Dynamic 3D-Cartographic Symbols for VR Geovisualizations

Helge Olberding[1] · **Mark Vetter**[1]

Received: 14 September 2023 / Accepted: 9 November 2023 / Published online: 24 November 2023

Abstract

Geovisualizations that use a virtual reality headset can have a more immersive effect on users than when only a desktop application is viewed. However, to have an overview, a VR 3D map is usually viewed from a top-down perspective. On the other hand, the first-person perspective is used when the scenario is visualized as realistically as possible, and the ultimate goal is to feel present inside the 3D city. With the help of dynamic cartographic visualization methods, such as animated symbols appearing or a dynamic change of selected or complete visualization, it is possible to convey targeted information while being inside the 3D. For this purpose, techniques and ideas from cartography and video games were utilized and combined. The combination resulted in dynamically changeable symbol objects, which can be placed automatically using tables or building information, and dynamically changeable post-processing methods. Users can interactively select a predefined set of visualization methods and show/hide dynamic cartographic symbols in the running application via an interface. The study aims to develop cartographic visualization methods that can be technically used within a game engine for VR applications to improve spatial perception.

Keywords 3D · Cartographic symbols · Geovisualization · Interactive · NPR · Virtual reality

Dynamische 3D-Kartensymbole für VR-Geovisualisierungen

Zusammenfassung

Geovisualisierungen mit einem Virtual-Reality-Headset können auf die Nutzenden eine immersivere Wirkung haben als reine Desktop-Anwendungen. Um einen guten Überblick zu erlangen, wird eine VR-3D-Karte in der Regel aus einer Top-down-Perspektive betrachtet. In VR-Szenarien kann eine Ego-Perspektive verwendet werden, um sich direkt in die 3D-Stadt hineinzuversetzen. Dynamische kartographische Visualisierungsmethoden im 3D-Raum, wie z.B. animierte Symbole oder ein dynamischer Wechsel ausgewählter oder vollständiger Visualisierungen ermöglichen die gezielte Vermittlung von Rauminformationen. Zu diesem Zweck wurden Techniken und Ideen aus der Kartographie und aus Videospielen genutzt und kombiniert. Dynamisch veränderbaren Symbolobjekte, die anhand von Tabellen oder Gebäudeinformationen automatisch platziert werden können, wurden kombiniert mit dynamisch veränderbaren Nachbearbeitungsmethoden. Der Benutzer kann interaktiv einen vordefinierten Satz von Visualisierungsmethoden auswählen und über eine Schnittstelle dynamische kartographische Symbole in der laufenden Anwendung ein- und ausblenden. Die Studie zielt darauf ab, kartographische Visualisierungsmethoden zu entwickeln, die technisch innerhalb einer Game Engine für VR-Anwendungen eingesetzt werden können, um letztlich die räumliche Wahrnehmung zu verbessern.

1 Introduction

Virtual reality (VR) headsets and head-mounted displays (HMD) are part of a technology that has become increasingly important for the visualization of geodata in recent years (Edler and Kersten 2021). Video game developers have used cartographic representations for a long time. In

✉ Helge Olberding
helge.olberding@thws.de

Mark Vetter
mark.vetter@thws.de

[1] Geovisualization, Technical University of Applied Sciences Würzburg-Schweinfurt, Würzburg, Germany

recent years, cartography research has increasingly explored using map-like elements in video game contexts (Horbiński and Zagata 2022b; Edler and Dickmann 2017). Especially if one wants to develop cartographic representations with a game engine (Vetter 2020), implementation techniques can be adopted from video games (Toups et al. 2019) and computer vision (Semmo et al. 2015). In this case, people's general interest in topics through a gaming approach and the benefits of gamification could be used (Shaffer et al. 2005; Laamarti et al. 2014).

Spatial perception depends mainly on how relevant information is shaped and cognitively processed (MacEachren 1995). One primary goal of cartography is to structure and reduce geoinformation and visualize spatial relationships in a map. Information reduction can be achieved with the help of alternative or abstract visualization approaches. The functionalities of game engines can be used to implement this information reduction in an automated way, considering concepts of cartographic generalization (Schiewe 2022).

Several methods exist for map symbols that can modify the symbols, including position, size, shape, or color. These predefined modifications are called visual variables. The visual variables describe the graphical dimensions in which a map or other visualization can be varied to encode information. Each of these variables can have specific properties (Bertin 1982). These visual variables can be divided into twelve dimensions. These variables are location, size, shape, orientation, color hue, color value, texture, color saturation, arrangement, crispness, resolution, and transparency (Roth 2016).

There is a need to combine the new technologies of representation and interaction made possible by game engines and VR with the traditional concepts of cartography.

The opposite of photorealistic visualization for 3D visualizations is non-photorealistic rendering (NPR) (Gooch and Gooch 2001). These visualizations can also be referred to as abstract visualizations or stylized rendering. An NPR is implemented by modifying the rendering and the textures and using filters originating in image processing. A non-photorealistic rendering uses a variety of renderings to either reduce complexity in the 3D visualization to mimic an art style or to highlight selected objects (Akenine-Möller et al. 2018). In general, photorealistic representations are not used in thematic cartographic mapping. Highlighting methods are needed for 3D objects to be selected, placed, moved, or displayed potentially. In doing so, the highlighting methods can vary in their presentation for cartographic purposes(Trapp et al. 2011; Semmo and Döllner 2014). NPR can use filters to recreate the esthetics of thematic maps in a 3D space (Semmo et al. 2012) and can visually abstract complex spatial information, such as 3D city models (Jobst et al. 2008). The focus of the use of NPR in research is strongly represented in computer vision, which takes up cartographic representation methods. NPR has not yet played a role in most cartographic applications, especially VR. Therefore, investigating the possible realization and potential outlook of possible applications is interesting.

Since 2016, VR headsets have entered the consumer market, making them accessible to a broad mass, including the first consumer variants of the Oculus Rift and HTC Vive. VR applications are often developed with the help of a game engine. The Unreal Engine has been available free of charge since 2015. Due to the more accessible game engines and VR headsets, more geovisualizations use them. The Unreal Engine 5 (UE5) is next to Unity, the most widely used game engine (Vohera et al. 2021). UE5 stands out due to the various possibilities to create interactive large virtual worlds.

Game engines were first developed for video games. Games use maps or map-like representations in different ways and all the time. From this, what is possible for cartography applications with a game engine can be deduced. Many games use fictional spaces with reference to real topographic spaces. Games combine non-linear visual, audio, and interactive elements to represent spatial issues. In addition, the presentation of animated fictional dynamic maps in video games is usually targeted at a younger audience (Edler and Dickmann 2016). Various perspectives in video games can be divided into map categories, for example, overview maps or mini-maps (Edler et al. 2018). More and more people, especially young people can handle video games. Consequently, cartographic applications from a game engine can build on the prior knowledge of young users.

Map symbols are used in video games to help gamers understand the spatial situation more quickly (Horbiński and Zagata 2022a, 2021). However, there are also various approaches to the design and interaction of a mapping interface (Toups et al. 2019). Different developments depend on technological and graphical possibilities (Horbiński and Zagata 2022b).

A wide variety of approaches to VR cartography have already been implemented based on a diverse range of geospatial data, e.g., Open Geospatial Data (Keil et al. 2021). This includes VR geovisualizations with the use of minimaps (Zagata et al. 2021) and the use of a cartographic scale (Hruby et al. 2021). Furthermore, combinations of cartography and elements of comic strips in VR (Moore et al. 2018) or embodied gesture interaction for immersive maps were implemented (Newbury et al. 2021). Similarly, there is increasing research on how VR cartography differs from desktop cartography regarding readability (Dong et al. 2020).

Data and display methods may be better suited for VR, depending on the case or goal. Potential difficulties of conventional 2D desktop visualizations and physical limitations can be overcome (Lochhead and Hedley 2021).

The objectives of the study are to establish a basis for testing the technical and graphic possibilities of different cartographic design methods in VR:

- The implementation of different visualization approaches to display cartographic signatures in terms of animations, perspective, and three-dimensionality.
- It will explore to what extent interactive NPR variants can be implemented, activated dynamically, and combined with realistic rendering methods.
- Realization of tools that allow the adjustment of map symbols and rendering these manually or based on input data.
- An exemplary realization of all techniques based on two application scenarios.
- Analysis of which signatures can be used for cartographic applications in VR and which visual properties must be considered from the first-person perspective.
- Investigate whether the visualization results yield the potential to explore follow-up research questions.

2 Data and Methods

City models and geodata result in large data. UE5 can display several thousand assets simultaneously as an engine, which is ideal for rendering many building models and signature assets. Every symbol that is in the 3D scene is an asset. Even though the Unreal Engine can display many assets, in any optimized visualization, assets should be grouped, hidden, or replaced with lower-detailed assets to ensure smooth rendering. A game engine or game development environment is a software framework that enables the development of applications that require real-time visualization. A wide variety of components are used for this purpose. In addition to real-time visualizations, it is also possible to render videos and individual images (Trenholme and Smith 2008). Within the software framework, it is possible to use different tools or systems for interactions, animations, audio, and artificial intelligence (Gregory 2019). In addition, there is a wide variation of rendering options, e.g., lighting, depth effects, and edge smoothing (Akenine-Möller et al. 2018).

Camera models simulate the properties of a photographic camera so that 3D objects can be displayed on two-dimensional surfaces. In addition, other optical and physical camera properties are also considered. The first-person camera perspective was chosen for this project for the VR application. First-person means that the camera represents the perspective of the controlling avatar, and the avatar is in the 3D world. The camera is bound to the user. Consequently, the camera's movement depends on the avatar's movement possibilities. The perspective is mainly used with direct control (Helbing 2004). First-person has a great immersion for the user (Denisova and Cairns 2015).

Alternative ways of viewing a scene within the virtual space can be outside of a fixed personality, the avatar's perspective in the virtual world. In addition to a ground-level personalized view, there is the bird's eye view alternative called god view or top-down perspective (Taylor 2002). It allows a more extensive overview as if the user had a 3D miniature landscape in front of the user (Lochhead and Hedley 2021). A combination allows simultaneous multiple perspectives (Wang et al. 2022).

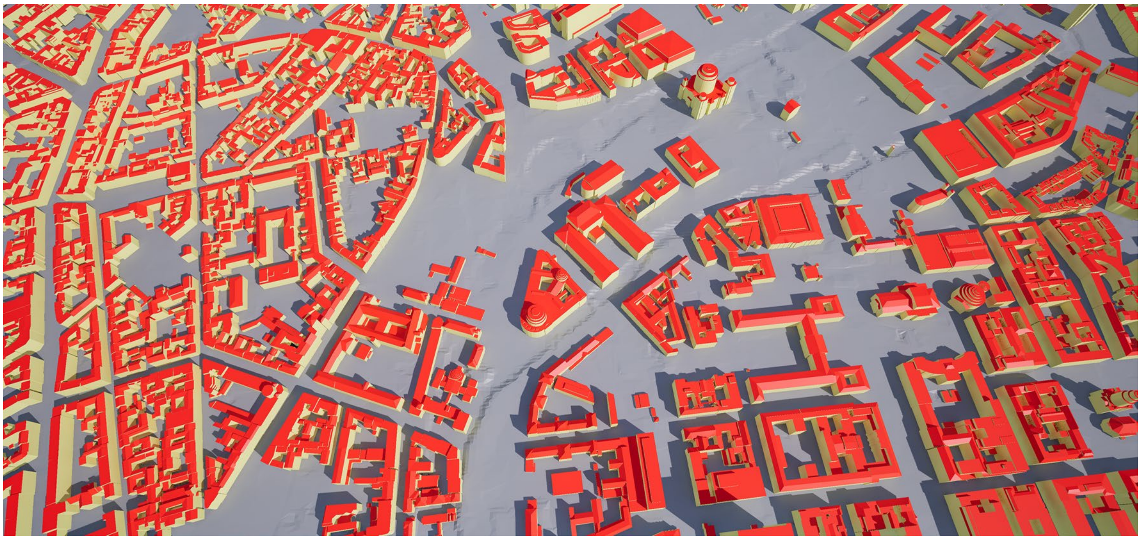

Fig. 1 A 3D city model of Berlin in CityGML format, in LoD2 resolution level

From the first-person perspective, the city is a 1:1 scale. Accordingly, suitable methods for visibility and navigation must be developed for the application.

The UE5 enables a form of visual scripting with the help of blueprints, which allows visual programming. Within the investigation, only Blueprints were used. This results in a wide variety of Blueprint Actors. Blueprint Actors are assets for the 3D scene, which can contain interactive elements. These Blueprint Actors can be used for symbols and interactions.

The applications were optimized for the VR headset HTC VIVE Pro 2. Here, the VR headset was constantly connected to the computer. The UE5 automatically recognizes these VR headsets and, thus, enables easy implementation of the motion controller interactivity.

As a basis for the 3D scenes, 3D city models in the form of the data type CityGML were used. CityGML data cannot be imported regularly into UE5. The plugin was developed with the University of Würzburg in a student project. For this purpose, the geo competence from THWS was combined with the game engine experience from the University of Würzburg (Zang and Schüßler 2022). The plugin allows thousands of LoD2 buildings to be imported into the 3D scene via drag-and-drop. The geometries and the attributes remain the same (Fig. 1).

Besides the CityGML data, additional assets from the UE5 marketplace were used, including various 3D models and blueprints. All datasets are based on the freely available data of Berlin. This dataset includes CityGML models, the digital terrain model (DTM), and additional metadata such as pollution levels.

The symbols used are based on map symbols. Map symbols are graphic signs that express the object's location, type, and qualitative and quantitative characteristics through a symbolic or iconic representation. The same map symbol shapes (circle, square, triangle, etc.) are repeatedly linked to other terms from the map to map. The initial focus was primarily on point-based classification.

Rather than exchanging different display methods, animations can be implemented in several ways. One visual possibility of interactive VR cartography is using a graphical user interface (GUI). In a GUI, 2D elements are integrated into the 3D visualizations. For this purpose, there are head-up displays (HUD) in Unreal Engine 5. A HUD provides additional information, e.g., displays, bars, texts, and images. Within the application, the HUD displays information about the scenario, coordinates information of the user, fixed teleportation points, and information about active highlighting techniques and can change the complete scenario.

3 Visualization Results

For this study, we created a cartographic VR application with different display variations depending mainly on two topics. The two topics are lighting simulation and CO_2 pollution simulation. Different representations and animation techniques were used for both topics.

3.1 Cartographic Point and Line Symbols

Three technical variants were created for point-based classification: 2D-, 3D-, and particle Symbols. The presence and location of symbols are based on attribute tables or specified information from CityGML buildings. It is possible to change the shape and color of symbols dynamically.

To transfer 2D symbols into 3D, the symbols can be sculpted as 3D models, e.g., markers, arrows, or objects. The advantage of these symbols is that they can be integrated into a 3D scene depending on the level of detail and texturing. However, creating a new 3D model can be time-consuming, depending on its complexity. At the same time, objects that are not equally well visible from all viewing angles should be aligned with the user. If 3D models are constantly aligned to the user, the use of complex asymmetric 3D symbols is questionable.

An alternative would be the use of 2D symbols in a 3D space. This approach can already be found in the origins of 3D video games (Arsenault 2009). For this, 2D symbols are used as sprites. A sprite is a graphic object viewed from one side in a 3D space. When this sprite is oriented toward the user, the 2D symbol can be seen equally well from all perspectives.

A completely different approach is the use of a particle symbol. A particle rendering system (particle system) handles the visualization of atmospheric effects. A unique feature of a particle system is that it is not one object but rather many simple graphical objects (particles) representing sparks, smoke, fire, water, etc., as particle effects. Not all representations generally have definable, smooth, and unbounded surfaces. Due to its unique properties, a special rendering system is implemented in the game engine. On the one hand, particle systems are characterized by their graphical objects, usually simple two-dimensional objects composed of two triangles. These are also called a quad. Particles, on the other hand, can be animated in different ways. Parameters can vary from image to image. This includes position, orientation, size, and texture coordinates. Animations from single particles are primarily procedural. For this purpose, different parameter ranges are defined, for example, the weighting, distribution, quantity, and velocity of a single particle. Keypoints can also manually adjust individual elements in a timeline (Akenine-Möller et al. 2018).

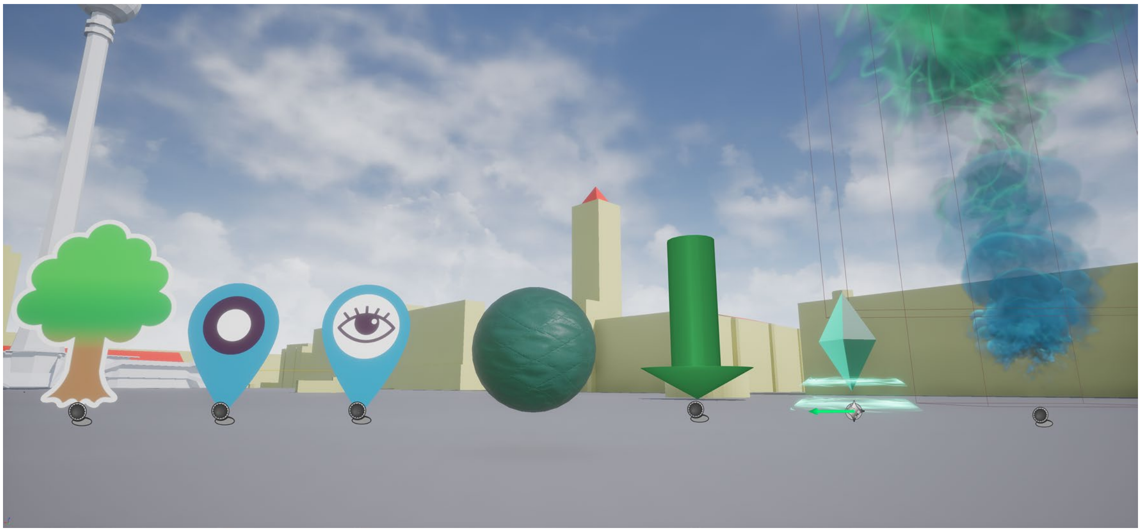

Fig. 2 From left to right: an exemplary representation of three 2D, two 3D, and two particle point symbols

Furthermore, individual particles are mainly generated and removed sequentially. Emitters allow particles to be spawned in the system at a user-defined rate. Particles are removed if they leave the defined region or their period expires (Gregory 2019). With the help of a particle system, 3D models can also be combined, or any 3D shapes can be mapped by arranging the particles (Fig. 4.). The cascade particle and Niagara particle systems are within the Unreal Engine. There are two particle systems within the Unreal Engine, Niagara and Cascade. Niagara is the newer system, allowing more variables to change and creating more complex effects. Cascade is reduced in its application possibilities, and functionalities are not transferable 1 to 1 in both particle systems (Epic Games, Inc. 2020). Other unique particle systems act as additional plugins for the UE5, like "Fluid Flux" by (Komisarek 2021). The plugin allows the creation of real-time dynamic water simulations based on Shallow Water Equations, an algorithm developed by Chentanez and Mueller (2010). The Niagara Particle System was used for the application's particle symbols.

The advantage of a 2D symbol is the direct transfer of a signature from a 2D map into a 3D space. On the other hand, the advantage of a 3D symbol is the possibility of more direct customization of the coloring through textures or materials. Unlike the other two, the particle's most significant benefit is allowing complex additional animations.

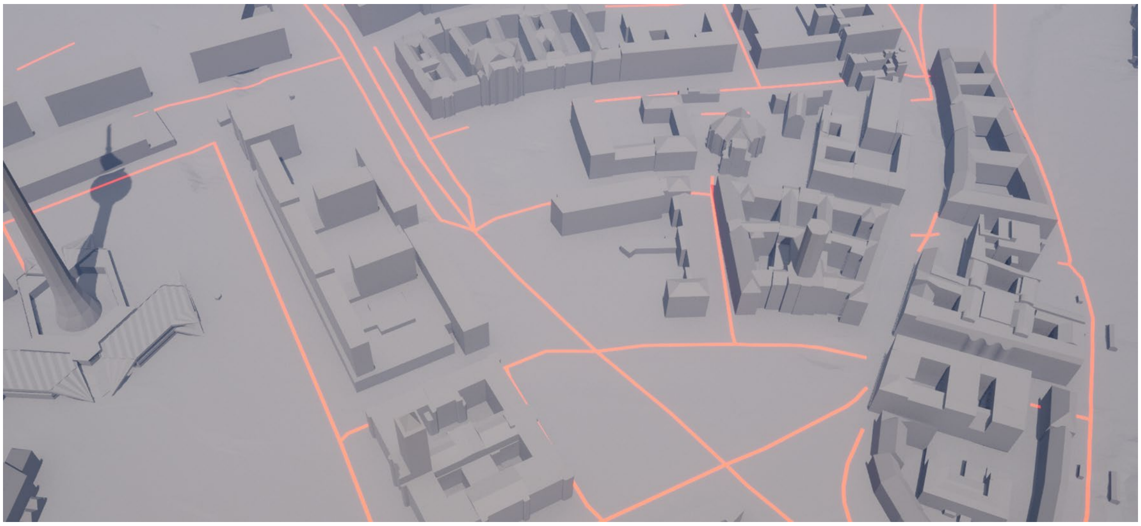

Fig. 3 The linear symbol is an example of a road network using materials on the surface

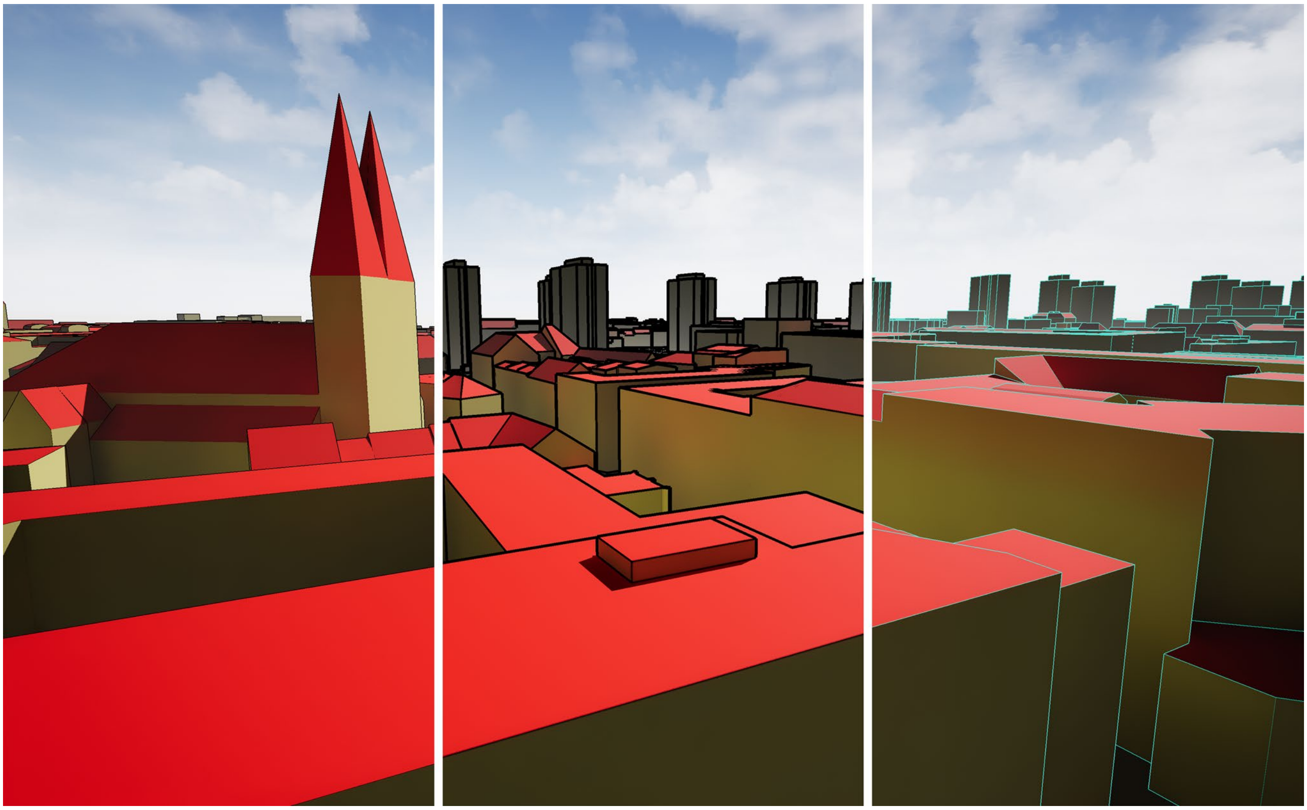

Fig. 4 Different parameters adjustments of the same outline material, left: line thickness = 1, middle: line thickness = 2, right: line color is light blue

In the implementation, the three variants are each in a Blueprint Actor. Blueprint Actors have the unique feature that they can be adapted dynamically. For this purpose, Blueprint Actors are identical in the form of use, although different graphical elements can be used. These Blueprint Actors can then be linked to an attribute table so that dynamic adaptation can occur depending on usage activities (dynamic map symbol creation). With the help of the Blueprint Actor, all three variants can be combined into one symbol asset. The Fig. 2 shows various examples, including location markers for trees, landmarks, accidents, selected stores, and authorities. Depending on the various attributes applied, these icons change their size, color, texture, or animation speed. Single, significantly larger icons and or faster-moving icons draw more of the viewer's attention.

In addition to point visualizations, line and area visualizations are also part of the cartographic representation methods. For this, however, additional either boxes or splines are required. Customized Blueprint Actors can then be linked to these. For example, displaying a line with materials, 3D splines, and particles is possible.

A way to avoid placing more assets in the 3D scene is to use dynamic material changes. Each symbol is an asset, and the more assets are displayed in the camera section, the more expensive the rendering is. The more expensive the rendering is, the more likely the frames per second will decrease. Overlaying textures onto existing textures is an alternative. This feature is called tagging, spraying, or stickering in video games. To do this, users are given a selection of textures and can use a button to place the graphics on selected 3D models. For example, this procedure can "draw" road networks onto the DTM (Fig. 3).

Different symbols were created for each topic, displayed based on previously defined metadata. This includes highlighting relevant buildings or streets in both applications. For the lighting simulation, different signatures were placed for important regions. Different particle effects in the form of smoke or colored spheres were used to represent CO_2 intensity. Linear and point signatures were used for pollution visualization in 3D cities.

3.2 NPR for Thematic-Orientated Cartographic Representations

With the help of rendering techniques such as post-processing, it is also possible to highlight or change selected areas or the entire 3D scene. It allows the entire scene representation to be changed in the last rendering step. In the UE5,

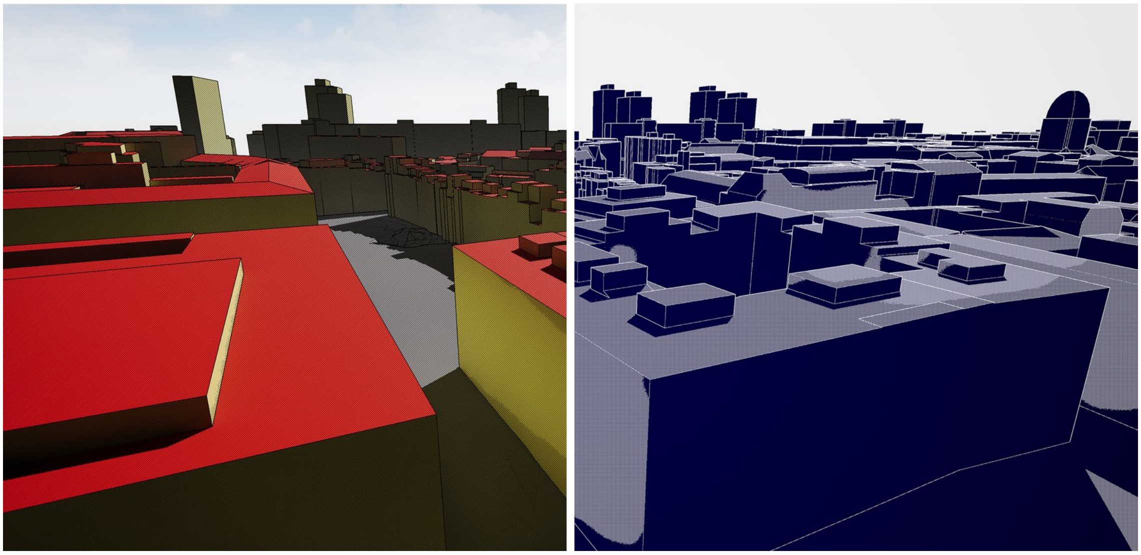

Fig. 5 Two different NPR effects are applied with the help of various materials for a 3D city, with different rasterization, color values, and shading

various visualizations and effects can be implemented for this cartographic visualization. The results are, by default, only applied when the camera is within a “PostProcessVolume”. The limitation of the “PostProcessVolume” can be solved in the settings, and a “PostProcessVolume” can influence everything without everything contained in the volume. Using the “PostProcessVolume,” the most diverse parameters can be adapted. These include simulated effects of physical camera properties such as depth of field and lens reflections, edge smoothing, or color correction is possible (Epic Games, Inc. 2022).

Individual post-processing materials can implement additional effects. Mapping methods can be used here, as can texturing individual objects. This all occurs in UE5 as materials. An example in this context is a black silhouette that serves as an additional outline for 3D models. Outlines can help to distinguish 3D objects more clearly from each other. For the creation of an outline material, e.g., for the detection of geometric edges and borders can implement different filtering methods in the material. Here, depending on the implemented material, the representation can differ significantly. Lines can be thicker and in different colors (Fig. 4) or include more complex methods to emulate different brushes in their texture and raster (Fig. 5).

The SceneDepth function creates an Outline material, e.g., for detecting geometric edges and borders. Depending on the camera's scene position, the function allows the display of distances between individual objects with color information. The required distance at which an edge is displayed can be specified by implementing a filter. Only two color values can be the output by rounding to 0 or 1. All values below the required distance are displayed in white; otherwise, they are black. All further scene information must be integrated with the used material and then united with the black lines. By the edge emphasis, the result shows a more abstract drawing representation in which adjacent buildings are more easily distinguishable.

The advantage of visually changing the scene through post-processing settings and materials is, on the one hand, the variety of different methods of displaying the same scene, independent of any 3D object present. In addition, post-processing effects can be easily implemented in interactions.

In addition to the outlines, shadows can be made simpler, and unique textures can be used over the entire scene or selected depths. This makes a wide variety of NPR display options possible. In UE5, the SceneDepth expression can be enabled for each mesh. SceneDepth outputs the existing scene depth for the respective mesh. SceneDepth can sample the depth at any position, even if, e.g., other objects are in front of the mesh. Using the SceneDepth function and adjusting the post-processing materials, it is possible to apply filters only to specific 3D models. This makes it possible to visualize effects such as a variant of an X-ray view. Outline visualization is ideal for this purpose.

Using multiple post-process volumes or post-process materials simultaneously is also possible. This allows the combination of different reduction or highlighting approaches. By unlocking parameters in the PostProcessMaterial, the line thickness or color can be adjusted more specifically. Furthermore, a PostProcessVolume does not have to be directly located in the 3D scene but can also be contained in a blueprint actor. With all these possibilities,

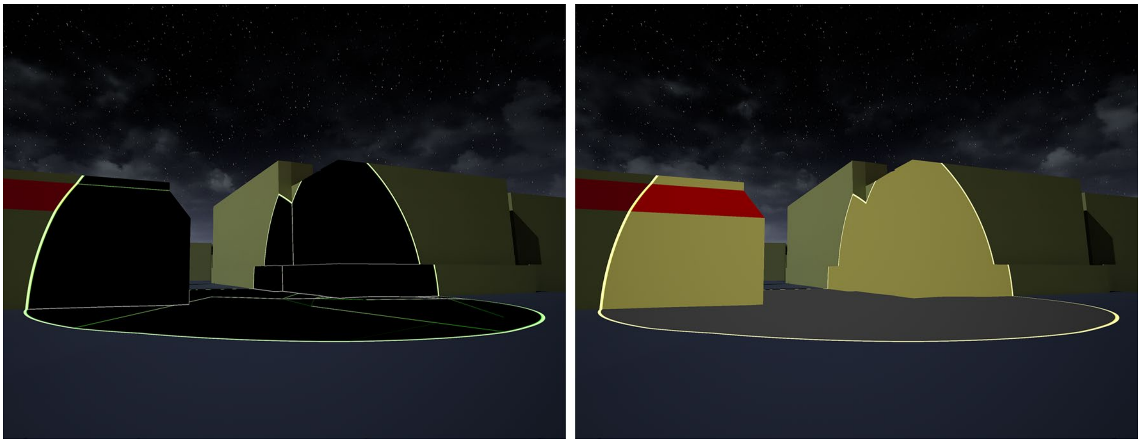

Fig. 6 Two different NPR materials, each affecting only a selected zone in the 3D environment with PostProcessing

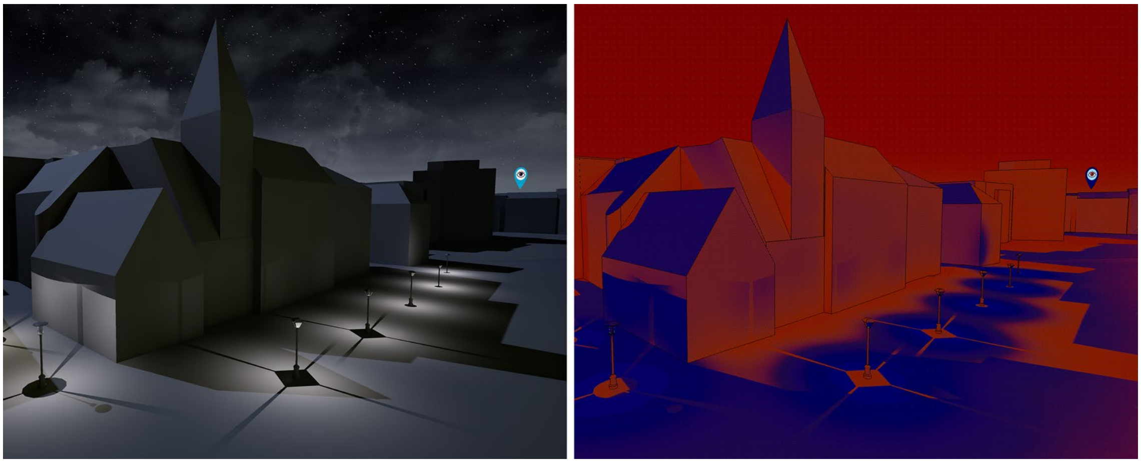

Fig. 7 3D lighting map example. Left is a night visualization with realistic lighting. On the right is an alternative visualization method in which the red areas represent the dark areas

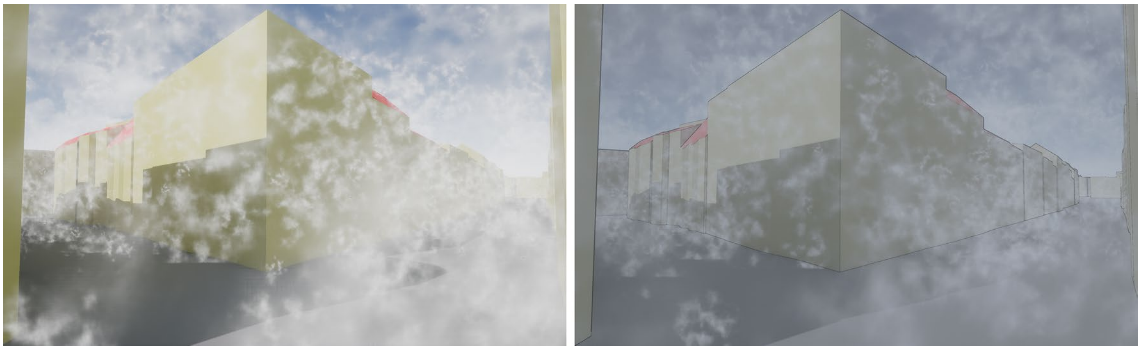

Fig. 8 3D pollutant map example. In both pictures, "Smok" particle systems run along the road. In the right image, one filter highlights buildings, and when the user is inside the particle systems, another filter grays out the scene

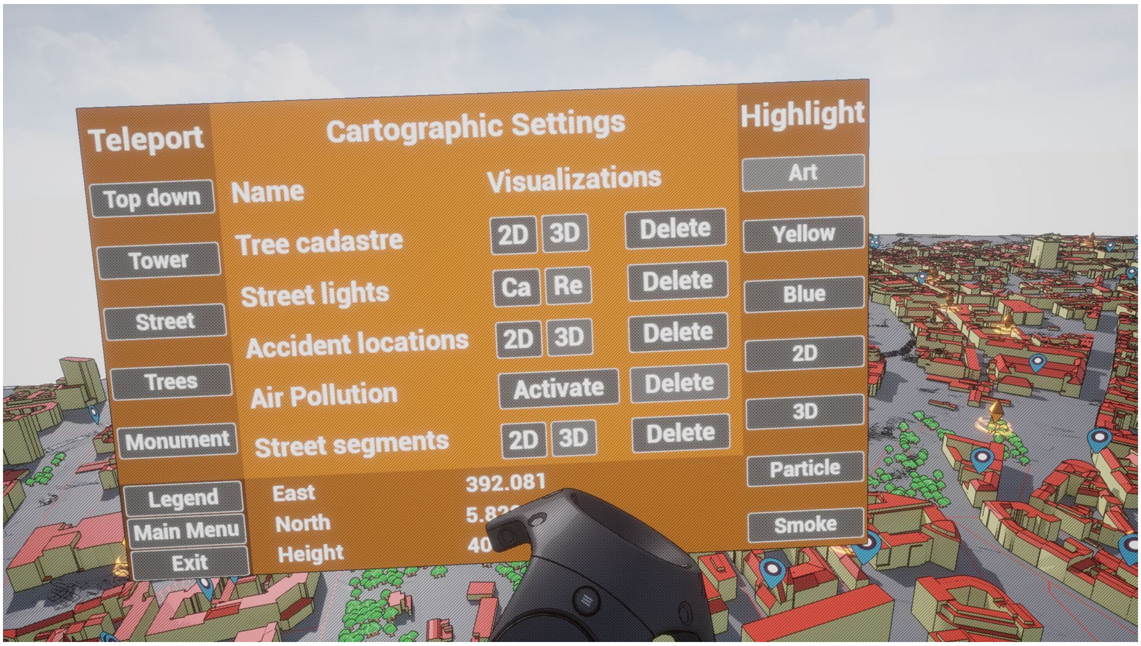

Fig. 9 An activated HUD with motion controls to change the visualization of the 3D scene

interactions or dynamic changes of the whole 3D scene can be displayed.

In addition, it is possible that an NPR effect is not displayed in the whole 3D scene but only in a section of the image or defined area. Thus, in addition to implementing a map-like representation and the targeted reduction or highlighting of certain information, NPR materials can additionally depict area signatures in a 3D space (Fig. 6).

For the lighting simulation, a night visualization can be realized with the help of the adjustment of the light and the sky box. With the PostProcess Volume, the brightness values can be changed to blue and red instead of white and black. In addition, an NPR effect can be used to reduce the gradation steps from light to dark. The combination can highlight potential darker areas. By making it easier to identify dark sections, the lighting planner has a visualization tool and can then test actions with light sources that are implemented in the virtual world (Fig. 7). Besides using particle systems such as smoke, the 3D scene becomes more grayish near a higher CO_2 content. Also, a textured surface was given to each surface. For the spheres, SceneDepth was used so that they are always detectable by buildings with an outline. The grayer the scene, the higher the CO_2 load in the air, allowing the user to experience the problem virtually (Fig. 8.).

3.3 Dynamic Objects as Cartographic Representations

When using a VR headset, seeing HUD elements in the peripheral areas is also more complicated. Unlike a desktop application, with an HMD, the head cannot orient itself differently to view edge elements on the VR headset lenses. There are two solutions. The first is to move the HUD elements more toward the center. The alternative is to create a HUD element as an object in the 3D scene. Implementing HUD elements as objects makes it possible to enable interactions with the motion controller. Also, it creates a feeling that the HUD exists in the virtual world (Fig. 9).

A selection of symbols, NPR visualization methods, additional points of view, and level selection options exist on the HUD. Point of view allows users to teleport to relevant or relevant fixed coordinates. Moreover, various cities and scenarios can be selected in an application with multiple levels. Otherwise, a person can use movements or short-range teleportation to change the viewed section.

UE5 has various possibilities to animate symbols after they have appeared. Animations or movements of the symbols can create additional attention. 3D symbols can be rotated or morphed. This can be used, for example, to represent non-symmetrical 3D models. With 2D symbols, a sprite animation can be created using several sprites. A sequence of the different sprites can simulate movement. Particle symbols are basically in motion due to their potentially large number of graphic objects. In the case of particle systems, this makes it possible to emphasize specific points in a more pronounced and attention-grabbing way.

4 Conclusion

In a game engine like Unreal Engine 5, it is possible to use realistic and non-realistic visualizations and combine them. Video games have always used cartographic signatures in various ways, so cartography can use the video game engine to create interactive 3D maps. Cartographic signatures in VR can then be used for this purpose. In addition to the classic graphical variables based on Bertin (Ellsiepen and Morgenstern 2007), other graphical effects can be achieved for this purpose, which can be used for quantitative or qualitative data representations as geovisualization.

NPR can massively visually change the whole or parts of the 3D scene, depending on the implemented filters and parameter settings. With NPR, it is possible to see through buildings, swap colors, change line thicknesses, create blurs, change color saturation, or add hatching and different shaders. NPR materials allow many new rendering variations without affecting the geometry of the 3D scene. Thus, once elaborated, NPR materials can be transferred to any subsequent project in the same game engine. However, an NPR visualization can also quickly degenerate into a gimmick if, for example, too many are activated, or a 3D scene becomes more unrecognizable due to the materials. Therefore, the form of the NPR should be well thought out beforehand.

2D, 3D, or particle styles for a signature can be exchanged or changed at any time and are based on the same blueprint edition. Therefore, a simultaneous combination of styles is also conceivable, e.g., a 3D object with additional particle effects. Complex dynamic materials, different animation techniques, and particle effects combined with VR headsets could allow more meaningful options to direct the user's gaze when examined more closely. In addition, it is possible to change the display variant multiple times in a running application using only one HUD.

For the representation of pollutants, particles have lent themselves to point and linear symbols. In the case of an illumination simulation, individual points are less to be emphasized. However, the 3D scene is viewed more as a surface, and here, on the other hand, post-processing has a more significant influence. Thus, both representations are scenarios that profit enormously differently from the symbols and the NPR. Further application scenarios can be implemented with adjustments to the parameters for the NPR material. The technical basis of the signature objects can be adopted, but new visual assets must be created.

In a VR application, it is always important to consider the different perspectives available in first-person VR applications. Viewing a city model from a first-person perspective differs from most geovisualizations on a desktop screen (mostly top-down). Cartographic signatures combined with animations can stand out in a 3D scene, especially the more realistic the rest of the environment is implemented.

The question is, in which form can these representation methods be qualitatively tested? Following this, specific thematic VR visualizations benefit more from cartographic effects and symbolism than others. Furthermore, does an additional cartographic visualization to the realistic representation significantly add value?

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The CityGML The plugin for UE5 is available at https://maxhunter007.itch.io/citygml-import-plugin.

Declarations

Conflict of Interest The authors report no potential conflicts of interest.

References

Akenine-Möller T, Haines E, Hoffman N (2018) Real-time rendering, fourth edition, 4th edn. Chapman and Hall/CRC, Milton

Arsenault D (2009) Video game genre. Evolut Innov EJCGC 3(2):149–176. https://doi.org/10.7557/23.6003

Bertin J (1982) Graphische Darstellungen und die graphische Weiterverarbeitung der Information. De Gruyter, Berlin

Chentanez N, Mueller M (2010) Real-time simulation of large bodies of water with small scale details

Denisova A, Cairns P (2015) First person vs. third person perspective in digital games. In: Bo Begole JK, Kori I, Woontack W (eds) CHI 2015 crossings. CHI 2015; proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems; April 18–23, 2015. Association for Computing Machinery; Annual CHI Conference on Human Factors in Computing Systems; CHI, Seoul, Republic of Korea. ACM, New York, pp. 145–148

Dong W, Yang T, Liao H, Meng L (2020) How does map use differ in virtual reality and desktop-based environments? Int J Digit Earth 13(12):1484–1503. https://doi.org/10.1080/17538947.2020.1731617

Edler D, Dickmann F (2016) Interaktive Multimediakartographie in frühen Videospielwelten—das beispiel "super mario world." In J Cartogr Geogr Inf 66(2):51–58. https://doi.org/10.1007/BF03545205

Edler D, Dickmann F (2017) The impact of 1980s and 1990s video games on multimedia cartography. Cartographica 52(2):168–177. https://doi.org/10.3138/cart.52.2.3823

Edler D, Kersten TP (2021) Virtual and augmented reality in spatial visualization. J Cartogr Geogr Inf 71(4):221–222. https://doi.org/10.1007/s42489-021-00094-z

Edler D, Keil J, Dickmann F (2018) Varianten interaktiver Karten in Video- und Computerspielen—eine Übersicht. KN (Kartographische Nachrichten J Cartograph Geograph Inform 68(2): 57–65. (Checked on 8/12/2019)

Ellsiepen I, Morgenstern D (2007) Der einsatz des bildschirms erweitert die kartographischen gestaltungsmittel. J Cartogr Geogr Inf 57(6):303–309. https://doi.org/10.1007/BF03544040

Epic Games Inc (2020) Unreal engine. In: Niagara overview. Available at https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Niagara/Overview/. Updated on 2020, checked on 10/16/2023

Epic Games Inc (2022) Unreal engine. In: Post process effects. Available at https://docs.unrealengine.com/5.1/en-US/post-process-effects-in-unreal-engine/. Updated on 2022, checked on 1/15/2023

Gooch B, Gooch A (2001) Non-photorealistic rendering. A K Peters (ITpro collection), Natick

Gregory J (2019) Game engine architecture. Third edition. In: An A.K. Peters book. CRC Press Taylor & Francis Group, Boca Raton, London, New York

Helbing R (2004) Ein erweitertes kameramodell methoden und werkzeuge für die dynamische kamerasteuerung in interaktiven systemen. In: Dissertation. Otto-von-Guericke-Universität Magdeburg, Magdeburg. Fakultät für Informatik. Available at https://pdfs.semanticscholar.org/d880/85c252ed14ccfdbd53111a9dad8f60c04393.pdf. Checked on 22.06.17

Horbiński T, Zagata K (2021) Map symbols in video games: the example of "valheim." J Cartogr Geogr Inf 71(4):269–283. https://doi.org/10.1007/s42489-021-00085-0

Horbiński T, Zagata K (2022a) League of legends—interpretation of spatial situations. Abstr Int Cartogr Assoc 5:1. https://doi.org/10.5194/ica-abs-5-106-2022

Horbiński T, Zagata K (2022b) View of cartography in video games: literature review and examples of specific solutions. J Cartogr Geogr Inf 72(2):117–128. https://doi.org/10.1007/s42489-022-00104-8

Hruby F, Castellanos I, Ressl R (2021) Cartographic scale in immersive virtual environments. J Cartogr Geogr Inf 71(1):45–51. https://doi.org/10.1007/s42489-020-00055-y

Jobst M, Kyprianidis JE, Döllner J (2008) Mechanisms on graphical core variables in the design of cartographic 3d city presentations. In: Antoni M, Igor D (eds) Geospatial vision (Lecture Notes in Geoinformation and Cartography). Springer Berlin Heidelberg, pp 45–59

Keil J, Edler D, Schmitt T, Dickmann F (2021) Creating immersive virtual environments based on open geospatial data and game engines. J Cartogr Geogr Inf 71(1):53–65. https://doi.org/10.1007/s42489-020-00069-6

Komisarek K (2021) Fluid flux. Documentation. Available at http://imaginaryblend.com/2021/09/26/fluid-flux/. Updated on 9/26/2021

Laamarti F, Eid M, El Saddik A (2014) An overview of serious games. Int J Comput Games Technol 2014:1–15. https://doi.org/10.1155/2014/358152

Lochhead I, Hedley N (2021) Designing virtual spaces for immersive visual analytics. J Cartogr Geogr Inf 71(4):223–240. https://doi.org/10.1007/s42489-021-00087-y

MacEachren AM (1995) How maps work. Representation, visualization, and design. The Guilford Press, New York, London

Moore A, Nowostawski M, Frantz C, Hulbe C (2018) Comic strip narratives in time geography. IJGI 7(7):245. https://doi.org/10.3390/ijgi7070245

Newbury R, Satriadi KA, Bolton J, Liu J, Cordeil M, Prouzeau A, Jenny B (2021) Embodied gesture interaction for immersive maps. Cartograph Geograph Inform Sci 48(5):417–431. https://doi.org/10.1080/15230406.2021.1929492

Roth RE (2016) Visual Variables. In: Richardson D, Castree N, Goodchild MF, Kobayashi A, Liu W, Marston RA (eds) International encyclopedia of geography: people, the earth, environment and technology. Wiley, Oxford, pp 1–11

Schiewe J (2022) Kartographie. Visualisierung georäumlicher Daten. Springer Spektrum (Lehrbuch), Berlin

Semmo A, Döllner J (2014) An interaction framework for level-of-abstraction visualization of 3D geovirtual environments. Hasso Plattner Institute, Germany. In: Falko S, Chris K, Holger F (eds) MapInteract '14. Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Interacting with Maps. Association for Computing Machinery, Inc, New York, pp 43–49. Available at http://dl.acm.org/citation.cfm?id=2677072&CFID=829364891&CFTOKEN=88586288

Semmo A, Trapp M, Kyprianidis JE, Döllner J (2012) Interactive visualization of generalized virtual 3d city models using level-of-abstraction transitions. Comput Graph Forum 31:885–894. https://doi.org/10.1111/j.1467-8659.2012.03081.x

Semmo A, Trapp M, Jobst M, Döllner J (2015) Cartography-oriented design of 3d geospatial information visualization—overview and techniques. Cartograph J 52(2):95–106. https://doi.org/10.1080/00087041.2015.1119462

Shaffer DW, Squire KR, Halverson R, Gee JP (2005) Video Games and the Future of Learning. Phi Delta Kappan 87(2):105–111. https://doi.org/10.1177/003172170508700205

Taylor LN (2002) Video games: perspective, point-of-view, and immersion. Masterthesis. University of Florida, Florida. Available at http://etd.fcla.edu/UF/UFE1000166/taylor_l.pdf. Checked on 1/20/2017

Toups ZO, Lalone N, Alharthi SA, Sharma HN, Webb AM (2019) Making maps available for play. ACM Trans Comput Hum Interact 26(5):1–43. https://doi.org/10.1145/3336144

Trapp M, Beesk C, Pasewaldt S, Döllner J (2011) Interactive rendering techniques for highlighting in 3d geovirtual environments. In: Thomas HK, Gerhard K, Claus N (eds) Advances in 3D geoinformation sciences (Lecture Notes in Geoinformation and Cartography). Springer-Verlag, Berlin Heidelberg, pp. 197–210

Trenholme D, Smith SP (2008) Computer game engines for developing first-person virtual environments. Virtual Rea 12(3):181–187. https://doi.org/10.1007/s10055-008-0092-z

Vetter M (2020) Technical potentials for the visualization in virtual reality. In: Dennis E, Corinna J, Olaf K (eds) Modern approaches to the visualization of landscapes. Springer, Wiesbaden, pp 307–317. https://doi.org/10.1007/978-3-658-30956-5_17

Vohera C, Chheda H, Chouhan D, Desai A, Jain V (2021) Game engine architecture and comparative study of different game engines. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). Kharagpur, India, 7/6/2021–7/8/2021. IEEE, Piscataway, pp. 1–6

Wang Yu, Hu Z, Li P, Yao S, Liu H (2022) Multiple perspectives integration for virtual reality-aided assemblability assessment in narrow assembly spaces. Int J Adv Manuf Technol 119(3–4):2495–2508. https://doi.org/10.1007/s00170-021-08292-9

Zagata K, Gulij J, Halik Ł, Medyńska-Gulij B (2021) Mini-map for gamers who walk and teleport in a virtual stronghold. IJGI 10(2):96. https://doi.org/10.3390/ijgi10020096

Zang M, von Schüßler RiV (2022) Citygml-import-plugin. Julius-Maximilians-Universität Würzburg. Available online at https://maxhunter007.itch.io/citygml-import-plugin. Updated on 8/2/2023