Palacký University Olomouc,
Faculty of Science, Department of Geoinformatics

Paris Lodron University Salzburg,
Faculty of Digital and Analytical Sciences, Department of Geoinformatics

DIGITAL TWINS IN THE CONTEXT OF DISASTER PREPAREDNESS: FUSION OF GIS AND GAME ENGINES

Diploma thesis

Author
Anatole DELIGANT

Supervisor (Palacký University Olomouc)
RNDr. Jan BRUS, Ph.D.

Co-supervisor (Paris Lodron University Salzburg)
Assoc. Prof. Dr. Dirk TIEDE

Erasmus Mundus Joint Master Degree Programme
Copernicus Master in Digital Earth
Specialization Track Geovisualization & Geocommunication
Olomouc, Czech Republic, 2022

COPERNICUS MASTER IN DIGITAL EARTH
Palacký University Olomouc
PARIS LODRON UNIVERSITY SALZBURG
With the support of the Erasmus+ Programme of the European Union

ANNOTATION

Digital twins are complex systems aiming at reproducing selected aspects of the real world into a virtual replica. One of their essential roles lies in the visualisation of their output through various means. Game engines offer powerful visualisation capabilities, including extended reality, but lack the capacity to ingest the numerous geospatial data formats used within digital twins. On the other hand, GIS allow for the native processing of said formats, though they are not meant to be visualisation platforms. Breaching the gap between GIS and game engines could provide an innovative solution usable for specific needs and use-cases. Disaster preparedness appears as a suitable application area due to the availability of data and its global reach. Additionally, it could benefit from raised awareness. This thesis examines available interoperability capabilities and integration tools, in particular the ArcGIS Maps SDK for game engines. It does so through the implementation of a virtual reality visualisation framework for urban flood data developed with Unity. The implementation is evaluated through a questionnaire to understand better the strengths and weaknesses of the underlying tools and methods, such as how geospatial data is perceived in a virtual reality environment. Current integration tools are shown to be capable and growing, albeit their commercial nature combined with the absence of major open formats and interfaces hinder their scalability.

KEYWORDS

arcgis maps sdk, data integration, interoperability, flood impact, unity, virtual reality

Number of pages	44
Number of attachments	6

I wish to thank the ESA Phi-Lab in Frascati for kickstarting this whole project, my supervisors at UPOL and PLUS for their reviews and advice, all participants to the questionnaire and all of my CDE classmates.

This thesis has been composed by Anatole Deligant for the Erasmus Mundus Joint Master's Degree Program in Copernicus Master in Digital Earth for the academic year 2021/2022 at the Department of Geoinformatics, Faculty of Digital and Analytical Sciences, Paris Lodron University Salzburg, and Department of Geoinformatics, Faculty of Science, Palacký University Olomouc.

Hereby, I declare that this piece of work is entirely my own, the references cited have been acknowledged and the thesis has not been previously submitted to the fulfilment of the higher degree.

Deligant

19.05.2022, Olomouc, Czech Republic Anatole Deligant

Palacký University Olomouc
Faculty of Science
Academic year: 2021/2022

ASSIGNMENT OF DIPLOMA THESIS

(project, art work, art performance)

Name and surname: **Anatole Matthias DELIGANT**
Personal number: **R200693**
Study programme: **N0532A330010 Geoinformatics and Cartography**
Work topic: **Digital twins in the context of disaster preparedness: fusion of GIS and game engines**
Assigning department: **Department of Geoinformatics**

Theses guidelines

The thesis aims to analyse, test, and describe the integration between GIS and game engines in the context of digital twins for disaster preparedness. The student will demonstrate the limits and possibilities of the selected approach in a pilot study within a selected urban area by simulating a disaster. The emphasis will be put on 3D and XR visualizations in preparation for a crisis scenario. The results will be a workflow and set of recommendations for the practical implementation of these technologies.

The student will attach all the collected datasets and all the animations to the thesis in digital form. The student will create a website about the thesis following the rules available on the department's website and a poster about the diploma thesis in A2 format. The student will submit the entire text (text, attachments, poster, outputs, input and output data) in digital form on a storage medium and the text of the thesis in two bound copies to the department secretary.

Extent of work report: **max. 50 stran**
Extent of graphics content: **as needed**
Form processing of diploma thesis: **printed**
Language of elaboration: **English**

Recommended resources:

FAN, Chao, Cheng ZHANG, Alex YAHJA, and Ali MOSTAFAVI. Disaster City Digital Twin: a vision for integrating artificial and human intelligence for disaster management. Online. *International Journal of Information Management*, vol. 56 (February 2021), p. 102049. ISSN 0268-4012.
JANG, Young-Hoon, Sang I. PARK, Tae Ho KWON, and Sang-Ho LEE. CityGML urban model generation using national public datasets for flood damage simulations: A case study in Korea. *Journal of Environmental Management*, vol. 297 (November 2021), p. 113236. ISSN 0301-4797.
MEERA, P., M. L. MCLAIN, Kamal BIJLANI, R. JAYAKRISHNAN, and Bhavani R. RAO. Serious game on flood risk management. In: *Emerging research in computing, information, communication and applications*, pp. 197–206. New Delhi: Springer India, 2015. ISBN 9788132225522.
QI, Shuhua, Daniel G. BROWN, Qing TIAN, Luguang JIANG, Tingting ZHAO, et al. Inundation extent and flood frequency mapping using LANDSAT imagery and digital elevation models. Online. *GIScience & Remote Sensing*, vol. 46 (January 2009), no. 1, pp. 101–127. ISSN 1943-7226.
WHITE, Gary, Anna ZINK, Lara CODECÁ, and Siobhán CLARKE. A digital twin smart city for citizen feedback. Online. *Cities*, vol. 110 (March 2021), p. 103064. ISSN 0264-2751.

Supervisors of diploma thesis: **RNDr. Jan Brus, Ph.D.**
Department of Geoinformatics

Date of assignment of diploma thesis:	**December 9, 2021**
Submission deadline of diploma thesis:	**May 5, 2022**

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA GEOINFORMATIKY
17. listopadu 50, 771 46 Olomouc

L.S.

doc. RNDr. Martin Kubala, Ph.D.
Dean

prof. RNDr. Vít Voženílek, CSc.
Head of Department

Olomouc December 16, 2021

CONTENT

LIST OF ABBREVIATIONS

Abbreviation	**Meaning**
2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
AR	Augmented Reality
DEM	Digital Elevation Model
DSM	Digital Surface Model
DXF	Drawing Exchange Format
ESA	European Space Agency
FBX	FilmBoX
FME	Feature Manipulation Engine
FOSS	Free and Open-Source Software
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information System
glTF	graphics language Transmission Format
HMD	Head-Mounted Display
LOD	Level of Detail
OBJ	OBJect
OGC	Open Geospatial Consortium
OS	Operating System
PDA	Personal Digital Assistant
RGB	Red Green Blue (colour model)
SDK	Software Development Kit
UI	User Interface
VR	Virtual Reality
XR	eXtended Reality

INTRODUCTION

Digital twins are at the heart of the European Union's Destination Earth initiative, whose objective is to develop a high-precision model of the Earth in order to model natural phenomena, human activities and explore their systemic consequences. One of the key challenges it faces is the need for a technological and conceptual framework providing visualisation and interaction capabilities, relying on both well-established and state-of-the-art technologies. The representation of information at different places on the spatial and temporal axis and at vastly different scales, from a single tree to the entire globe, will have to be achieved to ensure the usability of digital twins. There are still significant research gaps regarding visualisation systems, which will have to provide for lightweight, scalable, and relevant solutions, particularly on usability and integration.

Simultaneously, Earth observation and geospatial data are becoming increasingly available quantitatively and qualitatively. Many cities of all sizes across the world now have a data portal that allows the public to freely download and use the products they offer. The software involved, GIS and game engines, have become mature and accessible, offering unprecedented processing and analysis capabilities, while hardware has become sufficiently performant for any consumer device to run them. In a context of recurrent natural hazards increasing in frequency and intensity, specifically floodings in Europe, this stream of data can be disseminated to citizens and would benefit disaster preparedness and mitigation efforts globally by raising the awareness and understanding of disasters.

We are still a long way from seamlessly integrating geospatial data in game engines. While they could act as robust systems for visualising and interacting with geospatial data, thus greatly facilitating development by reducing the resources needed, they were not designed as interoperable. The interfaces and standards they support are limited, hence the need for integration tools such as the ArcGIS Maps SDK for game engines. At the intersection of the aforementioned challenges, this diploma thesis aims at filling the gaps in integration and visualisation by demonstrating how new tools and techniques can be linked together and applying them to a relevant use-case, in this case flooding visualisation in European cities. It is in direct continuation of an internship at the ESA Phi-Lab, building upon the Digital Twin ESRIN prototype that focused on integrating photogrammetry models and real-time environmental data into Unity for VR visualisation.

1 OBJECTIVES

The goal of this diploma thesis is to analyse, test, and describe the integration between GIS and game engines for digital twins in the context of disaster preparedness. The subgoals are implementing data integration methods applied to a relevant thematic area, developing an XR visualisation framework, and evaluating both its practical implementation and alternatives. By leveraging new and mature technologies to produce an XR visualisation system capable of depicting natural hazards such as floods, users will be able to explore the impacts on cities according to various scenarios in an immersive manner. At the same time, developers and experts will be able to build upon the concepts outlined in his work.

The completion of these objectives will require moving a step forward towards solving the current limitations of GIS and game engines integration. As the technology stands today, game engines do not support out-of-the-box global and local projections, georeferencing, and commonly used geospatial data formats. Developing these capabilities from scratch is doable but hardly accessible, requiring a lot of resources to invest. This is where the ArcGIS Maps SDK for game engines comes in. The first beta release was announced in October 2020 and a 1.0 release is expected in May 2022. This very recent tool has tremendous potential for the GIS world by expanding data visualisation possibilities in a manner accessible to individuals.

In that context, these technologies were applied to a thematic area suited to this type of visualisation, namely flood analysis, due to the need for diverse data types (elevation model, city model, satellite imagery), its applicability to XR through the empathy it generates and potential real-life applications for disaster preparedness. To realise these objectives, the research work is divided into 2 parts:

Technical implementation:

- Data collection and processing
- Data integration
- XR development

Conceptual evaluation:

- Alternatives to Esri
- User testing

The results of this work will strengthen the current state of the art knowledge in this very recent field of applications through its methodological outputs, offer a standalone application for applied VR visualisation which can be incrementally upgraded in the future, and detail the workflow necessary to achieve these results in other use-cases, i.e., transferability.

2 METHODOLOGY

The following chapter describes the tools and processes involved in solving the objectives. It provides an overview of their roles and usage while skipping over the details described in the research sections.

Methods

GIS and game engine integration is a relatively new field, especially within the context of digital twins. Defining the scope of the work was necessary to build upon previous research efforts and existing tools in order to achieve an output of interest to future developers. With this in mind, about two-thirds of the work focused on exploring existing integration solutions and implementing one of them, with disaster preparedness as a selected use-case, while the last third is evaluating this implementation and conceptual aspect.

Data collection and processing was made easy by the availability of all required data for Prague and Brno in a free and open manner. Some GIS work was required to project and prepare them for integration, including in a semi-automated manner using the ArcGIS Model Builder. Once the data was ready, it was integrated within Unity, starting with a single data type before expanding it to all.

Data

Data is at the centre of the integration process. In order to demonstrate the functionalities of the selected tools, the datasets should ideally be representative of various use-cases while staying relevant for disaster preparedness. Both 2D and 3D data were tested.

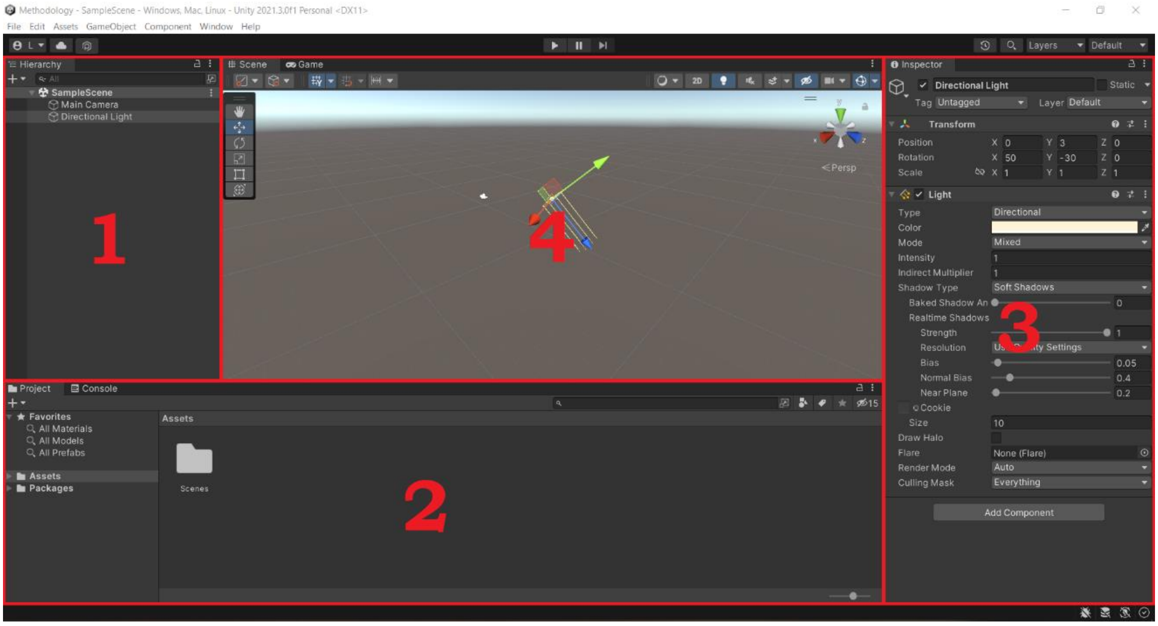
For 2D data, the main thematic layers were the flood extents, also referred to as floodplains or flood zones. They came from the Prague and Brno open data hubs and were not present directly in the final output but instead used for to process and enrich the 3D data. As vector layers, the format used was shapefile. This time, two other 2D datasets were required as raster files: elevation and imagery layer. They are available as hosted tile layers, either with an extent specific to a city or on a national scale.

For 3D data, the main visualisation components were 3D city building models (also referred to as 3D models, city models or building models) for Prague and Brno. Both cities offer them as multipatch shapefiles in LOD2. They were enriched by the 2D flood data and were, the datasets that had to be integrated to the game engine, along with the raster layers.

Software

A very prominent software in the GIS world, ArcGIS Pro in its latest version (2.9) was used for the data processing and export to an ArcGIS Online server for hosted scene layers. The specific geoprocessing tools used were varied and described in detail in the technical implementation part.

Unity is a general-purpose game engine developed by Unity Technologies, usable to create 2D and 3D experiences through a visual editor and the C# language (interpreted to C++ at runtime). It is multi-platform and the personal plan is free to use. Development with Unity began with version 2021.1.16f1, the latest at the time, later upgraded to 2021.3.0f1, the most recent LTS version. Development in Unity happens through scenes, each defining a “level” or “environment”. Each scene has a hierarchy made of game objects. Game objects can have various components, objects in themselves, with different settings. Game objects and components can be created within a scene directly or come from the project window, a folder view of imported assets (Unity Technologies, 2022). Even though Unity and Unreal Engine are similarly capable, the former was chosen for development due to the ease of learning (C# as opposed to C++, user interface), faster loading, and compilation times, all better suited to fast prototyping.



Methodology - SampleScene - Windows, Mac, Linux - Unity 2021.3.0f1 Personal <DX11>

Directional Light
Tag Untagged
Layer Default
Position X 0 Y 3 Z 0
Rotation X 50 Y -30 Z 0
Scale X 1 Y 1 Z 1
Type Directional
Mode Mixed
Intensity 1
Indirect Multiplier 1
Shadow Type Soft Shadows
Baked Shadow An 0
Strength 1
Bias 0.05
Normal Bias 0.4
Near Plane 0.2
Size 10
Flare None (Flare)
Render Mode Auto
Culling Mask Everything

Figure 1: Default Unity engine editor layout windows, as of version 2021.3.0f1: hierarchy (1), project (2), inspector (3), scene view (4).

The selected integration solution for the implementation was the ArcGIS Maps SDK for game engines, also referred to as “ArcGIS Maps SDK” or “Maps SDK”. This SDK aims to enable developers to bring GIS data into the Unity and Unreal game engines. As a pre-release product, it is already very capable, allowing the integration of raster layers and 3D objects directly into the engines for visualisation purposes. It offers a prefab Unity game object for rapid testing and a C# API for scripting.

In addition to the Maps SDK, other integration tools were explored and described as potential alternatives depending on the needs. First, besides its “native” integration capabilities, i.e., interoperability capabilities, Unreal Engine with the Cesium for Unreal plugin provides features similar to the Maps SDK through the integration of 3D content as 3D tiles and the ability to stream them from the cloud. Second, as a FOSS game engine, Godot has much potential with growing capabilities, even though it cannot yet compete with Unity and Unreal for large projects. Lastly, Google Forms was used to evaluate the integration solution through a questionnaire due to its simplicity.

Processing procedure

The research question should be approached methodologically. This starts by defining the scope, requirements, inputs, and outputs, along with the available tools and solutions. Since various tools are required, development can be started in parallel until all reach the integration step, at which point the outputs can be horizontally merged. In turns, this leads to the refinement and polishing of the application, so it is ready for the evaluation step.

This work can be conceptually visualised as a table with the software used in the x-axis columns and development steps in the y-axis rows. The columns represent the three main software used for development in an incremental manner that began in parallel: GIS, integration tools, and game engines. Consequently, the project went from the definition step to the development step for all three software, with their work complementing each other, before the evaluation of the final product.

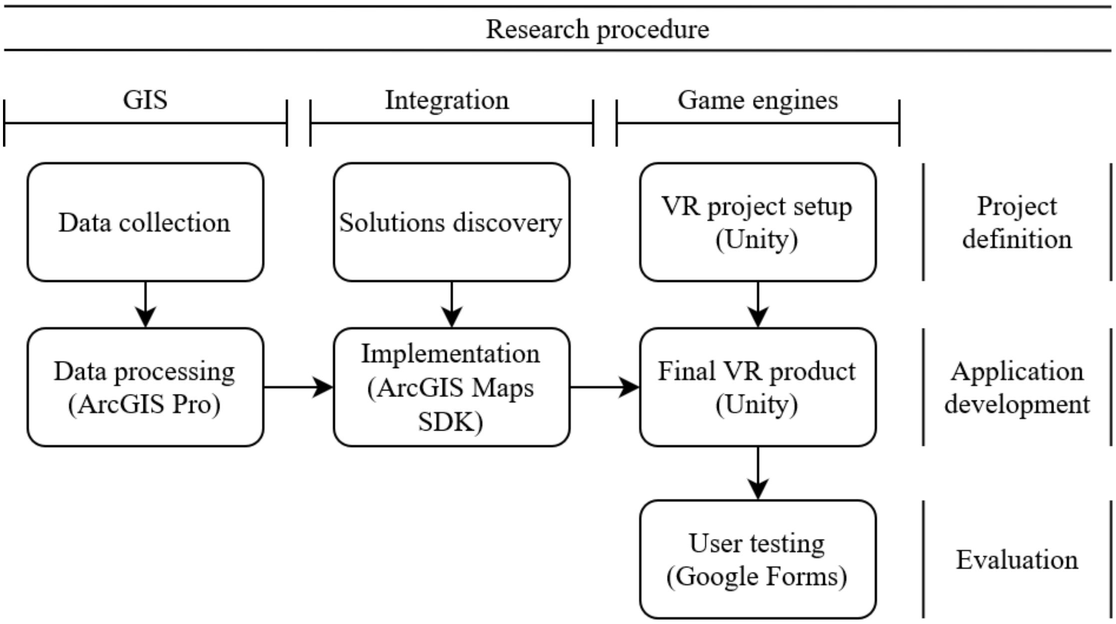


Figure 2: Summarised research workflow and processing steps, from the project definition to the evaluation and from GIS to game engines.

3 STATE OF THE ART

Game engines as visualisation frameworks for geospatial data

The earliest research on using game engines as visualisation tools for geospatial data dates back to 2002. The advantages of game engines were described as their capabilities to "simulate virtual environments" in a real-time, low-cost, and collaborative manner (Herwig, 2002). They relied on limited interoperability capabilities through the support of some 3D formats like DXF. In a paper from the University of Stuttgart, the free movement of players and interaction with data were added to the visualisation (Frisch, 2004) and the same conclusion was reached: "A problem is still […] to get existing spatial data into a format that the game or the accompanying editors and tools understand". Despite the understandable outdatedness of technologies involved (PDA, Unreal Engine 2…), their analysis remains relevant to this day, especially regarding the need for interfaces with GIS and databases along with more interaction capabilities. They correctly anticipated the drastic improvement of computer graphics, exemplified by the rendering of vegetation and the interdisciplinary nature of the field.

As the technology behind these software progresses, larger entities outside of academia can evaluate their use for practical products and services. Over two decades, game engines have matured and been made accessible to anyone, often for free. Unity offers its engine for free with minor limitations such as a splash screen and Unreal has a royalty system for profits over one million dollars.

Keeping in mind the scope of the research questions, the advantages of game engines can be summarised as follows:

- Multi-platform and XR support: an application can be coded once and then deployed on various platforms (PC, consoles, mobile, web…), including XR applications
- Real-time 3D rendering: allow for realistic interactions with physics and animations, lighting and shadows that drastically increase immersion
- Editor: an integrated development environment for authoring content and speeding up development by providing common functions as GUI tools
- Extensibility with plugins, packages, and assets adding new content and features

While the disadvantages of integration with GIS are:

- Lack of native georeferencing and projection system: location-based products have to be implemented from scratch, including local and global projections necessary for precise display (in Unity) or rely on plugins (in Unreal Engine with the "Georeferencing" plugin, Epic Games 2022)
- Asset import pipeline performance: even for supported 3D formats, their size can make their use impractical for large areas
- Limited interoperability: very few formats are natively supported by game engines, FBX is the most common one and is not suited to GIS-based 3D modelling (loss of semantic attributes, textures, geolocation) (Buyuksalih, 2017)

There has been a strong interest in providing easier access to GIS data in game engines, with Unreal Engine leading the way. Through their "Epic Mega Grants", Epic Games supports projects that enhance the 3D ecosystem and provide funding to companies developing new plugins such as Blackshark.ai or Cesium GS (Blackshark.ai, 2021 and Cozzi, 2020). Both aim at representing 3D data on a global scale, which could provide numerous benefits to the development of digital twins for the entire planet.

Floods as relevant use-cases

Disaster preparedness aims at reducing the impact of disaster events through preparation that helps cope with hazards. Natural hazards happen regularly everywhere on the globe, sometimes leading to disaster events when combined with exposure and aggravated by the local vulnerability. Due to their global reach and important impact on human lives, disasters are extensively studied and appropriate for use in exploratory visualisation in part due to the empathy they generate. Applied to a specific use-case, it would help understand the threats faced by communities and how to raise awareness about their scope, including to decision-makers.

For this project, the modelling of a tsunami on a coastal city such as Tokyo was initially envisioned, with animations showing incoming waves interacting with a city model realistically. Delft3D, a popular modelling suite for hydrodynamic simulation from Deltares, was selected and tested due to its proven capabilities (Deltares System, 2011). This proved to be out of scope for our work, which focuses on GIS and game engine integration, as implementing a realistic simulation within the software would not have made use of the same technologies (Le, 2019 and Josiah, 2020). Instead, pre-calculated urban flood extents were chosen due to their availability and compatibility with GIS.

Flood predictions can be used for cities such as Prague or Brno that publish them as open data to analyse the risk they pose to infrastructures. These predictions consist in flood extent data for a specific annual peak discharge, a measure of the highest water flow in a year in cubic metres per second. Frequency curves are used to determine the recurrence intervals, also known as the return period in years, depending on the intensity of the predicted flood event: this is called flood frequency analysis (Dalrymple, 1960). This probability corresponds to the likelihood of that event happening in a year: for instance, a 1-year flood has a 100% (1/1) probability of happening in a year, while a 25-years flood has a 4% chance (1/25) and a 100-years flood a 1% chance (1/100). Combined with elevation models, this can be transformed into the data used here, predicting the maximum extent a flood will have (Qi, 2009). With that said, it does not consider all modelisation parameters like the urban drainage system or state of vegetation.

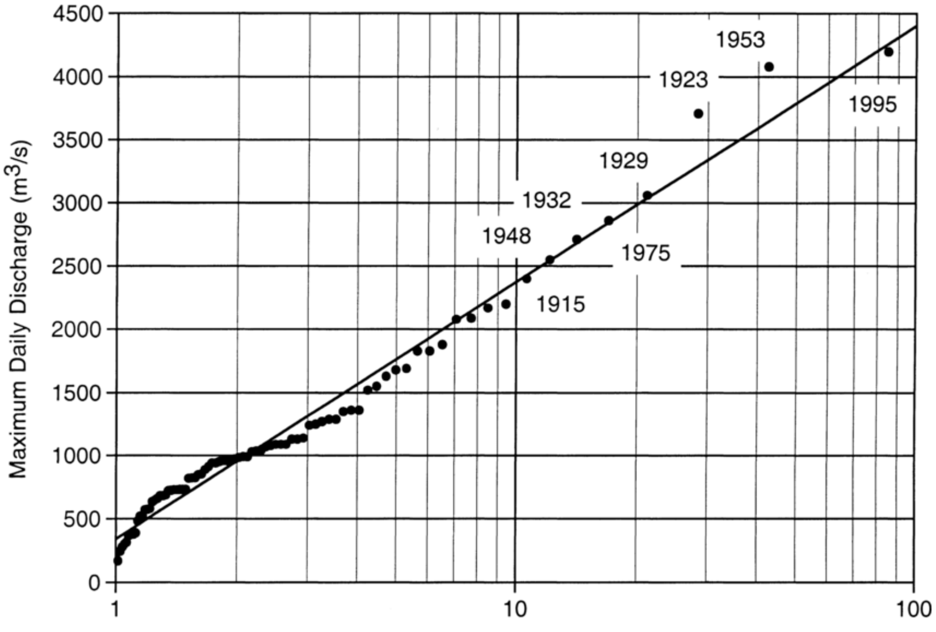


Figure 3: Example peak discharge versus return interval curve for the South Saskatchewan River in Canada (Kalischuk, 2001).

Flood events are particularly suitable for visualisation since they occur at frequent intervals on all continents, impact people’s livelihoods directly and can be depicted spatially at various complexity scales, from localised sea-level rise (NOAA, 2022 and Copeland, 2012) to underground infrastructure risk assessment (Vamvakeridou-Lyroudia, 2020). The focus is on visualising the impact of the aforementioned flood extents in a 3D manner that can be adapted to VR or AR. Flood Action VR, developed at the University of Iowa, is described as a “3D gaming environment for awareness and training about disaster preparedness” where a player has to solve tasks in VR in a flooded environment (Sermet, 2019). Unfortunately, the article does not detail the implementation and user feedback gathered during the 2017 Samsung Developer Conference. In another example, Zhi et al. describe urban flood risk assessment methods. They rely on a 3D visualisation of flood risk, which has advantages in transparency and user understanding, including for stakeholders (Zhi, 2020). Both examples illustrate the interest in the 3D and VR visualisation of flood data, even though the means are very different: in the first case a gamified experience, in the second an in-depth risk exploration.

Immersive visualisation through virtual reality

The transition from a 3D game to a VR application is made simple in Unity from a development standpoint through pre-made components that can be interlocked to produce a complete system, from head tracking to controller buttons press. The main implementation difference between a monitor and VR lies in interactions, i.e., from keyboard and mouse to position tracking and controllers. Because different manufacturers have their own VR runtime, developing a product focused on a single type of hardware would be inefficient. This is where the OpenXR open standard can help. Developed by the Kronos Group, its 1.0 specifications were released in 2019; according to Brad Insko, Intel’s Lead XR Architect, “OpenXR seeks to simplify AR/VR software development, enabling applications to reach a wider array of hardware platforms without having to port or rewrite their code” (Khronos Group, 2019). In other words, OpenXR solves AR/VR fragmentation by providing a cross-platform API that hardware and software companies can use, creating a platform-agnostic experience for XR products. Unity and Unreal Engine have been supporting XR applications and enabling their proliferation for many years. In addition, both now support OpenXR, further simplifying development.

As a relatively recent technology, VR can generate interest amongst users by itself, no matter the thematic area. Depending on the task and its execution, the added immersion can lead to a lower mental workload and better usability (Filho, 2020). VR is also better suited to learning than traditional text-based methods. Indeed, in a 2015 study on serious games, a group played a serious game on disaster risk management before their evaluation and comparison to a control group that used textbooks. The game-based learning environment proved better for motivation, engagement, and effectiveness, among other advantages (Meera, 2015). However, serious games still have the disadvantage of being more expensive to develop than text or static images. For development and testing, the Oculus Quest 2 headset was used.

User experience evaluation

Considering the exploratory nature of this visualisation work and its state of development, the methods mentioned are experimental. As such, evaluating their effectiveness will be helpful to future readers wishing to implement similar methods and build upon this work by giving it a critical look. Among the several ways of doing so, which include eye-tracking, direct questions, or thinking aloud (Freire, 2012), user questionnaires appeared appropriate for two reasons: they provide a window into the user's mind and opinion and allow for the simplified gathering of large quantities of data.

An in-VR questionnaire was considered, as opposed to a web-based, out-VR questionnaire. One fully integrated with the VR application would improve the immersion and the feeling of participating in an experiment, which users would prefer (Safikhani, 2021). However, there is no significant benefit to an in-VR questionnaire beside that. Furthermore, this would require a redesign of the application through the gamification of the experience, necessary for successful integration, plus additional UI design. For this reason, a web-based questionnaire was instead developed.

Analysing the user perception of geospatial data in a VR application developed in a game engine through questionnaires is not a new endeavour. It has been found to increase the perception of 3D environments and user motivation, however more research is necessary to understand the benefits in terms of usability (Carbonell-Carrera, 2021).

Digital twins

The term digital twin encompasses all types of virtual replicas, ranging from the human body to vehicles. The definition is broad as they are complex systems unique to their use-cases, including healthcare, urban planning or even manufacturing. Of interest for this thesis are smart cities digital twins, which are multi-layered replicas requiring the integration of terrain, building, infrastructure, mobility, and real-time data (White, 2021). While the whole design process is not part of this exercise, their use of geospatial data and techniques can be studied and re-created using different methods. Digital twins are systems whose individual components are assembled in the final phase. Implementing these components, most crucially the ones related to integration and interoperability, solves the challenge of interfacing between the replica's backend and frontend.

For disaster preparedness, conceptual research from Chao Fan et al. details how digital twins could be the core of a "*Disaster City Digital Twin*". The first step is sensing, collecting the data from various sources, including crowdsourcing. This step is followed by the integration step, with pre-processing and projection for further analysis. Lastly, using the collected data and observations, information valuable to decision-makers can be extracted before leading to real-life actions. One of the challenges with this approach is the filtering step, as the data is not necessarily authoritative, yet accuracy is crucial for any disaster response approach to avoid wasting resources or missing important news (Fan, 2021). An advanced gamified decision-making experience nonetheless provides benefits, especially for training purposes. No matter the chosen approach, game engines are key integration components, acting as the recipient for processed data and with the capabilities of adding their own logic to output information in a visual manner.

Semantic perspective

When reading this document, a concept that may come to mind is that of serious game, which can be defined as the combination of a utilitarian function with the main elements of a video game. The G/P/S model (Gameplay, Purpose, Sector) is a proposed model for classifying serious games (Alvarez, 2011). Applied to our standalone visualisation tool, the lack of gameplay elements (stated goal, possibility of winning), combined with the informative and research purpose, as opposed to entertainment, means it cannot be classified as a “serious game” because it is not a game in the first place. On the other hand, the classification “serious play” proposed by the author of the G/P/S model is appropriate. With that said, the scientific consensus appears to be the lack of a clear definition. It is thus up to the writer to assign their project to a category based on their understanding of the concepts and apparent relevance to the task at hand.

Another semantic clarification is the distinction between extended, augmented, virtual and mixed reality. Virtual reality is the most known term: it relies on an HMD for a full immersion into a virtual world and can include other components such as audio or haptic feedback. Augmented reality overlays digital information over the real world, enhancing our perception of reality. Mixed reality merges the real and virtual world by enabling interaction with one affecting the other (Intel, 2020). Finally, extended reality is an umbrella term that encompasses all previously mentioned “reality types”.

Lastly, defining integration, interoperability and fusion is beneficial to ensure a common understanding of the topic and implications. According to the Merriam-Webster dictionary, integrate corresponds to the process of “coordinating or blending into a functioning or unified whole”; interoperability is the “ability of a system to work with or use parts or equipment of another system”; fusion is the “union by or as if by melting”.

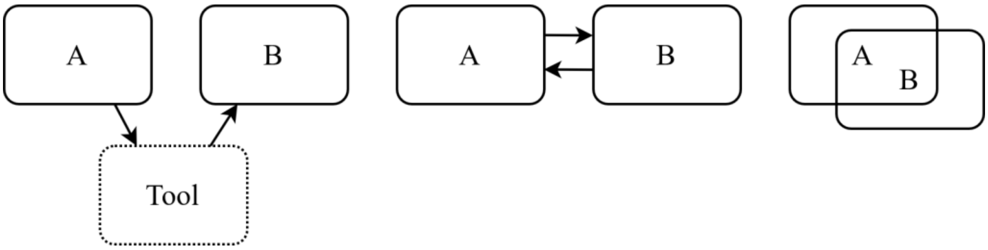


Figure 4: Schematic description of integration (left), interoperability (middle), and fusion (right) through two abstract systems A and B along with their relations.

When talking about data or software, the definitions may differ slightly even though the main ideas remain. The integration of two systems could be defined as using a translator for them to understand each other despite speaking different languages. On the other hand, interoperability is their ability to communicate together using a common language: standards and interfaces. As an illustration: streaming raster data from a GIS server to a game engine is not possible as is, so the systems are not interoperable; using a tool like FME enables the conversion of GIS data to a format understood by game engines, so the systems can be integrated. Fusion can be understood as the union of two distinct datasets or tools into a single functional one, combining all tools into one. In summary, integration describes the process of having two incompatible systems communicate with each other and is what this thesis is about, while interoperability and fusion are ideal future steps.

4 TECHNICAL IMPLEMENTATION

After a preliminary analysis of requirements and available tools, the implementation is the core part of this thesis. A VR application is developed by processing, integrating and visualising various 2D and 3D data. The following chapters describe these processes, explain the decisions made and provide development notes to allow the replication of part or all of it.

4.1 Data collection

For the scope of this work, the implementation required at least two types of thematic data: floodplain extents and building models. Since the host university for this thesis is the Palacký University Olomouc, the two largest cities in the Czech Republic, Prague and Brno, were initially selected. Both cities have an open data hub that provides all the necessary datasets, although most of their documentation is only available in the Czech language. As long as the data exists or can be created, any other city can be implemented.

All datasets are provided either in WGS84 coordinate system (EPSG:4326), the Czechoslovak S-JTSK (EPSG:4156) or both. They were converted to Web Mercator Auxiliary Sphere (EPSG:3857) since, at the time of writing, it is the only coordinate system supported by local scenes in the ArcGIS Maps SDK. Global scenes support WGS84 natively in addition to Web Mercator.

The data used in this initial implementation follows the Creative Commons Attribution 4.0 International (CC BY 4.0) licence. Under its terms, the data can be shared, redistributed, and adapted for any purpose, as long as appropriate credit is given to the authors and no additional restriction is placed on the derived work. The software used and their source code are covered by their respective licence policy, most importantly the commercial licenses of ArcGIS Pro and Unity.

4.1.1 2D data

Floodplains are the thematic data used for analysis and visualisation in this thesis. They are the maximum water extent during a natural flood event and are divided into different discharge rates. As this work is focused first on visualisation, 100-years floods are the most interesting since they have the most significant impact footprint over a city, though different return intervals are also covered. The processing methods for creating floodplains in the Czech Republic are officially defined by the country's Ministry of Environment (Czech Republic, 2018). The data is redistributed through the portals of municipalities, including Prague and Brno (IPR Praha Database, 2019 and Statutární město Brno, 2021). All flood datasets were downloaded directly as shapefiles.

In addition to flood extents, elevation and base imagery layers are necessary to provide context and an accurate, immersive representation of reality. The 5th generation DRM 5G (DTM) of the Czech Republic is provided by the Czech Office for Surveying, Mapping and Cadastre, has a 2-metre resolution, is produced through photogrammetry and LiDAR, and is continuously updated (Czech Land Survey Office, 2017). The imagery basemap of the Czech Republic was acquired from the same office and available in a similar manner, albeit through a map server instead of an image server (Czech Land Survey Office, 2022). The elevation and imagery layers are accessible as hosted web services in the raster tile layers format for cloud-based use within Unity. Higher-resolution data might be available on a per-city basis, in which case they can be replaced easily.

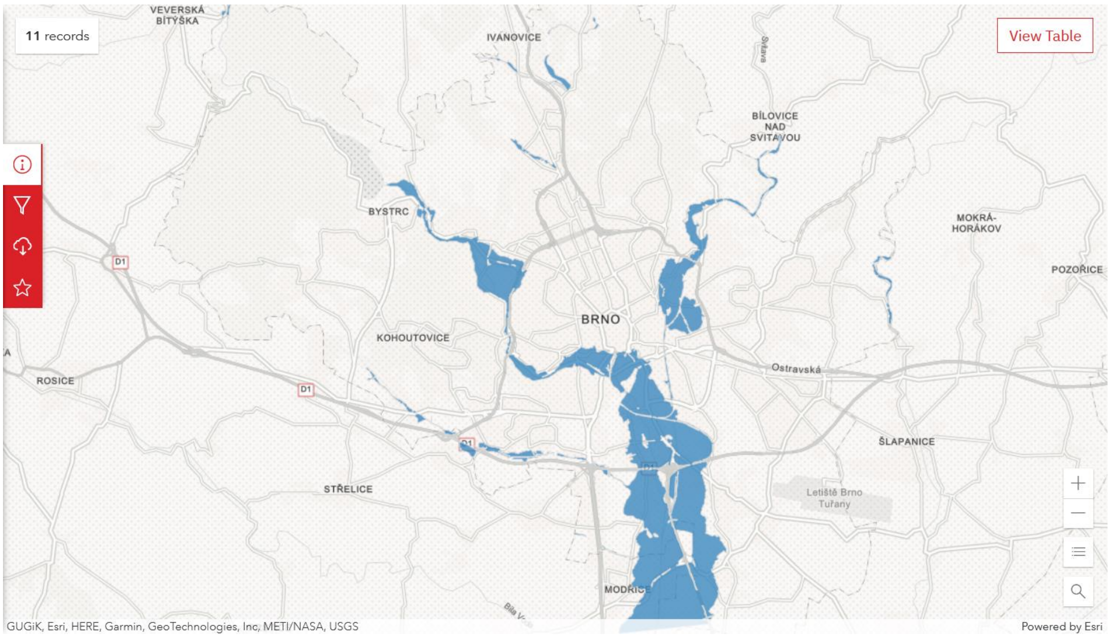


Figure 5: 100-years flood zones over the city of Brno, as displayed online in the Brno datahub.

It is possible to convert the 2D flood extents to a 3D object by enriching the vector layer as a polyline with elevation data, then interpolate it to a multipatch shapefile. However, this is not straightforward and complexifies the processing part. Because a river is also not a simple plane, it requires very precise elevation data for the output to look realistic.

4.1.2 3D data

The main visualisation component in this experiment is the 3D city building models. Every building in the city is represented without textures in the LOD2 representation. Levels of Detail are described in the OGC CityGML conceptual model and represent four consecutive levels of spatial abstraction (Open Geospatial Consortium, 2021). LOD2 consists of the generalised roof shape over an extruded building footprint. Because more detailed data is less common and visual aspect is the main selection criteria, LOD2 strikes a good balance between availability and visual fidelity.

The city of Brno offers two LOD2 building models. One comes from photogrammetry methods and is the most detailed (Statutární město Brno, 2022a), however its spatial extent is limited and does not cover all of our area of interest, namely the riverbanks. For that reason, the second dataset, produced through a DSM and building footprints, was used (Statutární město Brno, 2022b). Prague provides a model similar to Brno, with high-resolution images processed through photogrammetry and covering the entire city in LOD2 (IPR Praha Database, 2022). All buildings datasets are shapefiles in the multipatch geometry type, which enables the boundary of 3D objects to be represented and were downloaded as Esri file geodatabases.

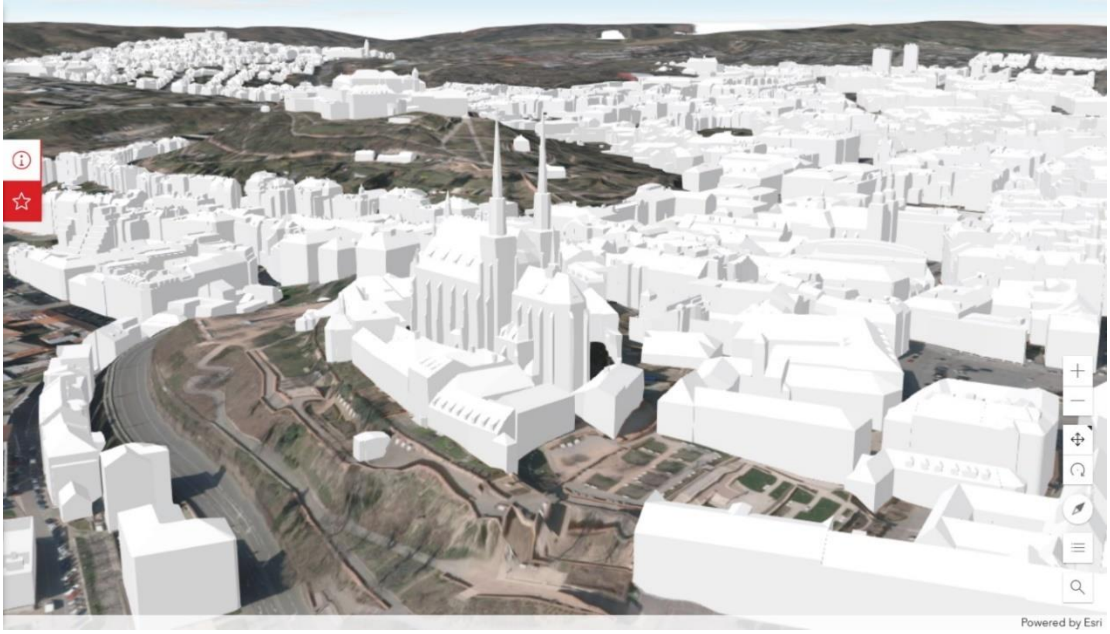


Figure 6: 3D building model of Brno, as displayed online in the Brno datahub.

4.2 Project preparation

Once the data has been identified and collected, it must be processed for the information of interest to be extracted. In parallel, the Unity project must be ready to receive and ingest the incoming data.

4.2.1 Data processing

Depending on the extent of the city model data, it might be necessary to ignore buildings that are too distant and do not contain meaningful information for the visualisation. For instance, because the model for Prague is so large, features farther than 2km away from the flood zones were filtered out in ArcGIS Pro with a *Selection by Location* operation prior to any further processing. At this stage, it is also important to make sure the data is in the correct coordinate system (WGS84).

The next step was to create a numerical attribute column named for example FLOODED. The previous operation, *Selection by Location*, is used again though this time with the condition that buildings selected must intersect with a flood extent. First, the largest, i.e., less common, return interval is selected. If buildings intersect with the flood extent, it is then possible to change the value of the FLOODED attribute and add the return interval in years as a value. It is assumed that all buildings flooded at a specific return interval are also flooded during less common return intervals, e.g., all buildings affected by a 5-years flood are also affected by 20-years and 100-years floods. As such, the value 100 is first added to buildings affected by 100-years floods, then 20, and lastly 5-years floods. Thanks to the overlap of the return intervals, the information for all flood events is kept within a single attribute column, with the added benefits of knowing at which return interval a specific building is first flooded (Table 1). These values will be used by the Unity shader later on.

Table 1 Description of return intervals and required attribute values for their display

Return interval	**Attribute values to select**
5-years floods	5
20-years floods	5 + 20
100-years floods	5 + 20 + 100

The precise centre for our area of interest will be useful in the ArcGIS Maps SDK as the centre of local scenes. To calculate it, a *Multipatch Footprint* operation is performed on the building layers, outputting a multipolygon shapefile layer with the footprint of each building. Changing the geometry type is necessary for the next operation: calculating the layer's *Mean Center*. The coordinates of the resulting point correspond to the mean centre of the building layer. While the precision of coordinates varies depending on the latitude, six digits after the decimal place are enough for decimetre-scale accuracy. Lastly, the building layer enriched with flood information must be exported in the Scene Layer Package format (.slpk), as this is the only 3D objects format read by the SDK. The .slpk files can be read as a local file or as a hosted scene layer on ArcGIS Online.

A toolbox was built using the ArcGIS Model Builder to illustrate the various GIS processing steps and facilitate its transferability (Figure 7). It is partially automated, requiring input layers and parameters such as flood return intervals to be correctly specified, and outputs a 3D object scene layer package file along with the mean centre for that layer. The complexity of models depends on the number of cities and flood return intervals to be integrated. Included below is the simplified model graphic with a single return interval selected. The toolbox itself is available as an attachment.

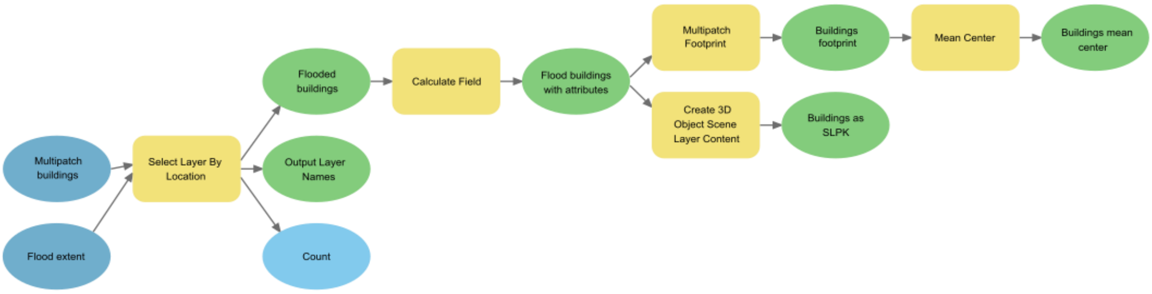


Figure 7: ArcGIS Pro Model Builder graph for the building enrichment process, exported from ArcGIS Pro. It contains the following operations: *Select Layer by Location*, *Calculate Field*, *Create 3D Object Scene Layer Content*, and in parallel to the latter, *Multipatch Footprint* and *Mean Center*.

4.2.2 Unity XR project setup

The second preparation step is the Unity project, as it must be ready to receive the data and visualise it in VR. First, the project should use the *Universal Render Pipeline*, the *High Definition Rendering Pipeline* having not been tested. For compatibility reasons, the *Active Input Handling* parameter in the player settings should be set to "Both", enabling both the "Old" and "New" input systems. Three plugins are required: the *XR Plugin Management* and *XR Interaction Toolkit*, which can be installed from the project settings, and the *OpenXR Plugin* from the package manager. OpenXR will be the plugin provider for XR, acting as the interface between the code and major headsets brands.

Within OpenXR settings, the *Render Mode* has to be set to “Multipass” in order to work with the ArcGIS Maps SDK. If web services are required and a mobile app is developed, *Internet Access* must be set to “Required" in the player settings.

For XR movement and interaction, a new scene has to be created. One game object must hold the *XR Origin* and *XR Interaction Manager* components, with a main camera as a direct child game object. Said camera must have a *Tracked Pose Driver* component. With these elements, the project is ready for the import of the ArcGIS Maps SDK for Unity package.

4.3 Data integration

4.3.1 Introduction to the ArcGIS Maps SDK for Unity

The ArcGIS Maps SDK for Unity is being developed by Esri and enables access to real-world geospatial data within this game engine. At the time of writing, it is in pre-release version 0.3.0 and should be fully released in May 2022. As such, more features are likely to be added but the core ones will stay the same.

Its key feature is the ability to consume geospatial content from an ArcGIS platform, either from the web or offline using local services. Web services can be in the form of raster tile layers hosted on a map server, an elevation layer from an image server, and scene layers in the form of 3D objects or integrated mesh scene layers. Raster tile layers can act as either the basemap or a layer on top of it. Local services support two types of layers, raster tiles, and 3D objects. For offline use, raster tiles must be available as a single tile package file (.tpk/.tpkx), while 3D objects must be in the scene layer package format (.slpk). 3D objects scene layers support 3D attributes filtering, which means Unity and its shader rendering are able to access the attribute table of those layers and enrich the visualisation. As in ArcGIS Pro, both local and global scenes are supported.

The ArcGIS Maps SDK works by providing various key components that offer functionalities necessary to the visualisation of localised geospatial data. There are eight main components which form the interface between the plugin’s backend and the user (Table 2).

Table 2 Main components provided by the ArcGIS Maps SDK for Unity with their description. Note that the prefix “ArcGIS” and suffix “Component” were removed for readability

Component	**Role**	**Condition**
Camera	Connects the camera object to the real-world data and manages the viewport	Attached to a camera object
Camera Controller	Controls the camera during play mode	Attached to a camera object
Location	Places a game object at a specific geographical position	At least one must be attached to a camera object
Map	Connects the layer with the map view component in the background	Requires a renderer component
Map View	Setups the basemap, elevation, and added layers	Required in all scenes, as a parent of the camera object

Component	**Role**	**Condition**
Rebase	Updates the position of the world when the camera moves	Attached to a camera object
Renderer	Links the map view to the rendering backend	Attached to a camera object
Sky Reposition	Updates the position of a volumetric fog when the camera moves	Requires a rebase component and a sky and fog volume object

Besides these, numerous dependencies and utilities are included in the plugin. Among those, it is useful to understand how the High Precision Framework, developed by Unity and incorporated into this plugin, works. Unity uses 32-bits floating-point numbers for its built-in coordinates, based on the *Transform* component which is present on every game object. While this precision is enough for most applications, this becomes an issue for global scenes and large local scenes as the imprecision can amount to metres of differences. For that reason, Esri needed double-precision floating-point numbers (64 bits), which is what the High Precision Framework offers (Esri, 2021). It does so through the *HPRoot* component, which translates real-world coordinates into Unity's universe coordinates, and the *HPTransform* component that replaces the normal Transform component. The replacement of the default *Transform* means the application must be developed with *HPTransform* in mind. It would also complexify the development of AR solutions, especially if those rely on GPS integration.

4.3.2 Manual implementation workflow

The ArcGIS Maps SDK plugin must be downloaded as an asset package file and imported into Unity. The plugin offers a prefab object which can be added to a scene and contains the *ArcGIS Map Controller* component. This object facilitates the testing of features through a UI. However, while it is very useful in discovering the functionalities and testing them, it does not provide us with enough control for a personalised implementation. To solve this problem, a custom C# script must be written using their API, preferably one with no hard-coded values for more flexibility when adding data for another city.

Table 3 List of variables and their role from the custom ArcGIS Scene Setup script

Variable	**Type**	**Description**
isVRScene	Boolean	Changes the camera controller type if the scene uses Virtual Reality
isLocalScene	Boolean	Changes the scene type from global to local
basemapURL	String	Set an image tile layer as the basemap
elevationURL	String	Set an elevation layer as the elevation source
latitude	Double	Set the latitude of the extent centre and origin location

Variable	**Type**	**Description**
longitude	Double	Set the longitude of the extent centre and origin location
altitude	Double	Set the altitude (m) of origin location
radius	Double	For local scenes, set the radius (m) of the extent
APIKey	String	ArcGIS Developer API key
buildingLayerFilePath	String	Path to the 3D object layers file
buildingMaterialFilePath	String	Path to the 3D object layers material

The *ArcGIS Scene Setup* component should be attached to a root game object, such as the *XR Origin*, and must have a camera tagged “MainCamera''. This is because the *ArcGIS Map View* component is added to that specific root object at runtime and various scripts will be looking for this component in their parent game object. Furthermore, the script will look for the main camera and attach it with the required components, namely the *Camera*, *Location*, *Rebase*, *Renderer*, and depending on the selected control type, either the *Camera Controller* or our custom *VR Controller* component. The script assumes the only layer of interest is a 3D objects layer with its own material. However, it can be modified to accommodate more layers and of a different type. An API key is necessary to stream private content from ArcGIS Online and can be obtained from the ArcGIS Developer website, requiring an Esri account.

4.4 3D and XR visualisation

4.4.1 Exploratory visualisation analysis

Placing our visualisation application within MacEachren and Kraak’s Goals of Map Use Cube (MacEachren, 1997) provides a conceptual framework for understanding the type of visualisation through its purpose, audience, and interaction level. Even though it was designed with 2D cartographic products in mind, it is still applicable to this context of 3D visualisation, especially as it is a form of exploratory visualisation.

In terms of purpose, the goal is to reveal unknown information to the user and create new knowledge. The level of interaction is medium due to the 3D nature of the visualisation and the various layers available, albeit no advanced control is possible. Lastly, the audience is diverse, with users having a wide range of experience in GIS and disaster preparedness; overall, it sits in the middle. With these parameters defined and according to the map use cube, this visualisation is made for the analysis of data.

4.4.2 3D attributes and shaders

By default, both ArcGIS Pro and Unity render 3D objects within scene layer packages as white objects. This would be enough if the visualisation was only focused on city buildings with no added thematic content. However, the goal is to convey information through the building layer: affected buildings must be identified during flood events to evacuate the population and minimise damage. Changing the building’s colour achieves this goal.

This is made possible by the Maps SDK as it implements 3D attributes filtering. By specifying the material used for rendering 3D objects layers, it is possible to use a shader attached to this material with a condition-based rendering. Shaders are programs defining the rendering pipeline for a program. The attributes to be forwarded from the scene layer package to the material have to be manually specified, hindering scalability as it requires knowing what attributes are present in the layer. Furthermore, only numerical attributes (floats, integers, doubles) are supported directly. For attributes such as strings or dates, it is required to use an additional function that will convert a non-numerical value into a numerical one, necessitating numerous additional steps and lines of codes.

Once the attribute has been forwarded from the layer to Unity, it can be used within the shader's logic. Shaders can be coded directly in C# or built using the Shader Graph tool, a node-based framework facilitating their development. While this graph tool does not have all of the functionalities offered by the code nor the same level of customisation, it is sufficient for simple processing. Each shader must be used by a material, and it is this material that will be applied to the buildings using the shader's logic. The rendering part comes from the default URP shader in the Maps SDK samples, so there is no need to modify that. The only part that needs tweaking is the logic outputting the base colour of the material: three consecutive if-else loops check whether the attribute of the FLOODED column is equal to the return interval, from the most common to the least common (Figure 8). The chosen colours were yellow for the most common flood extent to red for the least common flood extent since the most damaging flood event should be categorised using a more alarming colour. However, the opposite logic would be valid as well: buildings always affected by floods, no matter the return interval, are the most vulnerable.

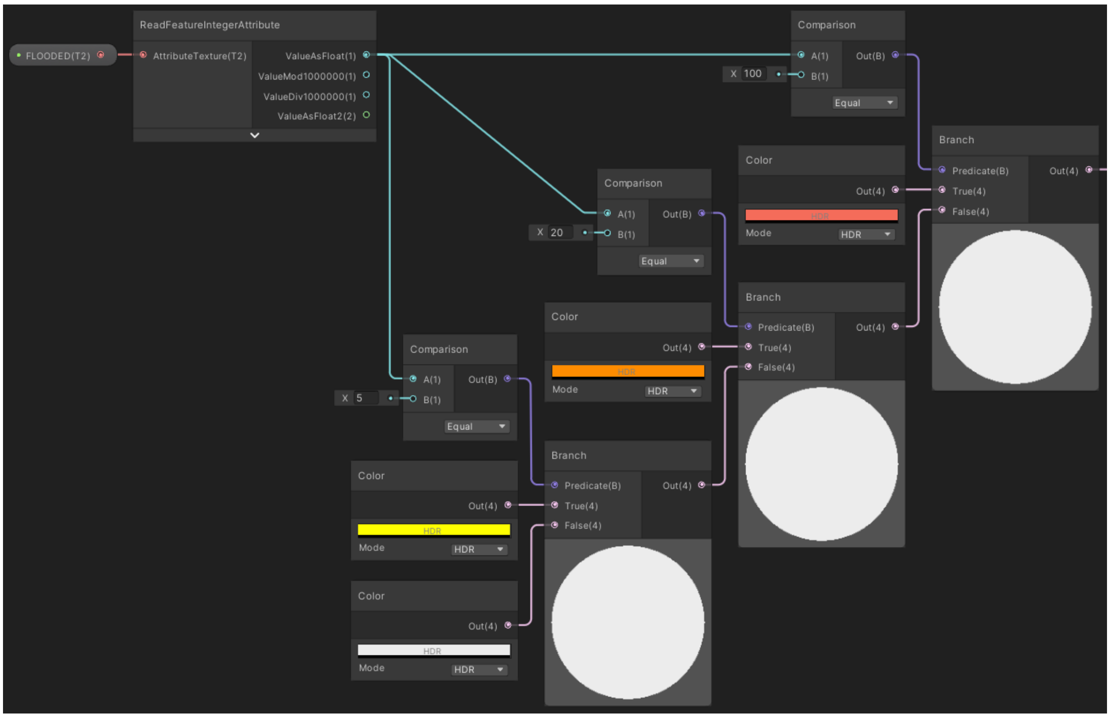


Figure 8: Unity flood shader logic for outputting the base colour of the material.

4.4.3 VR movement

The default camera controller component provided by Esri allows for the movement of the camera object in three directions: up-down, left-right, and forward-backward. The camera rotation happens through the mouse, with a raycast function identifying where the mouse cursor is clicking on the map and rotating the view based on this position. Mouse scroll is also used for zooming in and out. This system is not usable with a VR headset because of the way the camera is rotated. To solve this problem, a different way of rotating it has to be implemented. The following section does not describe how to add and move visible controller objects.

The location component, created by the setup script and added to the camera at runtime, deals with the camera's location and rotation. The expected result of 3D movement in VR is that the player will move either in the direction it is looking in or in the direction the controller is pointing in. For 2D movement, the direction of the joystick is sufficient, but this is not applicable here. As such, the heading and the pitch, respectively the angle between the default camera's direction on the horizontal plane and the vertical plane, must be determined from the look direction and forwarded to the location component. The roll can be ignored because it is unnecessary for smooth movement and could increase dizziness. Using the default component, the VR camera can be used to look around, but the game object will be pointing in the same direction and it will thus only be possible to move in that direction vector.

```
void FixedUpdate()
{
      var locationComponent =
origin.Camera.GetComponent<ArcGISLocationComponent>();
      var heading = origin.Camera.transform.eulerAngles.y;
      var pitch = origin.Camera.transform.eulerAngles.x;
      if (heading <= 180f) {
            pitch = (90f - pitch);
      } else {
            pitch = (90f + Math.Abs(pitch - 360f));
      }
      locationComponent.Rotation = new Rotator(heading, pitch, 0);
}
```

The code snippet above is the core of the VR controller. First, the logic happens within a `FixedUpdate()` function, running at a fixed frame rate instead of the default `Update()`, which has a variable frame rate dependant on the computer running it. This is necessary for the script to work as otherwise the movement would simply not happen. Three local variables are then initialised: the `locationComponent` of the *XR Origin* camera, defined at runtime and whose rotation will be modified, and the `heading` and `pitch` of the main camera through their Euler angles. The pitch has to be corrected depending on the look direction for it not to be reversed. Finally, the heading and corrected pitch can be fed into the location component to update the look direction. The rest of the code comes from the *Camera Controller* component, with unneeded functions like mouse raycasting or scrolling removed. The keyword var is used for defining the type of variables: they could be strongly typed but it is already used almost everywhere in the source code of the ArcGIS Maps SDK, providing more consistency and readability across the code-base. Instead of getting the camera from the *XR Origin* object, it is also possible to use `Camera.main.gameObject` to get the main camera directly.

The hierarchical placement of components between the various game objects is important. Incorrect placement of the main camera can lead to visual artefacts. For instance, by default, creating an *XR Origin* in Unity automatically creates a game object called *Camera Offset* as a child, and the main camera is located as a child of this camera offset, i.e., there is one empty object between the camera. If the *Map View* component is not located on the same object as the *XR Origin* and the camera is not directly a child of that, this will cause flickering when getting near the 3D object, possibly since the *HPTransform* component will not be updated correctly. Furthermore, it must be noted that the Maps SDK does not yet offer mesh collision for its layers, meaning it will be possible to fly through the terrain and buildings.

4.4.4 VR interaction

In order to navigate between the different cities, some level of interaction is required. As more than two cities are available, a dropdown menu is an easy way to choose between different choices. The dropdown can either be part of a menu or always be visible if it does not take up a lot of space. In any case, a canvas game object must be created containing the menu interactable objects as children and an event system object for input detection. In parallel, a menu script must be created to contain the function to be executed. The dropdown object can then be configured to contain the different cities and map each button to a menu function: when switching cities, the function will change the scene.

In addition, the canvas can be rendered either in *Screen Space - Camera* mode or *World Space*. The latter will display the menu in-game as a world object: it will appear at a fixed position like a floating sign. The former will display the UI in front of the camera, no matter where it is looking. Because this implementation is exploratory, in 3D, and over a large area, it is impractical to have a world space canvas at a fixed point and the only solution would be to create a new object in front of the camera every time the menu is open. Rather, the screen space camera overlay is good enough; its usual disadvantage is the flickering and difficulty in reading it in VR, though due to its simplicity it is not an issue and provides the required functionalities in a simple manner.

Lastly, the event system object should use the *Input System UI Input Module* component. This will make it use the new input system, which is more complicated to implement but offers more flexibility in terms of input mapping once this is done. The sample XR Interaction Toolkit input action map can be imported from the package manager, then modified to add a new UI action map with the navigate, submit and cancel action mapped respectively to the primary 2D axis of the right hand XR controller joystick, its primary button, and its secondary button.

5 CONCEPTUAL EVALUATION

The ArcGIS Maps SDK is already a usable integration solution for digital twin development. Despite its strengths, this tool shows a few weaknesses, some by design, others due to its pre-release state. Therefore, it is important to be aware of alternative tools and methods that can achieve similar goals, in addition to evaluating the implementation and gathering feedback from users. In the following chapters, three alternatives will be presented and explored while the previously implemented tool will be evaluated through a questionnaire.

5.1 Alternatives to the ArcGIS Maps SDK

5.1.1 Cesium for Unreal

Similar to Unity, Unreal Engine is a multi-platform commercial game engine developed by Epic Games and focuses on first-person 3D games. It is free with a royalty model and its source code is available. It uses C++ and visual scripting through “blueprints” for development (Epic Games, 2022). Unreal Engine 5, released in 2022, is the latest version available. It is for this engine that Cesium for Unreal was developed.

Cesium GS describes itself as a "platform for 3D geospatial”: it is a company that offers various free and paid services. One of these is Cesium Ion, a paid platform with a free tier for hosting and delivering 3D content in the cloud (Cesium GS, 2022). Similar to ArcGIS Online in that regard, it supports imagery and elevation layers, though the main difference is the additional support for 3D tiles. This open 3D tiles specification, an OGC community standard (Open Geospatial Consortium, 2019), defines a tile format optimised for delivering 3D content on the web where bandwidth is the main constraint. The Cesium for Unreal plugin combines the functionalities of Unreal and Cesium Ion by streaming geospatial content on a planetary scale, allowing it to be further visualised and developed into a fully-fledged gamified experience.

Figure 9: Unreal Engine 4 running the Cesium for Unreal plugin. The scene depicts a combined elevation and imagery layer for Brno. The Cesium Ion watermark is visible since the content is streamed from this platform without a commercial license.

By providing these integration capabilities, Cesium for Unreal positions itself as a major tool in building future digital twins. Streaming precise geospatial data for the entire world is precisely the type of capability needed for that. Development is ongoing and the various minor issues or bugs that users can encounter are being addressed promptly and publicly (Cesium GS, 2020).

The main downside of this tool is its incompatibility with game engines other than Unreal. It is linked to the engine's downsides, such as difficulty in developing mobile applications or the higher learning curve compared to Unity, illustrated by the complexity of C++ compared to C#. It is also tricky to export 3D scenes from ArcGIS Pro directly to Cesium Ion, as multipatch shapefile are not supported. This would require the additional hassle of using a tool such as FME to convert them to glTF, the preferred format for Cesium Ion, or another one like FBX or CityGML. Lastly, and despite its support for open standards and interfaces, Cesium Ion remains a commercial platform: the free tier only allows for 5GB of storage with restrictions on commercial use and funded educational projects.

5.1.2 Interoperability in Unity: Digital Twin ESRIN

Over a period of five months in 2021, the Digital Twin ESRIN prototype was developed in collaboration with the ESRIN/ESA Phi-Lab. It formed the preliminary work and main idea for this thesis, which leverages the know-how and resources involved. Its goal was to demonstrate functionalities for the visualisation of and interaction with environmental data in an XR environment representing a local digital twin of the ESRIN site (Deligant, 2021). Using Agisoft Metashape, a photogrammetry software, 3D models of the site of various sizes and quality were generated using drone images (Agisoft, 2022). After this step, they were directly imported into Unity. A simple first-person movement system was added to navigate within the game, besides other mandatory elements such as a main menu. REST APIs were also used to integrate live air quality and vegetation health data into Unity through a canvas, as shown in the code snippet below. Once this work was ready, the VR movement system was added on top with OpenXR as an interface.

```
public static string GetAPIData(string API_URL, string APIKey = null) {
      HttpWebRequest request =
(HttpWebRequest)WebRequest.Create(API_URL);
      if (APIKey != null) {
            request.Headers.Add("Authorization", "APIKey " + APIKey);
      }
      HttpWebResponse response = (HttpWebResponse)request.GetResponse();
      StreamReader reader = new
StreamReader(response.GetResponseStream());
      string json = reader.ReadToEnd();
      response.Close();
      reader.Close();
      return json;
}
```

Several limitations were encountered during the integration of the photogrammetry models. Only a handful of 3D formats are supported by Unity, such as FBX or Wavefront OBJ (Unity Technologies, 2022). However, these formats are not ideal for large, complex, and detailed models: their size is measured in hundreds or thousands of megabytes, not counting the texture files which can reach the same size. Detailed models and textures are required for an immersive experience, especially for a first-person walkable environment, since the player can get very close to the textures. The Unity asset import pipeline is not optimised for very large models. Their initial import can take hours and any modification to the model's setting will require re-importing.

Figure 10: Comparison of two pictures showing a part of ESRIN within Unity at different texture resolutions (top: 8192x4, bottom 4096x4). The reduction by a factor of four leaves bricks, leaves, and grass completely smoothed and reduced the file size for each individual texture from 256MB to 64MB.

Another tentative consisted in importing PLATEAU data for the entire city of Tokyo. Project PLATEAU aims to create 3D models of major Japanese cities and make them available for free for research and commercial purposes (MLIT Japan, 2020). The models are LOD1 in the FBX format, theoretically lightweight with no textures and natively supported by Unity. Unfortunately, direct import is also impractical as it takes about 24h for the entire city, a time during which no other action can be performed within the editor. Once that is done, opening the folder where the imported assets were located also takes a very long time, possibly because the editor needs to generate asset miniatures by reading the models.

All in all, this shows that importing 3D models of large areas into Unity only works well for areas of a limited size. This method does not scale well beyond an area covering a few buildings with detailed textures, i.e., for files larger than a few gigabytes. For these reasons, PLATEAU data was not used in the implementation.

There are still some advantages to the direct import approach. Interoperability is possible on a limited scale without georeferencing. If the requirements take into account the limitation of the technologies, game engines are mature enough to be usable for technical and scientific applications without necessitating a background in game development. For instance, integrating real-time thematic data is straightforward and produces usable results. Lastly, VR was confirmed as a powerful and accessible tool for visualisation.

5.1.3 Godot, a FOSS option

Godot is a FOSS multi-platform game engine released in 2014. It provides an editor for 2D and 3D development, supports the C#, C++ and native GDScript language. Its open-source nature under an open license theoretically makes it the most extendable engine: by comparison, Unity is closed-source and Unreal Engine open-source with a commercial license. As of version 4, Godot natively supports OpenXR (Linietsky, 2022).

The asset import pipeline for large 3D models felt notably faster in Godot than in Unity, although no benchmarks were done. For instance, importing the Digital Twin ESRIN model of medium quality could take almost an hour in Unity, but only a few minutes in Godot. This is possibly because Godot natively supports glTF, a highly compressed 3D format developed by Khronos Group aimed at improving the transmission of 3D scenes over the web (Khronos Group, 2021). The glTF format is recommended, though others such as OBJ and FBX are also supported. As such, the interoperability capabilities of Godot appear similar to that of Unity.

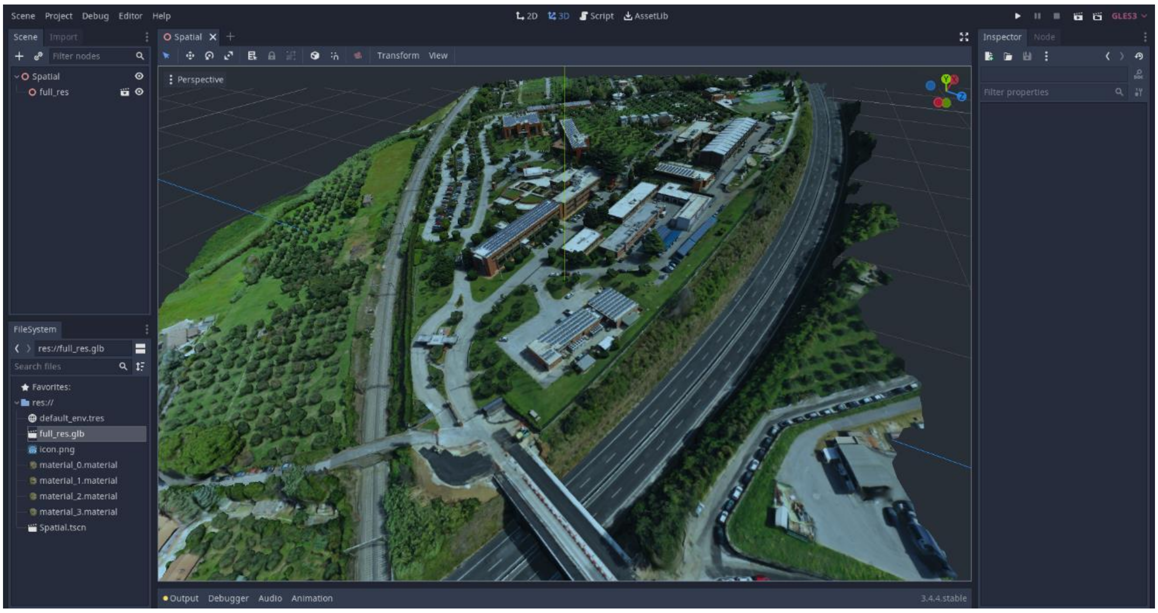


Figure 11: Full resolution glTF model of Digital Twin ESRIN imported in a Godot scene.

Godot has no built-in support for the integration of geospatial data, but it can be added via plugins. The geodot-plugin does exactly that: relying on GDAL, it allows georeferenced raster and vector data to be loaded in the engine, even in real-time (Institute of Landscape Development, Recreation and Conservation Planning, 2020). However, the plugin does not work out of the box, documentation is scarce, and overall learning and development require more work than Unity. All in all, Godot appears as a potentially attractive alternative to using Unity for interoperability work, although few to no major integration tools, such as for ArcGIS Online or Cesium Ion, are available.

5.2 User testing

5.2.1 Questionnaire design

Designing a questionnaire requires strong preparation to make it as neutral and attractive as possible. The form must be unambiguous, requiring appropriate question formats and a preliminary review. For instance, whenever a numerical scale is appropriate, having the scale go from zero to ten, instead of one to ten, provides a middle point for participants who neither agree nor disagree with it (Thwaites Bee, 2016). When the input is a number or a percentage, configuring the form to ensure only numerical values can be entered validates inputs. Avoiding ambiguities is also necessary to achieve the most accurate results: about Prague's model, not all buildings appearing on the basemap are also present as 3D objects. The questions on that city thus specify that "all buildings" include only buildings visible in the 3D model. Google Forms was selected due to its popularity, a tool most participants will be familiar with and allowing them to focus on the questions themselves. Testing happened over two consecutive days, with one batch of participants each day.

One type of content missing from the implementation is waypoints or markers for orientation. This would make it difficult for participants to convey localisation information or pinpoint a location. Considering that this was a requirement for the question on elevation, three waypoints were temporarily added to the project during the questionnaire. They are cylinders of three colours: purple, green, and blue, created as multipatch shapefile within ArcGIS Pro and exported as an additional scene layer package. Each had an attribute for their colour with a shader in Unity applying the right one to each. The participant can now mention a specific waypoint. Other temporary modifications to the application included disabling the UI, since each participant should try only one city in VR (with the other being in ArcGIS Online) and adding a shortcut to show and hide the waypoints manually.

The questions deal with the perception of flooding data, more broadly with the perception of GIS data through VR. Participants have to guess the percentage of flooded buildings of all colours or a specific one (orange) and are separated into two groups: one will do so for city A in VR and city B in ArcGIS Online, the other with city B in VR and city A in ArcGIS Online (Figure 12). This minimises the importance of the order in which cities are shown as the estimates will be very different. For cities viewed in VR, the three waypoints previously mentioned were placed across them at different heights and one question is about finding out which one sits on the highest point on the ground. Lastly, there are a handful of questions to provide background info on the participant's experience, asking about their opinion on the navigation in both cases and general feedback.

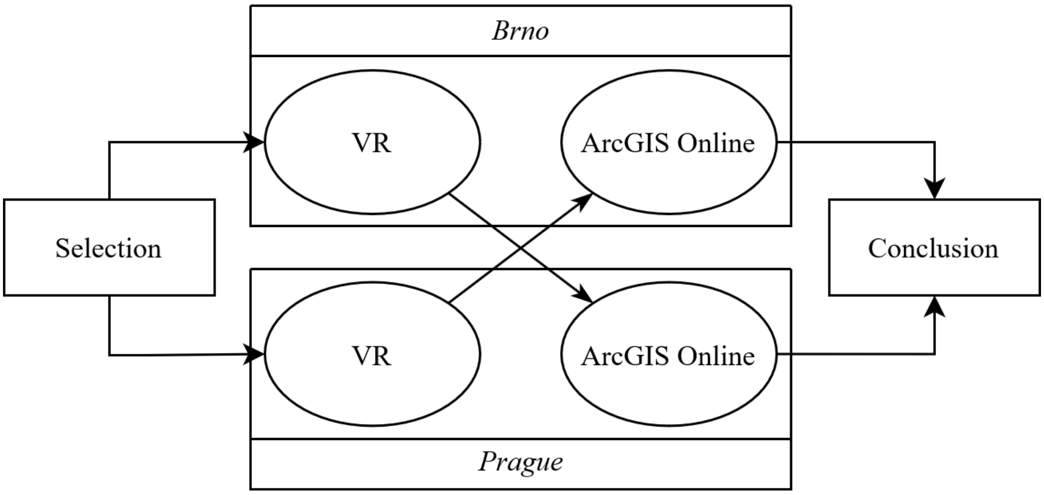


Figure 12: Schematic representation of the two possible questionnaire paths: each group sees one city in VR and the other in ArcGIS Online, while both share another common set of questions.

The hypothesis is that participants will overestimate the number of flooded buildings (of all colours) in VR because of its immersive aspect. The number of orange buildings (20-years floods) is expected to be overestimated in all cases, as the number of affected buildings is very small. Regarding the waypoints and their elevation, the correct one is not expected to be easily spottable without a careful examination. Lastly, it is expected that participants will find ArcGIS Online easier to control, though will prefer the fun aspect of VR.

Table 4 List of questions from the survey along with possible answers

Number	**Question**	**Answer**
1	What is your name? (ID)	*Text*
2	What group of questions will this be?	Brno VR/Prague ArcGIS Prague VR/Brno ArcGIS
3-6-9-11	During a 100-years flood (red), what percentage of the buildings are flooded?	*Number between 1 and 100*
4-7-10-12	During a 20-years flood (orange), what percentage of the buildings are flooded?	*Number between 1 and 100*
5-8	What waypoint is located on the highest ground?	Blue Purple Green
13	How much experience do you have with GIS?	No experience 1-5 years 5+ years
14	How much experience do you have with game engine development (Unity, Unreal)?	No experience 1-5 years 5+ years
15	Had you ever used ArcGIS Online?	Yes No

Number	Question	Answer
16	Had you ever used a Virtual Reality headset?	Yes, many times (5+) Yes, a few times (1-5) No, never
17	Were you broadly familiar with the cities beforehand?	Yes No
18	What system felt easier to navigate in terms of controls?	Oculus Quest ArcGIS Online
19	In VR, were you limited by the controls (speed, movement directions)?	Yes No
20	Do you see any added value in VR compared to ArcGIS Online?	*Text*
21	Do you have any other feedback?	*Text*

5.2.2 Participants demographics

Having a basic understanding of who the participants are is necessary to assess the validity of the questionnaire and prevent bias in the analysis of their responses.

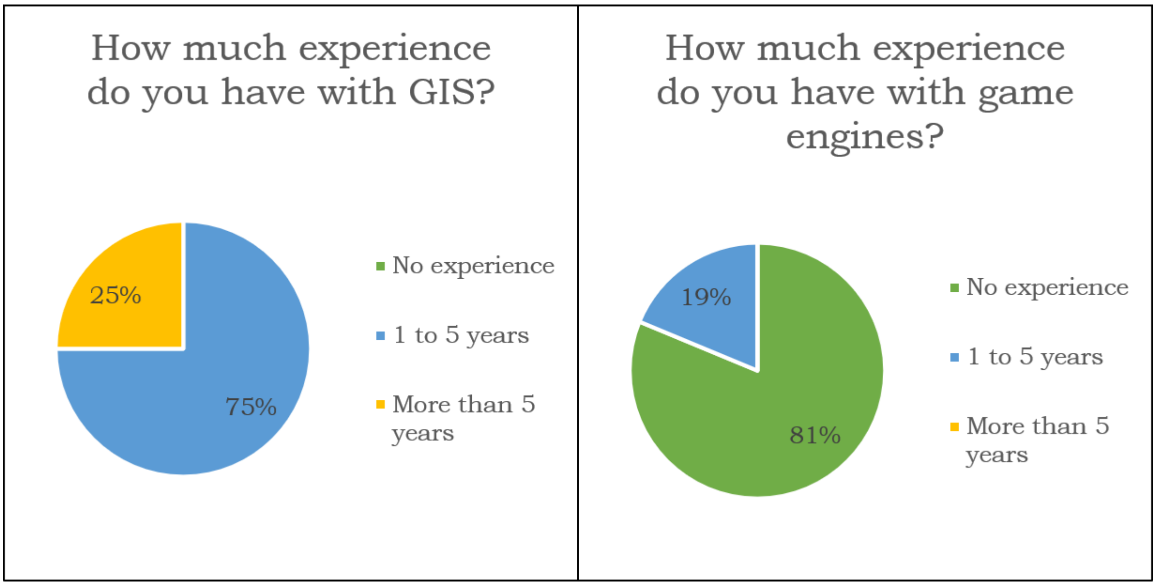


Figure 13: Pie charts showing the experience level of participants with GIS (left) and game engines (right).

All participants are expected to have at least one year of experience with GIS, with a few having more than five years of experience, while the majority are expected to have no experience with game engines. All were part of the Department of Geoinformatics at the Palacký University Olomouc: no diversity was required in that regard since the control test consisted of the ArcGIS Online questions. The work builds on top of this existing 3D visualization solution, plus users should be familiar with GIS to be able to provide meaningful feedback.

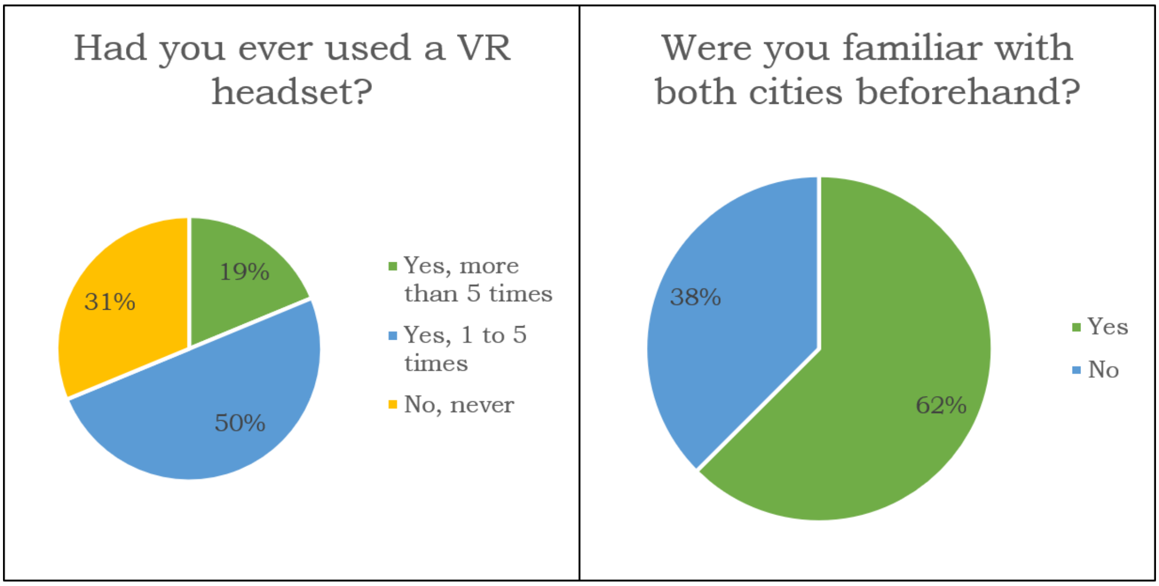


Figure 14: Pie charts showing the experience of users with VR headsets (left) and their background knowledge of both cities (right).

More than half of the participants had already tried a VR headset at least once, a handful had a lot of experience, and a third had never tried it whatsoever. Virtually all participants had already used ArcGIS Online, confirming its role as a control and validation part. Their name was collected initially and later converted to an identifier so that no personal information remained available.

Figure 15: Thesis supervisor Jan Brus wearing the Oculus Quest 2 during the questionnaire.

6 RESULTS

The objectives of this thesis were to analyse, test and describe the integration between GIS and game engines in the context of digital twins for disaster preparedness. Throughout two research parts, these goals were explored, providing a detailed explanation of the processes that lead to various relevant findings. This section presents them in a summarised manner.

6.1.1 Technical implementation

As the selected integration tool, the ArcGIS Maps SDK for Unity appeared powerful and capable despite its pre-release state and lack of important functionalities. All currently supported layer types were successfully integrated, including imagery basemap, elevation, and 3D objects scenes. The ability to stream files from ArcGIS Online with an API key adds flexibility and scalability, while local files can be convenient and reduce network strain. Advanced visualisation is not yet completely scalable: for instance, 3D attributes require defining a Unity shader and knowing the attributes of the layer in advance. It has a few peculiar characteristics that complexify development in Unity, most importantly the need for high-precision coordinates through the *HPTransform* component, meaning applications should be tailored to that tool or need to be converted to support it. With a 1.0 release expected in May 2022, the addition of new features should further increase its capabilities.

Developing a VR application in Unity is straightforward, thanks to the detailed documentation available and support for OpenXR, including interaction capabilities. Adapting it to the ArcGIS Maps SDK plugin required dealing with the aforementioned specificities. The movement script had to be reverse-engineered and adapted to work with an HMD.

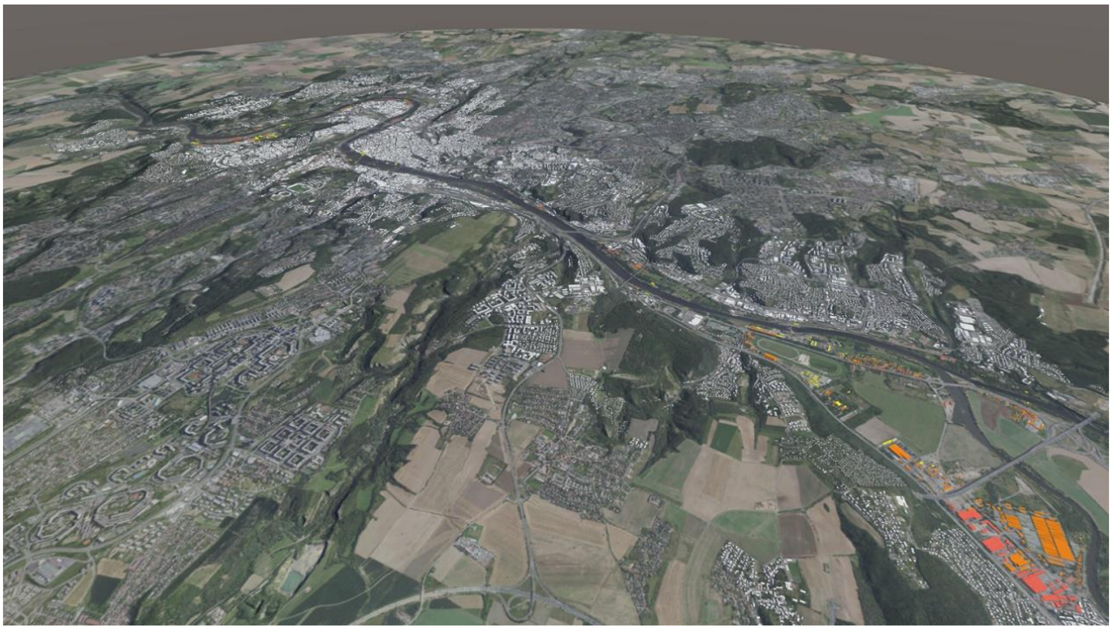


Figure 16: Prague as visible in the Unity application. The imagery basemap shows buildings that are not present in the 3D model.

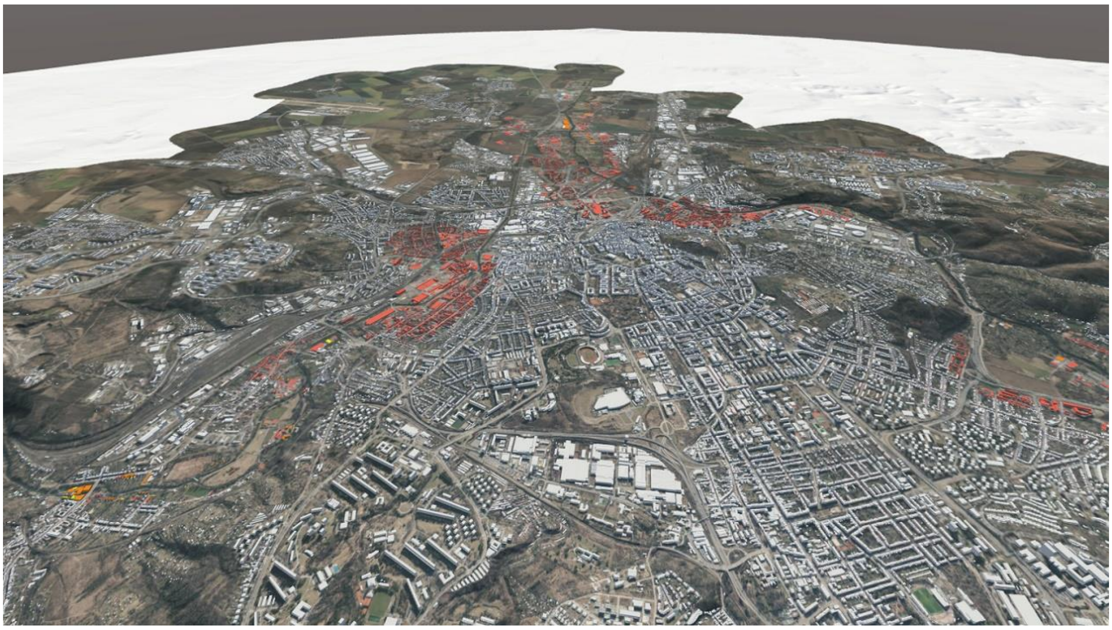


Figure 17: Brno as visible in the Unity application. The imagery's coverage does not extend to the whole area, leaving the outskirts in white with only the elevation displayed.

Available on the attached SD card, the output is the standalone VR application for the Windows OS, along with the project files necessary to modify and compile it for a different OS. It is ready to be used with any OpenXR-supported HMD and controller, although the application can also be compiled for use with a keyboard and mouse. The two largest cities in the Czech Republic, Prague and Brno, were implemented.

6.1.2 Alternatives methods and tools

Despite the powerful capabilities of the ArcGIS Maps SDK for game engines, namely the seamless integration of game engines with Esri platforms and interfaces, it is not suited to all use-cases. Reliance on this ecosystem is both a strength and a weakness, as the limited availability of formats and its commercial nature hinder it. There are nonetheless alternative integration tools and interoperability methods.

Cesium for Unreal is one of the most developed and accessible options, with its own set of advantages and disadvantages. By streaming content into Unreal from the Cesium Ion platform, it supports a variety of open formats and can scale to the entire globe. However, it is equally limited in terms of interfaces, with for instance no easy integration of Esri formats or the obligation to host the content on Cesium Ion, a commercial platform first and foremost.

Unlike the two previous examples, interoperability methods rely on the existing capabilities of game engines for reading geospatial data. Unlike Unity, Unreal Engine offers a georeferencing plugin, otherwise they are pretty similar in the formats they support, asset import pipeline performance, and scalability. The latter is often an issue due to the size of high-resolution and high-coverage models. Godot does a better job in that regard with native support for the streaming-optimised glTF specification, even though it is less accessible than its competitors due to limited documentation. Reliance on plugins and established engines is necessary for individuals or small teams without the resources to build a complete geospatial integration and visualisation system. Home-built plugins are an option for larger projects, especially on open-source engines such as Unreal Engine or Godot.

6.1.3 User testing

The goal of the questionnaire is to analyse the perception of 3D geospatial data in VR, compare it to the perception of 3D data on the web, and gather feedback on the implementation. Through two sets of questions, it inquired about how geospatial data is perceived, the ease of extracting information from a visualisation, and the potential biases caused by VR.

Table 5 Results from the questions on buildings count estimation for the city of Brno

Brno	**Flooded buildings (%)**		**Orange buildings (%)**	
	VR	**ArcGIS Online**	**VR**	**ArcGIS Online**
Correct value	10		1	
Average reply	23.6	12.6	7.1	2.9
Standard deviation	8.9	6.8	6.8	2.0

Table 6 Results from the questions on buildings count estimation for the city of Prague

Prague	**Flooded buildings (%)**		**Orange buildings (%)**	
	VR	**ArcGIS Online**	**VR**	**ArcGIS Online**
Correct value	3.4		1.2	
Average reply	5.4	7.0	2.6	3.0
Standard deviation	2.8	5.2	1.7	2.7

For virtually all questions, a majority of participants overestimated the number of buildings affected by floods. The difference is greater in Brno with an overestimation of 9% on average, compared to Prague with only 3%. The overestimation was the same in VR and ArcGIS Online at 6%. According to the standard deviation, the answers were more disparate for Brno than for Prague. The most interesting distinction appears when comparing the differences between VR and ArcGIS Online in each city. Participants overestimated affected buildings in Brno in VR compared to ArcGIS Online by a factor of 2, while they slightly underestimated the affected buildings in Prague in VR compared to ArcGIS Online (Table 5, Table 6).

VR thus appears to act as a perception amplifier: if many buildings are affected, this fact will be exaggerated; if few buildings are affected, it will be diminished. Due to the limited sample size, more research is necessary to confirm this trend.

Table 7 Height of all waypoints for both cities

Waypoint/City	**Brno**	**Prague**
Blue	227m	192m
Green	197m	187m
Purple	208m	179m

The questions on localising waypoints provided very equally interesting insights on the usefulness of the system. As seen in Table 7, the blue waypoints were the highest ones in both cases.

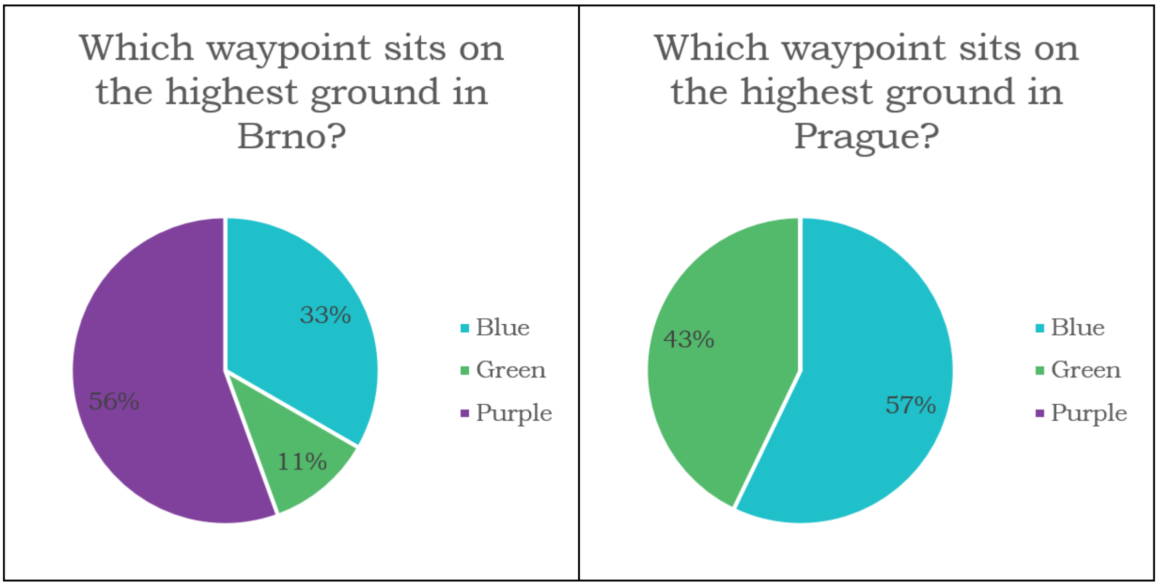


Figure 18: Pie charts showing the responses on which waypoint was the highest in Brno (left) and Prague (right).

According to the results for Brno and Prague, it was indeed possible to differentiate between the elevation represented by these towers (Figure 18). Only a third of participants chose the correct answer for Brno, although this can be explained by the presence of nearby hills giving the purple waypoint the illusion of height, leading more participants to select it. For Prague, the difference in height between the highest and second-highest was only 5m, yet a majority chose the correct answer and the lowest waypoint was never selected. Overall, the accuracy appears quite good with the highest or second-highest possibility selected, their clear differentiation requiring in one case to overcome an unconscious bias and in the second to spot a 5m difference in elevation. This finding is of interest as it demonstrates how VR is able to accurately convey subtle relative differences in elevation, a thematic layer extremely common in 3D visualisation, even though geographic features such as hills can cause bias.

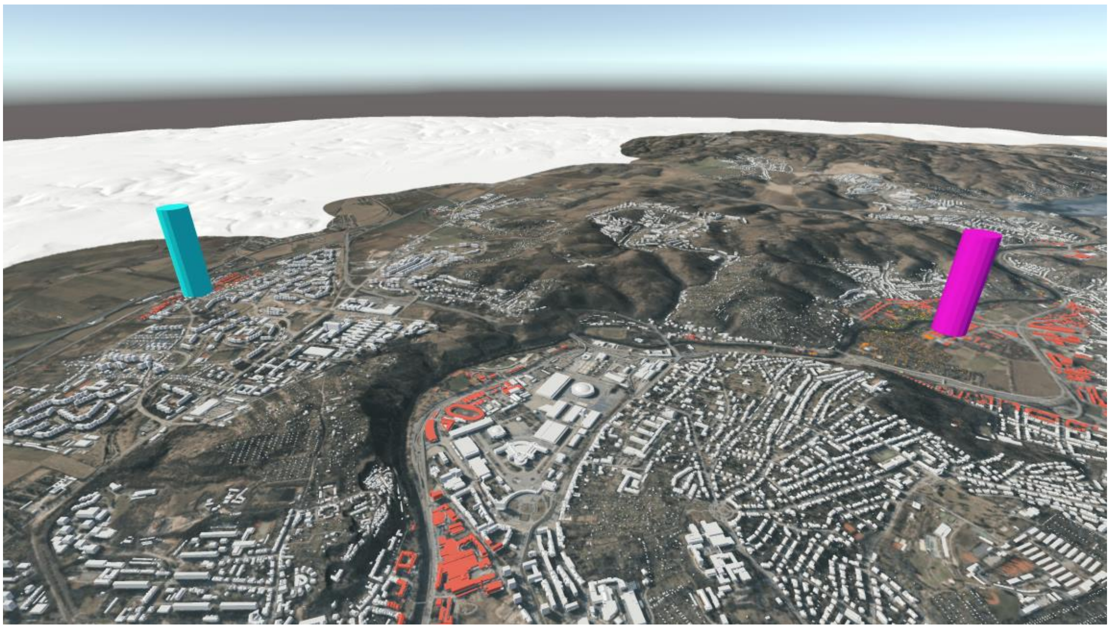


Figure 19: Blue and purple waypoints over the city of Brno as visible within Unity. The hills in the middle-ground give an illusion of height to the purple waypoint.

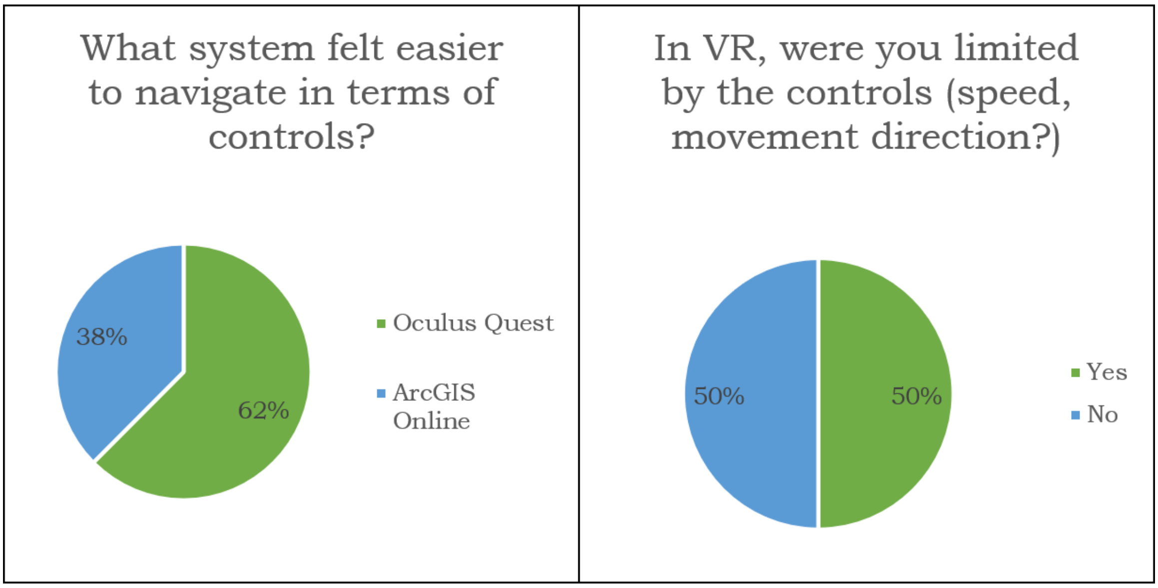


Figure 20: Pie charts showing the feedback of users on their navigation preferences (left) and complaints (right).

Disproving the initial hypothesis, two thirds of the people preferred the navigation through the Quest 2 to using ArcGIS Online. Despite this slight preference, half of all participants found themselves limited by the controls in their exploration of the data. Speed was the most common complaint, as seen in the textual feedback, luckily it is a duo of variables that can be easily tweaked. Movement directions are part of broader possible improvements on navigation, which would include using the whole controller and its position rather than just its joystick for movement. This would also be relatively easy to implement.

Table 8 Words appearing more than twice in the textual feedback on the advantages of VR, with similar words merged (e.g., view-viewing)

Word	**Frequency**
View	6
Angle	3
Immersion	3
Wide	3
Bird	2
Engagement	2
Realism	2

The results of the feedback section on the added value of VR give us information on advantages that are not necessarily evident nor easy to convey through text or a standard 2D screen. The word “view” is found in more than a third of all answers. It is often present in combination with “angle”, “wide” and “bird”, through the sentences “wide view angle” and “bird view”. Indeed, when using a VR headset, our eyes can enjoy a viewing angle much wider than when looking at a computer screen, in fact as wide as in real life. In turn, this improves the “immersion”, “engagement” and “realism” aspects, which are all found in the feedback responses.

Lastly, in terms of general feedback, a word frequency analysis is not relevant due to the low number of answers and their heterogeneity. At least two participants felt dizzy when trying the VR headset, a symptom of motion sickness. The aforementioned network issues affected a handful of participants, degrading their experience. Some suggested increasing the movement speed and moving in the direction of the controller as opposed to that of the camera.

There are various elements of interest noted while observing the participants that could not be formally recorded:

- When asked to estimate the percentage of flooded buildings, few participants looked at the cities from “nadir”, in VR as in ArcGIS Online, to try replicating the experience of a static map
- The default extent is crucial, especially in ArcGIS Online, as some will not try to look beyond what is initially visible or move around a lot, thus missing some parts of the cities
- The control scheme (forward movement direction aligned with the camera look direction) was unintuitive for some people, who did not often look behind them or up and down, while for others this was perfectly intuitive

Overall, geospatial data is well perceived in game engines through VR. The information was conveyed to users with a GIS background in an entertaining manner and the current implementation proved sufficient for that goal. The application could nonetheless be tweaked to be more user-friendly and accommodate more interaction functionalities. Besides, the importance of questionnaire preparation and setup was emphasised by the bandwidth issues present initially and the ambiguity of some questions.

7 DISCUSSION

During the analysis and implementation of this project, various technical and conceptual issues were encountered. This had to be solved or worked around to fulfil the initial objectives. This section describes the main challenges of the chosen approach, possible solutions, and future outlook.

Application development

As part of a digital twin system, GIS and game engine integration is a single component, not usable in and of itself. It requires at least input data and output visualisation. Game engines have made the development of said visualisation accessible to anyone without specialised skills, yet the process of creating a usable application for professional use is complex. For instance, no main or settings menu was implemented, the control scheme is static, and interaction is limited. Building these capabilities is a matter of resource investment with specific goals and use-cases. Unity, Unreal Engine, and Godot all make this possible with various levels of preliminary work required for each. The need for double precision for coordinates forces the ArcGIS Maps SDK for Unity to replace the default *Transform* component with its own high-precision transform. This limits the transferability of applications as they would require some level of modification to adapt to this change.

Plugin features and interoperability status

While powerful, both the Maps SDK and Cesium for Unreal plugin have limits to their capability which could prevent the implementation of some use-cases. Relying on them creates a dependency on the company behind them. In the case of Esri, it is a dependency on an entire ecosystem with its own standards and interfaces. In addition, the possibility of adding or requesting custom features is not guaranteed, for instance mesh collision in the Maps SDK. For large and complex projects, unless the chosen use-case fits within the capabilities they offer, relying on them is a risk. Finally, their commercial nature means an absence of up-front development costs and an increase in the cost of sustaining the product. Scalability could become an issue with the dependency moved to the ability of the host platform to host and update content and how well the plugins could deal with it.

Examples of important missing features include the lack of support for OGC web services or the absence of vector features and styling. Client-to-server communication has not been explored but should be possible in the future, allowing for advanced interactions with the real-time transmission of client data, such as their position. Simulations could also be a great use of the capabilities of game engines, helping to breach the barrier between serious play and serious game.

Conceptual aspect

Game engines have been shown to be perfectly usable for geospatial data visualisation. VR can amplify the message depicted by the data while being very entertaining to use. However, depending on the use-case, other ways of conveying information might be more appropriate, like 3D web visualisation, images, or text. A lot of progress has been made in the accessibility of VR, but it still requires a relatively expensive HMD and, if the HMD requires it, a performant computer for rendering. Unlike the web and as of yet, VR is not a technology for the masses.

In-person testing

For participants to fill in the questionnaire, they had to try the VR system, meaning they should come one by one to test the system and interact with the tool. This limits the number of participants due to the constraints of physically having many people and the time each takes, at about ten minutes.

A number of technical limitations were encountered during testing. First of all, the Oculus Quest 2 requires setting up a guardian boundary, which is a space manually defined in which the user can move without encountering any obstacles. This did not work well as the boundaries were not recognised every-time in-between participants and had to be manually reset. The HMD also used Air Link, a functionality allowing it to be linked to a computer without a cable. While the added convenience is of great importance, it requires the computer and the headset to be on the same local network, is limited by the performance of the router, and limits the battery life to less than two hours. Concretely, the viewing experience was not always smooth, with the view sometimes stuttering due to network congestion. Lastly, the application used 3D scene layer packages hosted on ArcGIS Online, a requirement to make it work on different computers without manually inputting the file path. However, as the scenes can be hundreds of megabytes large, this further strangled the network bandwidth, meaning local files were used in the second batch of participants for a smoother experience.

All of these points mean that a great deal of effort has to go into the preparation to ensure a smooth experience for all volunteers. In addition, two questions were found to be still ambiguous after testing. In question 17, "broadly familiar" meant whether participants had general background knowledge of the cities, their layouts, and geography, however this required an explanation. Question 20 also implied that benefits to VR existed due to the textual nature of the answer; only one person wrote that there was no benefit.

Relation to Copernicus

As a study program requirement, investigating possible links to Copernicus data. No direct use was made in this thesis, but various are imaginable. Imagery layers from the space segment can be directly visualised through layers hosted on the public ArcGIS Living Atlas, such as Sentinel-5P Ozone concentration or Sentinel-2 views, and products derived from the various services can also be integrated as non-imagery raster tile layers, such as the Corine Land Cover from the Copernicus Land Monitoring Service. One product particularly relevant to disaster management is the Global Flood Awareness System (GloFAS) from the Copernicus Emergency Management Service, although it uses OGC web services that cannot yet be integrated. As a methodological work, the framework developed allowed such data to be easily implemented, providing the right interfaces.

Digital twins

Designing a full digital twin is a complex endeavour requiring numerous layer components working together to produce a virtual replica of the chosen element. This work did not aim at creating an entire digital twin. Rather, it provides insights into using innovative technologies on two interlinked components: data integration and visualisation. By assessing the usability of these new tools and methods, future work can build upon this research to get a preliminary understanding of their strengths and weaknesses to determine whether they would be usable for their use-case.

8 CONCLUSION

Throughout this thesis, the state of integration between GIS and game engines for digital twins in the context of disaster preparedness was investigated and evaluated. XR visualisation was an integral component of the integration as it made use of Unity's capabilities to produce a standalone, testable product. This current state is moving rapidly, with many tools in development or being released. They add powerful capabilities on top of the ones already present in game engines, making them usable for the creation of digital twins.

Interoperability between GIS and game engines is relatively limited as it stands. Developers have the possibility of building their own capabilities though with a lot of resources investment. For instance, using the ArcGIS Maps SDK for Unity provides various integration features, speeding up development and blurring the line between fusion and integration. On top of that, VR can emphasise the message being propagated via the visualisation while keeping users entertained and immersed. This is particularly relevant for disaster preparedness through raising awareness, increasing disaster understanding and propagating the information to decision-makers. Notwithstanding, current integration tools have some important weaknesses, most notably their commercial aspect and limitations in the supported formats and interfaces. Besides, building a fully gamified experience with a VR element requires many resources invested in application development, UI, and hardware accessibility.

Future work should focus on defining whether integration tools are suited for their use-cases and implementing them in a digital twin system, for example as a main visualisation tool or as an added outreach instrument. Interoperability with open interfaces such as OGC web services could allow for more scalable and interactable applications but will require custom implementation or dependency on external providers. Moreover, while VR has been thoroughly explored, AR and MR have numerous advantages that could be put to use, most importantly improved interactivity and accessibility at the cost of degraded immersion.

Having taken part in the discovery and exploration of recent integration processes at an early stage of their development, this thesis will promote and facilitate their use for tasks requiring the interaction and visualisation possibilities offered by game engines. From products and services designed and built in entirely different worlds can stem new combinations and solutions to many problems, ranging from the dissemination of disaster preparedness information to the simulation of past and future hazards and their consequences. It is thus another stepping stone in the advance towards the realisation of Destination Earth and its modelisation of the entire planet.

REFERENCES

AGISOFT. Agisoft metashape user manual professional edition, version 1.8. Online. Agisoft LLC, 2022. Available from: https://www.agisoft.com/pdf/metashape-pro_1_8_en.pdf. [viewed 2022-04-22].

ALVAREZ, Julian, and Damien DJAOUTI. An introduction to Serious game Definitions and concepts. In: Philippe FAUQUET-ALEKHINE, and Luc SOLER (eds.). Proceedings of the serious games & simulation workshop, pp. 11–15. Paris, France. Paris: LARSEN Science 2011 - LAboratoire de Recherche pour les Sciences de l'ENergie, 2011. ISBN 9782954143002.

BLACKSHARK.AI. Blackshark.ai receives epic mega grant | blackshark.ai. Online. Blackshark.ai. 2021-11-02. Available from: https://blackshark.ai/blackshark-ai-receives-epic-mega-grant/. [viewed 2022-04-28].

BUYUKSALIH, I., S. BAYBURT, G. BUYUKSALIH, A. P. BASKARACA, H. KARIM, et al. 3d modelling and visualization based on the unity game engine – advantages and challenges. Online. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-4/W4 (November 2017), pp. 161–166. ISSN 2194-9050. Available from: https://doi.org/10.5194/isprs-annals-iv-4-w4-161-2017. [viewed 2022-04-17].

CARBONELL-CARRERA, Carlos, Jose Luis SAORIN, and Dámari MELIÁN DÍAZ. User VR experience and motivation study in an immersive 3D geovisualization environment using a game engine for landscape design teaching. Online. Land, vol. 10 (May 2021), no. 5, p. 492. ISSN 2073-445X. Available from: https://doi.org/10.3390/land10050492. [viewed 2022-04-17].

CESIUM GS. Cesium-unreal: bringing the 3D geospatial ecosystem to unreal engine. Online. GitHub. 2020. Available from: https://github.com/CesiumGS/cesium-unreal. [viewed 2022-04-22].

CESIUM GS. Cesium ion. Online. Cesium. 2022. Available from: https://cesium.com/platform/cesium-ion/. [viewed 2022-04-22].

COZZI, Patrick. Announcing our collaboration with epic games to create cesium for unreal engine. Online. Cesium. 2020-06-20. Available from: https://cesium.com/blog/2020/06/04/cesium-for-unreal-engine/. [viewed 2022-04-28].

COPELAND, Baden, Josh KELLER, and Bill MARSH. What could disappear. Online. The New York Times Web Archive. 2012-11-24. Available from: https://archive.nytimes.com/www.nytimes.com/interactive/2012/11/24/opinion/sunday/what-could-disappear.html. [viewed 2022-04-17].

CZECH LAND SURVEY OFFICE. ZABAGED® - altimetry - DMR 5G. digital terrain model of the czech republic of the 5th generation (DMR 5G). Geoportal ČÚZK. 2017-01-01.

CZECH LAND SURVEY OFFICE. Orthophoto of the czech republic. Geoportal ČÚZK. 2022-12-31.

CZECH REPUBLIC. Parlament České republiky. Vyhláška o způsobu a rozsahu zpracovávání návrhu a stanovování záplavových území a jejich dokumentace. Online. Regulation no. 79/2018 Sb. of 2018-05-18. Available from: https://public.psp.cz/en/sqw/sbirka.sqw?cz=79&r=2018. [viewed 2022-04-17].

DALRYMPLE, Tate. Flood-frequency analyses, manual of hydrology: part 3. Online. Water Supply Paper 1543-A. Washington: US Geological Survey, 1960. Available from: USGS Publication Warehouse, https://doi.org/10.3133/wsp1543a. [viewed 2022-04-17].

DELIGANT, Anatole. ID 146 digital twin ESRIN. Online. 2021-09-18. Available from: YouTube, https://www.youtube.com/watch?v=CO-TwjIwUDc. [viewed 2022-04-17].

DELTARES SYSTEM. Delft3D 4 suite (structured) - deltares. Online. Deltares. 2011-06-28. Available from: https://www.deltares.nl/en/software/delft3d-4-suite/. [viewed 2022-04-17].

EPIC GAMES. Unreal engine 5 documentation. Online. Unreal Engine 5 Documentation | Unreal Engine Documentation. 2022. Available from: https://docs.unrealengine.com/5.0/en-US/. [viewed 2022-04-22].

ESRI. ArcGIS maps SDK for unity | arcgis developer. Online. ArcGIS Developer. September 2021. Available from: https://developers.arcgis.com/unity-sdk/. [viewed 2022-04-17].

FAN, Chao, Cheng ZHANG, Alex YAHJA, and Ali MOSTAFAVI. Disaster City Digital Twin: a vision for integrating artificial and human intelligence for disaster management. Online. International Journal of Information Management, vol. 56 (February 2021), p. 102049. ISSN 0268-4012. Available from: https://doi.org/10.1016/j.ijinfomgt.2019.102049. [viewed 2022-04-18].

FILHO, Jorge A. Wagner, Wolfgang STUERZLINGER, and Luciana NEDEL. Evaluating an immersive space-time cube geovisualization for intuitive trajectory data exploration. Online. IEEE Transactions on Visualization and Computer Graphics, vol. 26 (January 2020), no. 1, pp. 514–524. ISSN 2160-9306. Available from: https://doi.org/10.1109/tvcg.2019.2934415. [viewed 2022-04-17].

FREIRE, Luciana Lopes, Pedro Miguel AREZES, and José Creissac CAMPOS. A literature review about usability evaluation methods for e-learning platforms. Online. Work, vol. 41 (2012), pp. 1038–1044. ISSN 1875-9270. Available from: https://doi.org/10.3233/wor-2012-0281-1038. [viewed 2022-04-28].

FRISCH, Dieter, and Martin KADA. Visualisation using game engines. Online. In: Orhan ALTAN (ed.). XX ISPRS congress, commission V, pp. 621–625. Istanbul, Turkey, 2004-07-12 – 2004-07-23. Hannover: International Society for Photogrammetry and Remote Sensing, 2004. Available from: https://www.isprs.org/proceedings/XXXV/congress/comm5/papers/627.pdf. [viewed 2022-04-17].

HERWIG, Adrian, and Philip PAAR. Game engines: tools for landscape visualization and planning? In: Erich BUHMANN, Ursula NOTHHELFER, and Matthias PIETSCH (eds.). Trends in GIS and virtualization in environmental planning and design, pp. 162–171. Heidelberg: Wichmann Verlag, 2002. ISBN 389907386X.

INSTITUTE OF LANDSCAPE DEVELOPMENT, RECREATION AND CONSERVATION PLANNING. Geodot-plugin: godot plugin for loading geospatial data. Online. GitHub. 2020. Available from: https://github.com/boku-ilen/geodot-plugin. [viewed 2022-04-23].

INTEL. Virtual reality vs. augmented reality vs. mixed reality - intel. Online. Intel. 2020. Available from: https://www.intel.com/content/www/us/en/tech-tips-and-tricks/virtual-reality-vs-augmented-reality.html. [viewed 2022-04-17].

IPR PRAHA DATABASE. Záplavové území (Q100). Online. Úvod | Geoportál hl. m. Prahy. 2019-02-25. Available from: https://www.geoportalpraha.cz/en/data/opendata/43744DE1-7128-4389-811A-AEEB9E0CE74F. [viewed 2022-04-17].

IPR PRAHA DATABASE. Budovy 3D. Online. Úvod | Geoportál hl. m. Prahy. 2022-03-31. Available from: https://www.geoportalpraha.cz/en/data/opendata/44EE8B0A-641A-45E8-8DC9-CF209ED00897. [viewed 2022-04-17].

JOSIAH, N. R., D. P. C. LAKNATH, and S. ARAKI. Assesment of tsunami preparedness measures in east coast of sri lanka based on 2004 tsunami event. Online. In: Yasuyuki SHIMIZU (ed.). Proceedings of the 22nd IAHR-APD congress 2020. Sapporo, Japan. Beijing: International Association for Hydro-Environmental Engineering and Research, 2020. Available from: https://iahr-apd2020.eng.hokudai.ac.jp/htdocs/static/mirror/proceedings/pdf/6-5-2.pdf. [viewed 2022-04-17].

KALISCHUK, Andrea R., Stewart B. ROOD, and John M. MAHONEY. Environmental influences on seedling growth of cottonwood species following a major flood. Online. Forest Ecology and Management, vol. 144 (April 2001), no. 1-3, pp. 75–89. ISSN 0378-1127. Available from: https://doi.org/10.1016/s0378-1127(00)00359-5. [viewed 2022-04-17].

KHRONOS GROUP. OpenXR - High-performance access to AR and VR —collectively known as XR— platforms and devices. Online. The Khronos Group. 2019-07-29. Available from: https://www.khronos.org/api/index_2017/OpenXR. [viewed 2022-04-17].

KHRONOS GROUP. GlTF - runtime 3D asset delivery. Online. The Khronos Group. 2021. Available from: https://www.khronos.org/gltf/. [viewed 2022-04-23].

LE, Tuan Anh, Hiroshi TAKAGI, Mohammad HEIDARZADEH, Yoshihumi TAKATA, and Atsuhei TAKAHASHI. Field surveys and numerical simulation of the 2018 typhoon jebi: impact of high waves and storm surge in semi-enclosed osaka bay, japan. Online. Pure and Applied Geophysics, vol. 176 (August 2019), no. 10, pp. 4139–4160. ISSN 1420-9136. Available from: https://doi.org/10.1007/s00024-019-02295-0. [viewed 2022-04-17].

LINIETSKY, Juan, and Ariel MANZUR. Free and open source 2D and 3D game engine. Online. Godot Engine. 2022. Available from: https://godotengine.org/. [viewed 2022-04-23].

MACEACHREN, Alan M., and Menno-Jan KRAAK. Exploratory cartographic visualization: advancing the agenda. Online. Computers & Geosciences, vol. 23 (May 1997), no. 4, pp. 335–343. ISSN 0098-3004. Available from: https://doi.org/10.1016/s0098-3004(97)00018-6. [viewed 2022-04-17].

MEERA, P., M. L. MCLAIN, Kamal BIJLANI, R. JAYAKRISHNAN, and Bhavani R. RAO. Serious game on flood risk management. Online. In: Emerging research in computing, information, communication and applications, pp. 197–206. New Delhi: Springer India, 2015. ISBN 9788132225522. Available from: https://doi.org/10.1007/978-81-322-2553-9_19. [viewed 2022-04-17].

MERRIAM-WEBSTER. Online. Merriam-Webster.com dictionary. [n.d.]. Available from: https://www.merriam-webster.com/dictionary/. [viewed 2022-04-18].

MLIT JAPAN. Plateau. Online. Plateau. April 2020. Available from: https://www.mlit.go.jp/plateau/. [viewed 2022-04-17].

NOAA. Sea level rise and coastal flooding impacts. Online. NOAA Office for Coastal Management. 2022. Available from: https://coast.noaa.gov/slr/. [viewed 2022-04-17].

OPEN GEOSPATIAL CONSORTIUM. 3D tiles specification 1.0. Online. OGC Public Document Repository - Open Geospatial Consortium. 2019-01-31. Available from: https://docs.opengeospatial.org/cs/18-053r2/18-053r2.html. [viewed 2022-04-22].

OPEN GEOSPATIAL CONSORTIUM. OGC city geography markup language (citygml) 3.0 conceptual model users guide. Online. OGC Public Document Repository - Open Geospatial Consortium. 2021-09-13. Available from: https://docs.opengeospatial.org/guides/20-066.html#overview-section-levelsofdetail. [viewed 2022-04-17].

QI, Shuhua, Daniel G. BROWN, Qing TIAN, Luguang JIANG, Tingting ZHAO, et al. Inundation extent and flood frequency mapping using LANDSAT imagery and digital elevation models. Online. GIScience & Remote Sensing, vol. 46 (January 2009), no. 1, pp. 101–127. ISSN 1943-7226. Available from: https://doi.org/10.2747/1548-1603.46.1.101. [viewed 2022-04-17].

SAFIKHANI, Saeed, Michael HOLLY, Alexander KAINZ, and Johanna PIRKER. The influence of in-vr questionnaire design on the user experience. Online. In: VRST '21: proceedings of the 27th ACM symposium on virtual reality software and technology, pp. 1–8. Osaka, Japan, 2021-12-08 – 2021-12-10. New York: Association for Computing Machinery, 2021. ISBN 978-1-4503-9092-7. Available from: https://doi.org/10.1145/3489849.3489884. [viewed 2022-04-17].

SERMET, Yusuf, and Ibrahim DEMIR. Flood action VR. Online. In: SIGGRAPH '19: special interest group on computer graphics and interactive techniques conference, pp. 1–2. Los Angeles, United States of America. New York: ACM, 2019. ISBN 978-1-4503-

6314-3. Available from: https://doi.org/10.1145/3306214.3338550. [viewed 2022-04-17].

STATUTÁRNÍ MĚSTO BRNO. Záplavová území Q100 / Flood zones - 100 year floods. Online. Data.Brno. 2021-04-26. Available from: https://datahub.brno.cz/datasets/mestobrno::záplavová-území-q100-flood-zones-100-year-floods/about. [viewed 2022-04-17].

STATUTÁRNÍ MĚSTO BRNO. Orientační 3D model budov / indicative 3D building model. Online. Data.Brno. 2022-01-17. Available from: https://datahub.brno.cz/maps/mestobrno::orienta%C4%8Dn%C3%AD-3d-model-budov-indicative-3d-building-model/about. [viewed 2022-04-17].

STATUTÁRNÍ MĚSTO BRNO. 3D model budov / 3D Building Model. Online. Data.Brno. 2022-01-19. Available from: https://datahub.brno.cz/maps/mestobrno::3d-model-budov-3d-building-model/about. [viewed 2022-04-17].

THWAITES BEE, Denise, and Deborah MURDOCH-EATON. Questionnaire design: the good, the bad and the pitfalls. Online. Archives of disease in childhood - Education & practice edition, vol. 101 (February 2016), no. 4, pp. 210–212. ISSN 1743-0593. Available from: https://doi.org/10.1136/archdischild-2015-309450. [viewed 2022-04-17].

UNITY TECHNOLOGIES. Unity documentation. Online. Unity Documentation. 2022. Available from: https://docs.unity.com/. [viewed 2022-04-17].

VAMVAKERIDOU-LYROUDIA, L. S., A. S. CHEN, M. KHOURY, M. J. GIBSON, A. KOSTARIDIS, et al. Assessing and visualising hazard impacts to enhance the resilience of Critical Infrastructures to urban flooding. Online. Science of The Total Environment, vol. 707 (March 2020), p. 136078. ISSN 0048-9697. Available from: https://doi.org/10.1016/j.scitotenv.2019.136078. [viewed 2022-04-17].

WHITE, Gary, Anna ZINK, Lara CODECÁ, and Siobhán CLARKE. A digital twin smart city for citizen feedback. Online. Cities, vol. 110 (March 2021), p. 103064. ISSN 0264-2751. Available from: https://doi.org/10.1016/j.cities.2020.103064. [viewed 2022-04-17].

ZHI, Guozheng, Zhenliang LIAO, Wenchong TIAN, and Jiang WU. Urban flood risk assessment and analysis with a 3D visualization method coupling the PP-PSO algorithm and building data. Online. Journal of Environmental Management, vol. 268 (August 2020), p. 110521. ISSN 0301-4797. Available from: https://doi.org/10.1016/j.jenvman.2020.110521. [viewed 2022-04-17].

ATTACHMENTS

LIST OF ATTACHMENTS

Free attachments

Attachment 1	Poster
Attachment 2	SD card

SD card structure

Root folder	**Subdirectory**
input_data	
	brno
	prague
output_data	
	brno
	prague
poster	
questionnaire	
thesis_text	
unity_project	
vr_application	
website	