The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-M-9-2025
30th CIPA Symposium "Heritage Conservation from Bits:
From Digital Documentation to Data-driven Heritage Conservation", 25–29 August 2025, Seoul, Republic of Korea

Creation of an Immersive Game for a CAVE Environment, around the Rhine Castles 3D and 4D Models

Jade-Emmanuelle Heitz [1], Mathieu Koehl[1]

[1] Université de Strasbourg, CNRS, INSA Strasbourg, ICube Laboratory UMR 7357, Photogrammetry and Geomatics Group
Strasbourg, France - (jade-emmanuelle.heitz , mathieu.koehl)@insa-strasbourg.fr

Keywords: VR, Cave, Cultural Heritage, Unity, Blender

Abstract

As part of the Interreg VI Châteaux rhénans – Burgen Am Oberrhein project, INSA Strasbourg is carrying out heritage conservation work on several castles on the Rhine plain. This work consists of 3D topographic surveys of the sites to crystallize them at a given time. Then, in a second phase, this work focuses on the modelling of 3D historical reconstructions of the sites. This challenge is carried out in collaboration with specialised archaeologists, to bring these castles back to life. The objective of this work is to promote the Rhineland castle heritage through digital technology. To increase the influence of these 4D models integrating different temporalities, a Unity game that can be used in a virtual reality cave has been created. Indeed, this method of valuation increases the public's immersion in the model and allows them to project themselves into the reality of the field. The methodology involves converting the models made with *Blender* to *Unity*, then creating a game with *Unity*. Finally, a self-guided game of a set of castles modelled at different historical periods is proposed. This game, focused on the player's immersion, gives access to menus allowing to choose the castle or the desired historical era. It also provides a range of fun tools allowing the player to move objects and take photos. It is also intended to be educational, by offering informative bubbles or interactions with NPCs, while roaming the castles to the rhythm of medieval music. VR is used here as a fun way to enhance the project's digital content in a more immersive form to the public.

1. Introduction

Heritage conservation is an important issue. It allows us to bring to life and revive stories that have disappeared, are not accessible and are not well known. Today, this conservation is increasingly being done digitally, a simple and accessible means for a wide audience. Indeed, as part of the Châteaux rhénans – Burgen Am Oberrhein project, a dozen castles on the Rhine plain are being surveyed and their various historical states are reconstructed in 3D. These are called 4D models, implying the notion of time. These models are made with the aim of creating promotional renderings highlighting this 4D. In order to create a more immersive experience in castles, this paper will focus on virtual reality within a virtual reality cave.

The VR cave is an immersive system installed at INSA Strasbourg as part of the INSA 2025 project (Insa2025, 2025) and allowing virtual reality visualization simultaneously with several people. The objective of this paper is to develop the methodology used for the transfer of 3D models created in *Blender*, in a VR game developed in *Unity* and usable in the VR cave.

2. First project deliverables: 3D models

The basic data to be used is the 3D model. During the project, two types of models were created: the digital twin of the current state of the castles and the 3D historical restoration model.

First, the 3D model of the current state is generated on the basis of topographic surveys. The final deliverable is a textured triangulated mesh (Figure 1). The texture is photorealistic and is based on photogrammetric surveys.

Figure 1. Mesh of the castle of Wasenbourg. Processing based on lasergrammetric and photogrammetric surveys.

In a second phase, the historical restorations of the castles are carried out on the basis of scientific or non-scientific data, provided by the partner archaeologist (Figure 2). This data can be archaeological excavation reports, topographical plans or iconography (engravings, paintings, images, drawings, sketches, texts, etc.). The restorations are also based on archaeological hypotheses based on analogies to contemporary or geographically nearby castles.

These 3D historical renderings are made using the *Blender* software, which is a free software for 3D modelling, UV mapping, texturing and animation. It is powered by a large community of users, creating a considerable variety of tools (Blender, 2025a). The entire modelling processing chain is then carried out there (Koehl et al., 2024).

Figure 2. Historical reconstruction of the castle of Wasenbourg (13th century), carried out entirely with *Blender*.

An important step in creating castle templates is texturing. Indeed, the project is intended for a wide audience and therefore requires deliverables that can be interpreted by all, which is not always the case with white models. In fact, a textured model must be created.

3. Texturing in *Blender*

Texturing is an important step in the dressing of 3D historical restorations and can take on different aspects depending on the needs of the work. In the case of the project, the texturing style chosen is photorealistic texturing. Indeed, it promotes a better correspondence between the current state of the castle and its 3D historical restorations for the public.
Blender, like other modelling/UV/texturing software, offers several texturing solutions.

3.1 Blender materials for the project

A material is composed of shaders that allow you to interpret and edit textures (Figure 3). A Shader is a parametric computer program that participates in the rendering process. It tells the computer how to calculate the rendering of each pixel of a material in an object. They are used for the manufacture of materials. As for materials, they are a set of textures/maps interpreted thanks to shaders, giving an appearance to an object. Textures are images or patterns applied to an object. It allows to give an aspect to the object that can include colour, detail, reflexivity, etc. (Blender, 2025a).

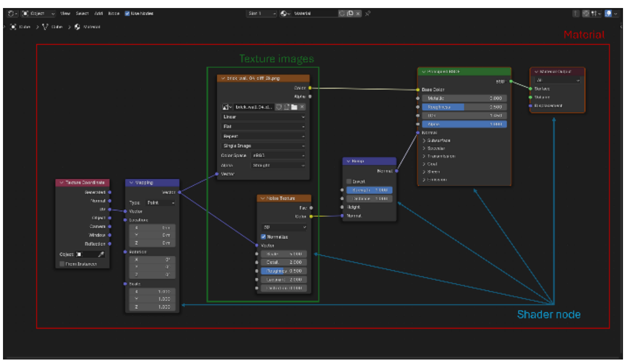

Figure 3. Building a Material within a Shader Editor/Shader Graph Editor (*Blender*)

Texturing can be done in two ways: using images or in a parametric way, called procedural.
For all the restitution modelling undertaken during the project, the texturing was mainly carried out using procedural materials for its many advantages (Sommer et al., 2025). Indeed, within the framework of the project, the procedural materials make it possible to adapt to the reality on the ground. It allows for greater flexibility in settings compared to image-based texturing. In addition, it allows you to create totally individualized patterns or textures for each need. Moreover, the procedural material will be 'seamless' without the need for pre-processing like the image texture, thus simplifying its application on the object.

3.2 Some procedural materials of the project

A key procedural material of the project is the sandstone rusticated block (Figure 4). The rusticated block was widely used in the thirteenth century and later, as a demonstrative sign on the castles or on the main towers in Alsace and in the Rhine plain.
As this type of material is specific, the best solution was parametric texturing.

Figure 4. (left) Rendering of the restitution of the Wasenbourg castle (procedural material of a block with a rusticated stone with holes in sandstone); (right) Image of the ruin of the Wasenbourg castle (block wall with rusticated sandstone holes)

Other procedural materials such as wood (Figure 5), raw sandstone, cob were generated for the project, thus constituting a library of materials specific to Rhine castles.

Figure 5. Latrines textured using a wooden procedural material

4. 4D Models: increase the Scope of the 4D

Finally, we obtain, by combining all these 3D models, a model designated as 4D.
In our case, a 4D model is a time-coherent set of 3D models of the same castle or site. This 4D model consists of a succession of frozen states of the castle in question (Figure 6), thus allowing the analysis of changes and variations of the castle over time.

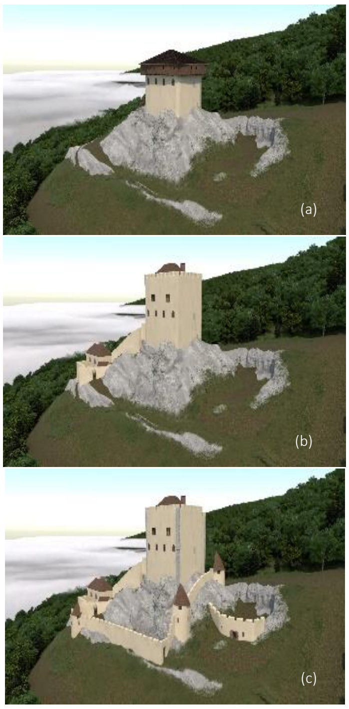

Figure 6. 4D image rendering of Ramstein Castle: representation of the historical evolution of the castle (a) 13th century, (b) 14th century, (c) 15th century

4D can then be expressed through different prisms, depending in particular on the target audience. For simple accessibility, 4D video rendering or in the form of images is the best solution. For a more sensory experience, 3D printing of the different historical phases of a castle is possible. Finally, the last type of deliverable offered is the 3D digital deliverable. Less accessible, but more exhaustive, it can consist of a point cloud, a mesh, etc. These types of deliverables can be made available through visualization platforms such as SketchFab (2025) or immersive VR games, as offered here. In this case, the basic models must undergo processing to adapt them to the reading/viewing/handling media.

5. Adapting 4D *Blender* models to other software

The objective of virtual reality was added during the project, resulting in initial 4D models that were not necessarily adapted to this ambition. A stage of adaptation of the models was therefore necessary. It is linked to the use of different 3D software during the project, but also to a need to optimize models for interoperability between platforms.

5.1 Interoperability of materials between 3D software

As described earlier, 3D models are textured using procedural materials created in *Blender*. For this software, as for other 3D texturing environments, a material is built on the basis of shader nodes (Figure 7).
The problem encountered here is the adaptability of these procedural materials to other texturing software. Generally, any Blender-specific shader will not be understood by other modelling/texturing software. Some main shaders such as the *Principled BSDF* can be understood or even converted, but the transfer of materials between software is very limited, especially between *Blender* and *Unity* (Bikmullina and Garaeva, 2020).
For this, several addons have been created for *Blender*. Martin Sánchez et al. (2024), as well as Texture Baking for Beginners - Blender Tutorial, 2022 offer *Blender* and *Unity* plugins that allow both software to export *Blender* shader files that can be used by *Unity*. However, as our models have quite complex shaders and are not adapted to the formats managed by the plugins, the baking method was chosen (Blender to Unity - Importing a procedural texture, 2023).

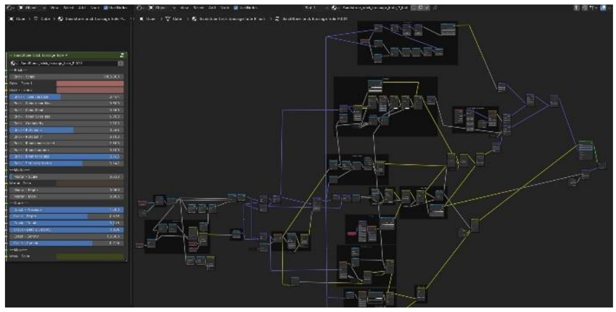

Figure 7. Blender Shader Editor showing the procedural sandstone bossed block material (on the left the user "interface" giving access to the main parameters, on the right the shader graph linked to the material)

5.1.1 Baking in Blender: Baking is a process aimed at speeding up or facilitating a calculation by pre-calculating and then storing data. Baking makes it possible to optimize textured 3D models on several points, including poly count and rendering time (Karlsson, 2013; Verhoeven, 2017).

It is possible to pre-compute (bake) many types of data: physical simulations, lights, animations, materials (Unity, 2023).
In the case of material baking, shader information is computed and stored in a texture file in raster format, also known as a bitmap. This process must be performed on a 3D model with UV maps (Figure 8).

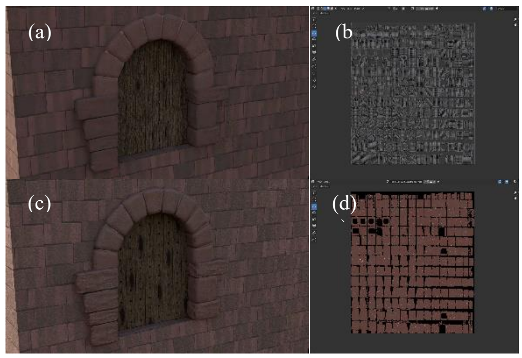

Figure 8. (a) Textured model with procedural materials; (b) UV map of the selected object; (c) Textured pattern with baked materials; (d) Image storing the information baked for the diffuse map.

Blender offers a baking tool that allows you to process many types of textures: the diffuse map, the ambient map, the occlusion map, the normal map, etc. (Blender, 2025b).
The manual baking process in Blender consists of several steps: adapting the Shader node to bake, baking each desired map, and creating and modelling the new material/shader node built with the baked texture maps.
For every material in every 3D object in the models, we need to repeat this process. In the case of castles processed for the project, each 4D model includes hundreds of 3D objects. The work is therefore considerable. In order to facilitate baking, the

processing chain was automated using a python script for Blender.

5.1.2 Automation of the baking process in *Blender*: The automation of the texture baking process in *Blender* was therefore carried out in *Python* for *Blender*. The script encompasses all the baking steps mentioned above.
The entries of the script were then: the collection of objects to be baked and the directory where the images will be saved. In addition, functions such as duplicating the .blend file before starting the baking operations or choosing the resolution of the baking image according to the size of the object have been implemented. Indeed, for the use of the baked model in a game, optimizing the resolution of the textures is quite important. Small objects were therefore textured with images at lower resolutions and vice versa for large objects. It should be noted that this method provides a result that is not very visible on the rendering of the textured model (Figure 8).
After these preparatory processes, the textured 4D model is ready to be exported in the desired format and therefore to be used within a game/rendering engine.

5.1.3 3D model export formats: To guarantee the compatibility of files between software, it is necessary to have models in the right format. Unity can read .fbx, .dae, .dxf, and .obj file formats. Alternatively, formats native to some software such as *Autodesk Maya*, *Blender*, *Modo* or *Cheeth3D* are readable by *Unity*. However, the *Unity* documentation recommends using the .fbx format whenever possible (Unity, 2019). This format contains the 3D data of the model, but also its textures and materials. All the elements required to create the game in *Unity* are then available.

6. CAVE environment - The ConceptCube by Virtual Concept

The VR cave studied in this paper is the one proposed by the French company Virtual Concept (2025) (Figure 9). This company, founded in 2017, offers many VR solutions, including the ConceptCube, a 3x3x3 meter virtual reality cave that can accommodate up to five players.
This system works thanks to 5 projectors which, linked by homography, allow a real-time visualization of the game. The VR experience is possible using 3D glasses, as well as connected controllers (Figure 10). These three components (glasses and controllers) are equipped with spherical targets (six for the glasses and three to four for each controller) to capture movement, but only the main player is equipped with them and controls the game. This system is designed by Optitrack and works with four infrared sensors (Optitrack, 2025).

To manage all the components, the ConceptCube uses several software programs:

- Motion tracking and target calibration are performed by the Motive software developed by Optitrack
- Projector monitoring and game management are carried out by the DEC software (DEC Industry, 2025)
- Creating and editing game scenes is supported by *Unity*. The cave works with Unity version 2019.40f1
- The management of the different Unity scenes is done using Unity Hub

Figure 9. ConceptCube by Virtual Concept (2025)

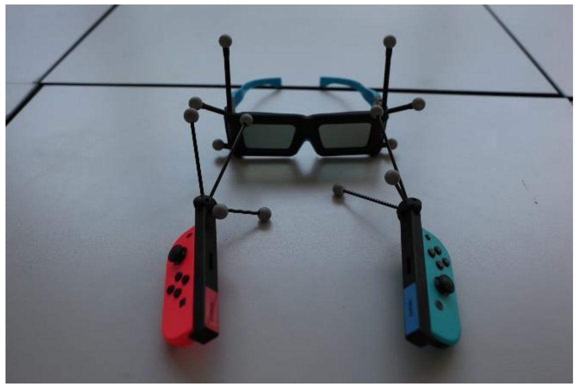

Figure 10. Main player's glasses and controllers, with spherical targets

7. Game designing in *Unity*

Unity is Virtual Concept's recommended software. This software is a game engine that allows you to create interfaces on many devices (smartphone, computer, web, VR hardware, etc.). It has the advantage of being free and having a large community of users around the world. Unity uses programming languages C++ et C# (Unity, 2019).
Virtual Concept has pre-designed a set of tools adapted to *Unity* for the creation of games adapted to the cave.

7.1 SDK and Virtual Concept

Virtual concept provides an SDK for Unity with the VR cave via the DEC app. The SDK is a template containing packages, which are themselves composed of assets. These assets can be scenes (parameterized game environments) or prefabs (sets of parameterized objects) for example. The interest of assets, and in particular prefabs, is their shareability between scenes or between projects.
The Virtual Concept SDK is adapted to the use of the virtual reality cave and provides tailor-made prefabs. In particular, it allows to adapt renderings to the Cave format. As a result, this template is used for the design of our game. The rest of the paper will allow us to develop the main functions of the DEC SDK.

7.1.1 Prefab XR player: The XR player is a prefab that simulates the player and therefore manages the player's navigation and interaction in the game scene. In particular, it allows you to control the controllers and the glasses.
The prefab also gives access to a pre-programmed inventory attached to the player's controllers. This inventory then lists buttons and texts that can be edited by the developer.
The XR player is also linked to the Avatar, a 3D model that allows the player to be visualized. Other objects are attached to the XR player, such as the 'Slot Start' allowing the player to set the starting position in the scene.

7.1.2 Prefab UI Dialog: The UI Dialog prefab consists of a pre-programmed canvas consisting of buttons, icons, and features. The Dialog UI allows you to display messages and sounds when interacting with the player. This information can be set to repeat, close, hide, etc.

7.1.3 Scripts Floor and Wall: Several scripts are also implemented in the DEC SDK, such as Floor and Wall. These scripts must be attached to GameObjects and allow them to be given floor and wall behavior, respectively. The GamesObjects concerned must obviously include a Component Collider (Unity, 2019) for the scripts to work.

7.1.4 Script Door: An interesting script in the context of the project is the Door script allowing interactions between the player and doors (sliding or swinging). This script makes it easier to set up doors, a lock if necessary, but also many aesthetic aspects for the game, such as sounds and open/close movements.

7.2 Creating a game with Unity for the project for Cave applications

7.2.1 Import a 3D model: After baking, templates can be imported into Unity. We choose the Blender format (.blend) accompanied by the folder containing the textures baked in image format. The models are then recognized as Prefab, for which the materials are automatically adapted to the Unity format. This format was favourited in contrast to the *Unity* recommendations, as it allows for easy editing and without the need to re-save the 3D file.

7.2.2 Staging of the game: The game created consists of three levels (Figure 11): the beginning tutorial, the menus and the castle visit.

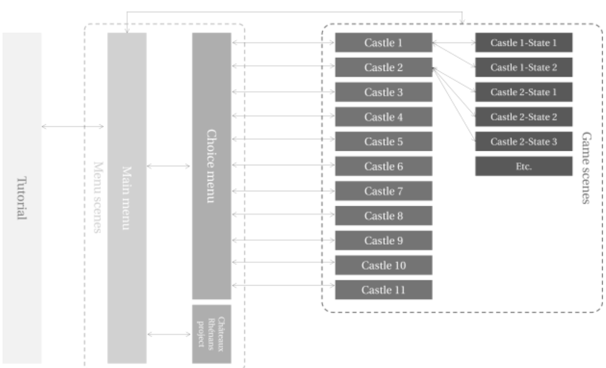

Figure 11. Game organization in the VR cave, each rectangle represents a Unity scene. The links between these scenes are represented by arrows and are made by scripts and functions.

The first step in the game is the tutorial. It consists of UI Dialog scrolling down a series of information related to the use of controllers in the game. It is then possible to go to the main menu via a programmed button.
The 'Main menu' is the home menu with three buttons that allow the player to start the game, to find out about the Châteaux rhénans - Burgen Am Oberrhein project or to make the tutorial on how to use the controllers.
For an immersive aspect from the beginning of the game, a universe reminiscent of a torchlit dungeon is created for the 'Tutorial' and 'Main menu' scenes. To do this, we modelled and textured this part in *Blender* and imported the models in .blend format as an asset of our game. To increase the immersive aspect, we added sound effects of crackling fire, and an adapted play of light (Figure 12).

Figure 12. Scene of the 'Main menu' in the cave

Technically, to create links between scenes, we use a script called 'MenuController' applied to 'Buttons'. This script gives access to functions of the UnityEngine.SceneManagement class such as 'ChangeScene', 'CancelInvoke', etc. (Unity, 2019). This method allows to optimize the size of scenes by dividing objects by scene and calling a scene as needed.
In a third step, the 'Choice menu' is offered to the player. It works on the same principle of 'Menu Controller' script associated with buttons. This scene offers an overview of the castles available to the player to visit. Indeed, the player finds himself in the middle of miniatures of the castles on offer. In order to visit the desired castle, the player makes his choice on the associated panel (button).
Once the castle has been chosen, the player can then walk through the centuries in the castle he has selected.

7.2.3 Immersion in a castle: When the player is in a scene dedicated to a castle, he has many tools and activities at his disposal. He can walk around the model via three modes of travel. This is possible thanks to the 'Floor' and 'Wall' scripts described above. The player also has an inventory attached to the controllers and can be called up using pre-defined buttons (Figure 13). This canvas is a child of the XR Player prefab and therefore benefits from the functions of the SDK. A canvas has been created to switch between the different eras of the castle, but also to return to the 'Choice menu' as well as the 'Main menu', or to open a 'Help' window explaining the different buttons of the controllers.

Also for more immersion, a real-time map has been added allowing the player to position himself in the castle. The player is then represented by a red arrow. This map is actually the rendering of a fixed camera placed above the scene. The arrow representing the player is attached to the XR Player and therefore moves at the same time as him. The card is defined as a child of the XR Player (Figure 13).

Figure 13. (a) Inventory related to the player's controllers; (b) Map allowing the player to view his position in real time during the game

In the educational context of the game, many UI Dialog have been added within the 'Game scenes'. These floating information bubbles communicate information about the location (Figure 14) or give game quests to the player. It starts automatically when the player approaches, the text is played via an audio track, the dialog follows the player for a short distance and it closes when the player clicks on the 'close' button or moves away. The associated functions are specific to the SDK developed by Virtual Concept. They work thanks to the Dialog Controller script which collects information such as Dialog Data (managing sound, text, text actions, etc.), the anchoring of the Dialog UI, the speed of the text, etc. (Unity, 2019).

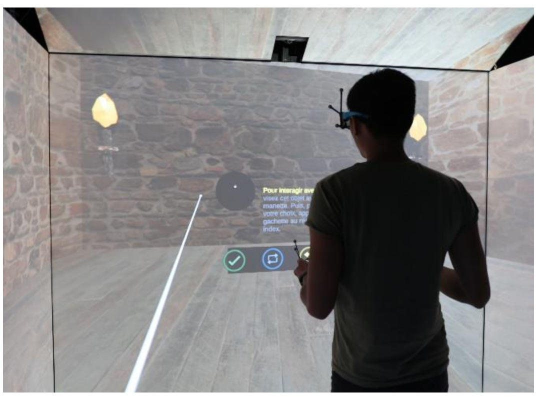

Figure 14. Launching a UI Dialog

The DEC SDK also provides an interactive Tablet offering several tools such as a camera, a clock, a rangefinder, and others (Figure 15).

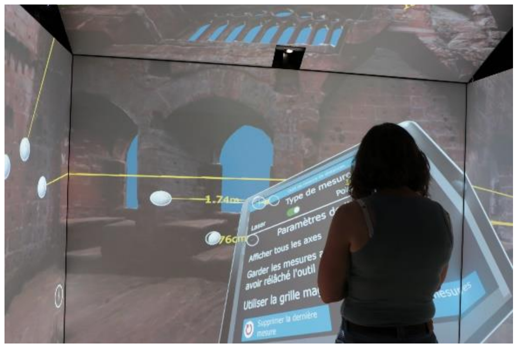

Figure 15. Measurement thanks to the prefab 'Tablet' provided by DEC

For interactive purposes, the SDK's 'Door' script is used on doors so that they can be opened or closed (Figure 16). For this interaction purpose, we have also added a skill game where the player can interact with sandstone blocks by moving, stacking them, etc.

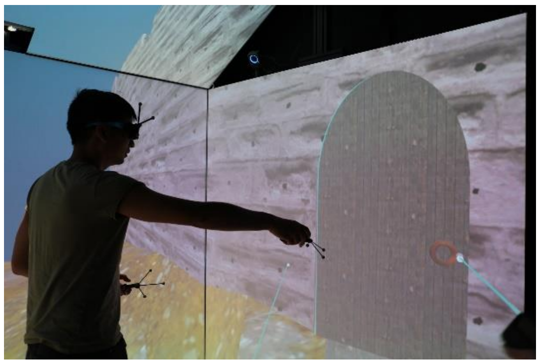

Figure 16. Interaction with a door thanks to the 'Door' script provided by Virtual Concept

7.2.4 Player experience and immersion: In order to increase the player's immersion, details such as a sky in the form of a Sky box, music, and sound effects have been added.

The immersion aspect is increased thanks to the implementation of NPCs (Non Playable Character) in the game scenes (Figure 17). These help to give a lively look to the game. These NPCs are 3D models downloaded from Mixamo.com (2025) and animated in *Unity* thanks to animations also from *Mixamo*.

Figure 17. NPCs imported into the game

Finally, the player experience is enhanced with an "emergency exit" in case of difficulty during the game. Indeed, the DEC SDK offers partial control over the game via the computer used. A 'classic' interface allows the game administrator to interfere in the VR part via this computer. This is a set of preconfigured canvases. From a technical point of view, it grants to change the player's mode of movement (teleportation, flying or on the ground), to modify the rendering parameters (quality, fps), etc. It also lets to guide the player through the game using a light beam, or to interact with the game's buttons. This device is very interesting, especially for the reception of people who are not experienced in VR games. It permits to provide help or to unblock a complicated situation.

8. Evaluate the player immersion in a castle with CAVE environment

As part of the Châteaux rhénans – Burgen am Oberrhein project, the game created is intended to be exhibited and offered to a wide audience. It is therefore essential to evaluate the accessibility of the VR game for this objective.

To do this, the game has been subjected to several player profiles: experts, intermediates and beginners. These three categories consider the player's knowledge of standard video games and the use of controllers, but also the apprehension of 3D.

The three-part evaluation protocol (training, play and evaluation) was submitted to 14 players (Table 1). During the training phase, the player makes the tutorial proposed at the beginning of the game. During the game, the game starts in the 'Main menu' and the player can visit a castle. Then, the evaluation is carried out using an evaluation table.

Evaluating player level	Number of players
Beginner	2
Intermediate	5
Expert	7

Table 1. Distribution of the number of players according to their level

The evaluation table consists of three topics. First of all, the 'Experience' of the game which is based on the use of controllers and buttons as well as on the game play (switching between historical phases, menus, dialogs and tools). Then, the player evaluates the 'Immersion' in the game with more sensory and personal notions such as the mode of movement, the appearance of textures and morphologies, the atmosphere of the game (music, lights) and of course the sensations of nausea or vertigo.

The 'Experience' and the 'Immersion' are evaluated according to a scale of integers between 1 and 4, with 1 being negative and 4 being positive. The score of 0 is used for the criteria of nausea and vertigo (0 = no nausea or vertigo).

Finally, the player is offered a comparison with other immersion systems such as simple video such as drone flyovers, panoramic video in VR, panoramic image in VR, manipulation on *SketchFab* or painted 3D printing.

The results were then cross-referenced according to the criteria evaluated, but also the players' experience. Generally, the 'Experience' of the game is rated at 3.3 and the 'Immersion' is rated 2.7. Players therefore find the game fun and accessible (72% of players find the game accessible to the public). The overall average of the game is 3 on the scale used. The weak points are first the sometimes-difficult use of the tools, with a score of 2.7. This is the case for the tablet and the inventory, i.e. tools attached to the player's controllers. Indeed, the distance between the two virtual controllers is too small, which forces the player to spread his arms in reality to see the canvases correctly, but also to aim and validate a choice. This setting is directly related to the SDK and is therefore difficult to access. Another weak point is also located on the quality of the model (texture and level of detail). In fact, the models (of the current state, as well as of the historical reproduction) have quite heterogeneous textures, because we kept simple models for the test version of the game. This point can easily be improved by the creation of more detailed models, and by baking more adapted to an aesthetic textured rendering objective. Moreover, these criteria are not related to the development of the game itself. We also note a significant number of players who suffer from feelings of nausea and the beginnings of dizziness despite our attempts to reduce these effects. Indeed, during the development phase of the game, the synchronization of head movements with the direction of movement was disabled. This game mode caused the camera to move too quickly to the player, without any possibility to control. It seemed wiser to allow the player to perform the rotation movements thanks to the joystick. However, this solution is more suitable for experienced players, as it requires the use of an additional command.

On the evaluation of the 'Experience', we observe a binary on the categories 'Adaptation to the controllers' and 'Adaptation to the game play' (Figure 18). Indeed, beginner players have more difficulty using the controllers (0.4 to 0.8 rating delta between beginners and intermediates/experts) due to their inexperience and find the game play much more satisfying than experts (0.4 to 1 rating delta between beginners and experts). Despite the small number of beginner evaluators, this observation is quite consistent, because a beginner player will have more difficulty adapting to the controllers but will be less intransigent on the game play due to his lack of knowledge than an expert player used to game play patterns.

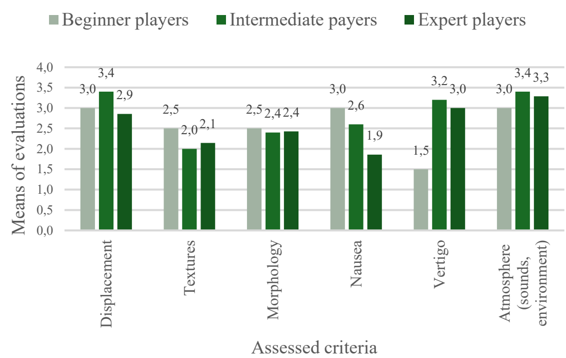

Figure 18. Means of evaluation to the established game experience criteria (by players level)

Regarding the comparison with other systems that can offer an immersive experience, the results are quite mixed. The majority of gamers find that viewing a 360 virtual tour video in a VR headset is more immersive or equivalent to playing in the VR cave. Gamers generally explain this preference for the ease of the VR headset, without controllers and only with head movements. These deductions are perfectly consistent with the observations made during the evaluation tests. Indeed, the controllers and associated tools are quite badly configured, which causes difficulties for the player.

However, the simple video (the example of a drone flight projected on a giant screen is given), the 360° image in a VR headset or *SketchFab* in first-person navigation are rated lower or even much lower than the game.

3D printing is out of the game, considered incomparable to VR gaming in the cave from an immersive point of view.

In the end, the game was enjoyed and players stayed five minutes or more on the discovery of a single castle. In addition, this evaluation phase was very beneficial to the development of the game. Testimonials and player reviews provide perspectives for correction, evolution, and new ideas. Despite the relatively low number of testers, there is a fairly positive and amused feeling of the evaluations.

9. Conclusion

There are hundreds of Rhine castles in the Rhine plain. Sometimes difficult to access, these ruins are the trace of an ancient history that is not well known. The aim of this project was to create a fun way for the general public to discover the Rhine castles as they have never seen them before. One of the solutions is the creation of a virtual reality game applied to a virtual reality cave, presenting several castles during their lifetime. The objective was to offer the public a simple, immersive and interactive tool, allowing them to travel through the centuries.

The game was therefore created in *Unity* for an application in a virtual reality cave. Thanks to the many prefabs provided by Virtual Concept (suppliers of the VR cave), as well as *Unity*'s own tools, a game could be created. It incorporates 4D models designed during the Châteaux rhénans – Burgen Am Oberrhein project and many immersive and educational elements.

The game was deemed 'accessible' by the evaluators, with a focus on the means of travel and the atmosphere of the game.

Many points still need to be improved in order to perfect the user experience, such as the use of controllers and the addition of educational games. However, the VR cave remains a fun and multi-player tool that allows you to discover the Rhine castles from all angles.

Acknowledgements

Best thanks to the Châteaux Rhénans - Burgen am Oberrhein Interreg VI Project (2023-2025). Our work aligns with Action 4.6, focusing on 3D valorisation of heritage sites.

Thank you to all the topographic engineering students at INSA Strasbourg who contributed to the development through the VR cave: Daéren Rigaud (Rigaud, 2023), for initiating the cave work at INSA Strasbourg in 2023 during his diploma thesis; Mamady Kourouma (Kourouma, 2024) and Luca Joseph for their contributions, respectively in 2024 and 2025, during their technological research projects, Luc Laurent for his developments around texture baking during his technological research project in 2025.

We also thank the INSA 2025 project for their technical support during the various projects.

Finally, thank you to all the volunteer testers of the game, especially Hélène Macher and Arnadi Murtiyoso for their visual exposure of the game's tools.

References

Blender tutorial, 2022. Texture Baking for Beginners. https://youtu.be/Se8GdHptD4A (Accessed: June 2025).

Blender, 2025. Blender manual.
https://www.blender.org/ (Accessed: June 2025).

Blender, 2025. Render Baking.
https://docs.blender.org/manual/en/latest/render/cycles/baking.html (Accessed: June 2025).

DEC Industrie. (2025) DEC Industrie. https://decgroupe.com (Accessed: June 2025).

Bikmullina, I. and Garaeva, E. 2020. The Development of 3D Object Modeling Techniques for Use in the Unity Environment. In: *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon).* pp. 1–6. doi: 10.1109/FarEastCon50210.2020.9271568.

Insa2025, 2025. Project INSA 2025. https://openinsa.groupe-insa.fr/nos-projets/insa-2025 (Accessed: June 2025).

Karlsson R., 2013. Improving Baked Textures as a Content Creator Dissertation.
https://urn.kb.se/resolve?urn=urn:nbn:se:bth-4832

Koehl, M., Heitz, J.-E., Rigaud, D. and Guillemin, S. 2024. Process of creating and comparing 4D models of Rhenish castles in different modelling environments. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W4-2024, pp. 263–270. doi: 10.5194/isprs-archives-XLVIII-2-W4-2024-263-2024.

Kourouma, M., 2024. Création des maquettes numériques des châteaux allemands du Landeck, Madenburg et Trifels dans leur état actuel et constitution des supports numériques pour la réalisation de maquettes tactiles. *Diploma Thesis - INSA Strasbourg*, 80 pages.

Martín Sánchez, M., Alonso, P. and Todd Rodríguez, E., 2024. Plugin de Blender Para Exportación de Materiales a Unity.

Mixamo. 2025. Mixamo.
https://www.mixamo.com/#/ (Accessed: June 2025).

Optitrack, 2025. Optitrack.
https://optitrack.com (Accessed: June 2025).

Rigaud, D., 2023. Restitution 4D du château de Wasenbourg dans le cadre du projet Interreg VI. *Diploma Thesis - INSA Strasbourg*, 69 pages.

SketchFab, 2025. https://sketchfab.com/feed (Accessed: June 2025)

Sommer, E., Koehl, M. and Grussenmeyer, P., 2025. Parametric Modelling Techniques for Rhine Castle Models in Blender. Heritage, 8(1). doi: 10.3390/heritage8010031.

Unity, 2023. Model file formats.
https://docs.unity3d.com/2023.1/Documentation/Manual/3D-formats.html (Accessed: June 2025).

Unity, 2019. Unity Manual.
https://docs.unity3d.com/2019.4/Documentation/Manual/index.html (Accessed: June 2025).

Verhoeven, G. J., 2017. Computer graphics meet image fusion; the power of texture baking to simultaneously visualize 3D surface features and colour. In: *Annals Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-2/W2, pp. 295–302.
doi: 10.5194/isprs-annals-IV-2-W2-295-2017.

Vitual Concept, 2025.
https://virtuelconcept32.com/ (Accessed: June 2025).