-codes. In spite of the fact that designers of Babylon.js consider Babylon AR project being on the early stage, web-programmes designed are efficient only on WebXR-supporting devices [21]. This narrows the scope of applying Babylon AR to 8.1 and higher Android devices. In January 2021, there were 74% of such devices among Android ones in Ukraine [22].

Considering the trend of increasing this share, Babylon AR can be considered a promising project of WebAR software development starting from 2021–2022 [22].

ARToolKit, which is 20 years older than Babylon AR, is one of the most applicable AR libraries. **JSARToolKit** (ARToolKit.js) is currently supporting three types of square markers (with arbitrary drawing, a 2D code and markers) and NFT-markers (natural feature tracking) [8]. JSARToolKit is not specific for a WebGL library, yet, it is most frequently used together with Three.js: the example [23] of their joint work developed by Lee Stemkoski is given in figure 5.

The code size required for implementing functionality of the example makes about 100 rows and is much smaller than in case of JSARToolKit. This is due to applying THREEx library – extension of Three.js for designing computer games, this fact making its functionality similar to Babylon.js [24].

Its author, Jerome Etienne [25], is the main designer of the first and the second version of AR.js, which, as of January 2021, has versions adapted to both A-Frame and Three.js. In the same way as A Frame is a superstructure over Three.js, AR is based on JSARToolKit. Thus, this library supports all types of square (marker tracking) and NFT markers (image tracking) as in JSARToolKit, and also enables locating computer models according to their geospatial coordinates (location-based AR) [26].

When applied with A-Frame, the code required for the example similar to that in figure 5



Figure 5: The cube on the Hiro marker (joint application of Three.js and JSARToolKit).

takes several rows:

```
<a-scene embedded arjs>
  <a-marker preset="hiro">
    <a-box></a-box>
  </a-marker>
  <a-entity camera></a-entity>
</a-scene>
```

3. Conclusions

1. The review of WebAR software design enables recommendations for beginners to master the following combinations of visualization tools of computer models and tools for tracking real-life objects:
 - A-Frame and AR.js is API for fast prototyping most software of which is a HTML-like code. A Frame is applied to designing scenes, objects, animation and other 3D elements in the web browser. AR.js is aimed at tracking markers and enables a scene designed by A-Frame to be rendered directly on the marker;
 - Three.js and ARToolKit.js are used for a deeper level to design software using JavaScript. Three.js uses WebGL to create high-quality 3D scenes in the web browser. JSARToolKit is used for marker tracking and low-level access to data from a camera of a device.

2. Technological conditions of teaching software development with WebAR involve an arbitrary operating system, Internet access, using of cloud storages to deposit materials and availability of at least two classes of devices to test programmes. To master the basics of WebAR software development, it is reasonable to combine A-Frame and AR.js, the scene testing in VR proceeding its AR design. With the scene complexity increasing, JavaScript application to designing A-Frame objects and processing events is accentuated. Designing multi-marker scenes requires different types of matrix markers provided by JSAR-ToolKit, while joint functioning of models attached to different markers requires application of Three.js library. This reveals a dialectic character of teaching WebAR software development: new methods are introduced to overcome problems unsolved by available ones.

References

- [1] Statista, Mobile Games – Ukraine, 2021. URL: <https://www.statista.com/outlook/211/338/mobile-games/ukraine>.
- [2] D. Elagina, Ukraine: popularity of app categories by age 2018, 2020. URL: <https://www.statista.com/statistics/1023304/ukraine-popularity-app-categories/>.
- [3] Derzhavna sluzhba statystyky Ukrainy, Rozpodil postinoho naselennia Ukrainy za statiu ta vikom na 1 sichnia 2019 roku: statystychnyi zbirnyk (Distribution of the permanent population of Ukraine by sex and age as of January 1, 2019: statistical collection), 2019. URL: http://database.ukrcensus.gov.ua/PXWEB2007/ukr/publ_new1/2020/zb_chuselnist%202019.pdf.
- [4] StatCounter, Mobile Operating System Market Share Ukraine, 2021. URL: <https://gs.statcounter.com/os-market-share/mobile/ukraine>.
- [5] StatCounter, Mobile & Tablet Browser Market Share Ukraine, 2021. URL: <https://gs.statcounter.com/browser-market-share/mobile-tablet/ukraine/>.
- [6] Opera Software AS, webkit, 2019. URL: <https://dev.opera.com/tags/webkit/>.
- [7] WebAssembly, 2020. URL: <https://webassembly.org/>.
- [8] artoolkitX, artoolkitx/jsartoolkit5: Javascript ARToolKit v5.x, 2020. URL: <https://github.com/artoolkitx/jsartoolkit5>.
- [9] I. S. Mintii, V. N. Soloviev, Augmented reality: Ukrainian present business and future education, volume 2257, CEUR-WS, 2018, pp. 227–231. URL: <http://ceur-ws.org/Vol-2257/paper22.pdf>, 1st International Workshop on Augmented Reality in Education, AREdu 2018, 2 October 2018.
- [10] Mozilla, individual contributors, WebRTC API – Web APIs, 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API.
- [11] Mozilla, individual contributors, WebGL: 2D and 3D graphics for the web – Web APIs, 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API.
- [12] Microsoft, Mixed reality technology for business, 2021. URL: <https://www.microsoft.com/en-us/hololens>.
- [13] Google, Glass, 2020. URL: <https://www.google.com/glass/start/>.

- [14] Tutorialspoint, WebGL - Sample Application, 2020. URL: https://www.tutorialspoint.com/webgl/webgl_sample_application.htm.
- [15] Babylon.js, Babylon.js: Powerful, Beautiful, Simple, Open - Web-Based 3D At Its Best, 2021. URL: <https://www.babylonjs.com/>.
- [16] R. Cabello, three.js – JavaScript 3D library, 2021. URL: <https://threejs.org/>.
- [17] Supermedium, A-Frame – Make WebVR, 2020. URL: <https://aframe.io/>.
- [18] O. V. Syrovatskyi, S. O. Semerikov, Y. O. Modlo, Y. V. Yechkalo, S. O. Zelinska, Augmented reality software design for educational purposes, volume 2292, CEUR-WS, 2018, pp. 193–225. URL: <http://ceur-ws.org/Vol-2292/paper20.pdf>, student Workshop on Computer Science and Software Engineering, CS&SE@SW 2018, 30 November 2018.
- [19] Babylon.js, Babylon AR Preview CDN, 2019. URL: <https://ar.babylonjs.com/>.
- [20] J. Murray, Babylon.js, Babylon AR, 2019. URL: <https://medium.com/@babylonjs/babylon-ar-7823ab4a80c1>.
- [21] Mozilla, individual contributors, WebXR Device API – Web APIs, 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API.
- [22] StatCounter, Mobile Android Version Market Share Ukraine, 2021. URL: <https://gs.statcounter.com/android-version-market-share/mobile/ukraine>.
- [23] L. Stemkoski, Basic Cube, 2018. URL: <https://stemkoski.github.io/AR-Examples/hello-cube.html>.
- [24] J. Etienne, THREE.js Game Extensions for Three.js, 2017. URL: <https://www.threejsgames.com/extensions/>.
- [25] J. Etienne, jeromeetienne/threex: Game Extensions for three.js, 2017. URL: <https://github.com/jeromeetienne/threex>.
- [26] AR.js organization, AR.js-org/AR.js: Image tracking, Location Based AR, Marker tracking. All on the Web, 2020. URL: <https://github.com/AR.js-org/AR.js>.