Proceedings of the 58th Hawaii International Conference on System Sciences | 2025

Creating Virtual Reality Environments at The Edge Of The Metaverse

Mostafa Sadeghi
Aalto University, Finland
mostafa.sadeghi@aalto.fi

Abstract

This study describes the process of creating virtual reality (VR) environments. Following an exploratory approach, six senior VR developers were interviewed, the interview data was analyzed guided by the characteristics of the VR environment extracted from the literature, and a process model was synthesized that describes how different types of VR environments including Single-user VR applications, Multi-user VR applications, and virtual reality environments based on Metaverse platforms can be created. Then, by reflecting on two real cases, the applicability of the process model is demonstrated in practice. This study contributes to the IS literature by systematically describing the VR creation workflow.

Keywords: Virtual Reality, VR, Metaverse, Game engine, Digital Twin

1. Introduction

Virtual Reality (VR) is defined as “A computer-generated simulation of a lifelike environment that can be interacted with in a seemingly real or physical way by a person (Oxford English Dictionary, 2024)”. Different domains of real-world use cases of VR are proposed, such as education and training, virtual collaboration and remote working, socializing and fun activities, gaming, etc. During the last few years, giant tech companies including Meta and Apple have announced ambitious plans and substantial investments to develop a range of products and services (such as Oculus and Apple Vision Pro headsets) to promote VR. This has led to much interest and speculation among corporate leaders and technology enthusiasts about the business opportunities of VR. Consequently, the worldwide VR market size is predicted to broaden from $25.11 billion in 2023 to $244.84 billion by 2032 (Fortune Business Insights, 2024).

Since the 90s decade, a lot of technical frameworks (e.g., Carlsson and Hagsand, 1993; Behr and Froehlich, 1998; Bierbaum et al., 2001; Kelso et al., 2002; Sauter, 2003; Tecchia et al., 2010) have been proposed for creating VR environments. Most of these frameworks require developers to have strong programming knowledge and skills. To ease this issue, commercial VR development tools, such as unity.com and unrealengine.com emerged in the last decade. They provided new capabilities that made VR building easier, but still, developers need to utilize a wide range of complementary tools and skills (Ashtari et al., 2020).

According to Nebeling and Speicher (2018), the landscape of VR creation tools is fragmented, and creators need to utilize multiple different tools (Ashtari et al., 2020). This acts as a barrier for inexperienced creators due to the lack of understanding of crucial background knowledge, and figuring out what background knowledge is important before beginning a VR project (Ashtari et al., 2020). Ashtari et al. (2020) also mentioned that especially domain experts, hobbyists, and end-user developers have struggled with where to start due to the unavailability of appropriate learning opportunities and educational materials. They also mentioned the need for easing VR creation tools for this group of developers who do not have IT technical background.

To handle the issues reported by Ashtari et al. (2020), it is crucial to delineate the VR creation process in detail and discuss the state-of-the-art tools, competencies, and skills for VR building. Thus, this study aims to explore the following RQs:
RQ1. How can a VR environment be created?
RQ2. What tools, technologies, and skills are required for VR environment creation?

This study will provide a coherent overview of VR environment creation. It can guide inexperienced developers on what basic background knowledge is crucial in the VR creation workflow. It also can be used by educational instructors to generate learning opportunities for VR building.

2. Background

2.1. Virtual reality

VR is an advanced human-computer interaction that simulates a realistic 3D environment (Zheng, Chan, and Gibson, 1998). The core ideas are immersion and

URI: https://hdl.handle.net/10125/109696
978-0-9981331-8-8

interactivity (Zheng, Chan & Gibson, 1998). Immersion means to block the connection of users with the real world. Users are physically located in the real world but mentally they are captured by, imbibed in, and fascinated by stimuli from the virtual environment (VE) (Palmer, 1995). Interactivity means that users can act and react to things in the VE (Nevelsteen, 2018). Due to this characteristic, VR can imitate the real world and provide a sense of physical immersive presence (Vogel et al., 2021). "When users have higher levels of presence, they are more likely to behave in VR similar to their behavior in physical reality, blurring the line between these two realities (Dincelli and Yayla., 2022, P.2)."

A VR experience is made from two main cores, VR hardware and VR content (virtual environment). VR hardware includes a variety of devices such as VR headsets that are designed and developed specially for fully immersive VR experiences as well as general-purpose computers and laptops, tablets, game consoles, and mobile phones. The first generation of VR headsets was developed in the late 1980s (Schroeder, 1993). Since that time a lot of technological advancements evolved the devices and created the current commercial VR headset. According to the Gardner report (skarredghost, 2018), VR headset is a mature technology that is ready for mass adoption.

2.2. Virtual environment (VE)

Various researchers have tried to define VE and its characteristics. Girvan (2018, P.13) defined VE as "shared, simulated spaces which are inhabited and shaped by their inhabitants who are represented as avatars. These avatars mediate our experience of this space as we move, interact with objects, and interact with others, with whom we construct a shared understanding of the world at that time." Schroeder (2008. P.2) described VE as "a computer-generated display that allows or compels the user (or users) to have a sense of being present in an environment other than the one they are actually in and to interact with that environment." Nevelsteen (2018, P.11) defined it as "a simulated environment where many agents can virtually interact with each other, act and react to things, phenomena and the environment; agents can be zero or many human(s), each represented by entities called a virtual self (an avatar) software agent; all action/reaction/interaction must happen in a real-time shared spatiotemporal nonpausable virtual environment." Bartle (2010, P.2) mentioned that a VE is "an automated, shared, persistent environment with and through which people can interact in real time by means of a virtual self." According to these definitions, any VE should have a set of features. The table below (Table 1) provides an overview of the features discussed in various studies.

Table 1. VR environments features list.

Feature	Description	Source
F1. Automated (simulated) environment	The VE implements a coherent set of rules that entirely define what changes users can make to that environment.	Girvan (2018), Nevelsteen (2018), Bartle (2010)
F2. Spatial environment	The VE provides a sense of presence in a spatial environment.	Schroeder (2008), Nevelsteen (2018)
F3. Real-time	The VE allows users (avatars) to perform actions simultaneously, and actions immediacy is based on the designed scenario (F1) of the environment.	Girvan (2018), Nevelsteen (2018), Bartle (2010)
F4. Accessible for users	The VE should be accessed by zero or many users through avatars	Schroeder (2008), Girvan (2018), Nevelsteen (2018), Bartle (2010)
F5. P2P communication	Avatars can communicate together	Girvan (2018), Nevelsteen (2018), Bartle (2010)
F6. Interaction	Avatars can act and react to things and phenomena in the environment	Schroeder (2008), Girvan (2018), Nevelsteen (2018)
F7. Persistent, Nonpausable	The VE continues to exist even when there is no user interacting with it.	Nevelsteen (2018), Bartle (2010)

3. Research Methodology

In this study, an exploratory approach is adopted. This approach is suitable for examining a phenomenon that has not been studied in depth before (Swedberg, 2020). In social science, exploratory researchers attend to observe as much as possible to all that is going on and generalize among lines suggested by various data sources such as interviewing people or observing patterns of behavior in the phenomenon that is being studied (Stebbins, 2001). Different types of qualitative and quantitative data collection methods are proposed for exploratory research. In this study, a qualitative approach is adopted due to the nature of the RQs which aim to describe a workflow process.

3.1. Data collection and analysis

This study includes two phases of data collection and analysis: 1) Conducting semi-structured interviews, and 2) Investigating two real-world cases in depth.

In the first phase of data collection, the author conducted semi-structured interviews with six senior VR experts (see Table 2). The main purpose of the interviews was to understand the working process in a VR creation team and the roles that are engaged in the process. The author asked the interviewees about the VR applications they have developed as well as the skills, tools, and technologies that they have utilized to build these applications and the purpose and responsibility of each tool and skill. One of the interviews was in person, others were online through Zoom or Teams (based on the preferences of the interviewees). Each interview took around 65 minutes on average. Interviews were recorded and transcribed providing the data set for analysis. The data was coded by the author using the seven features of VR environments (see Table 1) as a navigator that guided the author on how to categorize data. First, the author read the transcribed data and extracted a set of excerpts from them. Then he assigned a tag (F1 to F7) to each excerpt. Excerpts could have one or more tags based on the evidence that they could provide and the connection between the evidence and seven features. For instance, an excerpt that provides information about the design aspects and logic of the VE is categorized as F1. Another one that provides information about 3D modeling is categorized as F2, and an excerpt that informs about hosting and distribution is categorized as F7 and F4. In the end, there was a set of categories each of which represented one (or more) of the features. Based on the logic that the author could find in the data, he surmised how categories are interrelated in the process, drew the arrows between them, and built the process model.

Table 2. Interviewees background

	Position	Educational background	Years of experience
1	Technical team lead	Software development with emphasis on VR development	11
2	Product Manager	Business administration and business law	9
3	Product Manager	Architect, urban Planner	6
4	Technical team lead	Studied physics in university, self-educated software developer	7
5	Product Manager	Fine Art with emphasis on Film Making	7
6	Product Manager	Arts with emphasis on digital narratives	8

In the second stage of data collection, the process model is applied to two real-world projects to be evaluated and its usability in practice be demonstrated. The first one is a fire safety training application (Haj-Bolouri and Rossi, 2022) that is being used as a training tool for firefighting. The technical team lead and the product manager of this project were interviewed in the first step of data collection. The data presented for this project come from the interviews as well as the commercial website of the project. The second application is designed and developed as an urban digital twin for collaborative urban design planning (Finestcentre, 2024). The product manager of this project was interviewed in the first phase of data collection. However, to obtain more details about the implementation of this project, the author had a demo session with one of the technical developers of the project in person. The data presented for this project comes from the conversation between the author and the technical developer in the demo meeting.

4. VR Creation Process Model

Any of the presented features in Table 1 demands a specific tool and skill to be implemented in practice. Figure 1 shows how features are interrelated in the process and Table 3 introduces tools and skills that are available to implement each feature.

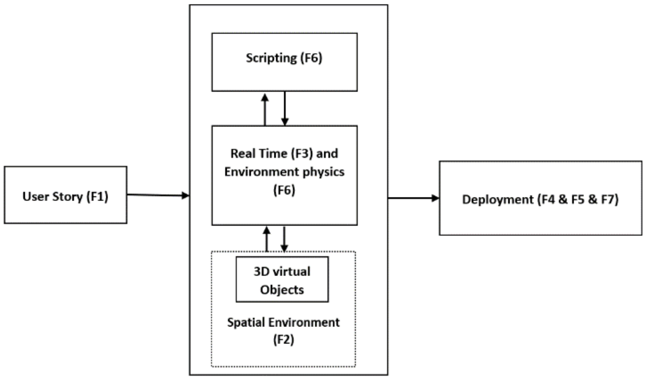

Figure 1. VR creation process model

Table 3. Necessary tools and skills for VR creation

Task	Skill	Tool	Examples of available tools and techniques
User Story	Concept Producer & Interaction Designer	Storytelling	User testing, User interviewing, critical thinking, benchmarking
3D Virtual Objects Design and Development	3D Artist	3D digital Content Creation Software	Blender, Autodesk, 3ds Max, Maya, Adobe, Substance 3D Collection, and Pixologic & ZBrush, etc.
3D Spatial Environment Design and Development	3D Artist	3D digital Content Creation Software	Blender, Autodesk, 3ds Max, Maya, Adobe, Substance 3D Collection, and Pixologic, Zbrush, Real-time engines editor, etc.
Real-Time and Environment Physics	Real-Time Engine Specialist	Real-Time Engine	Unity, Unreal, CryEngine, Babylon.js, Covise, etc.
Scripting	Programmer	Coding Language	C#, C++, java scripting, etc.
Deployment (Hosting and Networking)	Server and Network Engineer	Hardware Knowledge and Programming Language	AWS, Playfab, Photon, Vivox, Fusion, etc.

The process starts by generating the user story of a VR environment. A user story describes the purpose of the environment, its spatial figure, physics and atmosphere, and interactions between users and the environment. User stories are usually abstract features of a VR environment based on a natural language or some models and shapes written or drawn.

In the second step, the features are constructed, and the VR environment is built. In this step, 3D artists utilize 3D content creation software to develop 3D objects and shape the spatial environment (F2), programmers employ a programming language to script interactions (F6), and real-time engine specialists use a real-time engine to implement the physics of the environment and integrate codes and spatial environment to build the VR environment. There might be some iterations between different elements of this step. In practice, real-time engine specialists test codes and 3D objects to see how they fit together and might return them to refine.

The last step is deploying the VR environment. A VR environment can be deployed as a single-user stand-alone application that can be installed on a VR device without any further infrastructure requirement. Such a VR environment is accessible for only one user at a time and does not support F5 and F7. To support F5 and F7 and to be accessible to more users at a time, a VR environment should be hosted on a server (F7) and distributed through the Internet network to be accessible to users (F4). The server should utilize networking algorithms to transfer data such as voice (F5) between users and synchronize any changes inside the VR environment for all users. VR environments can be deployed independently using exclusive infrastructure or can be published inside a Metaverse platform. The way that an environment will be deployed affects the integration between the tools that can be used to construct the environment significantly which should be managed from the start of the project. Requirements of the user story and the scope of the project determine which alternative is more suitable for deploying a specific project.

4.1. User story (F1)

A user exploring a VR environment can do nothing unless the designer constructing the environment anticipates it (Willis, 2004). A user story is the anticipation of the designer. It defines the primary features and units of the functionality of the environment. The initial idea of stories can originate from various sources. It can be a real-world simulation such as a pilot cabin simulation or a gym environment, a fully creative imagination that does not have any mimic in the real world, or a combination of both.

> *"It [story generation] is simple. There are some needs in daily life or something in a game that is not VR, or you only have a vision in mind."*

User story creators need a profound understanding of the context in which the VR environment will be used. This understanding helps them to design an environment that meets users' and other stakeholders' expectations. Critical thinking, user testing, and user interviewing are examples of techniques that can be utilized by designers to obtain the desired understanding. In cases that the virtual environment is a simulation of a real-world environment such as a pilot cabin simulation or gym environment, the user story designer can explore the context in the real world, sympathize with the end-users, and design a user story that meets their desires.

> "*We meet customers and end users in the environment [environment is a firefighter training session] and get pictures of them before [when trainees are training in the real world] and after developing the app [when trainees are training inside the VR environment]. We receive their feedback, brainstorm them in a collaborative environment, and decide how to implement their needs.*"

4.2. Spatial environment (F2)

A spatial environment consists of 3D virtual objects that are incorporated next to each other to make the environment, quite similar to a real-world environment which consists of lots of real objects. 3D virtual objects can be created using 3D digital content creation software. Thanks to technological advancement, the design and development of 3D objects do not require coding skills anymore. Designers should know the basics of 3D modeling and should be able to use 3D digital content creation software. The necessary knowledge is sourced by 3D artists (Pratama and Dossick, 2019).

> *"Several tools, such as Blender, are used for developing 3D assets [virtual objects]. Beyond using the software, you need to know how to model 3D objects. [I mean] the idea that helps you to design a model. It comes from your mental abilities. Some are very good at this, and some are not. But you can learn it and enhance it through experience."*

The user story elucidates what 3D virtual objects are required and how they should be incorporated next to each other to make a particular spatial environment. The same as architectures and interior designers who design and decorate indoor and outdoor real-world spaces and make them functional and beautiful, 3D artists utilize 3D virtual objects and arrange them next to each other to design and develop a 3D spatial environment (Lacity, Mullins, and Kuai, 2023). It can be done using 3D digital content creation software or a real-time engine editor. This means that designers can incorporate 3D virtual objects inside software such as Blender, Maya, etc., to create the spatial environment, and then import the whole environment to the real-time engine, or they can import 3D virtual objects to the real-time engine one by one and shape the spatial environment inside the real-time engine editor.

"We develop the environment in the blender and [then] import it to the engine."

"We use Maya for 3D modeling [3D virtual object development] and then import them to the unity. Unity [real-time engine] editor allows you to put assets [3D virtual objects] together and change or replace them whenever you want."

4.3. Real-time (F3) and environment physics (F6)

Nevelsteen (2018) stated a virtual environment is real-time that allows users (avatars) to perform actions simultaneously and the environment reaction to an action is immediate. An immediacy is accepted only when it is defined as part of the designed scenario (F1) of the environment. Developers utilize a real-time engine to 1) receive the input of the user which can trigger an action (the input is an interaction between the user and a VR device in the real world, for instance, pressing or clicking on a button), 2) provide the simultaneity of actions, and 3) set the immediacy and transposition between different actions and reactions in a virtual environment.

Real-time engines are software that originated from the game development industry and traditionally have been used for developing different types of games, especially 3D games. The real-time engine is the core element in constructing a VR environment that allows creators to integrate the VR device, the user story, the 3D spatial environment, and scripts to generate a VR experience.

"Engine allows you to bring story and assets [spatial environment, 3D virtual objects, codes, and VR device] together and develop the experience."

In addition to the main role that real-time engines play in the process, a real-time engine is used to implement the atmosphere of the environment and the interactions that are caused by it. For instance, in a realistic scenario, the physics, gravity, and lighting of the virtual environment and the actions and reactions caused by them are a critical component to the sense of real-time. In the real world dropping an object, we suppose it to fall immediately, pushing an object, we suppose it to move, blocking the direction of a light, we suppose to see shadow, etc. Commercial real-time engines, such as Unity, Unreal, and CryEngine, allow developers to develop the atmosphere and interactions caused by it without scripting.

"[In engine] You define what objects moving or user can move it. You define all the assets [virtual objects] physics, for example, a wall that the user should not pass through, or where is underwater like sea and river. You define how much gravity you want. You can define the direction of audio and light. All the physic simulation, audio, etc. are handled by the engine".

4.4. Scripting (F6)

In addition to the interactions that are caused by the atmosphere, there is another type of interaction in the real world that is designed or engineered by an agent based on a particular logic. For example, a key can open a locker based on the logic that the key maker defined for it. Or a driver can use a car pedal to increase its speed or the car steering wheel to change its direction based on the logic designed by a car maker. In a VR environment, this type of interactions that originate from a particular logic can be constructed by dedicated codes, attached to specific virtual objects (Willis, 2004).

"For anything that involves logic, we use C# to program it."

Each real-time engine supports only one programming language, for instance, unity supports C#. Therefore, the selected real-time engine for a specific project, defines which coding language can be used.

To ease the issues that programming can cause for non-technical creators such as domain experts, hobbyists, and end-user creators and make scripting simpler for them, new approaches such as visual scripting are proposed by commercial real-time engines which still demand developers to have some background skills in programming.

4.5. Deployment (F4, F5, and F7)

A VR environment can be deployed as a single-user stand-alone application that can be installed on a VR device and be used by a single user at a time. Such a VR environment does not support F5 and F7. To implement F5 and F7, some infrastructures are required. These infrastructures include hosting storage to host the virtual environment (F7), internet connection to make the environment accessible to end users (F4), and networking algorithms to transfer data such as voice (F5) between the devices of users and synchronize any change in the environment for all the users. To provide the infrastructures, VR environment creators have two architectural choices: 1) Independent multiuser VR application, and 2) Metaverse platforms.

Independent architecture is similar to most of the regular internet websites and mobile applications. To develop such a VR environment, after constructing the environment, developers host it on a server that they can control and modify. This server should be connected to the Internet to allow users to have access to the environment. It also provides computing power and networking algorithms that are necessary for transferring data such as voice between users' devices as well as synchronizing the changes inside the environment for all users.

> *"I have somebody on my team who is in charge of our servers. Once we have created the interactive environment, we host that on remote servers. Those servers use networking technologies to manage the network traffic."*

The alternative option is utilizing a Metaverse platform. A metaverse platform is a third party that provides the common infrastructures that are required for all VR environments. The features that were discussed in previous sections (F1, F2, F6) are unique properties of each environment. This means that if we compare two VR environments, they have different User Stories (F1), Spatial Environments (F2), and Interactions (F6). However, the remaining features (F4, F5, F7) are common among all VR environments. It allows third parties to provide the infrastructure and make the connection between VR creators and end users. VR creators can utilize the infrastructure to publish their VR environment inside the platform and make it accessible to end-users. Creators have no control over the platform, its processes, hosting servers, networking algorithms, etc. They are solely able to generate and publish content according to functionalities that the platform owner has provided for them.

There are multiple Metaverse platforms available in the market at the moment. These platforms can be categorized into two main groups. The first group provides a special software package (SDK) of the real-time engine that they support and asks creators to develop the VR environment using the SDK and then import the final product to the platform. Platforms such as immerse.io, glue.work, vrchat.com, spatial.io, etc., are among this group. These platforms usually support most functionalities of their real-time engine and could be used for developing highly customized and interactive environments. The second group of platforms integrates the real-time engine into some complementary services and provides an abstract instance of the real-time engine in the platform. Platforms such as sandbox.game and decentraland.org are among the second group. Using the second group, creators can directly import 3D virtual objects (or 3D spatial environment) to the platform and continue the development process inside the platform. It makes the development process simpler but on the other hand does not support all functionalities of the real-time engine and developing highly complex interactive environments is not easily doable.

The purpose of a VR environment and the scope of a project define which architectural type and which type of platform is more suitable for a specific case. Independent architecture is more appropriate for professional teams with higher financial power which the VR environment is the core element of their business. On the other hand, platforms are more appropriate for freelancers, domain experts, hobbyists, and end-user developers for whom the VR environment is not the core element of their business. Also, the user story elucidates which type of platform is more appropriate for a specific project. Platforms capabilities can be used to develop a wide spectrum of environments, from highly customized and unique for a specific purpose with a lot of interactions inside the environment to simple environments where avatars solely can communicate together and see some visual content such as photos or slides. For highly customized and complex user stories, the first group of platforms that provide the SDK of the real-time engine is more appropriate. For simple scenarios, the second type of platform is more suitable and can be used by non-technical domain experts, hobbyists, and end-user developers with a short instruction.

> "*You need to know the user story, to see which platform or path can handle the story. I would say if you have a lot of communication with other users, platform [second group of platforms] is a better option. If have a lot of customized interactions with the environment, it's doable in platforms [the second group] but cumbersome, you have to create a lot of new things. Independent path or platforms that allow to develop in the engine [first group that provides the real-time engine SDK] and then upload to the platform are better options.*"

5. Cases

5.1. VR fire training

Using VR as a tool for facilitating learning and training in industrial contexts has received increasing attention recently (Jensen and Konradsen, 2018; Feng et al., 2018), especially in contexts where training in the real environment is dangerous or expensive (Freina and Ott, 2015). One of these contexts is fire safety. Fire safety training in the real world can be dangerous for learners. It also needs an equipped environment that makes it costly. A simulated fire scene in which learners can immersed through a VR headset and fight the simulated fire can solve these issues. Table 4 shows how the VR creation process model is justifiable with an already-developed VR fire training application.

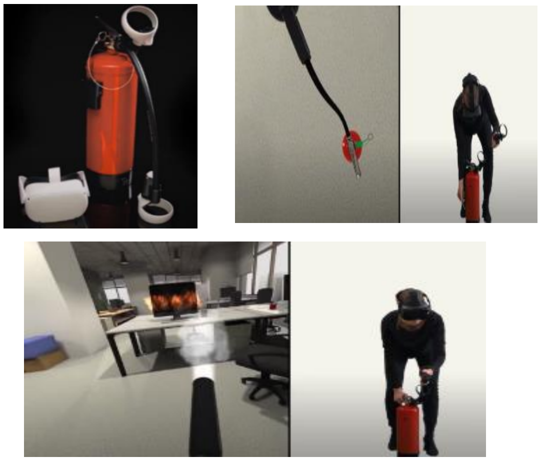

Figure 2. VR fire training extinguisher (top left), VR fire trainer activating extinguisher (top right), fighting fire in a virtual office environment (down) (Vobling, 2024)

Table 4. VR Fire training creation process

Features	Description
User story (F1)	This application is a firefighting tool. It simulates the behavior and reactions of each fire type, extinguisher agent, and smoke in a fire scene simulation. Users can choose which type of fire extinguisher agent such as water, foam, CO2, etc. they need for their training session and will see how any of them reacts to the fire. Along with the VR application, customers need to have a special extinguisher (Figure 2). This extinguisher is connected to the VR headset through Bluetooth. The trainee´s interactions with this extinguisher will be applied to the digital twin of the extinguisher in the VE that simulates a realistic firefighting sense.
3D virtual objects (F2)	The most important object in this scenario is the extinguisher and its elements such as the safety pin. This object is developed using MAYA. Other 3D virtual objects of environments are developed through a joint work of in-house development using MAYA for unique elements and utilizing virtual objects that are available in the market for general elements.
Spatial Environment (F2)	The application provides various spatial environments such as working office, hospital, home kitchen, etc. The working office has been shown in Figure 2. Trainees can switch between environments through the menu of the application. 3D virtual objects are incorporated next to each other in the Unity editor to build spatial environments.
Real Time (F3) and interactions (F6)	This application is developed using the Unity engine. Anything related to the atmosphere of the VE such as gravity, lighting, fire and smoke simulations, and interaction between extinguisher agent with the fire and its consequences are developed using the functionalities of the Unity editor. In addition, the transposition of actions and reactions is set by Unity functionalities.
Scripting (F6)	Interactions of the trainees with the extinguisher should be tracked and their effects should be shown instantly after the interaction in the VE. For instance, trainees need to release the safety pin (Figure 2) to be able to use the extinguisher. Then, they can use the extinguisher handler to fight the fire. Finally, they can replace the pin in the initial position to deactivate the extinguisher. These interactions are implemented using C#.
Deployment	Due to the business requirements, this application is deployed as a single-user standalone application without any external hosting or networking service. It only utilizes the computing power of the headset to operate.

5.2. Urban digital twin

Increasing citizens' engagement in the urban spaces design process has been a substantial topic in urban co-design in the last decades. Citizens are the main stakeholders of urban spaces, and their needs and desires should be reflected in the design process (Carmona, 2021). However, this domain is dominated by expert designers and planners (Munthe-Kaas, 2015). To ease the collaboration and communications between designers and public communities, researchers (Jamei et al., 2017; Wolf, Söbke, and Wehking, 2020) suggested using immersive artifacts such as VR and Augmented Reality (AR) applications for conducting co-design workshops. These technologies facilitate cooperation, providing more inclusive and efficient design results (Ehab and Heath, 2023).

Table 5 shows how the VR creation process model is justifiable with an already-developed urban co-design application. This application allows participants to visualize and foresight the seasonal and temporal transitions in urban vegetation, as well as to be involved in urban green area designing (Finestcentre, 2024). Consequently, it contributes to the development of participatory urban planning by enabling unexpert citizens to be engaged in the urban planning process, who have been traditionally excluded from this process. (Fabiandembski, 2024).

Figure 3. A scene of the urban digital twin application in two different seasons (Fabiandembski, 2024)

Table 5. Urban digital twin creation process

Features	Description
User story (F1)	The main idea of this project is to provide a collaborative environment for city planners and citizens to co-design the vegetation layer of the city environment together. Participants can use the environment to view the temporal and seasonal changes in the urban vegetation. They can change the season and see how the green layer of the environment is modified by seasonal change (Figure 3).
3D virtual objects (F2)	1) 3D models of plants in different stages of their growth as well as their shapes in different seasons. 2) Artificial elements in the city such as buildings, roads, etc. All elements are developed using 3D max.
Spatial Environment (F2)	The spatial environment mimics the outdoor environment of the city (Figure 3). The environment is built using 3D max
Real-Time (F3) and interactions (F6)	This project is based on COVISE (Hlrs, 2024).
Scripting (F6)	As it is mentioned, users can change the season, and see how the green layer of the environment changes by seasonal change. When they select a different season from the menu, a new web service is called that provides the related data of the green layer and the atmosphere (rain, snow, fog, etc.) of the environment in the chosen season. A bunch of codes integrates these web services to the VE and implement orders. C++ (the programming language that COVISE supports) is used for developing the codes.
Deployment	This application is deployed on private servers specially dedicated to this project.

6. Discussion, Limitations, and Future Research

This study provides a coherent landscape of VR creation workflow and state-of-the-art tools and competencies that VR creation requires. The VR creation process model is the main contribution of this study.

Based on the process model, the VR creation workflow starts by generating the user story of the VR environment. The user story defines the spatial figure of the VR environment, virtual objects that are located inside the environment, and interactions that users can do with the objects. To implement the user story, 3D artists design and develop virtual objects and the spatial environment using 3D content creation software, programmers use a coding language to script interactions, and real-time engine specialists integrate the spatial environment, virtual objects, and codes to generate the VR environment.

To be accessible to end users, the VR environment should be deployed. It can be deployed as a single-user standalone application that can be installed on a VR device without any extra infrastructure. To be accessible for more than one user at a time, the VR environment can be hosted on dedicated servers and distributed through the Internet connection or can be published in a Metaverse platform.

The VR creation process model can guide VR developers especially domain experts, hobbyists, and end-user developers on what basic knowledge is required for developing a VR application and how they should proceed step by step in the workflow of creating a VR environment. It can also be used by educational instructors to generate educational programs for VR development.

This study has limitations and leaves room for future studies. The main limitation is that, in this research, the researcher approached the phenomenon from the perspective of highly professional experts in the field who are leading professional VR creation teams which are built of professionals with different skills and backgrounds. Approaching the VR creation workflow from the perspective of solo creators such as domain experts, hobbyists, and end-user developers who must do all the tasks alone can provide new insights into this topic.

In addition, exploring the possibility of developing short-term VR creation educational programs for training solo creators such as hobbyists and domain experts is another potential path for future research. As it is obvious, various educational programs at different levels about digital artifact design, digital 3D art, and programming are available. However, short-term programs that integrate all the necessary knowledge and skills for VR creation with an emphasis on training solo creators and domain experts are not common yet. Following Ashtari et al. (2020), the author asks researchers and practitioners to investigate the possibility of providing such educational programs.

7. Conclusion

This study aims to describe how VR environments can be created, and what tools and skills are required for this purpose. To this aim, based on the seven characteristics of VEs extracted from the literature and through an exploratory qualitative study, a process model was developed which describes the workflow of creating VR environments. Then, the usability of the process model was demonstrated by investigating the creation workflow of two real cases. In the end, the limitations of the research and potential paths for future studies were discussed.

8. References

Ashtari, N., Bunt, A., McGrenere, J., Nebeling, M., & Chilana, P. K. (2020, April). Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1-13).

Bartle, R. A. (2010). From MUDs to MMORPGs: The history of virtual worlds. *International handbook of internet research*, 23-39.

Behr, J., & Froehlich, A. (1998). AVALON, an open VRML VR/AR system for dynamic application. *Topics*, *1*(1), 28.

Becker, K., Becker, & James. (2017). *Choosing and using digital games in the classroom* (pp. 175-214). Berlin/Heidelberg, Germany: Springer.

Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., & Cruz-Neira, C. (2001, March). VR Juggler: A virtual platform for virtual reality application development. In *Proceedings IEEE virtual reality 2001* (pp. 89-96). IEEE.

Carlsson, C., & Hagsand, O. (1993). DIVE—A platform for multi-user virtual environments. *Computers & graphics*, *17*(6), 663-669.

Carmona, M. (2021). *Public places urban spaces: The dimensions of urban design*. Routledge.

Chan, S. C., Wan, C. J., & Ko, S. (2019). Interactivity, active collaborative learning, and learning performance: The moderating role of perceived fun by using personal response systems. *The International Journal of Management Education*, *17*(1), 94-102.

Dincelli, E., & Yayla, A. (2022). Immersive virtual reality in the age of the Metaverse: A hybrid-narrative review based on the technology affordance perspective. *The Journal of Strategic Information Systems*, *31*(2), 101717.

Ehab, A., & Heath, T. (2023). Exploring Immersive Co-Design: Comparing Human Interaction in Real and Virtual Elevated Urban Spaces in London. *Sustainability*, *15*(12), 9184.

Fabiandembski. (2024). https://fabiandembski.com/2021/05/28/greentwins-tallinn-and-helsinki-digital-twins-for-more-democratic-resilient-and-greener-cities/ (visited on: June 2024)

Feng, Z., González, V. A., Amor, R., Lovreglio, R., & Cabrera-Guerrero, G. (2018). Immersive virtual reality serious games for evacuation training and research: A systematic literature review. *Computers & Education*, *127*, 252-266.

Finestcentre. (2024). https://finestcentre.eu/project-pilot/greentwins/ (visited on: June 2024)

Freina, L., & Ott, M. (2015, April). A literature review on immersive virtual reality in education: state of the art and perspectives. In *The international scientific conference elearning and software for education* (Vol. 1, No. 133, pp. 10-1007).

Fortune Business Insights, (2024). https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378 (visited on: June 2024)

Girvan, C. (2018). What is a virtual world? Definition and classification. *Educational Technology Research and Development*, *66*(5), 1087-1100.

Haj-Bolouri, A., & Rossi, M. (2022). Proposing design principles for sustainable fire safety training in immersive virtual reality.

Hlrs (2024). https://fs.hlrs.de/projects/covise/doc/html/index.html (visited on: Sept 2024)

Jamei, E., Mortimer, M., Seyedmahmoudian, M., Horan, B., & Stojcevski, A. (2017). Investigating the role of virtual reality in planning for sustainable smart cities. *Sustainability*, *9*(11), 2006.

Jensen, L., and Konradsen, F. (2018). A review of the use of virtual reality head-mounted displays in education and training. *Education and Information Technologies*, *23*, 1515-1529.

Karagiannis, S., Papaioannou, T., Magkos, E., & Tsohou, A. (2020, November). Game-based information security/privacy education and awareness: theory and practice. In *European, Mediterranean, and Middle Eastern Conference on Information Systems* (pp. 509-525). Cham: Springer International Publishing.

Kelso, J., Arsenault, L. E., Satterfield, S. G., & Kriz, R. D. (2002, March). Diverse: A framework for building extensible and reconfigurable device independent virtual environments. In *Proceedings IEEE Virtual Reality 2002* (pp. 183-190). IEEE.

Lacity, M., Mullins, J. K., & Kuai, L. (2023). Evolution of the Metaverse. *MIS Quarterly Executive*, *22*(2), 6.

Munthe-Kaas, P. (2015). Agonism and co-design of urban spaces. *Urban Research & Practice*, *8*(2), 218-237.

Nebeling, M., & Speicher, M. (2018, October). The trouble with augmented reality/virtual reality authoring tools. In *2018 IEEE international symposium on mixed and augmented reality adjunct (ISMAR-Adjunct)* (pp. 333-337). IEEE.

Nevelsteen, K. J. (2018). Virtual world, defined from a technological perspective and applied to video games, mixed reality, and the Metaverse. *Computer animation and virtual worlds*, *29*(1), e1752.

Oxford English Dictionary, 2024. https://www.oed.com/dictionary/virtual-reality_n?tab=meaning_and_use#15654690 (visited on: June 2024)

Palmer, M. T. (1995). Interpersonal communication and virtual reality: Mediating interpersonal relationships. *Communication in the age of virtual reality*, 277-299.

Pratama, L. A., & Dossick, C. S. (2019). Workflow in virtual reality tool development for AEC industry. In *Advances in Informatics and Computing in Civil and Construction Engineering: Proceedings of the 35th CIB W78 2018 Conference: IT in Design, Construction, and Management* (pp. 297-306). Springer International Publishing.

Sauter, P. M. (2003). VR2Go™: a new method for virtual reality development. *ACM SIGGRAPH Computer Graphics*, *37*(1), 19-24.

Schroeder, R. (1993). Virtual reality in the real world: history, applications and projections. *Futures*, *25*(9), 963-973.

Schroeder, R. (2008). Defining virtual worlds and virtual environments. *Journal For Virtual Worlds Research*, *1*(1).

Skarredghost. (2018). https://skarredghost.com/2018/08/27/virtual-reality-is-reaching-a-mature-state-according-to-gartner/ (visited on: June 2024)

Stebbins, R. A. (2001). *Exploratory research in the social sciences* (Vol. 48). Sage.

Swedberg, R. (2020). Exploratory research. *The production of knowledge: Enhancing progress in social science*, 17-41.

Tecchia, F., Carrozzino, M., Bacinelli, S., Rossi, F., Vercelli, D., Marino, G., ... & Bergamasco, M. (2010). A Flexible Framework for Wide-Spectrum VR Development. *Presence: Teleoperators and Virtual Environments*, *19*(4), 302-312.

Vobling. (2024). https://vobling.com/ (visited on: Jan 2024)

Vogel, J., Schuir, J., Koßmann, C., Thomas, O., Teuteberg, F., & Hamborg, K. C. (2021). Let's do design thinking virtually: design and evaluation of a virtual reality application for collaborative prototyping.

Willis, P. (2004, June). Virtual physics for virtual reality. In *Proceedings Theory and Practice of Computer Graphics, 2004.* (pp. 42-49). IEEE.

Wolf, M., Söbke, H., & Wehking, F. (2020). Mixed reality media-enabled public participation in urban planning: A literature review. *Augmented reality and virtual reality: Changing realities in a dynamic world*, 125-138.

Zheng, J. M., Chan, K. W., & Gibson, I. (1998). Virtual reality. *Ieee Potentials*, *17*(2), 20-23.