SoftwareX 10 (2019) 100269

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Workflow and tools to track and visualize behavioural data from a Virtual Reality environment using a lightweight GIS

José L. Soler-Domínguez[*], Manuel Contero, Mariano Alcañiz

Instituto de Investigación e Innovación en Bioingeniería (I3B),Universitat Politècnica de València, Camino de Vera s/n. 46022. Valencia, Spain

A R T I C L E I N F O

Article history:
Received 2 June 2018
Received in revised form 23 January 2019
Accepted 24 June 2019

Keywords:
Virtual reality
User experience
Geographic information system
Human–computer interaction

A B S T R A C T

Evaluating user behaviour in Virtual Reality is a challenge for every researcher involved in designing and executing experiments in immersive environments. Behavioural information could lead to relevant findings in presence, engagement or, for example, the mood of the player during a VR experience. Saving this kind of information and exploding it in an appropriate way could lead researchers or even game designers to identify relevant behavioural patterns or correlations. In this article, we are proposing a simple, replicable workflow and a set of scripted tools in order to acquire user's navigational data and visualize it using the inherent capabilities of a Geographic Information System. Our workflow goes from data acquisition in Unity3D with C# to the final representation in a map using Leaflet, an open-source GIS JavaScript based, passing through the pre-processing of XML files. Using a GIS to visualize navigational data is a flexible, ecological and effective solution that improves productivity and awareness on data storytelling capabilities.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_69
Legal Code License	CC0
Code versioning system used	none
Software code languages, tools, and services used	Javascript, C#,CSS
Compilation requirements, operating environments	Unity3D,Leaflet.js
If available Link to developer documentation/manual	
Support email for questions	jlsoler@florida-uni.es

1. Motivation and significance

Interest in virtual reality (VR) as a tool in different fields [1–4] has continuously raised over the last fifty years. Since its beginnings, lecturers, doctors, trainers, instructional designers, psychologist and several other professionals, have highlighted the VR potential to improve their activities, taking advantage of its immersion, engagement, measurement and feedback capabilities. Attending to this usage of Virtual Reality as a mean to achieve multidisciplinary goals, several assessing strategies have been developed. Most of them are focused on obtaining some metrics about users' feelings, emotions or perceptions. This group is usually post-experience survey based, with some inherent bias. Recently, this set of metrics it is being replaced by psychophysiological data in order to obtain fully objective measures [5]. Electromyography (EMG), electrodermal activity (EDA), electroencephalography (EEG), functional magnetic resonance imaging (fMRI) are the most used and validated psychophysiological properties and techniques. The other family of metrics is related to users' behaviour inside the virtual environment. VR researchers try to register every relevant interaction aiming to find out how is the user performing or, for example, how is he/she navigating the environment. Navigation is the most

[*] Corresponding author.
E-mail address: josodo@upv.es (J.L. Soler-Domínguez).

basic and ubiquitous interaction in every digital environment, immersive or non-immersive and it has strong implications in presence (the feeling of being actually inside the virtual world) [6] or cybersickness [7]. For this reason, researchers invest a significant amount of time designing and developing software solutions to acquire and exploit navigational data of users. Attending to this, we want to share our workflow and tools designed to capture navigational data of users in a virtual environment developed in a specific and popular game engine: Unity3D [8], the most versatile and spread game engine and to be visualized in an innovative way, using a Geographical Information System as Leaflet [9]. This features could save a lot of time to other researchers and also, could improve replicability of experiments focused on study user behaviour in virtual environments.

Furthermore, VR is becoming a collaborative space since technologies are supporting multi-user simultaneous interactions in virtual environments. These new capabilities allow users to be involved in a common task inside a virtual world [10]. One of the most challenging aspects of designing and developing Collaborative Virtual Environments (CVE) is to measure the awareness of being executing a task that is part of a collaborative work [11]. As with other parameters usually measured in VR environments (i.e. presence, co-presence, comfort), some questionnaires have been developed [12] but as that surveys are usually taken after the experience, they have significant bias. Our proposed workflow is inherently multi-user because each participant has its own tracker and could give light, both in real-time and post experience, about how users interact between them and with the environment.

2. Previous work

With the development of internet mapping, several GIS solutions were developed and immediately, the academic world highlighted the need to explore spatial data visualization in alternative ways. Since the beginning it was notorious the links between GIS and VR because of their complementarity: GIS deals natively with geographic data and VR has incredible capabilities of natural interaction. Firstly, this GIS-VR relationship was only established in one way: visualizing GIS data in VR [13–16].

After that, having demonstrated the transferability from VR experiences to the real world, several studies where developed where individuals had to identify some elements of their recent VR experience in a 2D representation of the environment [17, 18]. These studies, chosen as representative samples, asked the participants to write down in a paper map things that they saw in the HMD (Head Mounted Display) or decisions they made. This methodology is difficult and slow to process with high n (samples) values.

Some ad-hoc tools for the visualization of navigational data were created (i.e. [19]) but they were very limited in functionality, have high restrictions in re-usability and diverted research teams for their original research questions because of the effort needed to develop this kind of tools. In this paper we propose a replicable workflow and share some tools in order to make this analysis easier and with more potential possibilities of customization.

3. Software description

Our main contribution is the complete, tested, workflow description and two software tools represented by two scripts aiming, respectively, to acquire the data and to adapt it to be used by the GIS (see Fig. 1).

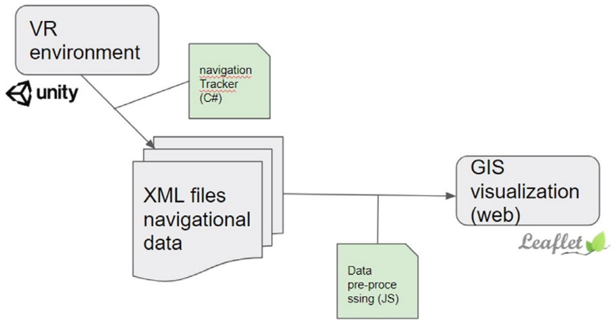


Fig. 1. Virtual Reality Environment diagram.

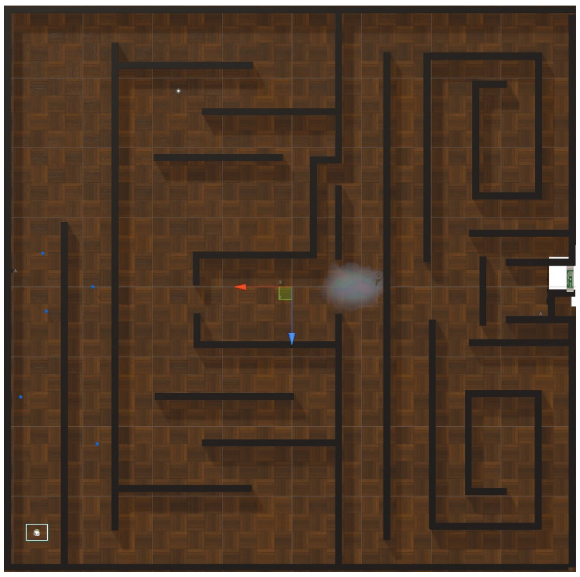


Fig. 2. Top view of the maze scene.

3.1. Software architecture

3.2. Software functionalities

3.2.1. Navigation tracker

This script, originally designed for HTC Vive Camera Rig is very simple and could be added to every Character Controller with minor modifications. It is based in two main events: StartMovement() and EndMovement(). Both functions get the position of user's avatar at the beginning and at the end of each movement. Also, we store the start and end time in order to calculate the time that the user is moving. With this info, is very easy to calculate the distance moved with each movement:

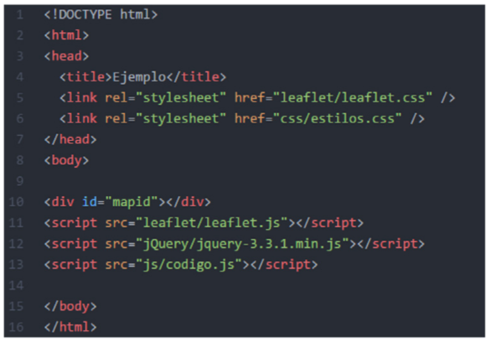
```
distanceMoved = Vector3.Distance(startPosition,
endPosition) ;
```

After each movement is ended, we can add it to our main data structure: the Movements collection, defined as a list.

```
public List<Movement> movements = new
List<Movement>() ;
```

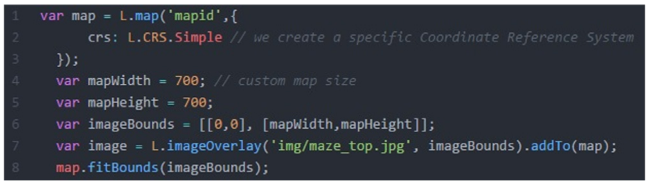
Their functionality is as follows:

- When StartMovement is detected, we add a new *movement* to the movements list.



```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo</title>
  <link rel="stylesheet" href="leaflet/leaflet.css" />
  <link rel="stylesheet" href="css/estilos.css" />
</head>
<body>

<div id="mapid"></div>
<script src="leaflet/leaflet.js"></script>
<script src="jQuery/jquery-3.3.1.min.js"></script>
<script src="js/codigo.js"></script>

</body>
</html>
```

Fig. 3. index.html.



```
var map = L.map('mapid',{
        crs: L.CRS.Simple // we create a specific Coordinate Reference System
  });
  var mapWidth = 700; // custom map size
  var mapHeight = 700;
  var imageBounds = [[0,0], [mapWidth,mapHeight]];
  var image = L.imageOverlay('img/maze_top.jpg', imageBounds).addTo(map);
  map.fitBounds(imageBounds);
```

Fig. 4. Map declaration and creation.

```
movements.Add(new Movement(position));
```

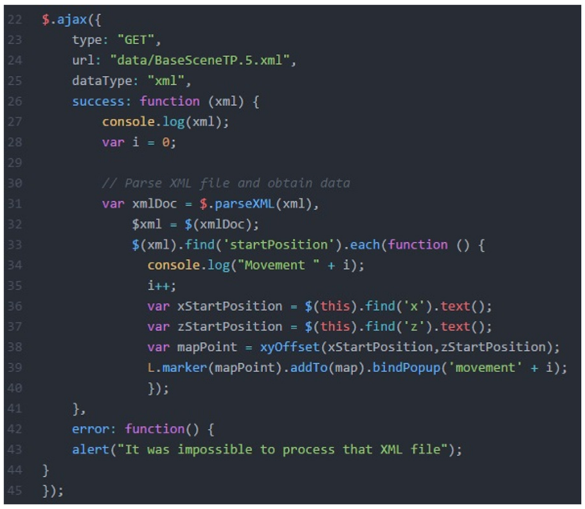
- When the EndMovement event is thrown, we update the list counter (*currentMovementIndex* and store the final position.

```
movements[currentMovementIndex].
EndMovement(position) ;
currentMovementIndex ++ ;
```

In order to make this approach work, you must custom your StartMovement() and EndMovement() events calling methods. For example, in this project we have two navigational metaphores: Pressing a touchpad and head bobbing (walking in place and detecting head vertical movements). For each navigation technique, we had to determine which is the action that started and ended the movement. For touchpad pressing it was obvious: when the user pressed the touchpad, the movement starts and when he/she releases it, that movement ended. For head bobbing it was far more complex because detecting a valid head bobbing movement deals with speed, angles and other parameters in order to discriminate regular head movements from those related to walking in place. But, at the end, we implemented a function HeadBobDetected() that was our StartMovement() flag and when we did not detect head bobbing, we launched the StopMovement() event.

The XML file generated with all this movements has this sub-structure (with sample data):

```
<movement>
  <startPosition>
    <x>0.8478197</x>
    <y>0.0833333358</y>
    <z>2.12983274</z>
  </startPosition>
  <endPosition>
    <x>0.854446</x>
    <y>0.0833333358</y>
    <z>2.07698369</z>
  </endPosition>
  <startTime>7.191995</startTime>
  <endTime>7.20538664</endTime>
  <totalTime>0.0133914948</totalTime>
  <distanceMoved>0.0532628447</distanceMoved>
</movement>
```



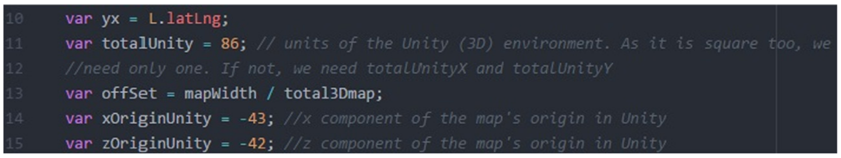
```
$.ajax({
    type: "GET",
    url: "data/BaseSceneTP.5.xml",
    dataType: "xml",
    success: function (xml) {
        console.log(xml);
        var i = 0;

        // Parse XML file and obtain data
        var xmlDoc = $.parseXML(xml),
            $xml = $(xmlDoc);
            $(xml).find('startPosition').each(function () {
              console.log("Movement " + i);
              i++;
              var xStartPosition = $(this).find('x').text();
              var zStartPosition = $(this).find('z').text();
              var mapPoint = xyOffset(xStartPosition,zStartPosition);
              L.marker(mapPoint).addTo(map).bindPopup('movement' + i);
              });
    },
    error: function() {
    alert("It was impossible to process that XML file");
}
});
```

Fig. 7. Parsing with AJAX from JQuery.

3.2.2. *Loading data to the GIS*

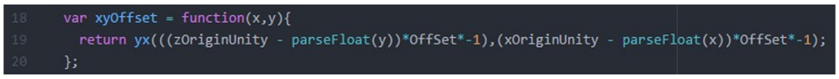
Aiming to represent all the data collected (usually, one XML per user) during the first stage of the workflow, we have to follow this steps:

1. **Create a 2D map of our 3D virtual environment.**



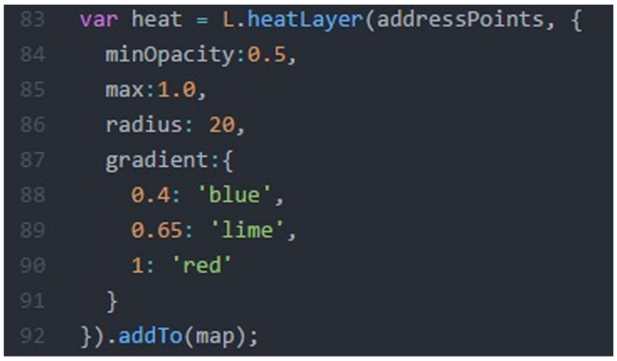
```
    var yx = L.latLng;
    var totalUnity = 86; // units of the Unity (3D) environment. As it is square too, we
    //need only one. If not, we need totalUnityX and totalUnityY
    var offSet = mapWidth / total3Dmap;
    var xOriginUnity = -43; //x component of the map's origin in Unity
    var zOriginUnity = -42; //z component of the map's origin in Unity
```

Fig. 5. Declaration of variables.



```
    var xyOffset = function(x,y){
      return yx(((zOriginUnity - parseFloat(y))*OffSet*-1),(xOriginUnity - parseFloat(x))*OffSet*-1);
    };
```

Fig. 6. Transforming function.



```
var heat = L.heatLayer(addressPoints, {
  minOpacity:0.5,
  max:1.0,
  radius: 20,
  gradient:{
    0.4: 'blue',
    0.65: 'lime',
    1: 'red'
  }
}).addTo(map);
```

Fig. 8. Declaration of heatLayer in code.js.

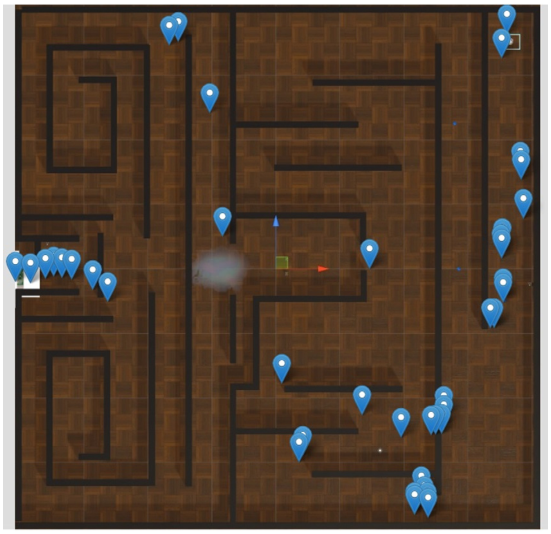


Fig. 9. Interactive map with movements as markers.

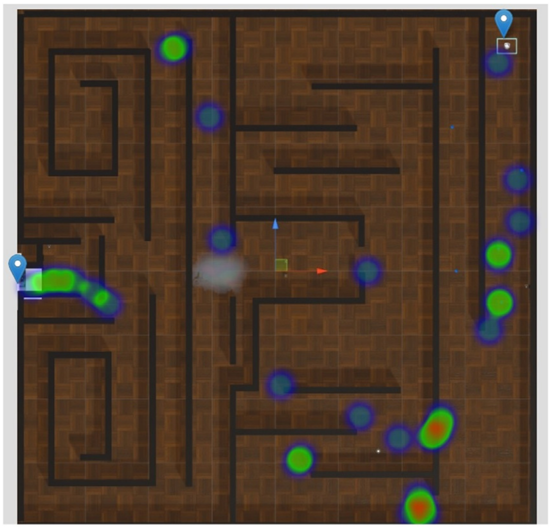


Fig. 10. Heatmap layer.

It has to respect proportions in order to make the translation easier. The fastest and easiest method is to capture a top view of the environment inside the game engine.
In Fig. 2 we can see an example from our project:

2. **Set up the Leaflet environment.**
Leaflet is a light-weight GIS consisting of the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 38 KB of JS, it has all the mapping features most developers ever need.
Leaflet is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms [9].
As Leaflet is JS native, it is intended to work in a web environment but you could set up a local instance that could be useful as a testing system. Always remember that XML files cannot be accessed locally by a javascript. If you want to configure this minimum environment you should install a local Tomcat server or similar in order to access your files from http.
The most recommendable configuration for this architecture is storing XML files in a remote server and serve them as a service to be called by your local host. In the same way, all this workflow could be implemented on JSON data, being JS native. We have chosen XML for its universality.
In our initial setup we have included:
   - **leaflet.cs**: main Leaflet library.
   - **jquery-3.3.1.min.js**: JQuery is a very well known, open-source, javascript library with lots of utilities to parse XML documents and to interact with large data sets.
3. **Create the interactive map, pre-process the XML files and add them to the map**
All this functionality is included in a single .js. We are going to explain it in detail.
As it is embedded in a web project, all the libraries have to be added to the main html file, *index.html* (Fig. 3).
Additionally, we have to create a master CSS file, which will contain the main map definition.

```
#mapid { height: 700px; }
```

After that, we should create our main js file, *code.js*. First of all, we have to declare and formalize the size of our map. In the same statement, we assign the 2D map we have previously created (Fig. 4).
We also have implemented a simple function to translate our notation for the coordinates of each point to the leaflet geographic style. The *offSet* variable represents the scale between our 2D map and the 3D environment (Fig. 5).
Again, if we had a non-squared map/environment, we should create horizontal and vertical offsets. XOriginUnity and zOriginUnity represent the coordinates where the (0,0) point of the 3D environment is placed. We do not use the Y axis because in our project, the user is not allowed to fly, climb or any action that could vary its Y position.
4. **Converting coordinates from 3D environment to 2D map**
At this point, we use a single function that converts text values parsed from the XML files to Leaflet Latitude/Longitude style coordinates. We use the *latLng* parameter to set this feature (Fig. 6).
With all this auxiliary tools working, we are able to parse our XML files (one by one or in bulk mode) using AJAX methods from jQuery and add all the points extracted from navigational behaviour of users to our GIS map. We have to search the coordinates useful for our representation (x,z) and, in the same loop, we can collect different data just searching by the name of the field (Fig. 7).

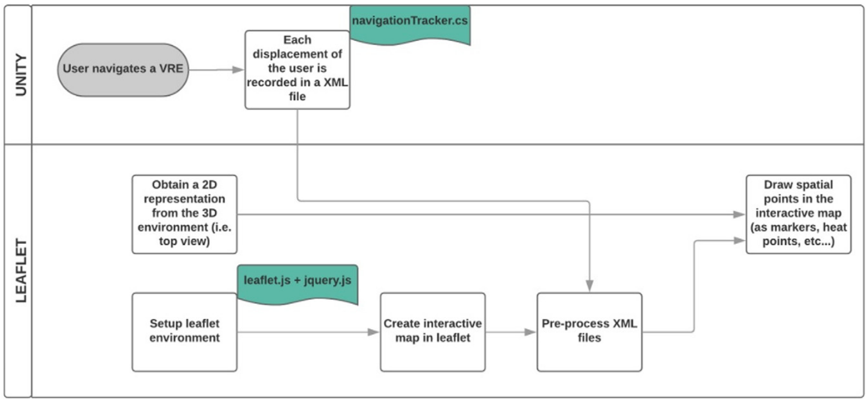

UNITY
LEAFLET
User navigates a VRE
Each displacement of the user is recorded in a XML file
navigationTracker.cs
Obtain a 2D representation from the 3D environment (i.e. top view)
Draw spatial points in the interactive map (as markers, heat points, etc...)
leaflet.js + jquery.js
Setup leaflet environment
Create interactive map in leaflet
Pre-process XML files

Fig. 11. Workflow explained.

4. Illustrative examples

After including all the movements into the map, visualizing them in different ways is really easy and fast: markers, routes, heatmaps... taking advantage of pre-built tools inside a GIS like Leaflet is simple and effort-effective.

We can easily create and interactive map like the one shown in Figs. 9 and 10.

In the same way, including one of the great number of available plugins for Leaflet, we can change our visualization mode. We are going to use the **leaflet-heat.js** [20] plugin in order to add a heatmap layer to our customized 2D map that will represent the density of points (movements) geographically distributed. We have also created an array, *addressPoints*, in order to store all the points before adding them to the heat Layer. These layers (markers, heatmap, etc...) could be drawn one over the others in order to create complex visualizations (Fig. 8).

In Fig. 11 the reader can have the big picture of the complete workflow.

5. Validation

We have tested this workflow and tools with 35 different sets of navigation data. In all 35 cases, the tools worked without any error or exception. In 5 of them, with high amounts of little displacements, the GIS marks were so close and this made their interpretation difficult. In those cases, re-sizing the marks solved the visibility problem. The metric we used was Spatial Accuracy (SA) based on the visual comparing of the original 3D path and the mapped 2D path. If both paths match, the variable 2D/3DMatch was set to 1 for this specific data set. If not, the variable was set to 0. If some divergences have to be studied, the variable should had to be set to 2. We obtained a value for SA $=$ 100%, since each individual data set was coded to 1, proving that the mapping function works appropriately. This validation procedure was executed by the authors of this paper.

6. Impact

This proposed workflow could improve significantly the processes of VR researchers making easy their data acquisition and analysis when we talk about visualising user behaviour which is one of the most recurrent interests for researchers.

Additionally, this replicable workflow and set tools gives an outstanding starting point for every researcher aiming to track and visualize behavioural data from participants in VR experiments using a flexible, powerful, customizable web lightweight GIS. Our workflow goes from data acquisition in Unity3D with C# to the final representation in a map using Leaflet, an open-source GIS JavaScript based, passing through the preprocessing of XML files. Using a GIS to visualize navigational data is an ecological and effective solution that improves productivity and awareness on data storytelling capabilities.

7. Conclusions and further work

The libraries and workflow presented in this paper are the tool set that may be used to track and to visually represent in a 2D interactive map the navigation data obtained from a VRE. Even when this tool set is mainly designed to represent navigation data, it provides an scalable, customizable, lightweight, ready-to-use solution that could be used to support VR behavioural analysis since could be modified to work with every single interaction. Its novelty relies on using a GIS for data visual representation, making easy and affordable translating spatial data from a 3D environment to a 2D map and analyse it in simple but powerful way. Further work could be done in order to automatize this workflow working only with Unity but, today, using an specialized third party GIS is the most productive and flexible architecture. Additionally, further studies have to be developed about how interaction data could inform other usual VR metrics like presence [21], co-presence [22], cognitive load [23] or affective states [24], in order to introduce new indicators in virtual environments. These metrics, far from biased questionnaires and complementary to complex to obtain physiological signals [5], could lead to objective assessment methods with high ecological validity.

Acknowledgments

This work was supported by the Spanish Ministry of Science, Innovation and Universities funded project ATHENEA- DPI2017-91537-EXP and by the European Commission project RHUMBO - H2020-MSCA-ITN- 2018-813234.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.softx.2019.100269.

References

[1] Psotka J. Immersive training systems: Virtual reality and education and training. Instruct Sci 1995;23(5):405–31. http://dx.doi.org/10.1007/BF00896880.
[2] Riva G. Virtual reality: an experiential tool for clinical psychology. British J Guid Counselling 2009;37(3):337–45.

[3] McCloy R, Stone R. Science, medicine, and the future: Virtual reality in surgery. BMJ: British Med J 2001;323(7318):912.

[4] Botella C, García Palacios A, Baños Rivera RM, Quero Castellano S, Bretón-López J. Virtual reality in the treatment of pain.

[5] Kivikangas JM, Chanel G, Cowley B, Ekman I, Salminen M, Järvelä S, Ravaja N. A review of the use of psychophysiological methods in game research. J Gaming Virtual Worlds 2011;3(3):181–99.

[6] Slater M, Usoh M, Steed A. Taking steps: the influence of a walking technique on presence in virtual reality. ACM Trans Comput Human Interact 1995;2(3):201–19.

[7] Rebenitsch L, Owen C. Review on cybersickness in applications and visual displays. Virtual Real 2016;20(2):101–25.

[8] Technologies U. Unity3d. 2018. URL https://unity3d.com/.

[9] Agafonkin V. Leaflet. 2014. URL https://leafletjs.com/.

[10] Sarmiento WJ, Collazos CA. Common-awareness artifacts. In: International conference on human-computer interaction. Springer; 2016, p. 376–81.

[11] Churchill EF, Snowdon D. Collaborative virtual environments: an introductory review of issues and systems. Virtual Real 1998;3(1):3–15.

[12] Nguyen TTH, Duval T. A survey of communication and awareness in collaborative virtual environments. In: Collaborative virtual environments (3DCVE), 2014 international workshop on. IEEE; 2014, p. 1–8.

[13] Huang B, Lin H. Geovr: a web-based tool for virtual reality presentation from 2d gis data. Comput Geosci 1999;25(10):1167–75.

[14] Rhyne TM. Going virtual with geographic information and scientific visualization. Comput Geosci 1997;23(4):489–91.

[15] Verbree E, Maren GV, Germs R, Jansen F, Kraak M-J. Interaction in virtual world views-linking 3d gis with vr. Int J Geogr Inf Sci 1999;13(4):385–96.

[16] Faust NL. The virtual reality of gis. Environ Plan B: Plann Des 1995;22(3):257–68.

[17] Bowman DA, Koller D, Hodges LF. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In: Virtual reality annual international symposium, IEEE. IEEE; 1997, p. 45–52.

[18] Cushman LA, Stein K, Duffy CJ. Detecting navigational deficits in cognitive aging and alzheimer disease using virtual reality. Neurology 2008;71(12):888–95.

[19] Gillner S, Mallot HA. Navigation and acquisition of spatial knowledge in a virtual maze. J Cogn Neurosci 1998;10(4):445–63.

[20] Agafonkin V. Leaflet.heat. 2014. URL https://github.com/Leaflet/Leaflet.heat.

[21] Sanchez-Vives MV, Slater M. From presence to consciousness through virtual reality. Nat Rev Neurosci 2005;6(4):332.

[22] Nowak KL, Biocca F. The effect of the agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments,. Presence: Teleoperators Virtual Environ 2003;12(5):481–94.

[23] Slater M, Wilbur S. A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. Presence: Teleoperators Virtual Environ 1997;6(6):603–16.

[24] Riva G, Mantovani F, Capideville CS, Preziosa A, Morganti F, Villani D, Gaggioli A, Botella C, Alcañiz M. Affective interactions using virtual reality: the link between presence and emotions. CyberPsychol Behav 2007;10(1):45–56.