The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-2-2023
29th CIPA Symposium "Documenting, Understanding, Preserving Cultural Heritage:
Humanities and Digital Technologies for Shaping the Future", 25–30 June 2023, Florence, Italy

COMPARISON OF CHARACTERISTICS OF BIM VISUALIZATION AND INTERACTIVE APPLICATION BASED ON WEBGL AND GAME ENGINE

Y. Zheng [1], A. Merchant [1], J. Laninga [2], Z. X. Xiang [1], K. Alshaebi [1], N. Arellano [2], H. Romaniuk [2], S. Fai [2], D.H. Sun [3]

[1] Carleton University - (yuzheng4, ArkounMerchant, zixunxiang, KhaledAlshaebi)@cmail.carleton.ca
[2] Carleton Immersive Media Studio, Carleton University, 1125 Colonel By Drive Ottawa, Canada – (jlaninga, narellano, hromaniuk, sfai)@cims.carleton.ca
[3] Beijing Huachuang Tonsing Tech Co., Ltd. - 738988486@qq.com

KEY WORDS: Game Engine, Unreal Engine, WebGL, BIM Visualization, Interactive Application, Pixel Streaming, Workflow

ABSTRACT:

How can we make the building information model accessible to all stakeholders on a project? An efficient approach is to access the building information model is to use the software that created the model. However, not all stakeholders will be able to use this highly specialized software—due to lack of training and expensive licences—even if some software specially developed a simplified version of the viewer to browse the model, however, it still failed to provide convenient access to these models for participants from a wide range of backgrounds. The current development of BIM model visualization and interactive applications is mainly based on two technologies: WebGL and game engines. What is the general workflow of WebGL and Game Engines supporting application development? What are their characteristics respectively? What conditions are restricted? There are no relevant academic papers to discuss and compare these two types of platforms. Therefore, this is the content of this essay. By comparing the workflow and characteristics of BIM visualization and interactive application development based on WebGL and Game Engine, it can provide a reference for heritage building managers when planning the development of relevant application tools and meet the participation needs of different stakeholders.

1. INTERACTIVE APPLICATION OF BUILDING INFORMATION MODEL HELPS STAKEHOLDERS PARTICIPATE IN HERITAGE CONSERVATION

Mario Carpo proposes "The Participatory Turn That Never Was" in the field of architectural design in the fourth chapter of his The Second Digital Turn: "the customer, client, or any other stakeholder would be called on to intervene and participate in the design process, to 'customize,' or co-design, the end product within the framework and limits set by the designers or administrators of the system." He considered that this is the turn brought by the rise of the participation spirit of web2.0 from the new millennium (Carpo, 2017). It also hints to how could heritage building conservation practice embraces the digital turn. Conservation process is the informed decision-making process, this means involving many stakeholders (people, institutions, and government agencies) in the process and working with them to understand why a particular place is important and to identify what physical evidence needs to be preserved (Letellier, Schmid, and LeBlanc, 2007). Semantically rich three-dimensional (3D) models such as building information modelling (BIM) are increasingly used in digital heritage. They provide the required information to varying stakeholders during the different stages of the historic buildings lifecycle which is crucial in the conservation process (Bassier, Vergauwen, and Van Genechten, 2017).

How can we make the building information model accessible to all stakeholders on a project? An efficient approach is to access the building information model is to use the software that created the model. However, not all stakeholders will be able to use this highly specialized software—due to lack of training and expensive licences—even if some software specially developed a simplified version of the Viewer[1] to browse the model, though, it still failed to provide convenient access to these models for participants from a wide range of backgrounds.

Many studies have focused on this issue and made corresponding exploration. For example, Achille et al. (2019) proposed that one of the modern methods adopted to achieve this goal is to use structured web-platform. With the help of BIM3DSG platform, they established a set of reality-based 3D models of the chapels linked to a Data Base (DB) of information for the world cultural heritage site "Sacri Monti of Piedmont and Lombardy". Used to help to update a section of the next UNESCO's Periodic Reporting (2018-2024) and will contribute to shaping "standard" best practices to monitor and safeguard Sacri Monti complex and all the similar case studies. Giovannini et al. (2019) used an interactive web-presentation portal of high-resolution 3D models enriched by historical and Archival set of content, from the digitization procedure applied to collection objects, to the digitization process of related data and information. From 2014 to 2017, Beijing Guowenyan Information Technology Co., Ltd. once entrusted Beijing Huachuang Tonsing Tech Co., Ltd. to complete the digital assets display system for the colour painted statues and murals of Shuanglin Temple and realized the visualization of dense point clouds and photogrammetry mesh models through WebGL (**Figure 1**). The above three cases are all BIM model visualization tools and interactive applications developed directly on the browser based on the WebGL.

[1] Such as Autodesk Viewer. It is a free browser application that lets you upload, view, and share designs for products such Revit and AutoCAD, or with one of the many supported file types. https://viewer.autodesk.com/designviews

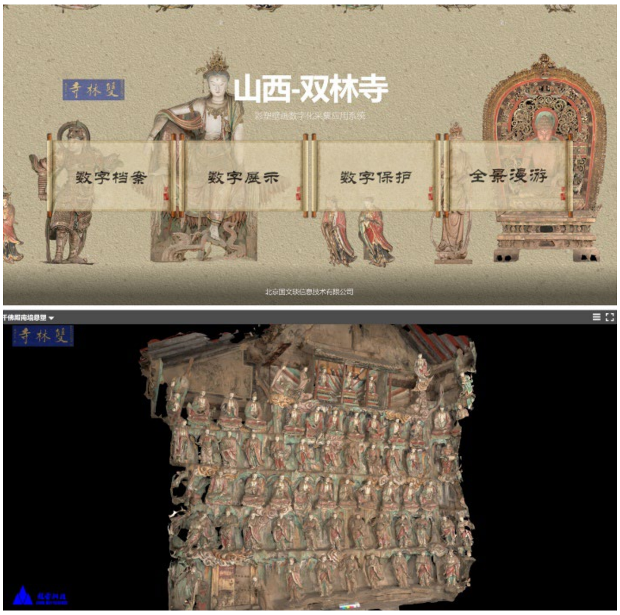

Figure 1. Digital assets display system for the colour painted statues and murals of Shuanglin Temple

However, the more common practical application in dealing with 3D models and making them available to a wider range of people is 3D games. Therefore, the use of game engines as a solution is one of the research options. For example, Pybus et al. (2019) identified six heritage spaces of the Centre Block of the Canadian Parliament which had been previously documented and modelled by the Carleton Immersive media Studio[2] (CIMS) were prepared for Unity 3D[3], Enabling their later use in a storytelling experience in VR. Rodriguez et al. (2021) introduced a binational research team from Mexico and the U.S. made up of historians, architects and animation engineers created the historical settings, characters, and interactivity necessary to offer the public this immersive experience which revisualized a historical event that occurred on March 11, 1554, in the city today known as Antigua, Guatemala. They imported the 3D model created by Autodesk 3DMAX and AutoCAD and the photogrammetry model generated by Metashape into Unreal to reproduce the historical events that happened at that time. Tryfonos et al. (2021) also described a realistic representation of Asinou Church in a 3D video game environment based on Unity. They merged the H-BIM categories such as doors, windows, roofs and floor to Rhinoceros 3D, creating a complete 3D model of the monument. Then, the model is exported to Autodesk FBX Technology format and imported to Unity.

From the above cases, we can see that the current development of BIM model visualization and interactive applications can be based on two platforms: WebGL and Game Engines. While these case studies amply illustrate how their respective 3D rendering tools can meet the needs of a wider range of stakeholders in heritage conservation to participate in decision-making. But what is the general workflow of WebGL and Game Engines supporting application development? What are their characteristics respectively? What conditions are restricted? There are no relevant academic papers to discuss and compare these two types of platforms. In the global geomatics industry, GIM International has published an article, "Emerging Web and Game Engine Tech for 3D Cities[4]", which discusses the considerations when choosing development platforms in the development of 3D city model applications, proposes WebGL and Games Engines are the two main streams of current development. However, the workflow of the two types of development platforms is not described in detail in the paper, and there is no point-to-point comparison of their characteristics of them. Therefore, this is the content of this essay. By comparing the workflow and characteristics of BIM visualization and interactive application development based on WebGL and Game Engine, it can provide a reference for heritage building managers when planning the development of relevant application tools and meet the participation needs of different stakeholders.

2. COMPARISON OF VISUALIZATION AND INTERACTIVE APPLICATION CASES OF BUILDING INFORMATION MODEL BASED ON WEBGL AND GAME ENGINE

Recently, CIMS has carried out two research projects on the development of BIM model visualization interactive applications. The starting point of the projects is the same set of BIM models of Carleton Digital Campus (CDC) developed using Autodesk Revit as an authoring BIM software. The two studies used WebGL and Unreal Engine as development tools respectively, and the goal of the interactive application is to make the BIM easily accessible to all stakeholders of the project. These two studies form an excellent contrast. By graphically illustrating the workflow of the two types of development processes, we can understand the differences between the two types of platforms in the development of BIM models visualization and interactive applications, especially in the development process. Compare their ability to achieve their goals, and the advantages and disadvantages of the development process and the end result.

2.1 CDC Digital Twin project workflow based on WebGL

The CDC Digital Twin project is the first attempt by the CIMS lab to develop a visual and interactive application of the BIM models of campus' buildings based on a previous CIMS' project called Digital Campus Innovation[5] (DCI). It is part of Imagining Canada's Digital Twin[6] (ICDT) project. In 2013, CIMS initiated the DCI project that examined integrating Building Performance Simulation (BPS) technologies with BIM on a campus scale. Over the next four years, the multi-disciplinary team produced a model of the Carleton University campus. The goal of the DCI project is to define, develop and evaluate new processes, methods and technologies for a systemic, integrative and collaborative approach to decision-making in planning, design, construction and operation of sustainable communities (**Figure 2**). The basis of the 3D visualization technology in this project is WebGL.

Interactive applications based on WebGL generally need to split data when processing BIM models such as Revit and divide BIM models into geometric and non-geometric data. The geometric

[2] https://cims.carleton.ca/#/home
[3] Unity is a cross-platform game engine developed by Unity Technologies.
[4] https://www.gim-international.com/content/article/emerging-web-and-game-engine-tech-for-3d-cities?utm_source=newsletter&utm_medium=email&utm_campaign=3D+Modelling+%26+Visualization+Weeks
[5] https://cims.carleton.ca/#/projects/DigitalCampusInnovation
[6] https://cims.carleton.ca/#/projects/imagining_canada's_digital_twin

prepossessed as glTF[7], which minimizes the size of 3D assets, and the runtime processing needed to unpack and use them. Non-geometric data, that is, attribute data, are exported as JavaScript Object Notation (JSON) files. JSON is a lightweight data-interchange format. Element IDs link the two types of data. Then 3D geometric data is further developed with Three.js [8] and rendered in real-time according to customer needs through WebGL. The attribute data is also called according to the need to match the element ID. In the CDC Digital Twin project, geographic information data comes from map renderers such as Mapbox GL JS and Open Street Map, integrated into the application by JavaScript programming and finally output to users through the Web browser (Figure 3).

In the CDC Digital Twin project, the IFC.js library has played a huge role. IFC.js is the first JavaScript library fully dedicated to parse IFC files so they can be displayed and manipulated in any web browser. They were released in December 2020. Their mission is to provide AEC professionals with easy and free methods to build their own BIM tools. IFC.js provides a viewer with examples of how to create your own BIM application: Scene navigation, material changes, element selection by clicking on them, section plans, etc.

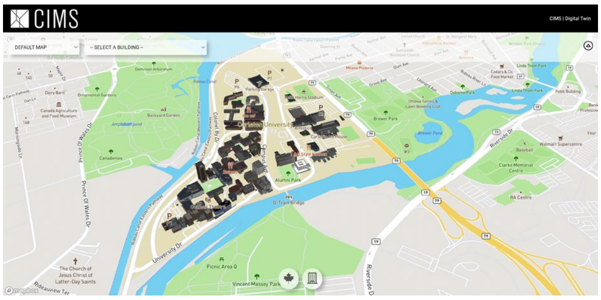

Figure 2. The CDC digital twin browser

2.2 CDC INCA project workflow based on Unreal Engine

The CDC-INCA project, namely CDC: Interactive "Digital" Campus Application, is another attempt by CIMS Lab to develop a visual and interactive application of the BIM model of campus buildings in Revit format on the basis of the DCI project. This project focuses on making the building information model accessible to all stakeholders on a project. The solution is a Streamable Assets Viewer (**Figure 4**). The Viewer is Built in Unreal Engine, Accessible in a browser via pixel streaming. With Pixel Streaming, we run a packaged Unreal Engine application on a local server or a server in the cloud, along with a small stack of web services that are included with the Unreal Engine. The final result of the project is very similar to video games. In a virtual 3D space scene with rich content, we can interact with the model through the UI, load the campus buildings in levels, look at the building in the section box, annotate by location, view MetaData etc.

Figure 5 is a diagram illustration of the workflow of the project. Starting from the Revit model, through model reorganization, each building is exported as Unreal DataSmith files based on geographic location information. The index address relationship of the exported data files is established through the PostgreSQL relational database. The UI design was also carried out and the menu functions in the application were implemented using Unreal Widget Blueprint. The project leader established various settings of the Unreal Project, loaded the data in the CIMS server, integrated it into an application of BIM Viewer, and finally streamed it to the Web client through Pixel Streaming and remotely controlled the server interaction through menu options. Or achieve the same function through a mobile device.

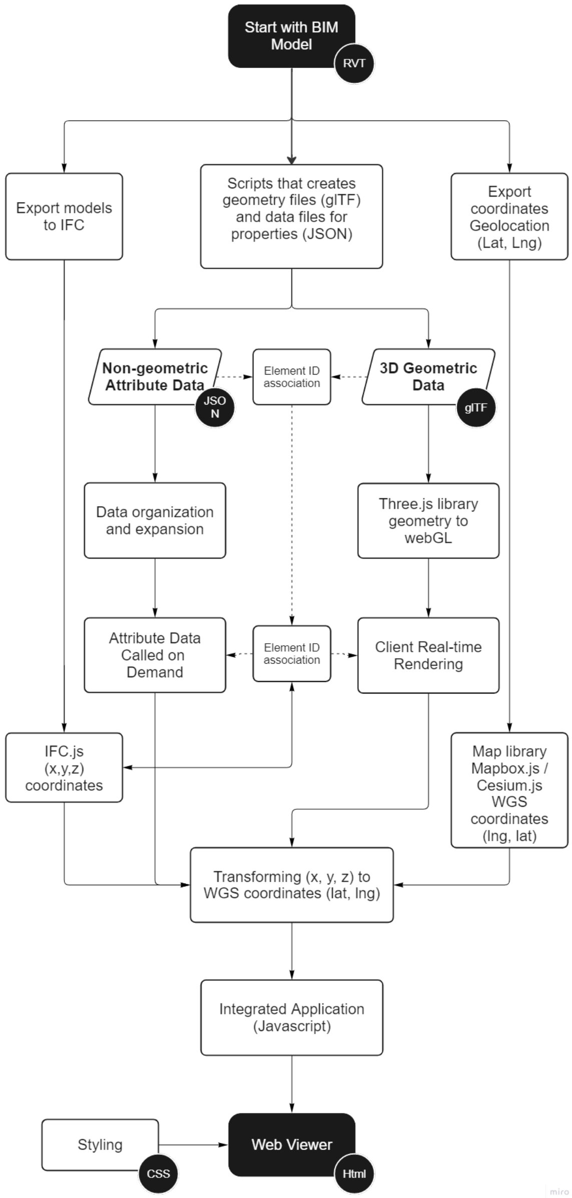

Figure 3. WebGL based interactive application workflow - BIM / GIS

[7] glTF™ is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by engines and applications. https://www.khronos.org/api/index_2017/gltf

[8] Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser using WebGL.

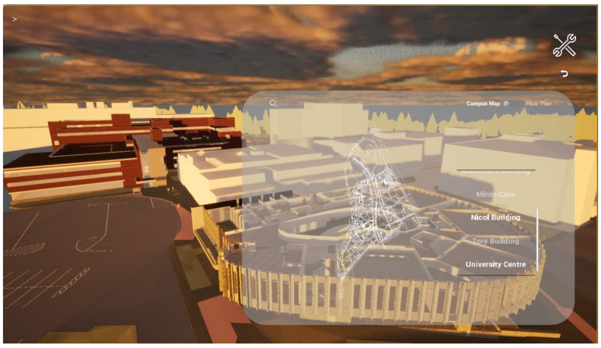

Figure 4. Streamable assets viewer demo of CDC-INCA

2.3 Comparison of two Approaches

2.3.1 Rendering features, or visualization capabilities: Many of the most advanced visualization techniques are driven by and applied in the video games industry. Hence, UE can provide AAA-level game visual effects, but it is not easy to achieve through WebGL. This is actually because WebGL is a lower-level general technology and standard than Game Engines, and frameworks like Three.js, Babylon.js and Playcanvas are needed on top of it to help developers achieve complex rendering effects, while a Game Engine such as Unreal is a software product, that is, a more mature commercial development has been done on the basis of the underlying rendering source code, so that users can more easily obtain better rendering effects. Therefore, when people with the same professional ability use these two approaches to develop applications, the Game Engine is significantly better than WebGL in terms of visual experience.

For the performance experience, WebGL and UE have their own advantages and disadvantages. WebGL has some delays when loading larger models, while UE is faster. For example, the overall campus bird's-eye view model of CDC Digital Twin is opened on a laptop whose GPU is NVIDIA GeForce GTX 1050. The test shows that the average time to load the model is two minutes (**Figure 2**), while the same model only takes a few seconds when opened on the CIMS Lab server through Pixel Streaming. Since WebGL applications need to transfer 3D models to the user's local web browser and render them in real-time through the GPU of the user terminal, the model loading time is affected by the scale of the model data, the GPU performance of the local device, and the network transmission speed. The Game Engine directly uses the server for loading and rendering. Generally, the performance of the server is much higher than that of the client's local terminal. The professional rendering capability design of the Game Engine also speeds up model visualization. Nevertheless, to make good use of the limited server resources, a Low LoD Model Mass is still established for the complex building model in the CDC INCA project, which saves computing resources when loading the bird's-eye view model of the entire campus. By building different levels of detail (LoD) models, loading the most economical model for visualization according to the needs of the scene is a commonly used method to improve the performance experience in both approaches.

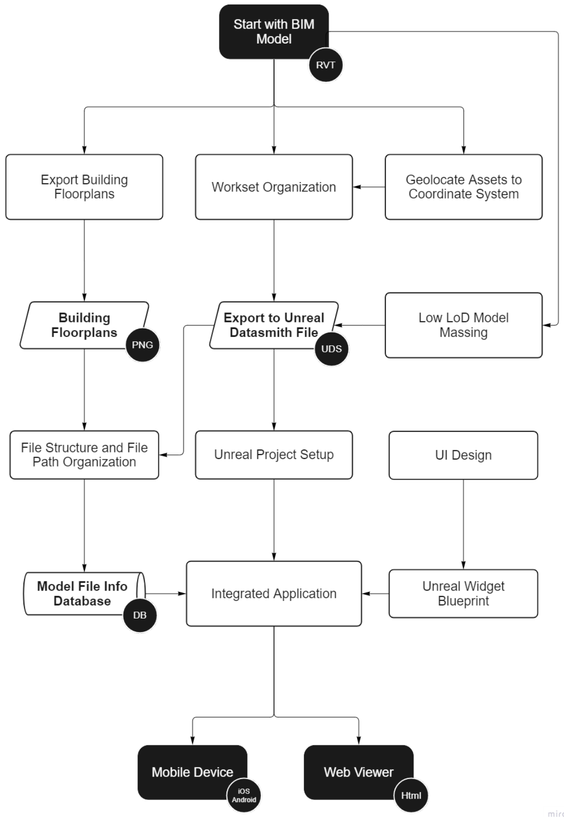

Figure 5. Unreal based interactive application workflow of CDC-INCA

2.3.2 Data integration and management capability: Data integration and management capability, WebGL needs separate the 3D model and attribute data, the data can be expanded and modified, and the data integration is flexible. But in the UE process, the attribute data is exported together with the model file, the ability to integrate and manage attribute information is weak. Data integration is at the core of a meaningful application and can be seen as a key enabler for building data analysis and the creation of digital twins. By generating a semantic information model of a building complex, integrating attribute information (e.g., material, performance, system function data) or other additional information (e.g., age, value, historical maintenance, etc.), the BIM Viewer can be more than just a visualization of a built environment. Furthermore, an increasing amount of data is being collected using real-time sensors related to traffic, weather and building HVAC systems, for example. BIM Viewer can act as a platform for integrating and visualizing these data sources.

2.3.3 Interoperability and flexibility of use: Interoperability and Flexibility of Use, both approaches do not need to install software when using, and the development software can basically be regarded as open source too. WebGL Capable of serving large-scale global users. In the contrast, UE Online users are limited by server capacity, and the ability to support a large number of public services is weak. WebGL could be completely open source, its "openness" plays an important role in the flexibility of use. However, compared with commercial software, there may be problems such as slow update iteration speed, componentization, and low standardization. While UE is professionally maintained by commercial companies, and open source to a certain extent, with a stable development roadmap.

2.3.4 Support for various 3D assets: In the BIM visualization and interactive application, not only the parametric building information model, but also many types of 3D assets need to be visualized and interactively operated in the application. These 3D assets include dense point clouds acquired by 3D laser scanners (e.g., LAS, E57), large textured mesh models obtained from photogrammetry and other professional 3D modelling software (e.g., OBJ, FBX, glTF), different formats of building information models (e.g., IFC), etc. Furthermore, support for flexible data formats that are suitable for several asset types, such as 3D tiles, is becoming increasingly important.

WebGL fully supports different types of 3D assets. Beijing Guowenyan Information Technology Co., Ltd. once entrusted Beijing Huachuang Tonsing Tech Co., Ltd. to complete the digital assets display system for the colour painted statues and murals of Shuanglin Temple (**Figure 1**) and realized the visualization of dense point clouds and photogrammetry mesh models through WebGL. Rendering of mesh models and point cloud models is different from BIM models. Meshes and point clouds are discrete, and the data volume of the model is particularly large, so a network rendering method similar to a map or satellite image, which is graded and divided into blocks and loaded in real time. A typical approach is to cut into blocks according to spatial relationships, such as octrees, which have been graded according to the density of the data. High-level LoD blocks perform data thinning. In this way, when the application of the client browser is loaded and rendered, the block and block level to be loaded can be calculated in real time according to the position of the camera during interaction, and then downloaded and rendered. Unreal Engine has a rich plug-in ecosystem and an active application market. Driven by commercial development, various plug-ins can also easily support various types of 3D assets such as dense point clouds and photogrammetric mesh models too.

2.3.5 Network, hardware, and software support requirements: WebGL needs to load the model and application data to the client, which requires high network bandwidth, but the network requirements can be reduced after the data is properly subdivided. While the UE bandwidth is very stable, which is determined according to the resolution of the output video streaming. On the other hand, Server requirements for WebGL are low, rendering relies on client GPU. The biggest limitation for the UE is the need for the server. The demand for cloud rendering services is high, and each server serves 1-3 cloud rendering clients. The cost of using the software can be said to be free for heritage conservation.

2.3.6 Data Security: Finally, from the perspective of data security, the WebGL solution faces challenges in multiple aspects, while there is no security risk for data in UE because it is based on pixel streaming.

2.4 Comparison Conclusion

Based on the above technical comparison and **Table 1**, the following conclusions can be drawn:

1. Creating a high-end 3D software that can only run on high-end hardware with limited access and small user base: develop it with Game Engine Pixel Streaming. If the application needs to be served to many thousands or even millions of users for free but does not require advanced features and visualisations: use WebGL.

2. Projects that are more sensitive to data security should choose Game Engine Pixel Streaming. Because it only loads the pixel stream to the client, while all the models and data are on the server side.

3. For applications with high requirements for attribute information management, such as heritage site management systems, WebGL will provide more flexible data integration and management functions.

Characteristics	WebGL	UE/UE Cloud
Visual Experience	Developers are required to have certain aesthetic skills to design visual effects.	Professional game rendering engine capable of providing AAA-level game visual effects.
Performance Experience	Fast, lightweight, no need to compile, simple 3D rendering, performance is better.	It is suitable for large and medium-sized scenarios, supports advanced interaction, and its performance is mainly related to network bandwidth.
	Open up the browser and graphics card GPU channel through OpenGL.	UE5 supports breakthrough technologies such as Nanite and Lumen and can directly generate SPIR-V without GLSL.
	Supports separate rendering of Canvas and DOM, which is helpful for clear logical structure and small performance improvement.	Based on pixel streaming, client operations are transmitted in real time, generally not layered rendering on the client side.
	There may be a long delay in scene loading time depending on the model and data volume loaded locally.	Scene load times are stable. Determined by the resolution of the loaded pixel stream and the response speed of the server, it is generally faster.
Data Integration and Management Capability	The 3D model and attribute data are separated and then matched, the data can be expanded and modified, and the data integration is flexible.	The attribute data is exported together with the model file, the ability to integrate and manage attribute information is weak.
	There are few scripting language encapsulation libraries and tool chains, and there is a certain component ecology.	Overall solution, complete tool chain, huge component group.
	GLSL belongs to the embedded syntax of JavaScript, lacks an advanced development and debugging environment, and has high requirements for developers.	A complete visual development environment, complete node-based development tools, and low requirements for developers' programming capabilities.

Interoperability and Flexibility of Use	No need to install software (just a modern browser), no plug-ins, directly embedded in most mainstream browsers, high flexibility.	Using pixel streaming, no software installation is required, and browser support ability is yet to be evaluated.
	Capable of serving large-scale global users.	Online users are limited by server capacity, and the ability to support a large number of public services is weak.
	Completely open source, its "openness" plays an important role in the flexibility of use.	Professional maintenance by commercial companies, while open source to a certain extent, with a stable development roadmap.
Support for Various 3D Assets	Practical cases confirm full support for different 3D asset types including dense, coloured point clouds (e.g. LAS, E57), large textured mesh models (e.g. OBJ, FBX, glTF), or building information models (e.g. IFC)	Based on the rich plugin ecology and active application market, it can support various types of 3D assets.
Network, Hardware and Software Support Requirements	Model and application data need to be loaded to the client, which requires high network bandwidth, but the network requirements can be reduced after the data is properly subdivided.	The requirement for network bandwidth is very stable, which is determined according to the resolution of the output video streaming.
	Server requirements are low, mainstream distributed service architecture is adopted, and rendering relies on client GPU.	The demand for cloud rendering services is high, and each server serves 1-3 cloud rendering clients.
	Free software.	Software is free for public welfare projects, free for game revenue below $1 million, and 5% for the portion over $1 million.
Data Security	Data security on multiple aspects such as browsers, shaders, and renderers face challenges.	Based on pixel streaming, there is no security risk for data.

Table 1. Comparison summary table

3. LIKELY FUTURE DEVELOPMENTS OF THE TWO APPROACHES

The most effective planning and design approach is an integrated one that combines heritage conservation with other planning and project goals and engages all partners and stakeholders early in the process and throughout (Parks Canada, 2010). The conservation of heritage buildings is inseparable from the participation of stakeholders. As building information models are more and more widely used in heritage conservation, to help the widest possible stakeholders participate in the heritage decision-making process, it is necessary to allow users to access and use BIM models more easily. Whether an interactive application based on WebGL or a Game Engine, helps to achieve this goal depends on several factors: 1. The characteristics of the heritage building itself, 2. The professional and technical background of the stakeholders, 3. The progress of the heritage affairs demands, 4. feasibility and cost of technical implementation. This essay mainly discusses the fourth factor. By comparing the workflow and technical characteristics of the CIMS Lab research projects, it summarizes the advantages and disadvantages of WebGL and Game Engine development approaches that we have experienced in person during the development process.

Some of these advantages and disadvantages are caused by the limitations of the current development of computer technology, such as: computing speed, network bandwidth, data volume, etc.; some are determined by the basic characteristics of different workflows, such as the flexibility of WebGL in non-geometric data processing when separate model and attribute data, and the data security safeguard brought by Pixel Streaming; while some are related to the professional and technical requirements of the participates, for example, WebGL requires developers to be familiar with computer languages such as JavaScript, while game engines often only require basic programming concepts to implement application functions through a visual development environment and node-based development tools. With the continuous development of emerging technologies in the future, it can be expected that some basic bottlenecks will gradually become less of a problem. As Carpo (2017) pointed out, data-compression technologies we don’t need anymore, no longer need algorithm simplification, and search don’t sort. The unbearable delay caused by computing speed and network bandwidth may be unbelievable in the future [9]. However, characteristics rooted in structural reasons will persist unless the working structure is changed. Issues related to people are much more complex, and it is difficult to predict how people will work and choose their expertise in the future.

However, through this comparison, we can see that WebGL and Game Engines are not on an equal level of comparison as 3D assets sharing solutions. WebGL is a lower-level general technology and standard than Game Engines, while Game Engines such as Unreal is a software product. Based on WebGL, it may be further developed into another Game Engine, and the application developed by the Game Engine may also be released as a tool based on WebGL to be used directly on the browser. That said, in the future, it is likely that multiple technologies will be mixed in an interactive application, rather than a single technical workflow. More importantly, the feasibility and cost of technical realization are only one factor when we consider the 3D interactive application of heritage buildings. The actual work also needs to take into account the characteristics of the heritage building itself, the professional and technical background of the stakeholders, and the progress of the heritage affairs demands too.

[9] Carpo, Mario. 2017. The Second Digital Turn: Design beyond Intelligence. Writing Architecture. Cambridge, Massachusetts: The MIT Press. P19

ACKNOWLEDGEMENTS

This work was supported by CIMS, a Carleton University research centre. The work has also been supported by the CDC project: Interactive "Digital" Campus Application'' (CDC-INCA) and New Paradigms/New Tools for Heritage Conservation in Canada. This project was funded in part by the Social Sciences and Humanities Research Council (SSHRC) of Canada.

REFERENCES

Achille, C., C. Tommasi, F. Rechichi, F. Fassi, and E. De Filippis, 2019: Towards an advanced conservation strategy: a structured database for sharing 3d documentation between expert users. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15:9–16. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-W15-9-2019.

Agustín, L., and M. Quintilla, 2019: Virtual reconstruction in bim technology and digital inventories of heritage. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15:25–31. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-W15-25-2019.

Angás, J., S. A. Jasim, M. Bea, M. Farjas, P. Uribe, P. Lanau, and X. Otero, 2019: Rock art recording in khatm al melaha (united arab emirates): multirange data scanning and web mapping technologies. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15:85–92. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-W15-85-2019.

Carpo, Mario. 2017: *The Second Digital Turn: Design beyond Intelligence*. Writing Architecture. Cambridge, Massachusetts: The MIT Press.

Giovannini, E. C., M. Lo Turco, and N. Mafrici, 2019: Documenting historical research for a collection information modelling. A proposal for a digital asset management system. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15:519–25. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-W15-519-2019.

Julin, Arttu, Kaisa Jaalama, Juho-Pekka Virtanen, Mikko Maksimainen, Matti Kurkela, Juha Hyyppä, and Hannu Hyyppä, 2019: Automated Multi-Sensor 3D Reconstruction for the Web. *ISPRS International Journal of Geo-Information* 8 (5): 221. https://doi.org/10.3390/ijgi8050221.

Julin, Arttu, Kaisa Jaalama, Juho-Pekka Virtanen, Matti Pouke, Johanna Ylipulli, Matti Vaaja, Juha Hyyppä, and Hannu Hyyppä, 2018: Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information* 7 (2): 55. https://doi.org/10.3390/ijgi7020055.

Kaimaris, D., T. Roustanis, K. Klimantakis, I. A. Karolos, and P. Patias, 2021: Possibilities of spatial correlation of 3d models in an archaeological augmented reality application. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-M-1–2021:355–59. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLVI-M-1-2021-355-2021.

Lanzara, E., S. Scandurra, C. Musella, D. Palomba, A. di Luggo, and D. Asprone, 2021: Documentation of structural damage and material decay phenomena in h-bim systems. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-M-1–2021:375–82. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLVI-M-1-2021-375-2021.

Letellier, Robin, Werner Schmid, and François LeBlanc, 2007: *Recording, Documentation, and Information Management for the Conservation of Heritage Places: Guiding Principles*. Los Angeles: Getty Conservation Institute.

Pybus, C., K. Graham, J. Doherty, N. Arellano, and S. Fai, 2019: New realities for canada's parliament: a workflow for preparing heritage bim for game engines and virtual reality. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2-W15:945–52. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLII-2-W15-945-2019.

Rechichi, F., A. Mandelli, C. Achille, and F. Fassi, 2016: Sharing high-resolution models and information on web: the web module of bim3dsg system. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B5 (June): 703–10. https://doi.org/10.5194/isprs-archives-XLI-B5-703-2016.

Rodriguez, A., J. F. Chuchiak, H. B. Erickson, and P. V. Ballicu, 2021: The great "auto de fe" at santiago de los caballeros, or how to achieve historical empathy with cultural heritage through virtual reality. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-M-1–2021:633–40. Copernicus GmbH. https://doi.org/10.5194/isprs-archives-XLVI-M-1-2021-633-2021.

Virtanen, Juho-Pekka, Sylvie Daniel, Tuomas Turppa, Lingli Zhu, Arttu Julin, Hannu Hyyppä, and Juha Hyyppä, 2020: Interactive Dense Point Clouds in a Game Engine. *ISPRS Journal of Photogrammetry and Remote Sensing* 163 (May): 375–89. https://doi.org/10.1016/j.isprsjprs.2020.03.007.