UNIVERSITY
OF OULU
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Ciprian Florea

VIRTUAL REALITY INTERFACE FOR THE PATIO USER INVOLVEMENT TOOL

Master's Thesis
Degree Programme in Computer Science and Engineering
June 2019

Florea C. (2019) Virtual Reality Interface for the PATIO User Involvement Tool. University of Oulu, Degree Programme in Computer Science and Engineering. Master's Thesis, 51 p.

ABSTRACT

The objective of this thesis was to explore, implement, and evaluate means to improve user participation using virtual reality. Thus, two applications were implemented: a VR client for user feedback collection, and a web tool that works as a moderator creation editor. These tools offer an approach to collecting feedback about activities taking place in VR 3D spaces and combine different functionalities such as geolocated points of interest and 3D user interfaces. Through these applications, users are able to explore spaces and answer geolocated surveys in an interactive and immersive way.

These two clients were developed side by side with other clients of the same user involvement tool and thus involved several iterations to achieve a good user experience. In addition, related work was studied to research about building VR experiences and interactive 3D user interfaces.

The VR client was tested with real users (n = 14) where their experiences were gathered using a five-point Likert scale questionnaire and an Adjective cards selection method, while also being watched and interviewed.

The study conducted for evaluation shows that the VR solution is important to users because they can be involved in research and product development even when it is not possible to be in a certain location or when an environment does not exist, or it no longer exists. However, better interaction methods in the virtual environment, as well as additional graphics performance optimization are needed for a better experience in the user involvement process.

Keywords: virtual reality, user involvement, 3D game engines, online survey tools, 3D user interfaces, geolocated points of interest

TABLE OF CONTENTS

FOREWORD

Many thanks to Prof. Timo Ojala, my papa here at the Centre for Ubiquitous Computing, for teaching, helping and guiding me through my work and study related endeavors, and many thanks to Dr. Matti Pouke and Dr. Paula Alavesa for acting as my supervisors for this masterpiece.

Oulu, 18.6.2019

Ciprian

ABBREVIATIONS

VR	virtual reality
3D	three-dimensional
WebGL	Web Graphics Library
app	application
AR	augmented reality
VE	virtual environment
2D	two-dimensional
HTML	Hypertext Markup Language
JS	JavaScript
HTTP	Hypertext Transport (or Transfer) Protocol
URL	uniform (or universal) resource locator
JSON	JavaScript Object Notation
MB	megabyte(s)
KB	kilobyte(s)
px	pixel
UI	user interface
MR	mixed reality
POI	point of interest
REST	Representational State Transfer
API	application programming interface
SM	SurveyMonkey
UX	user experience

1. INTRODUCTION

This master's thesis discusses and presents an immersive virtual reality (VR) and a web application for user involvement and co-creation, developed using 3D model representations of real places / sites, with a focus on utilizing powerful 3D game development platforms, and built for the Oculus Rift and WebGL platforms.

The motivation for this work came from a need of new technological solutions to the process of user involvement in developing better services and products by users, customers, and other stakeholders working together. Involving end-users in the development of such artefacts results in creating products that bring more value not only to the users, but also to the developers. One key concept in this work is how can VR technologies help develop new applications that users can be immersed in, seeing the strengths VR has over other technologies. Moreover, user involvement and co-creation is a concept explored and improved with acceleration, and researchers see the potential in using it for different purposes: smarter cities and urban planning, personalized services, and valuable products [1, 2].

A VR solution coupled with a moderator web tool, that are part of a bigger user involvement tool can engage users differently in the stages of the co-creation process. Thus, by involving all parties, products that bring more value are created while minimizing development risks, seeing that problems in the design and planning process can be found before major decisions are made. Considering that VR technologies are not utilized at an optimum level when evaluating real world sites and spaces, this thesis explores, implements, and evaluates means to take advantage of the immersive and interactive VR technology by integrating maps, accurate 3D model representations of existing spaces, an online survey software, 3D user interfaces, and geolocated 3D points of interest.

A user experience study was conducted to evaluate the users' subjective experiences of the VR client. The aim of this study was to find out how interesting, useful and fun is to participants to use the VR client in a user involvement context. The participants used the application to answer survey questions and thus give feedback while experiencing and exploring a familiar space represented in 3D. During this study, user feedback was collected at the same time the users experienced the 3D virtual environment and its content.

1.1. Overview of the Thesis

The content of this thesis is divided as follows. To begin with, the concept of immersive virtual reality and its applications, and existing knowledge related to user involvement and co-creation are investigated. Secondly, related works relevant to the subject matter of this work are considered. This thesis is quite substantial with regards to the design and implementation of its applications. Thus, we describe two applications for user involvement and co-creation. Firstly, a VR client for user feedback collection, and secondly, a web tool that works as a moderator creation editor for all the user clients included in the parent user involvement tool. Finally, the evaluation process and its results are talked through, and the outcomes are discussed.

1.2. Contributions

The PATIO ecosystem incorporates a few client applications that were developed for the same purpose of user involvement: the original website, a mobile client, an augmented reality (AR) client, a VR client, and a moderator tool. Although the first three are mentioned at some point in this thesis, my contributions are the VR client and the moderator tool, which are detailed in this work.

2. BACKGROUND AND RELATED WORK

2.1. Definitions and Related Concepts

2.1.1. Virtual Reality, Immersion, Presence

Virtual reality (VR) is a sensory-immersive type of computer-generated technology [3]. It is a sensory-immersive technology because it aims to convincingly engage human senses to create the perception of existing in another world and interacting with it. As Biocca and Delaney [3] nicely state, "virtual worlds are constructed by the senses and only really exist in the minds of users." This is one focus area when designing and developing VR applications: how can VR make users feel more immersed and fool their senses into overlooking that the virtual environment around them is not actually real, but computer-generated. This also brings back one of the challenges of this thesis' VR application that is, creating a feeling of immersion with regards to the implemented 3D environments and interactive 3D user interfaces, for example.

The simulated world or scenario can be a digital representation of a real-world environment, completely imaginary, or a blend of both. Therefore, VR has a wide range of possible applications. Fictional worlds are often used in entertainment applications such as VR gaming and 3D cinema, with movies like "The Matrix" and "Avatar" having done much to popularize the idea of experiencing fantasy/science fiction worlds.

To better understand what virtual reality is and how it differs from other related technologies, a short simplification of existing reality technologies is:

- Virtual reality takes place in an enclosed digital environment, user interacts with the virtual world.
- Augmented reality refers to virtual objects and information used as overlay to enhance the real environment.
- Mixed reality is virtual and real combined, where virtual objects interact with or are influenced by the real world. [4, 5, 6]

Virtual reality headsets are most often used to experience VR applications. These devices equip the user with a display, sound, and head motion tracking sensors. Additionally, controllers are used to interact with the digital world. [7, 8]

LaValle gives his definition of VR in his 2016 book: "Inducing targeted behavior in an organism by using artificial sensory stimulation, while the organism has little or no awareness of the interference." As he explains, this definition tries to capture the core of VR, without focusing on particular devices or technologies. In addition, since we mentioned immersion in the previous paragraph, it is important to understand what to consider when talking about VR systems and how to achieve the feeling of immersion. Thus, LaValle emphasizes that often the focus is only on the engineering parts, hardware and software. But, although the engineering aspect is an important part of the puzzle, it is essential to consider and take advantage of human psychology and perception [9].

Immersion and presence are two concepts that are associated with virtual reality technologies and are pivotal in relation to the development and success of VR. Immersion is defined in [10]: “Immersion is a description of a technology, and describes the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding and vivid illusion of reality to the senses of a human participant.” To achieve immersion, several essential conditions must be met. Firstly, interferences from the physical reality should be minimized. Secondly, the range of sensory modalities should be extensive. In addition, virtual reality should be panoramic instead of being restricted to a small narrow field of view. Displays should be vivid in terms of richness, information content, resolution and quality. [10, 11, 12]

Immersion can happen by different means. Since the applications detailed in this thesis were developed using 3D game engines and incorporate game mechanics, it is worth noting how Ernest W. Adams, founder of the International Game Developers Association (IGDA), describes three forms of immersion:

- Tactical immersion is produced when the user solves very small tasks, in the form of immediate interaction. The user does not worry about the overall strategy, other than the task at hand. To create and maintain this type of immersion, the controls, user interface and overall engagement must be intuitive, quick and reliable.
- Strategic immersion is intellectual rather than emotional or physical. It can happen when a participant undertakes exciting mental challenges. Compared with tactical immersion, strategic immersion is not about immediate interaction. It is about solving challenges through concentration, observation, calculation, and deduction. Chess players experience strategic immersion during intense mental concentration and visualization.
- Narrative immersion is produced by captivating and convincing storytelling. The user feels a connection with the story, its characters, and wants to know more. Narrative immersion is emotional rather than intellectual or physical, and thus similar to reading a book or watching a movie. [13]

Another form of immersion, more relevant to this thesis, is explained by Staffan and Holopainen in their 2004 article:

- Spatial immersion is produced when an artificial world is credible and convincing. The world is perceived as real and the user feels physically present. [14]

Regarding the concept of presence, it is described as “a state of consciousness, the (psychological) sense of being in the virtual environment” [10]. Then, this is related to but distinct from immersion which is considered to be objective and quantifiable. When someone is highly present in a virtual environment, they experience the environment that they see and interact with in virtual reality as a real place, and feel in that moment more engaged by the virtual world rather than the physical place they are currently in. [10, 11, 12]

2.1.2. Virtual Environments, User Participation, 3D Game Engines

Virtual Environments (VEs) are a relevant approach when the aim of a research or other computer simulations is to visualize change, collect user feedback, and bring improvements through better user participation [15, 16, 17, 18, 19, 20, 21, 22, 23]. VR brings a novel approach to represent and visualize 3D environments and spaces. Seeing the technological advancements of this technology, it is understandable that it has developed to a great extent and is nowadays considered to be a mature technology [24]. Thus, it has made its way into consumer products [25].

Research suggests that in the cases of user involvement and participatory urban planning, VR headsets and 3D rendering techniques bring several benefits to these processes. The reason for this is that VR better engages users and increases participation when compared to other 2D presentation technologies. With regards to visualization, VR helps participants make knowledgeable decisions and give feedback based on a better understanding of the situation, seeing that VR produces clearer images in the mind compared to other 2D technologies. [1]

User involvement and co-creation can be simply described as developing solutions with users, for users. It means them being part of the creation process not only as advisers but also as creatives in a way that the resulted product is closer to what end users need [26, 27]. More formally, co-creation is defined as "an active, creative and social collaboration process between producers (retailers) and customers (users), facilitated by the company. Customers become active participants in an open innovation process of a firm and take part in the development of new products or services." [28] The interactions between producers and customers bring value not only to the customers in the form of products that they want and need, but also to the producers. Thus, they create innovative products and competitive advantage. [29]

With regards to user participation and service design, companies, developers, and other stakeholders are becoming more aware of the role of users. Instead of looking at users as people who will ultimately use a product, they are now considered an active part of the design and creation processes. Users bring a lot to the table when it comes to innovation ideas, seeing that they are the best source of feedback when it comes to everyday use. Thus, involving users in the creation process leads to better quality products while minimizing development risks. Research suggests that it is worthwhile to consider users as a source of innovation and to increase their involvement in the design of products and services. Nowadays, more developers see the benefit in this and include users more and more, to design with them and for them. In co-creation, all stakeholders are included in the process and work towards innovation, including end-users. Traditionally, not many users have been involved in the process. With the acceleration of new technological methods and solutions, however, it has become easier to reach end-users and provide them with a platform for co-creation. [2]

Powerful 3D game development platforms are not only used for video game development anymore. With flexible and advanced real-time tools, applications of 3D game engines continue to grow from the Gaming industry, and branching out into Film, Automotive, Architecture, Engineering & Construction, Gambling, and more [30]. Thus, using such development platforms brings clear benefits when it comes to creating a variety of applications that make use of 3D model representations and visualizations, such as the applications detailed in this thesis. Modern game engines include a multitude of different purpose development tools integrated into one to empower creation and rapid development of interactive applications with complex models and environments. One of the main benefits of using such a 3D game engine is that they come with advanced render engines that enable developers to create visualization applications that include complex, detailed 3D environments [31]. Two very popular choices are Unity developed by Unity Technologies [32], and Unreal Engine developed by Epic Games [33].

2.1.3. WebVR in User Involvement

People use many types of Web tools and services every day, ranging from wikis and blogs to video sharing [34] and VEs [35, 36]. Web applications are a suitable approach also for the process of innovation creation, providing mediums for user participation, content creation, sharing, collaboration, and more [37].

Virtual reality is utilized in numerous use cases. One such use case is visualizing change. Research suggests that visualizing change in VR is a useful method, given that VR provides a high level of immersion. In VR, users embody the virtual world which results in a higher level of presence. [37] Moreover, undertaking activities in a virtual environment in VR is considered to be more natural, given that the users can observe and coordinate in the same manner they would in the real physical world [37]. Thus, exposure to graspable simulations and reconfigurable prototypes are key elements when it comes to user participation and living labs [38, 39].

WebVR, or VR on the Web, is a useful approach for the user involvement process, seeing that it can inherit the advantages of other Web applications, especially those used for the same purposes. Although WebVR is used for visualizations in other industries, the entertainment industry for instance, when it comes to user involvement and co-creation, its applications are rare. [37]

2.2. Related Works

Two related works were selected to be discussed in this chapter. Firstly, van Leeuwen et al. present in their 2018 paper “a case of using VR in a municipal process of civic participation concerning the redesign of a public park” [1]. Secondly, Kaasinen et al. describe in their 2012 paper three co-creation spaces that “manage to bring co-creation close to the users’ everyday life” [2].

2.2.1. Virtual Reality in Participatory Urban Planning: Public Ballot and Laboratory Study

In this work, the authors present user involvement and co-design activities that aimed to involve participants in the decision-making process by using VR headsets. The first activity presented was through a ballot and it made use of 3D-rendered versions of different urban designs. The second experiment took place in a controlled environment. The results indicate that VR applications offer higher immersion and engagement.

For the first part, a public ballot activity let citizens vote for their preferred design. The authors investigated whether VR technologies can improve the user involvement process through a more realistic and immersive approach. Photorealistic 3D rendering for design variants visualization, and a range of devices are used. These designs were also presented with a VR headset. As for results, the authors assess the level of engagement with the process and its cognitive benefits. For the public ballot, using VR applications does not seem to be significantly different than using smartphones with regards to the level of engagement. With regards to cognitive benefits, the type of device used for viewing the designs did not seem to influence the votes either.

The authors also describe a laboratory study that was conducted in public locations to assess the level of immersion of VR in a more controlled and accurate way. The goal was to compare a VR approach with a non-immersive approach. In this case, the results prove that the participants were more immersed by the VR experience compared to using a conventional laptop computer. The results prove that a VR headset performs better than a laptop computer in terms of engagement.

This work and the applications implemented in this thesis are similar seeing that they are all directed at user involvement and co-creation. In addition, the VR client is also designed to offer an immersive medium for user feedback collection. The differences however are that firstly, the VR client was developed as a WebVR application and is therefore intended to be used not only in organized, controlled experiments, but also by users on their own. In addition, the activities created with the web tool, and experienced using the VR client are more elaborate, with many survey questions and geolocated points of interest and 3D content.

2.2.2. Co-Creation Approaches: Owela, Ihme, Living Labs

The applications described in this work are web based Owela, physical showroom Ihme, and Living Labs which is a combination of both. The authors discuss and compare these three applications to evaluate their strengths and weaknesses in terms of user involvement.

Owela, or Open Web Lab, is an online platform that aims to help the process of co-creation by involving all interested parties: users, customers, developers, and other stakeholders. It is designed to be used as a social media platform and thus it is not time or place restricted. The authors say that over 40 kinds of cases have been conducted using Owela. Topics include multicultural social media service, city adventure service, and other. Owela offers mostly text-based communication.

Ihme is an innovation showroom concept for open public co-creation. The idea is to make available an environment where interested people can experience and test physical impressions of new products and services. Participants are involved in different sessions and give meaningful feedback. The authors argue that this type of playful interaction makes agile, iterative development possible. In Ihme, many different types of technologies are put forward to participants including virtual services, games, augmented reality applications, and more. Direct dialogue is one of the benefits of a showroom concept.

Living Labs are open innovation ecosystems for creation of new services and products. It is an approach that engages users in the innovation and development of such artifacts. The authors argue that user behavior can be observed before real money is spent in such developments. In Living Labs, user feedback is collected to improve ideas.

The authors compare these co-creation approaches from different points of view and conclude that the main differences relate to where the participation takes place and what is the role of participation. The authors argue that although these approaches differ from these points of view, they all manage to reach a significant amount of people and bring them closer to the innovation and development processes for new services, technologies, and products.

3. DESIGN AND IMPLEMENTATION

This chapter presents two applications: a virtual reality client and a web tool. Both were developed as a part of a user community and user involvement tool which aims to efficiently include users in the co-creation process and collect experiences regardless of time and place through diverse clients and methods.

The VR client does this by immersing the participant in 3D virtual environments where they go through a number of location-based 3D questions. The web tool is a web application developed to aid and accelerate the creation of content for the PATIO platform. Built for the WebGL platform, thus easily available, it is used as a simple and interactive creation editor.

3.1. VR Client

The VR client, or Virtual PATIO, places users in an interactive immersive environment. Taking advantage of the immersion quality of such a medium, the desired result is to efficiently collect valuable comments and feedback. The client was developed using 3D model representations of real places / sites and location-based surveys.

Figure 1. User feedback is collected using the VR client.

In Unity, an application or a game and their objects are built in Scenes. They contain elements such as environments, obstacles, decorations, and so on. Each Scene file is a level or a piece of the gameplay. [40]

All the scenes and their required assets put together make the build of an application. In short, it is a package which contains all the files needed to run the application outside of the engine's editor [41].

The VR client contains three scenes:

1. Scene 0 – Program
2. Scene 1 – Model
3. Scene 2 – Virtual Tool

3.1.1. Scene 0 – Program

This is the first loaded scene and its purpose is to provide the functionality needed to prepare and introduce the following scenes. It features a prefab with a script component attached. Some terms need to be defined for a proper understanding.

In Unity, the Prefab asset allows storing a GameObject object complete with components and properties. Thus, new instances can be created from it, speeding up the development process. For example, instead of creating several trees from scratch, it can be more practical to create just one, save it as a prefab, and simply create instances of it after. Overriding settings for each instance is also possible. [42]

GameObject is "the fundamental object in Unity scenes, which can represent characters, props, scenery, cameras, waypoints, and more. Its functionality is defined by the Components attached to it" [43].

Component is "a functional part of a GameObject. A GameObject can contain any number of components. Unity has many built-in components, and you can create your own by writing scripts that inherit from MonoBehaviour" [44].

Lastly, "MonoBehaviour is the base class from which every Unity script derives" [45].

The script introduced before firstly initializes three important parameters: display name, activity id and activity name. These are sent from the Web client as session storage items. The Web client, or Classic PATIO, is one of the clients offered as a participation tool in the development of products and services. The Web client is the environment where activities are introduced, and it is where users launch the VR client from.

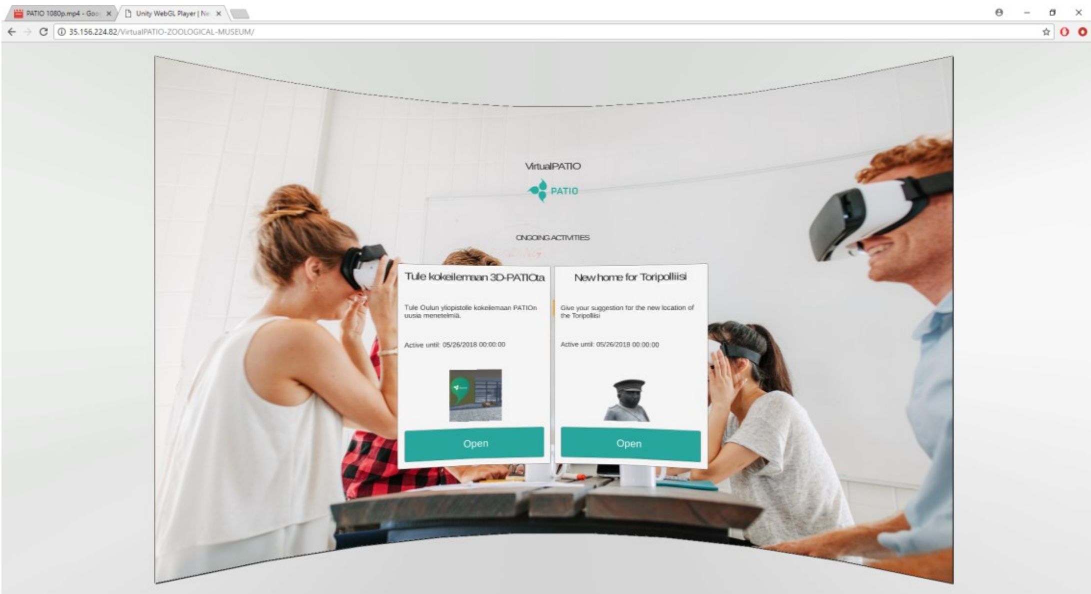

Figure 2. Screenshot of the VR client during the early stages of design and development. The VR client here cannot yet communicate with the Web client. Thus, the user had to manually select their desired activity again, using the available activity cards, and users were nameless.

The data is saved using HTML5 web storage, namely the sessionStorage object. With it, our data is stored as key/value pairs locally within the user's browser for only the current session. [46] This is a good alternative to using cookies as it is more secure, and only pages from the same origin can access the same data [47].

However, the VR client needs to be able to interact with browser scripting for this method to be successful. A way to do this is to directly interface with the browser's JavaScript engine through plugin files, and thus call JS functions from Unity scripts [48].

The plugin function returns a session storage item with a given key "str". More specifically, the keys are "display_name", "activity_id", and "activity_name". We use this returned data to retrieve the activity data in the next step.

The term activity is used to refer to any use case that users undertake related to a specific topic or the development of a product or service. For example, the main use case discussed later in this thesis applies to how 3D models of zoological animals could be utilized at campus area and inside the campus virtual environment.

To get information about this activity and, more importantly at this stage, the name of the 3D model associated with it, textual data is retrieved from our HTTP web server using the "UnityWebRequest.GET" call, with the target URL as a string argument [49].

The resulted textual data is deserialized into a list of Activity objects using Json.NET [50]. JSON stands for JavaScript Object Notation and is "a syntax for storing and exchanging data" [51].

The JSON data and *Activity* object share the same structure:

- ID, *int*
- Name, *string*
- List of *Client* objects, *List<Client>*. *Client* object structure:
  - Name, *string*
  - Survey title, *string*
  - Survey response URL, *string*
  - List of *Poi* objects, *List<Poi>*. *Poi* object structure:
    - Latitude*, double*
    - Longitude, *double*
    - Question href, *string*
    - Page ID, *string*
- Model*, string*

The name of the model with corresponding activity ID and name is returned.

Once the name of the model is returned, the remaining two scenes, Scene 1 – Model and Scene 2 – Virtual Tool, are loaded additively and thus keep the first scene, Scene 0 – Program, active. The model scene contains the corresponding 3D model.

In addition, one more functionality is provided by the script. Pressing down the Esc key closes all loaded scenes and loads the initial one. In other words, it reloads the application.

Lastly, the scene contains a directional light to simulate the sun and illuminate the objects [52]. It is the one and only light source and is used in a calculation process known as baking [53].

The Light object in the scene provides properties such as intensity, direction and color of the light to calculate the shading of illuminated 3D objects [53]. A 3D object is "a 3D GameObject such as a cube, terrain or ragdoll" [54].

3.1.2. Scene 1 – Model

The model scene contains the corresponding 3D environment of an activity.

Two complementary parts are implemented to form a model scene:

1. Mapbox map object [55]
2. 3D model representation of site, homemade with 3D creation software

Mapbox is “an open source mapping platform for custom designed maps” [55]. The map component covers a larger area and integrates location into the applications. Some properties are initialized to create such a map at a location of interest accurately. In this way, the two parts defined above match with regard to position, rotation and scale.

The properties are:

- Position
- Latitude and longitude
- Zoom
- Tile size
- West, north, east and south tile range

A map is loaded in tiles. To better suit the needs of this application, the tiles constructing a map are further configured. Mesh Collider components are added to handle physical collisions [56], and loading of buildings is skipped.

The VR client currently supports activities at two locations. However, the system is designed to easily support new locations. The minimum requirement is setting the latitude and longitude geographic coordinates.

The two locations are:

1. Oulu City Center
2. Campus / Tellus

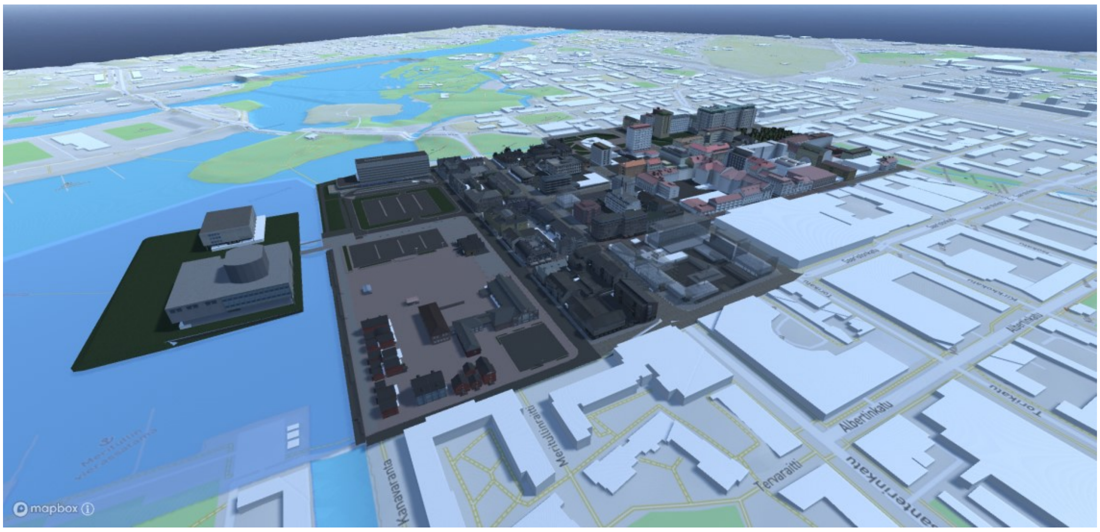

Figure 3. Oulu City Center model scene.

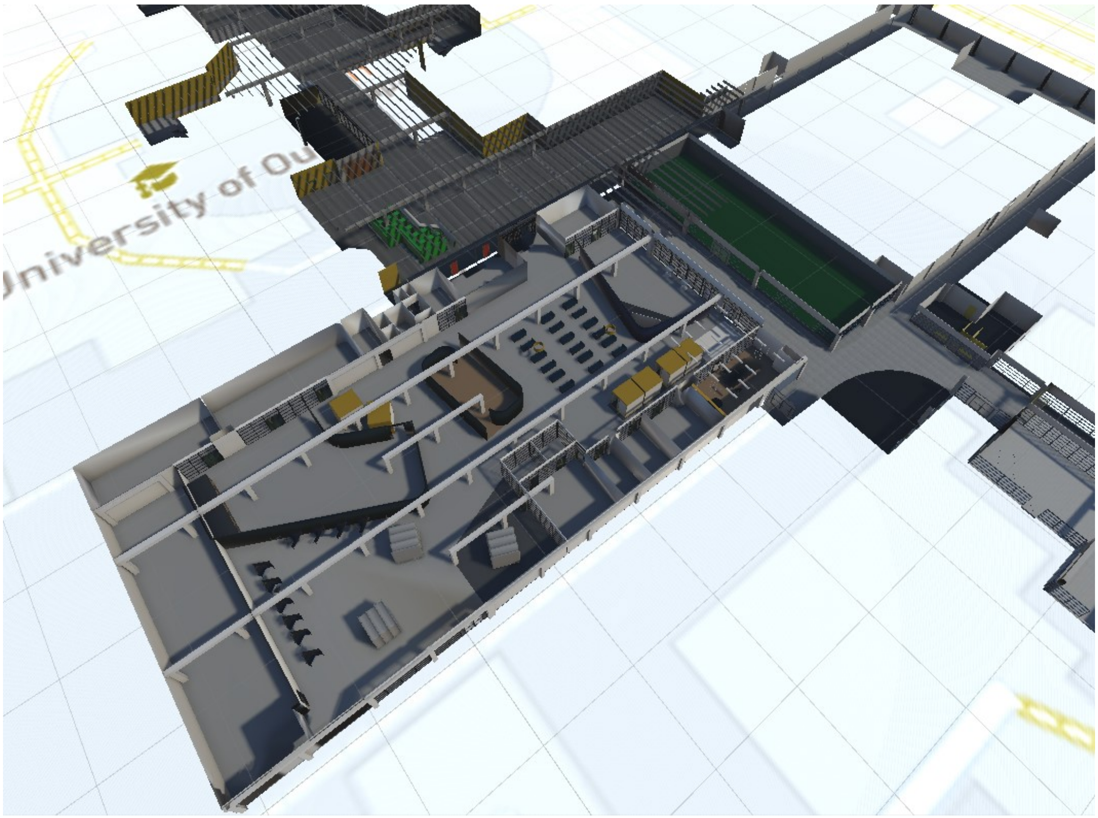

Figure 4. Campus / Tellus model scene.

The map object used for the Oulu City Center location has the following properties:

- Position = 0, -16, 0. Must be below the 3D model to avoid overlap
- Latitude and longitude = 65.01667, 25.46667
- Zoom = 16. Matches well the scale of the 3D model
- Tile size = 259.2116. Matches well the scale of the 3D model
- West, north, east and south tile range = 2, 0, 1, 3. These values are used to enclose the entire 3D model

The map object used for the Campus / Tellus location has the following properties:

- Position = 0, -16, 0
- Latitude and longitude = 65.0586733, 25.465562
- Zoom = 16
- Tile size = 259.2116
- West, north, east and south tile range = 0, 0, 1, 1. These values are used to enclose the entire 3D model

The 3D models are manually positioned and rotated to match their real-life counterparts.

3.1.3. Optimizing Graphics Performance

3.1.3.1. Cameras and Occlusion Culling

The Occlusion Culling feature is used to optimize the rendering process and increase the performance of the application. Occlusion Culling ensures that only the geometry visible to the camera, and not obscured (occluded) entirely by other nearer geometry, is rendered. It is especially useful to apply this feature when working with complex, static 3D environments, such as the cases discussed here. In Unity, Frustum Culling is enabled by default. Thus, any geometry outside the camera's viewing area is not rendered either [57, 58]. "Cameras are the devices that capture and display the world to the player" [59].

The first step to setting up Occlusion Culling was to mark all renderers in the scene as Static. The Static flag lets the system know that those renderers will not be moved and that they should be taken into account when baking the occlusion data [58].

The second step was to set the parameters and bake the Occlusion Culling data.

To this end, the default/recommended values for any scene were tried:

- Smallest Occluder = 5
- Smallest Hole = 0.25
- Backface Threshold = 100

The generated occlusion data, sized 1.3 MB for the Oulu City Center 3D model and 50.4 KB for the Campus / Tellus model, is stored and utilized by the application automatically.

3.1.3.2. Lighting and Lightmapping

Virtual PATIO makes use of a feature called lightmapping. Like Occlusion Culling, it increases the performance of a scene by considering the components (meshes, materials, textures, lights) in it to pre-calculate the brightness of surfaces. The resulted data is stored in lightmaps which the application uses automatically [60].

A similar configuration process to the Occlusion Culling one is required to prepare the scene and bake the lightmaps. The first step was to generate lightmap UVs for selected models. Next, renderers in the scene were marked as Static to be included in the lightmapping process.

Default lighting and lightmapping settings were tweaked to balance lighting quality, bake time and size of the generated lightmaps:

- Realtime Global Illumination = false
- Baked Global Illumination = true
- Indirect Samples = 256
- Lightmap Resolution = 32
- Lightmap Size = 512
- Ambient Occlusion = true
- Directional Mode = Non-Directional. [60, 61]

The lightmapping bake process generated 188 Non-Directional Lightmaps: 188x512x512px, 62.7 MB for the Oulu City Center 3D model and 52 Non-Directional Lightmaps: 52x512x512px, 17.3 MB for the Campus / Tellus model. Data is stored and utilized by the application automatically.

3.1.4. Scene 2 – Virtual Tool

This scene adds interactions and interactive objects into the application: user controls, 3D user interfaces (UIs), surveys, and more.

Figure 5. An example of a 3D survey question from an early development iteration. User gives feedback via a thumb signal.

3.1.4.1. Player

The Player object loads together with the scene. Here, Mozilla's assets for creating WebVR-enabled experiences with Unity are utilized. The web VR camera set prefab provides the camera functionality [62, 63, 64].

In addition to the readymade camera set prefab, a configuration logic was implemented to enable two modes to experience and interact with the environment, involving the three cameras. This gives a user the possibility to 1. experience an activity fully immersed into VR using a virtual reality headset and gaze input method, or 2. participate using the application as a normal web app with mouse input.

The modes are Normal and Enabled. The cameras are CameraMain, CameraL and CameraR. They are configured as follows:

1. Normal (vrState == WebVRState.NORMAL)
    a. CameraMain enabled
        i. Field of View = 80
    b. CameraL disabled

   c. CameraR disabled
   d. Interact using MOUSE
   e. VR graphical elements deactivated
   f. User interface responds to CameraMain events
2. Enabled (vrState == WebVRState.ENABLED)
   a. CameraMain disabled
   b. CameraL enabled
      i. Position X = -0.032
      ii. Viewport Rect X = 0, W = 0.5
   c. CameraR enabled
      i. Position X = 0.032
      ii. Viewport Rect X = 0.5, W = 0.5
   d. Interact using GAZE
   e. VR graphical elements activated and progress image set
   f. User interface responds to CameraL events

All these settings are required to support both modes, normal web use and VR, both user inputs, mouse and gaze, and create a seamless transition between these modes.

A world space UI canvas is attached to CameraL and provides some important graphical elements for when the VR state is enabled. These are: a panel that is faded in or out at times, a circle image that serves as the user's gaze (cursor), and a progress bar that fills during interactions. The user's gaze in this application is in the center of the screen, marking where they are looking, and is used to interact with elements [65].

Gaze is a commonly used form of user interaction in mixed reality (MR) and VR. It is used to mark where the user is looking and manipulate interactable objects. [66]

3.1.4.2. Map, Minimap, Camera Map

A minimap shows the surrounding area and points of interest to the user. It helps them orient in large areas, visualize the environment as a 2D image, and see where the questions are in the 3D world. The map uses an orthographic camera that captures the user's surroundings, positioned above the Player object, and rotated to look at it. What the camera sees is rendered into a Render Texture during runtime [67], and then showed to the user as a 2D image. The Player object and other points of interest (POIs) are marked with 2D icons positioned above these in the 3D environment to make the map useful. Using buttons added to the map's user interface (UI), users can decrease or increase the camera's orthographic size to zoom in or zoom out the map. For a better quality, the size of the texture used here was set to 1600x1000 pixels. The camera always follows the user, so the player icon is always centered. A red circle marks the position of the player.

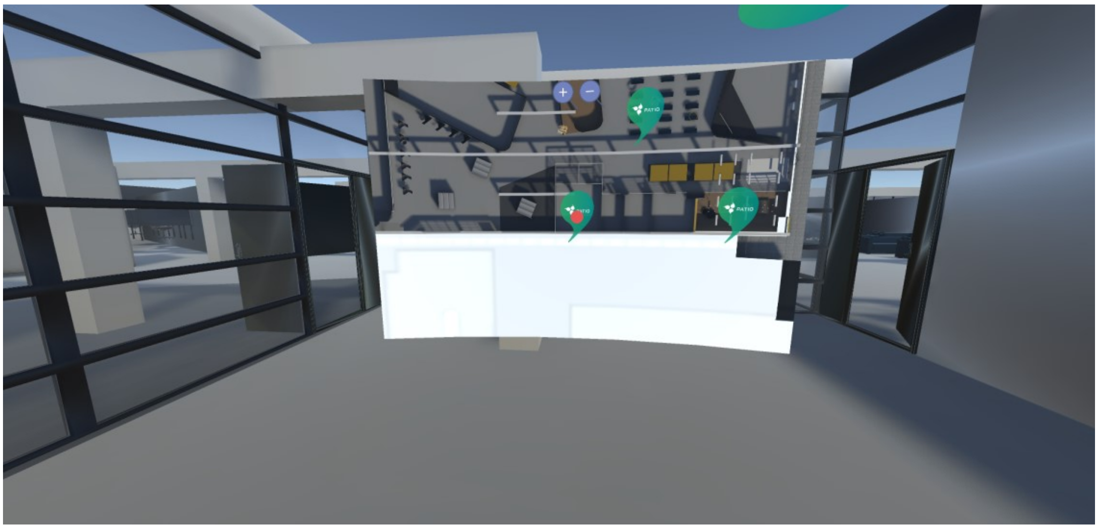

Figure 6. The minimap represented as a 2D image in 3D space shows the POIs marked with green bubbles, the player marked with a red circle, and the surrounding area.

3.1.4.3. Point of Interest (POI)

Generally, a point of interest is a location that may be of interest to someone. The concept of POI appears often in software applications that use digital maps. An example is the location of the Great Pyramid of Giza on a map, or the location of the grocery store nearby. [68]

For this application, a POI is a point location in the 3D environment where users go to, to give feedback in the form of survey questions.

The POI object defines the following data:

- Client name, *string*
- Latitude, double
- Longitude, double
- Question href, *string*
- Page ID, *string*

A green bubble image positioned above a POI marks its location in the 3D environment.

3.1.4.4. Survey Software Tool

Virtual PATIO is powered with a survey software called SurveyMonkey [69]. SurveyMonkey is used together to create surveys, collect responses and analyze results. Its REST-based API enables the VR application to access the features and data needed.

The survey software was integrated seamlessly. The VR client gets the survey questions, collects user responses and sends them back to the survey software where they are stored and analyzed.

The API returns responses in JSON. An access token is provided and required whenever a request is made.

3.1.4.5. Survey

When a VR activity is launched, the application gets that activity's related data in the same way as explained for the first scene: the application sends a GET web request to retrieve the textual data from our HTTP web server, and deserializes it into a list of Activity objects.
The JSON data associated with an activity has the following (example) structure:


```
{
  "id": 0,
  "name": "UNITY_EDITOR",
  "clients": [
    {
      "name": "MobilePATIO",
      "survey_title": "Title",
      "survey_responses_url": null,
      "pois": [

      ]
    },
    {
      "name": "VirtualPATIO",
      "survey_title": "VR-test-survey",
      "survey_responses_url":
            "https://api.surveymonkey.net/v3/collectors/211579625/responses",
      "pois": [
        {
          "lat": 65.05827150537264,
          "lng": 25.465615838358215,
          "question_href":
            "https://api.surveymonkey.net/v3/surveys/151788580/pages/29177633/que
            stions/79605686",
          "page_id": "29177633",
          "target_href": "-",
          "position": "1"
        }
      ]
    },
    {
      "name": "AR-PATIO",
      "survey_title": "",
      "survey_responses_url":
            "https://api.surveymonkey.net/v3/collectors/211579625/responses",
      "pois": [

      ]
    }
  ],
  "model": "Tellus"
}
```

Having stored the activity ID and name, the correct activity is brought into play. Each VR activity holds a list of POIs. The program iterates through the list to populate the scene with POI objects. For each POI, the latitude/longitude is converted to a world space Vector3, and a new 3D object is instantiated at this position. The VR client gets geographic coordinates for POIs and thus conversion is required to position them accurately within the 3D environment. The questions linked to the POIs are loaded and created next. To do this, a GET web request is sent to each POI's Question href to retrieve information about it. The request accesses the data of a question from SurveyMonkey (SM) through its REST-based API [70]. The Question href is the access URL for this request. The access token is also set here.

The request retrieves textual data in JSON that is deserialized, for the next operations, to a Question object:

- sorting, *object*
- family, *string*
- subtype, *string*
- required, *object*
- answers, *Answers*:
    - choices*, List<Choice>*
        - visible, *bool*
        - text, *string*
        - position, *int*
        - id, *string*
- visible*, bool*
- href, *string*
- headings, *List<Heading>*
    - heading, *string*
- position, *int*
- validation, *object*
- id, *string*
- forced_ranking, *bool*

The application instantiates 3D question objects based on the deserialized data. A question is made out from four components: a question holder, text component for everything text related, a grid layout group that holds and arranges the answers / choices, and toggle component for choices. A choice is either of type single choice, or multiple choice. All 3D game objects get instantiated during runtime. The data gets passed to create in game questions: question heading is set, question position is set, choices types, texts, and ids are set, and so on.

To create and submit a survey response, the program creates objects that match the required JSON format, reads and assigns the user's answers, and sends them to the online survey software (SurveyMonkey).

Firstly, an object of type RootObject is created:

- pages, *List<Page>*
- custom_variables, *CustomVariables*

This object holds all the information required to submit a response. Its list of Page objects is initialized. Next, the application finds all the question objects present in the scene. A question behavioral component attached to every 3D Question stores two essential variables needed for identification.

This ensures that each 3D Question is linked correctly to its SurveyMonkey question equivalent:

- id, *string*
- page_id, *string*

For each such object, a Question data object gets created. The very same id is assigned, and its list of answers is initialized:

- id, *string*
- answers, *List<Answer>*

The program creates Page objects to hold the questions. Each question is part of a page. All pages are part of the root object:

- id, *string*
- questions, *List<Question>*

Next, all the choices present in the scene are retrieved. A choice behavioral component attached to every 3D choice object (Toggle) stores one essential variable needed to know what a user answers.

This ensures that all the choices of a 3D question are linked correctly to their SurveyMonkey equivalents:

- id, *string*

For each such object, an Answer data object gets created. The very same id is assigned. An answer is part of a question:

- choice_id, *string*

The application uses a CustomVariables class to send the user's display name together with their survey response to the online survey software. Thus, a new object of this type gets created and the display name previously retrieved is assigned:

- var1, *string*

When all these prior operations are done, a response is ready to go. Thus, a new web request of method POST is created. The target URL is the survey's response URL previously stored in the Activity object. The root object is converted to JSON, the access token is set, and the request is sent to the survey software.

3.1.4.6. User Interface (UI)

The user interface gives users interactable elements that enable the application to be practical and effective. It is a collection of texts, images, buttons, toggles, and similar elements. These components provide essential functionality such as starting the VR experience, displaying instructions and questions, user movement from POI to POI, answering questions, interacting with the minimap, and submitting survey responses.

In Unity, there are three types of user interfaces regarding the render mode: Screen Space – Overlay, Screen Space – Camera, and World Space. In VR applications, world space UI is the preferred type since our eyes cannot focus on objects that are too close, and the Screen Space – Overlay mode is not possible at all in Unity VR apps [71].

The UI utilizes an asset called Curved UI, “a curved interface system designed to bend the canvas in world space, allowing the player to view and interact with it from any angle” [72]. The reason for using this tool is to enrich the user experience and make switching the control method nice and easy.

The first panel displayed to the user is the start-window panel. Its purpose is to introduce the user to the VR activity and start the 3D questionnaire.

Figure 7. The start-window panel UI introduces the user to the VR activity.

The next one is the survey panel. It contains a brief instructional text, survey questions, and navigation elements. At the bottom of the panel there are three interactable components. One is the Prev button used to get the user back to the previous question. Thus, it moves the user to the correct location in the 3D environment and sets active the associated question. Another component is the Next button which has the same functionality but gets the user to the following question. A Toggle component activates/deactivates the minimap. Two additional buttons are available to zoom-in/out the map.

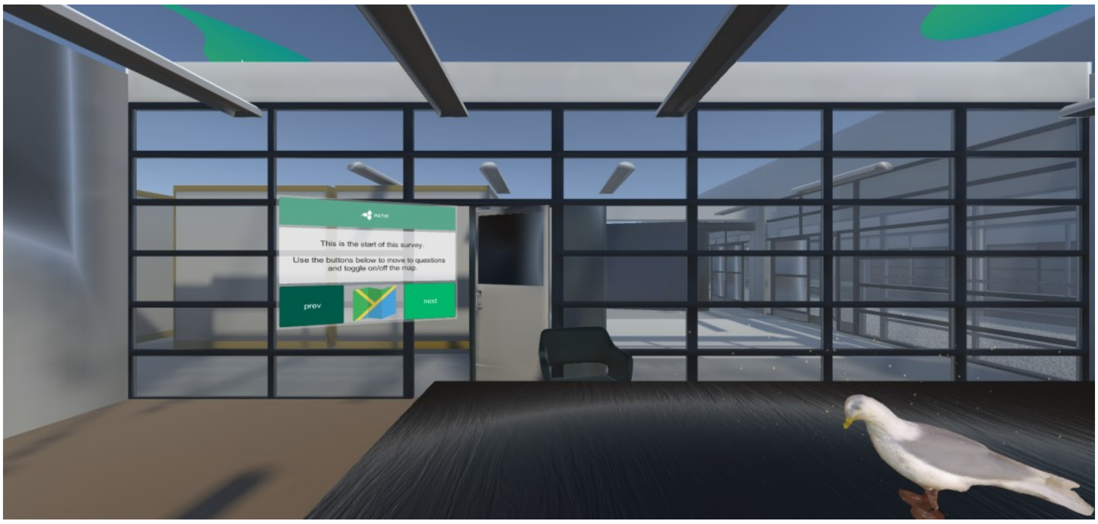

Figure 8. The survey panel UI contains a brief instructional text, survey questions, and navigation elements.

Each 3D question is its own scene object. The user chooses their answer by interacting with the available toggle components.

Figure 9. The user answers a 3D survey question by selecting choices.

A separate instructional panel is displayed at the end of a survey. Here, the user presses the Done button to complete the survey and submit their response. This initiates the process described previously in the Survey chapter.

Figure 10. The user submits a response by pressing the Done button.

To quit the VR activity, the user presses the Quit button enabled at the very end. This takes the user back to the Web client, the place where they launched the VR client from. Thus, the application interacts with browser scripting once again thanks to plugin functions. The button calls the browser’s history back method to load the previous URL [73]. The VR experience concludes here.

3.1.5. Architecture

The first architecture figure presented below is that of the whole living lab environment. It shows, from a high-level standpoint, how the different clients and tools connect and communicate with each other. The connections between these applications ensure that they receive the required data to function as intended. An example is an activity’s id and name, which a client must know to load its associated survey data. The connections also ensure, for instance, that the user data collected is sent successfully to the survey software. This figure is provided for a better general understanding of the living lab environment. However, it covers more than the implementation part of this thesis.

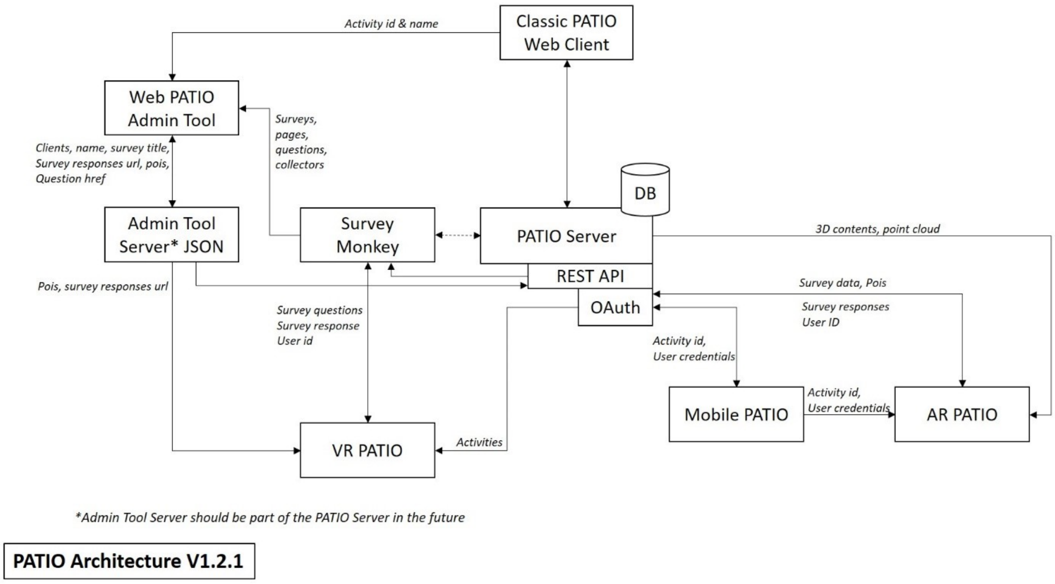

Figure 11. High-level standpoint architecture of the whole living lab environment.

The second architecture figure illustrates the VR client specifically. It shows, from a high-level standpoint, what entities the application communicates with, and what data is sent and received.

Step by step explanation:

1. Run / open application, VR Menu scene loads.
2. An HTTP request is sent and data from PATIO Server, through the REST API is received. The data is the list of activities.
3. Choose an activity to open from the group of ongoing activities. Activities are displayed as cards. The Virtual PATIO Activity scene loads, and the id & name of the activity are supplied.
4. An HTTP request is sent and data from Admin PATIO Server, JSON is received. The data includes the list of pois and the survey responses URL for this activity (id & name).
5. An HTTP request is sent and data from SurveyMonkey server is received through the REST API, for each poi. The data is the survey question of this poi. Each poi has a question href property to get the data.
6. Questions are answered, survey response is submitted. An HTTP request is sent, and data is supplied to SurveyMonkey server. The data is the survey response.

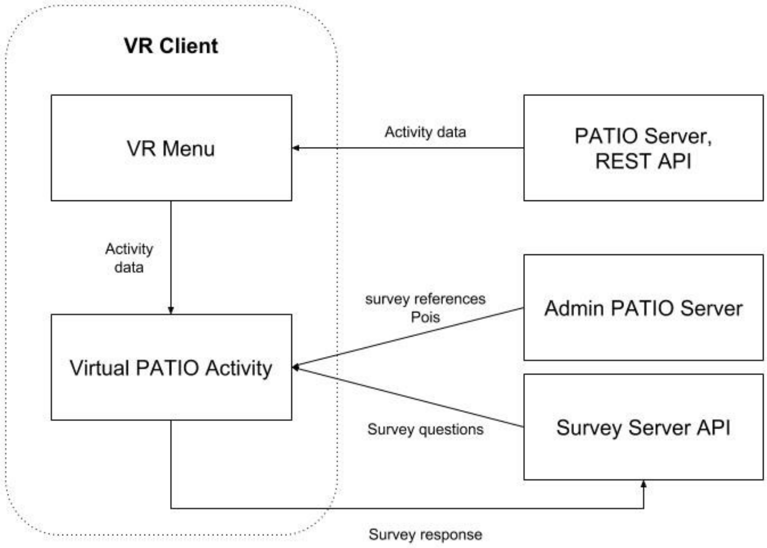

Figure 12. High-level standpoint architecture of the VR client.

3.1.6. Build & Platform

The scenes explained previously form the build of the VR client application. The build is the application package ready to be used outside of the development editor [41].

Virtual PATIO is built for the WebGL platform. Thus, it is published "as JavaScript programs which use HTML5 technologies and the WebGL rendering API" [74]. Virtual PATIO runs in a web browser.

Additionally, the application integrates a WebVR tool that provides support for virtual reality devices [62, 64]. Thus, users can run the VR client directly in a web browser, and use supported virtual reality headsets.

3.2. Web Tool

3.2.1. Introduction

The web tool, also known as Admin Tool, was developed and integrated so that it would be usable and useful for the administrator and moderator of the user involvement platform. Thus, Admin Tool is a web application developed to aid and accelerate the creation of content for the PATIO user involvement and co-creation platform. Built for the WebGL platform, thus easily available, it is used as a simple and interactive creation editor.

During the early stages of design and development it was resolved that building Admin Tool as a web application will be the most beneficial, for two main reasons. Firstly, the original service was being developed already as a website. Thus, it makes sense architecture-wise that our new application is included in the same environment. Secondly, the overall design of the system set out to keep things user-friendly and seamlessly integrated. This would not be possible if Admin Tool was developed for a different platform.

To successfully integrate Admin Tool, the app must communicate with the PATIO parent website certain information. Thus, given that every co-creation activity has its own identifiers, the information is passed forth in the form of session storage items. This solution is possible given that both apps run in the same environment.

Admin Tool was developed to be used by administrators and moderators of the user involvement platform (PATIO). Its usability and usefulness are in enabling creation of content, visually and interactively, for the three user involvement clients at the same time, without requiring admins to have a lot of understanding of how these clients work.

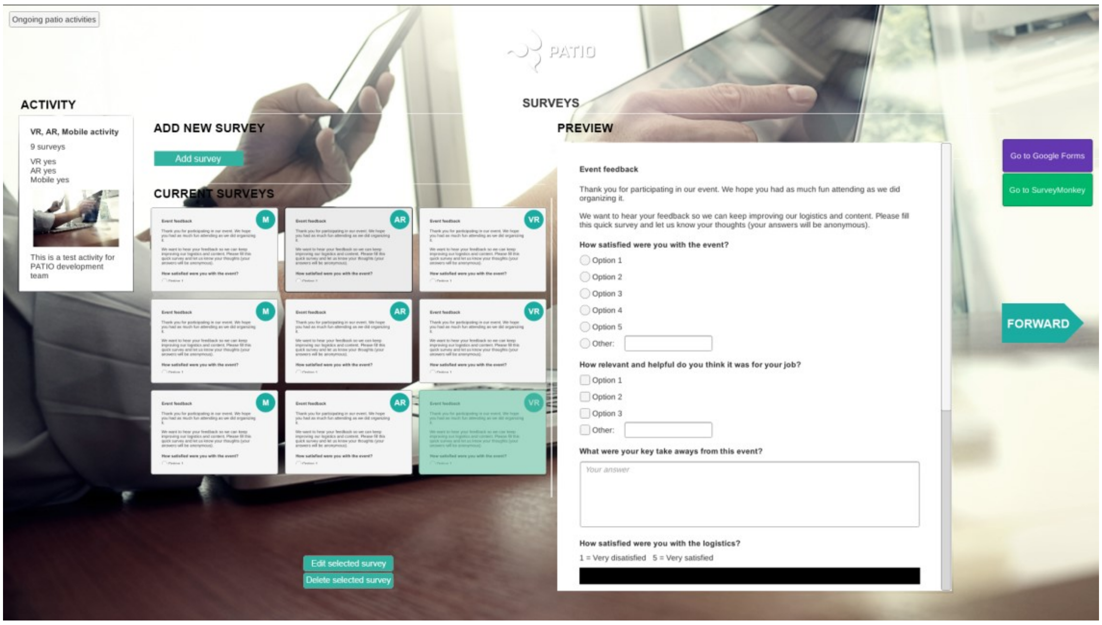

Figure 13. Screenshot of Web Tool during the earlier stages of design and implementation. Although the UI was simplified and the map functionality better integrated, it nicely shows the design thinking behind this tool.

3.2.2. Clients and Content

User involvement is realized through three clients: mobile, VR and AR. Users use these applications to experience / take part in different activities, and ultimately give meaningful feedback to involved parties. The clients are different in the way they reach users and how they experience activities through different technologies, while taking advantage of what these mediums offer. For example, mixed reality apps are generally more immersive than mobile apps. Each of these applications require certain content to be useful and thus fulfil their purpose.

Content is what forms an activity and it is defined as a list of data, containing information such as data about the activity in question, points of interest (POIs), survey question references, etc.

The web tool contains four scenes:

1. Scene 0 – Program
2. Scene 1 – Select Model
3. Scene 2 – Model
4. Scene 3 – Admin Tool

3.2.3. Scene 0 – Program

Overall, this scene works in the same manner as the Program scene of the VR client. It is the first loaded scene and its purpose is to provide the functionality needed to prepare and introduce the following scenes. It features a prefab with a script component attached. For initialization, the display name, activity id and activity name are assigned. This information comes from the user involvement tool's web client in the form of session storage items. For this purpose, Unity WebGL provides methods to interact with browser scripting and call JavaScript functions from Unity scripts. We get the session storage items by calling our JavaScript plugin function.

There are some things that happen differently, however, compared to the VR client. Firstly, the next scene, Scene 1 – Select Model, is loaded immediately after in additive mode, where user input is required to load the remaining scenes. In addition, the model property is set when the user selects a model which then sets the 3D environment of an activity, and loads the remaining scenes, Scene 2 – Model and Scene 3 – Admin Tool, additively.

3.2.4. Scene 1 – Select Model

This part of the application loads immediately after and features a UI canvas with two buttons representing the two available 3D environments. It gives users the option to select a model / environment. The same one will be used, for example, in the VR client once the activity is finalized and saved. At the moment, there are two options, Tellus and Oulu City Center. More models could be added in the future. When a selection is made, the program unloads this scene and loads the following two. The model property is stored.

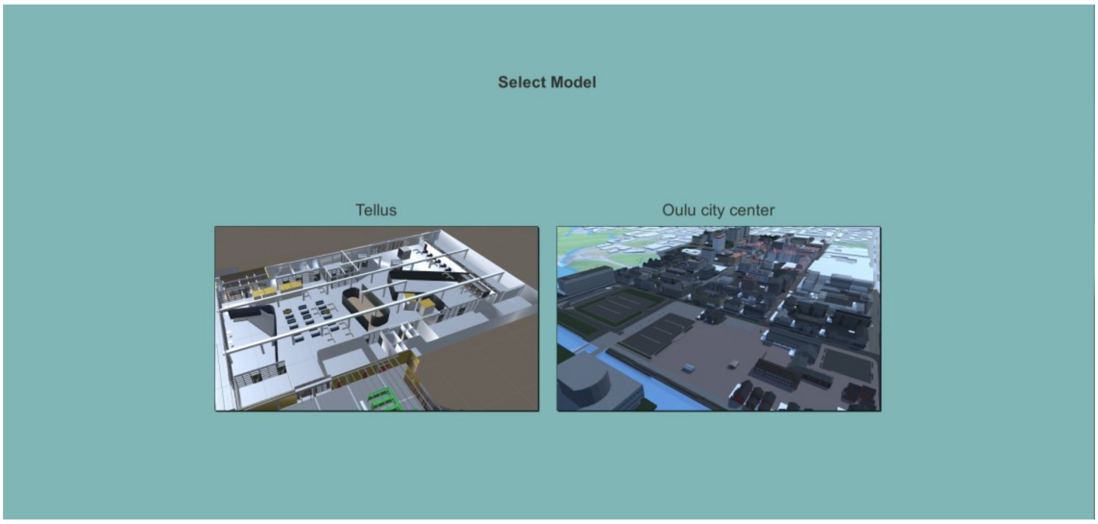

Figure 14. The user selects one of the two available 3D environments.

3.2.5. Scene 2 – Model

The design and implementation of this scene is identical to that of the VR client's Model scene.

3.2.6. Scene 3 – Admin Tool

The Admin Tool scene contains the bulk of the application's functionality. Here users interact with the 3D environment and available creation tools to create and edit user involvement activities. Moderators can add surveys, remove surveys, create, delete and move POIs at desired (geo)locations, add POIs to one or several user clients, add a survey question and link 3D content to it. When the creation process is finalized, a moderator saves and makes the activity available to the user clients.

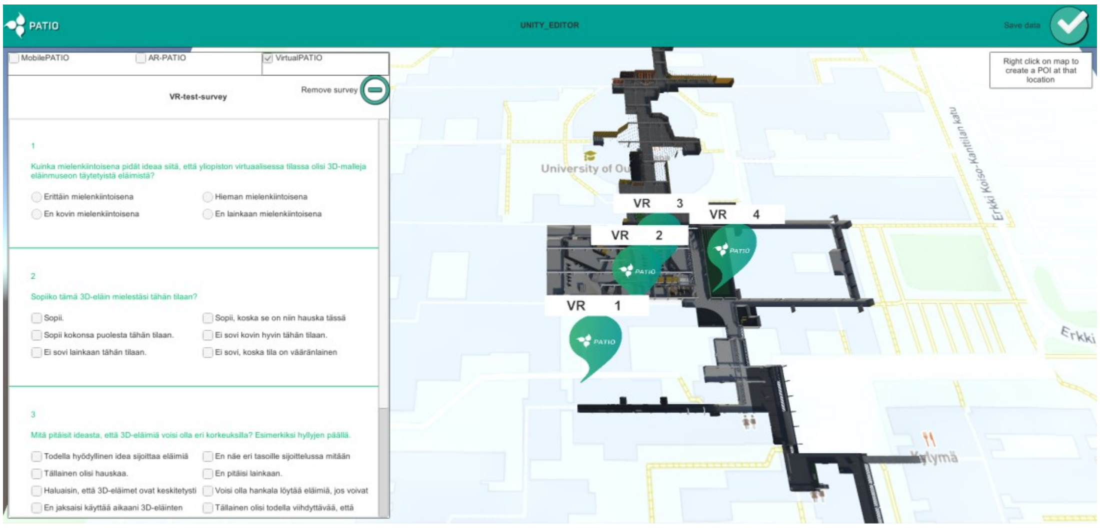

Figure 15. Moderators use creation tools to create and edit activities.

3.2.6.1. Camera

A perspective projection camera is used to render the world from above. It enables the user to navigate within the environment with panning and zooming functionality. Thus, a moderator creates the content of an activity from a map-like viewpoint.

3.2.6.2. POI

A POI is created at a location of interest by right clicking on the map. In Admin Tool, a POI is represented with a 2D bubble image. When the user first selects a location, a placeholder bubble image appears, and a menu gives the options to create a POI there or cancel the selection. Clicking a POI highlights it. In addition, two properties are displayed together with a POI, the target client and question number.

A POI 3D object holds information such as client name, latitude and longitude, question href, page id, target href and question position in the associated survey. These properties are used to modify and customize POIs.

The user can move a POI from one location to another by dragging the bubble across the map. To remove a POI, the user clicks the Delete button in the associated POI panel. The POI panel is used to customize a POI. It shows and keeps track of the target client, question number, 3D content URL and latitude and longitude properties of a POI.

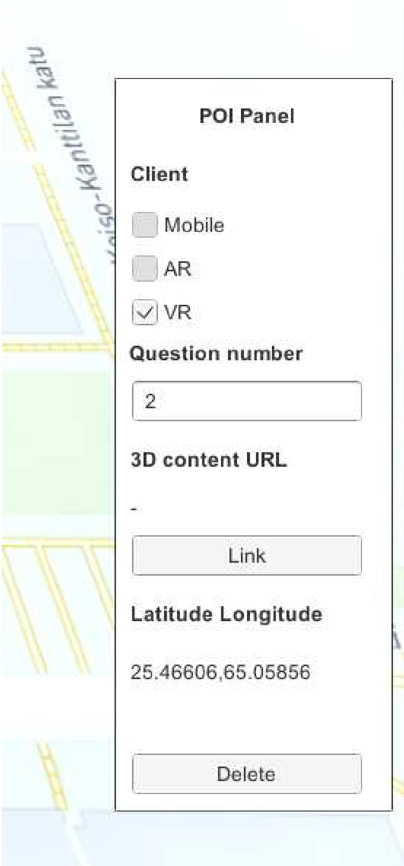

Figure 16. A POI's settings are configured via the POI panel UI.

3.2.6.3. UI

An element that provides important functionality to the creation process is the survey UI. It always shows the surveys associated with the activity in question, for each target client. The moderator navigates through using three toggles, one for each client. Each survey panel also has an Add survey button that activates an input field where the title of a requested survey is inputted. The text inputted must match the title of a survey in the external survey software tool. To remove a survey, the moderator clicks the Remove survey button. Each survey panel displays the survey title and its questions. The moderator can thus visualize and manage surveys, questions and POIs in one place.

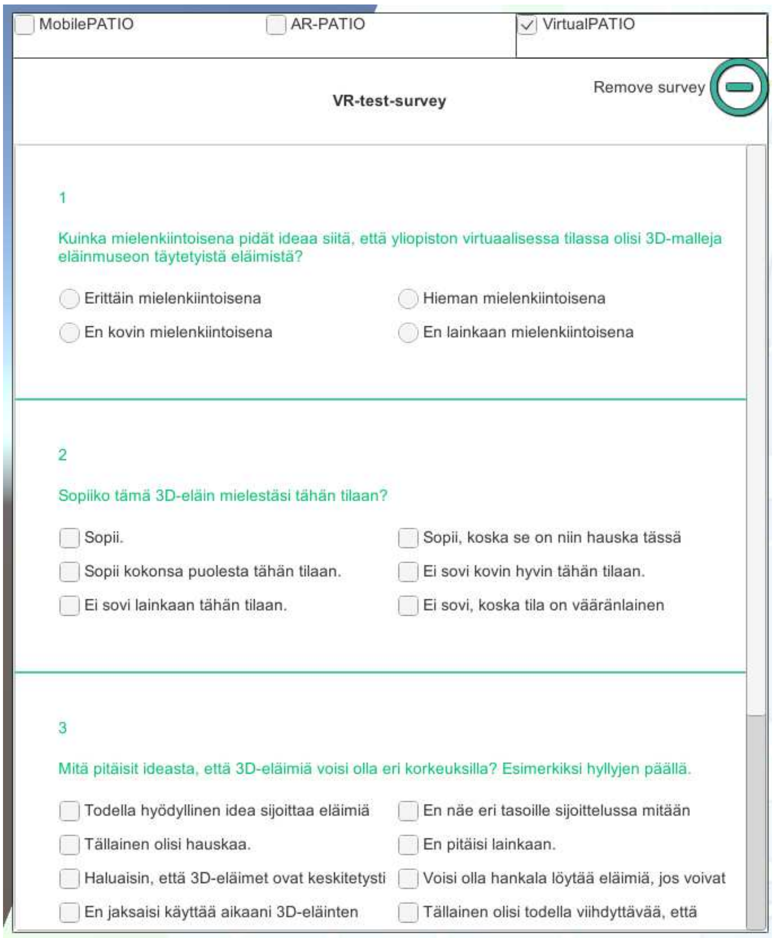

Figure 17. The survey panel UI has one survey for each client and displays these surveys that come from the external survey platform to the moderator.

The Save data button in the top right corner saves and pushes an activity's associated data to the server and thus make it available to the user clients.

3.2.6.4. Activity

An activity is a setting or experience that the moderators of the user involvement tool create to involve users and receive feedback, related to a selected topic, from them.

For instance, if future renovations of an area require temporarily moving a sculpture located there, where would be a good location to move it to?

The Activity blueprint is defined as follows:

- ID, *int*
- Name, *string*
- List of *Client* objects, *List<Client>*. *Client* object structure:
  - Name, *string*
  - Survey title, *string*
  - Survey response URL, *string*
  - List of *Poi* objects, *List<Poi>*. *Poi* object structure:
    - Latitude*, double*
    - Longitude, *double*
    - Question href, *string*
    - Page ID, *string*
    - Target href, *string*
    - Position, *string*
- Model*, string*

When an activity is opened, the application retrieves information about it from the web server. The program checks for an activity that matches the activity's id and name. If such information exists, the title is displayed together with each client's (AR, VR, Mobile) associated survey in the UI. Moreover, for each of the three clients, a list of POIs is retrieved. 3D POI objects get created and initialized at correct geolocations, with their associated content: client name, survey question data, linked 3D content. If the opened activity does not have any associated data stored on the web server, so it is a new activity, the moderator begins creating one from scratch.

To put the associated surveys of an activity in the application, information about these must be requested from the online survey software tool (SurveyMonkey). This is done by sending a web request to the API endpoint. A survey's title and the access token provided are enough for this purpose. Once the data is retrieved, the surveys are put into play. Thus, UI elements get instantiated to show a survey's content.

Another important feature is the possibility to save an activity so that it can be used by participants of a study using the available clients. To do this, a moderator clicks the Save data button to push the data to the web server. The program saves the surveys' related information and creates a new Activity object. The surveys' titles and response URLs are first set. The response URL is necessary so that the user clients know where to send the participants' feedback. This feedback is sent from the user clients to the online survey software where it can be further analyzed and utilized. If an activity already has associated data, this data is updated. Otherwise, a new activity entry is created. The application also searches for information related to POIs. Thus, their data is saved in an easy to use way by the user clients. In addition, the model property is saved so that the clients know what 3D model to load when a user starts a VR activity, for instance. All this information is sent to the web server by a PUT method web request.

In the user clients, feedback is sent to online collectors of the survey software. The clients mentioned use a Web Link collector type, which is a quick and versatile way to collect responses.

The Collectors blueprint is defined as follows:

- Per page, *int*
- Total, *int*
- Data, *List<Datum>*:
    - Href, *string*
    - Name, *string*
    - ID, *string*
- Page*, int*
- Links, *Links*
    - Self, *string*

3.2.7. Architecture

The architecture figure below illustrates the web tool specifically. It shows, from a high-level standpoint, what entities the application communicates with, and what data is sent and received.

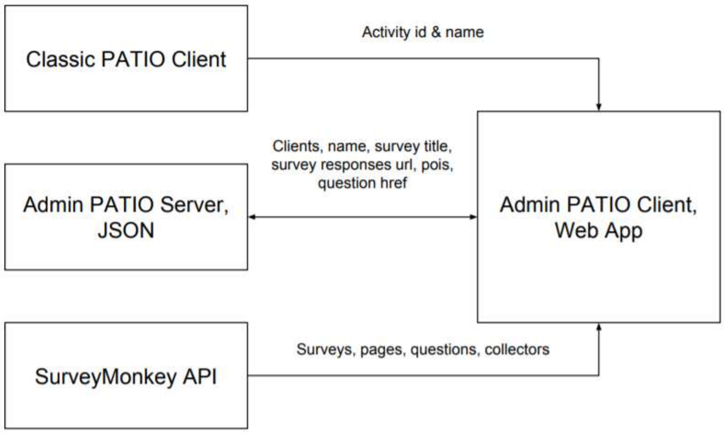

Figure 18. High-level architecture of the web tool.

Step by step explanation:

1. Run / open application. For initialization, the display name, activity id and activity name are assigned. This information comes from the web client, ClassicPATIO, in the form of session storage items.
2. An HTTP request is sent and data from Admin PATIO Server, JSON is received. The data includes everything is known about an activity: clients, name, survey title, survey responses url, pois and question href.

3. For each survey, an HTTP request is sent and data from SurveyMonkey server is received through the REST API. The data is the survey details, including pages, questions, answers, choices, etc.
4. A PUT HTTP request is sent to Admin PATIO Server, JSON, on Save data. The data is information to be saved about activity in question.

3.2.8. Build & Platform

The scenes explained previously form the build of the web tool application. Admin Tool is built for the WebGL platform and runs in a web browser. In addition, the application is embedded into the user involvement tool and launched from there.

4. EVALUATION

The evaluation process consisted in participants testing the VR client. A test activity was created for this purpose. The context of this activity was how 3D models of zoological animals could be utilized at campus area and inside the campus virtual environment. Thus, the campus area and a small number of zoological animals were represented with 3D models in this activity. The study included several locations, or points of interest, within the virtual environment. At each location, a 3D survey question was set, and user feedback was collected. The questions were with regards to the 3D environment space of the activity in question. Participants used an Oculus Rift VR headset to visualize and interact with the surrounding environment. They interacted with and gave answers to survey questions using a gaze input feature. No game controllers were used for interaction in the study. A reticle-like element was implemented in the center of user's vision. The study was conducted using a computer station in a selected location within the campus area. The 3D virtual environment depicted in the activity was a representation of the same campus area where the study was conducted. User feedback was collected and sent to the online survey tool, which was integrated in the application, at the end of each user test.

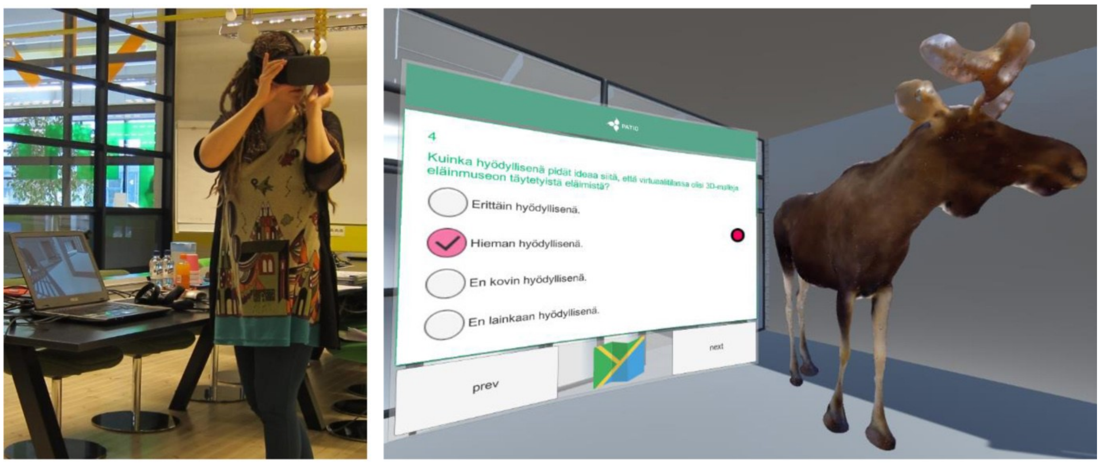

Figure 19. One of the study participants gives feedback while using the VR client, and what she sees [37, 75]. ((c) Authors. Used with permission).

The aim of this study was to evaluate the users' subjective experiences of the 3D virtual environment and UI. Thus, the evaluation focused on studying user experience by gathering descriptive qualities of the prototype while the users launched into the test activity and answered survey questions within the virtual environment. In addition, a focus was on the importance of this type of activity, where they can answer survey questions related to a 3D virtual space. User feedback was collected at the same time participants were using the VR application and encountering the 3D content associated with the activity in question.

The VR client was tested together with other user clients developed for the same user involvement tool. Participants filled a background questionnaire before the test session.

14 participants took part in our evaluation process. Out of 14, 11 participants joined our study through the user involvement tool's existing website, and 3 joined in through personal contacts. During evaluation, users were divided so that half of them

test the VR client first, and then the AR client, while the other half, the other way around. Users' ages were between 26 and 59 years old, with an average of 36 years old. Out of 14 participants, 43% were female. Each test lasted around one hour, during which the test users were watched and interviewed. The test activities have also been recorded. User experiences were also gathered using a five-point Likert scale questionnaire and an Adjective cards selection method [76]. User feedback was further collected through forum discussions and diaries in our website.

The evaluation process was conducted as follows:

1. A background questionnaire in paper format:

· This was an icebreaking and motivational phase. The users were presented a short video (2 min) of user involvement tool including four clients: Web, VR, Mobile and AR.

2. or 3. Use of VR Client

· The users were requested to explore the 3D environment and content and answer to questions in VR by gaze.

· After the use they we instructed to select four out of 42 adjective cards to describe their experience with VR or AR Client and comment their selections.

· Users filled the Likert scale questionnaire (10+11 statements)

3. or 2. Use of AR Client

· Users explored the 3D content by walking about the test location to find the 3D animals and questions. Both AR and VR options had a pop-up map to guide the users.

· After the use the users were asked to select four out of 42 adjective cards to describe their experiences relating to AR Client and comment their selections.

· Users were requested to fill the Likert Scale questionnaire (11+11 statements). [37, 75]

5. RESULTS

Overall, test users' involvement in the VR client's evaluation was very positive. Participants' experiences regarding this application were generally positive, according to the results of the Adjective Card Selection method [76]. Thus, the most selected adjectives to describe the VR client were Fun (5), Innovative (5), Entertaining (4), Easy to use (4) and Visually pleasant (4).

On the other hand, the negative adjectives selected by participants indicate usability issues or a need for better visual quality. One consideration for the VR application was to make it more accessible, from one's home for instance, thanks to WebVR. Thus, the visual quality of the 3D virtual environment was high but not particularly detailed.

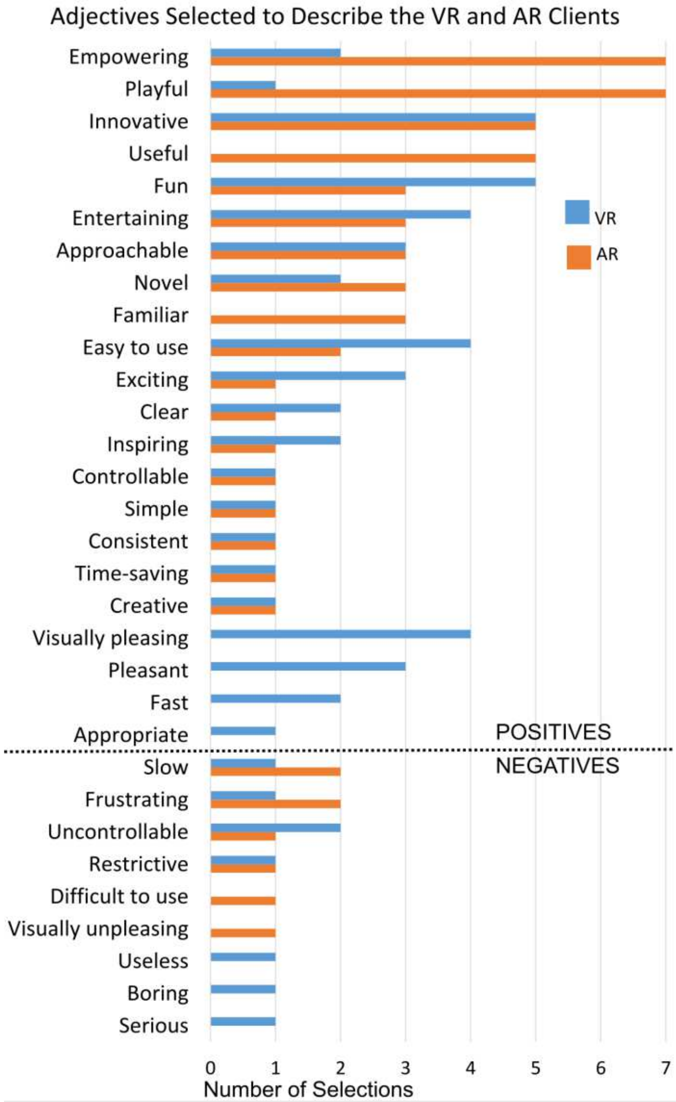

Figure 20. Adjectives selected by study participants. The figure shows adjectives for both VR and AR clients in a comparison. 4/42 adjectives from a shuffled selection of 21 positives and 21 negatives were selected by each user. [37, 75] The Adjective Card Selection method was used for evaluation [76]. ((c) Authors. Used with permission).

Regarding use of interaction, the gaze input method was perceived to be Easy (4.3), Fast (4.1), Interesting (4.4), Fun (4.4), and Useful (4.2), as Likert Scale answers show. However, most participants would have still liked to use VR controllers as means of interaction. Although answering using gaze in VR was quite easy and natural, some participants selected the Uncontrollable adjective to describe it.

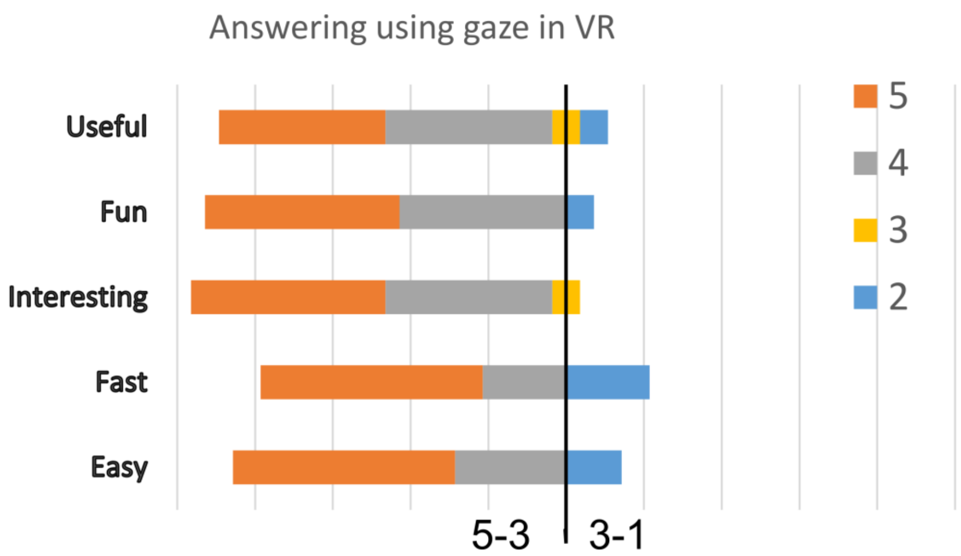

Figure 21. Answering questions using the gaze input method in VR was evaluated and produced positive results. The diverging chart shows the Likert Scale results. [37, 75] The Adjective Card Selection method was used for evaluation [76]. ((c) Authors. Used with permission).

The test users were eager to explore the VR activity and give comments related to how this type of VR application could be employed for different purposes, such as urban planning, education, or entertainment. They considered the VR activity and giving feedback via VR questions to be different from "answering to boring survey" because doing so in a virtual environment, while exploring different points of interest, was exciting and fun.

Our use case, "3D animals of the zoological museum", made the application feel exciting. One user (ID 1) said "It was so exciting that the sea gull was on the wall, I wondered when it starts flying over me and the moose was so big, although you know that that they are not real". Other participants, who selected Fun and Entertaining adjectives, had a hard time putting into words why they perceived it as such. It could be assumed that they found the overall experience of participating in the study fun and entertaining, considering that they used a VR headset to visualize and explore 3D virtual spaces while giving feedback in VR. The participants considered the concern of the study, what is the future of the zoological museum collection, cute. They considered the VR application as an honest attempt to utilizing 3D animals of the zoological museum in VR. In addition, they remarked that it was very interesting that they were in the same space physically and in VR. They also mentioned other ways to utilize a similar setup for their work or study activities, for example, virtual meetings or promoting. One such case remarked by a participant (ID

13) was using virtual learning environments to decrease performance anxiety. For instance, our use case could be employed to help people who have anxieties towards certain animals. VR therapy has been used to treat phobias [77, 78].

6. DISCUSSION

This thesis presents the development of two applications for user involvement. Firstly, a VR client was developed to engage participants in this process and collect meaningful feedback for the purpose of creating better services and products. The VR tool aims to improve the user involvement process by providing an alternate solution that utilizes VR technology. Secondly, an assistance web tool developed to help moderators of the online user involvement tool create content for its user clients. In addition, these applications focus on utilizing 3D model representations of existing spaces to put together immersive 3D virtual environments, which in turn produces some considerations.

The VR client gives end-users an interactive, immersive environment where they can explore and experience 3D virtual environments that represent spaces of interest and give valuable comments and feedback regarding a topic by responding to online surveys via 3D questions. Thanks to VR, end-users can participate in activities that take place in old, new or long-lost locations. For example, the actual zoological museum content, which was the theme of our user study, has been taken apart and put in a storage to make room for a new restaurant. Its content, however, could still be explored in VR. Thus, the use of VR offers an immersive medium that can improve the user involvement process. Our study shows that the VR solution is important to users because they can be involved in research and product development even when it is not possible to be in a certain location or when an environment does not exist, or it no longer exists.

6.1. Limitations, Challenges, Future Visions

The development of this thesis and its artefacts did not come without limitations and challenges. Firstly, the main focus was to develop a VR solution for an existing online user involvement platform. This meant that it should be integrated with this existing tool and not work just as a standalone application. Thus, more investigation and consideration were necessary to realize this requirement. The design and implementation sections explain how the VR client was integrated and used with the online user involvement tool. Communication was done through basic HTML5 Web Storage. In addition, a secondary application was soon required to help moderators create content for the VR client and other user clients. This tool was developed as a web application and communicates in the same manner with the parent user involvement tool. Secondly, because of the nature of these applications and their objective, it was required that they are developed to support different types of activities and their content dynamically. For example, the 3D survey questions implemented are not hard-coded, but they get their structure and content from an online survey platform instead, based on which activity is undertook. This consideration was kept in mind during development so that the solution is not made for one activity in particular, but for any activity that require a specific survey, a different 3D virtual environment, and differently configured POIs.

Target platform also played a big role. The VR client was initially developed to run as a standalone Windows application with VR support for Oculus Rift. Thus, the 3D virtual environments used ran quite well despite their size. Still, a powerful enough machine would be needed to run a VR application that utilizes such models.

Because one of the goals was to have this application easily accessible from users' homes, it was further developed for WebVR, an open specification for in-browser VR apps [79]. In that case, the application did not perform well, and additional performance optimization tasks were required, such as occlusion culling and lightmapping techniques. Even so, more graphics performance optimizations should be performed for a perfectly smooth experience.

Regarding evaluation, the sample size was relatively small. A larger sample size would have delivered better results. The study, however, tried to complement this fact by collecting rich data from the UX evaluation sessions. The VR client worked according to plan, from a technical perspective, throughout the evaluation process thanks to iterative and incremental development. The results are positive. During evaluation, we noticed that VR technology was still quite new to the participants and we suspect that the positive feedback received may be partly attributed to this novelty effect.

In the future, the applications could, or should, be further improved and evaluated to provide better user engagement and user involvement overall. Also, graphics performance should be further optimized to provide a smooth experience in WebVR. Other interaction methods could be investigated, seeing that although answering by gaze was considered quite easy and natural, results show that other methods may be preferred for user interaction.

7. CONCLUSION

This thesis discusses and presents two applications that were developed to bring new functionality to an existing user involvement service, and thus improve its user participation activities. These two applications, a user feedback collection VR client, and a moderator creation editor web tool, have been developed while exploring good approaches when it comes to user participation in VR and managing several clients through moderator tools. In addition, the applications were developed using 3D model representations of real places, using 3D game engines, and were built for the Oculus Rift and WebGL platforms. These aspects are also discussed as well as encountered challenges.

A map toolset, an online survey software, 3D user interfaces, and geolocated 3D points of interest have been implemented to facilitate immersion and functionality to these tools, and their design and implementation aspects are discussed.

Virtual reality aims to make users feel more immersed and thus fool their senses into overlooking that the virtual environment around them is not actually real, but computer-generated. This quality was considered and tried to emphasize while designing and developing this thesis' VR application. One of the reasons after all was to try to engage users differently using virtual reality and its capabilities. Thus, the implemented 3D environments and interactive 3D user interfaces try to maintain this quality.

When the applications were in good shape from a technical point of view, a user experience study was conducted. Participants used the VR application to answer survey questions and thus give feedback while experiencing and exploring a familiar space represented in 3D. During this study, user feedback was collected at the same time the users experienced the 3D virtual environment and its content. The goal was to firstly evaluate how interesting, useful and fun is to use the VR client in a user involvement context, but also to evaluate how well the system performs from a technical point of view considering the more challenging aspects of making the PATIO clients work together, and complement each other.

The results suggest that user participation in virtual reality, where users can immediately give feedback by answering geolocated 3D questions about activities taking place in 3D spaces is a relevant approach. The VR method was described as innovative, entertaining, and fun, amongst other things.

8. REFERENCES

[1] van Leeuwen, J. P., Hermans, K., Jylhä, A., Quanjer, A. J., & Nijman, H. (2018). Effectiveness of Virtual Reality in Participatory Urban Planning: A Case Study. In: Proceedings of the 4th Media Architecture Biennale Conference, 128-136, ACM.

[2] Kaasinen, E., Koskela-Huotari, K., Ikonen, V., & Niemeléi, M. (2013). Three approaches to co-creating services with users. Advances in the Human Side of Service Engineering, 286, 4343-4352.

[3] Biocca, F., & Delaney, B. (1995). Immersive virtual reality technology. Communication in the age of virtual reality, 15, 32.

[4] Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. IEICE TRANSACTIONS on Information and Systems, 77(12), 1321-1329.

[5] Milgram, P., Drascic, D., Grodski, J. J., Restogi, A., Zhai, S., & Zhou, C. (1995). Merging real and virtual worlds. In: Proceedings of IMAGINA, 95, 218-230.

[6] Feiner, S., Macintyre, B., & Seligmann, D. (1993). Knowledge-based augmented reality. Communications of the ACM, 36(7), 53-62.

[7] Sharples, S., Cobb, S., Moody, A., & Wilson, J. R. (2008). Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems. Displays, 29(2), 58-69.

[8] Santos, B. S., Dias, P., Pimentel, A., Baggerman, J. W., Ferreira, C., Silva, S., & Madeira, J. (2009). Head-mounted display versus desktop for 3D navigation in virtual reality: a user study. Multimedia Tools and Applications, 41(1), 161.

[9] LaValle, S. (2016). Virtual reality.

[10] Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. Presence: Teleoperators & Virtual Environments, 6(6), 603-616.

[11] Sanchez-Vives, M. V., & Slater, M. (2005). From presence to consciousness through virtual reality. Nature Reviews Neuroscience, 6(4), 332.

[12] Slater, M., Usoh, M., & Steed, A. (1994). Depth of presence in virtual environments. Presence: Teleoperators & Virtual Environments, 3(2), 130-144.

[13] Adams, E. (2004). Postmodernism and the three types of immersion. Gamasutra: The Art & Business of Making Games, 9.

[14] Bjork, S., & Holopainen, J. (2004). Patterns in game design (game development series).

[15] Alatalo, T., Pouke, M., Koskela, T., Hurskainen, T., Florea, C., & Ojala, T. (2017). Two real-world case studies on 3D web applications for participatory urban planning. In: Proceedings of the 22nd International Conference on 3D Web Technology, 11, ACM.

[16] Brooks, F. P. (1999). What's real about virtual reality?. IEEE Computer graphics and applications, 19(6), 16-27.

[17] Gotchev, A., Akar, G. B., Capin, T., Strohmeier, D., & Boev, A. (2011). Three-dimensional media for mobile devices. In: Proceedings of the IEEE, 99(4), 708-741.

[18] Kurtenbach, G., Fitzmaurice, G., Baudel, T., & Buxton, B. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. In: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, 35-42, ACM.

[19] Light, J., & Miller, J. D. (2002). Miramar: A 3D workplace. In: Proceedings of the IEEE International Professional Communication Conference, 271-282, IEEE.

[20] Patterson, D. (2007). 3D SPACE: using depth and movement for selection tasks. In: Proceedings of the twelfth international conference on 3D web technology, 147-155, ACM.

[21] Reifinger, S., Laquai, F., & Rigoll, G. (2008). Translation and rotation of virtual objects in Augmented Reality: A comparison of interaction devices. In 2008 IEEE International Conference on Systems, Man and Cybernetics, 2448-2453, IEEE.

[22] Vinson, N. G. (1999). Design guidelines for landmarks to support navigation in virtual environments. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, 278-285, ACM.

[23] Wang, S., Poturalski, M., & Vronay, D. (2005). Designing a generalized 3D carousel view. In CHI'05 Extended Abstracts on Human Factors in Computing Systems, 2017-2020, ACM.

[24] Carrozzino, M., & Bergamasco, M. (2010). Beyond virtual museums: Experiencing immersive virtual reality in real museums. Journal of Cultural Heritage, 11(4), 452-458.

[25] Moorhouse, N., tom Dieck, M. C., & Jung, T. (2018). Technological innovations transforming the consumer retail experience: a review of

literature. In Augmented Reality and Virtual Reality, 133-143, Springer, Cham.

[26] Kristensson, P., Matthing, J., & Johansson, N. (2008). Key strategies for the successful involvement of customers in the co-creation of new technology-based services. International journal of service industry management, 19(4), 474-491.

[27] Ind, N., & Coates, N. (2013). The meanings of co-creation. European Business Review, 25(1), 86-95.

[28] Piller, F. T., Ihl, C., & Vossen, A. (2010). A typology of customer co-creation in the innovation process. Available at SSRN 1732127.

[29] Prahalad, C. K., & Ramaswamy, V. (2004). Co-creation experiences: The next practice in value creation. Journal of interactive marketing, 18(3), 5-14.

[30] https://unity.com/solutions

[31] Greenwood, P., Sago, J., Richmond, S., & Chau, V. (2009). Using game engine technology to create real-time interactive environments to assist in planning and visual assessment for infrastructure. In 18th World IMACS/MODSIM Congress, 2229-2235.

[32] https://unity3d.com/

[33] https://www.unrealengine.com/

[34] Sadaf, A., Newby, T. J., & Ertmer, P. A. (2016). An investigation of the factors that influence preservice teachers' intentions and integration of Web 2.0 tools. Educational Technology Research and Development, 64(1), 37-64.

[35] Firpo, D., Kasemvilas, S., Ractham, P., & Zhang, X. (2009, April). Generating a sense of community in a graduate educational setting through persuasive technology. In: Proceedings of the 4th International Conference on Persuasive Technology, 41, ACM.

[36] Harris, A. L., & Rea, A. (2009). Web 2.0 and virtual world technologies: A growing impact on IS education. Journal of Information Systems Education, 20(2), 137.

[37] Florea C, Alavesa P, Arhippainen L, Pouke M, Huang W, Haukipuro L, Väinämö S, Niemelä A, Cortés Orduña M, Pakanen M, Ojala T (2019). Extending a User Involvement Tool with Virtual and Augmented Reality. In: Proceedings of the 26th IEEE Conference on Virtual Reality and 3D User Interfaces, IEEE.

[38] McNeese, M. D. (2004). How video informs cognitive systems engineering: making experience count. Cognition, Technology & Work, 6(3), 186-196.

[39] Pierson, J., & Lievens, B. (2005). Configuring living labs for a 'thick'understanding of innovation. In: Proceedings of the Ethnographic Praxis in Industry Conference, 2005(1), 114-127, Oxford, UK: Blackwell Publishing Ltd.

[40] https://docs.unity3d.com/Manual/CreatingScenes.html

[41] https://docs.unity3d.com/Manual/PublishingBuilds.html

[42] https://docs.unity3d.com/Manual/Prefabs.html

[43] https://docs.unity3d.com/Manual/Glossary.html#GameObject

[44] https://docs.unity3d.com/Manual/Glossary.html#component

[45] https://docs.unity3d.com/ScriptReference/MonoBehaviour.html

[46] https://www.w3schools.com/jsref/prop_win_sessionstorage.asp

[47] https://www.w3schools.com/html/html5_webstorage.asp

[48] https://docs.unity3d.com/Manual/webgl-interactingwithbrowserscripting.html

[49] https://docs.unity3d.com/Manual/UnityWebRequest-RetrievingTextBinaryData.html

[50] https://www.newtonsoft.com/json

[51] https://www.w3schools.com/js/js_json_intro.asp

[52] https://docs.unity3d.com/Manual/Lighting.html

[53] https://docs.unity3d.com/Manual/LightingInUnity.html

[54] https://docs.unity3d.com/Manual/Glossary.html#3DObject

[55] https://www.mapbox.com/

[56] https://docs.unity3d.com/Manual/class-MeshCollider.html

[57] https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html

[58] https://docs.unity3d.com/Manual/OcclusionCulling.html

[59] https://docs.unity3d.com/Manual/class-Camera.html

[60] https://docs.unity3d.com/Manual/Lightmapping.html

[61] https://docs.unity3d.com/Manual/ProgressiveLightmapper.html

[62] https://github.com/mozilla/unity-webvr-export

[63] https://hacks.mozilla.org/2018/02/create-vr-on-the-web-using-unity3d/

[64] https://assetstore.unity.com/packages/templates/systems/webvr-assets-109152

[65] https://unity3d.com/learn/tutorials/topics/virtual-reality/interaction-vr

[66] https://docs.microsoft.com/en-us/windows/mixed-reality/gaze

[67] https://docs.unity3d.com/Manual/class-RenderTexture.html

[68] https://en.wikipedia.org/wiki/Point_of_interest

[69] https://www.surveymonkey.com/

[70] https://developer.surveymonkey.com/

[71] https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr

[72] https://assetstore.unity.com/packages/tools/gui/curved-ui-vr-ready-solution-to-bend-warp-your-canvas-53258

[73] https://www.w3schools.com/js/js_window_history.asp

[74] https://docs.unity3d.com/Manual/webgl-gettingstarted.html

[75] Alavesa, P., Pakanen, M., Niemelä, A., Huang, W., Väinämö, S., Haukipuro, L., ... & Ojala, T. (2018). Mobile Augmented Reality Client as a UX Method for Living Lab's User Involvement Tool. In: Proceedings of the 22nd International Academic Mindtrek Conference, 135-142, ACM.

[76] Sunnari, M., Arhippainen, L., Pakanen, M., & Hickey, S. (2012). Studying user experiences of autostereoscopic 3D menu on touch screen mobile device. In: Proceedings of the 24th Australian Computer-Human Interaction Conference, 558-561, ACM.

[77] Harris, S. R., Kemmerling, R. L., & North, M. M. (2002). Brief virtual reality therapy for public speaking anxiety. Cyberpsychology & behavior, 5(6), 543-550.

[78] North, M., North, S., & Coble, J. (1998). Virtual reality therapy: an effective treatment for phobias in virtual environments. Clinical Psychology and Neuroscience, IOS Press, Amsterdam.

[79] https://webvr.info/