Low-Cost Virtual Reality Environment For Engineering And Construction

T. Hilfert and M. König

Chair of Computing in Engineering, Ruhr-Universität Bochum, Germany
E-mail: thomas.hilfert@rub.de, koenig@inf.bi.rub.de

ABSTRACT

Presenting significant building or engineering 3D-models is a crucial part of the planning, construction and maintenance phases in terms of collaboration and understanding. Especially in complex or large-scale models, immersion is one of the major key factors for being able to intuitively perceive all aspects of the scene.

A fully immersive system needs to give the user a large field-of-view (FOV) with reduced latency for lifelike impression. Technologies such as VR-walls and shutter glasses can deliver high refresh rates, yet fail to give a large FOV. Head-mounted-devices (HMD) for virtual reality (VR) fill this gap. Head tracking mechanisms translate movements of the user's head into virtual camera movements and enable a natural way of examining models. In contrast to a stereoscopic representation with projectors, point-of-view tracking can be achieved separately for each individual HMD user. Hardware costs for such systems were very high in the past, but have dropped due to virtual reality systems now gaining traction in the mainstream gaming community.

In this paper we present a way to build a low-cost, highly immersive virtual reality environment for engineering and construction applications. Using the Oculus Rift HMD and the Leap Motion hand-tracking device, we show the possibilities of naturally interacting within a virtual space in different use cases. The software, based on the popular game engine Unreal Engine 4 (UE4), will be used as a basis for further research and development.

Keywords –

Virtual Reality, Visualization, Construction, Engineering, Head-mounted devices

1 Introduction

Head mounted devices (HMD) for Virtual Reality (VR) are currently experiencing a renaissance. In the past, these systems were only accessible for large companies at high costs (starting at several thousand Euros) and using specialized systems. Even then, the user experience was under par. Devices were lacking the refresh rate needed to smoothly translate head movements in the virtual world and had inadequate resolution and a low field of view for realistic impressions. Sufficient refresh rates of displays and updates in the virtual world are key factors to consider when using VR, as the user will be otherwise likely prone to motion sickness on longer exposure.

In contrast to traditional surface based stereoscopic methods, such as 3D shutter glass monitors or beamers, HMDs enable individual rendering of the users perspective. Movements of the head and / or body are being translated into movements of the virtual avatar. The camera location and rotation will then be modified to the matching position. These techniques are similar to the field of Augmented Reality (AR), omitting the blending with the real world. AR will become more important in the future, but lacks the fully free designed environment without the boundaries of reality VR has.

While a designer or engineer has learned to use his imagination during the design phase to visualize the final product, this is difficult for outsiders. Being able to present 3D models and scenes to a wide range of audience is therefore beneficial in engineering and construction. We chose to use a game engine as basis of our research, as they are trimmed to maximum performance at a high level of detail. Also, in contrast to using a pure graphics engine, sound playback, physics and networking abilities are already present. Having game developers targeting the engine for plugin development is also a plus, as ongoing feature extensions and bug fixes will be available.

In this paper we present a solution for building a VR environment, consisting of different hardware and software components to achieve a deep level of immersion to users. We show applications for the environment via three different, common use cases. Additionally, all of the components mentioned are cheap to license or buy, enabling even non-professionals access to such simulations. Every part can be used independently, but play well together on maximizing the immersion experience and interaction.

2 Related research

In [1] the application of VR to visualize the construction of a bridge using two different construction methods is described. The target of this implementation was to give students a deeper understanding about how bridges are built, as they can't have the same level of insight on a real construction site due to safety reasons. They target mainly the creation and deployment of 3D models over traditional teaching practice (verbal description / pictures / diagrams) and come to the conclusion that the interaction is one main benefit of using VR. However, they don't elaborate on proper input methods for natural interaction with the system and the high level of immersion a HMD would give.

Grabowski and Jankowski [2] are testing VR training for coal miners using different HMD hardware setups in conjunction with joystick and VR glove input methods. They found out that the test subjects preferred high immersive VR, but the FOV was negligible. This may have been the reason, because of the hardware used for the 110° FOV is known to be prone to ghosting (artefacts from previously rendered frames still partly visible, causing motion sickness) and this may be diminishing the positive effects of having a high FOV. Nevertheless, the study showed that the use of a vision based system for detecting natural hand movements is better than wireless 3D joysticks and that the result of the training "[…] is maintained in the long term." ([2], p. 321).

Rüppel and Schatz [3] describe the creation of a serious game environment for evacuation simulation in case of a fire. They harness Building Information Modelling (BIM) data as a base for their building-modelling concept and describe the advantages of having material data included to simulate the structural damage. The VR-lab concept described uses ideally most human senses (visual, tactile, auditory, olfactory) and enables interaction with the scene. However, their proposal references expensive components for the environment, in means of VR-lab facilities. Using the low-cost equipment described in this paper may benefit to create an easier deployable and affordable setup at minor drawbacks (olfactory, tactile), but with more immersive visual representations. Surround headsets and / or binaural recordings can replace the audio installation described in their paper.

The authors in [4] explore different publications on virtual reality-based learning with regard to the outcomes in education. During the assessment they found out, that there are many examples of VR being beneficial for learning and "virtual reality-based instruction is an effective means of enhance learning outcomes" ([4], p. 37). As stated, one reason for not having widespread VR in education is financial feasibility. Also, VR environments based on desktop 3D computers do not provide fully immersive experiences, but enhance the learners' engagement ([4], p. 30). We conclude that having a low-cost VR environment at disposal, which works on a normal desktop level, may be favourable for both the Architecture, Engineering and Construction (AEC) industry, as well as for education outside of this field. There may be some applications where regular serious games on a computer monitor may be better to use, as mostly everyone can use a mouse or keyboard to operate traditional user interfaces (e.g., entering numeric values). However, operations, such as pushing buttons or levers, may manifest more naturally in a students' memory if real world movements are usable and being tracked into VR.

In [5] this statement is supported, in saying that "body movement can enhance navigation performance and experience" ([5], p. 43), but the ability of tracking parts of the body needs usually expensive equipment. The authors describe a system, using an Xbox Kinect sensor for navigating planned urban environments. By using body postures (leaning back and forth, turning the shoulders) participants are able to navigate through a VR-model of a city. They were able to perceive distances and scales inside the environment better, when using body movements. As translating postures to VR navigation is a first step, an optimal solution would be to map walking movements directly with dedicated hardware, if possible.

All studies show that there is demand for VR in many fields. It may have not been considered testing the actual deployment in some due to financial concerns of such a system. Giving a low-cost flexible environment, that delivers safe testing and natural interactions is desirable. Especially in fields where hazards limit trial and error testing, VR can gain a foothold.

3 Concept

The goal of our approach is to unify an environment for different use cases in engineering and construction. As BIM is being introduced throughout the construction industry, support for Industry Foundation Classes (IFC) is beneficial. The IFC enable data exchange between a wide range of software applications and design tools. IFC models consist of optional geometry information, attached metadata, such as materials used and product structure. By retaining this metadata information within the virtual environment (VE), interaction with the elements enables more profound experiences. If a user wants to, say, check which material a certain wall is composed of, this is seamlessly possible without breaking immersion.

A VE consists of a visual output element and one or multiple input elements to enable interaction with the system. Creating natural input methods are a hurdle at building such a system. They are either camera-based rigs with multiple cameras using visual detection algorithms, handheld devices or specialized depth sensors. Tracking the hands in relation to the virtual body position seems to be the most promising way to manipulate objects in the virtual world, as the learning curve for new users is modest and tracking reliability is not limited by hidden body parts due to an unfavourable camera position.

Ideally, off-the-shelf equipment should be usable for the whole system, as this will lower the total cost and increase deployment options.

3.1 Hardware Setup

Several different display technologies are possible to immerse users into a virtual world. In contrast to beamer supported solutions, namely CAVE (Cave Automatic Virtual Environment) or similar shutter / polarized 3D walls, HMDs allow for full first person immersion. Therefore we will only further elaborate only these kinds of display devices. For our application, the currently most widespread, low cost single device will be used.

3.1.1 Oculus Rift

Recent innovations in display technology enable us nowadays to have low cost HMDs by using display panels intended for mobile phones. The most prominent device is the Rift from Oculus (commonly referred to as "Oculus Rift") [6]. It uses a single mobile phone display, derived from the Samsung Galaxy Note 3 device (1920x1080 pixels, overclocked at a 75 Hz refresh rate). While older HMDs used independent displays, the picture displayed on a Rift screen is divided into two different parts and then separated for each eye by using lenses, providing an effective resolution of 960x1080 pixels per eye with 100° nominal field of view. An internal gyroscope, an accelerometer and a magnetometer are polled at 1000 Hz for detecting head movements and rotations in all three dimensions of space. Connection to the host system is realized with a regular HDMI connector and USB ports.

The current status of the Rift is the Development Kit 2 (Figure 1, a), which first introduced positional tracking, by employing infrared LEDs. An additional camera, mounted in front of the user, films these blinking emitters enabling the detection of positional changes of the head and body at 60 Hz. Also, ghosting effects have been reduced due to a technique called "low persistence", setting the screen to black (pixels off) between two rendered frames.

3.1.2 Leap Motion

The Leap Motion controller [7] is a hand-input device, which uses two cameras and infrared LEDs to capture stereoscopic images of the user's hands. With calibrated camera positions and proprietary software algorithms it is able to calculate the finger, hand and wrist positions. It is attachable to the Rift via a separately sold VR mount (Figure 1, b).

Tracking performance is dependent on lighting and environment. False positive detections of hands are possible when the distance is comparable to other nearby objects, such as monitors or keyboards. Under optimal tracking conditions it is possible to fluidly convert detected joint positions to a model in virtual space.

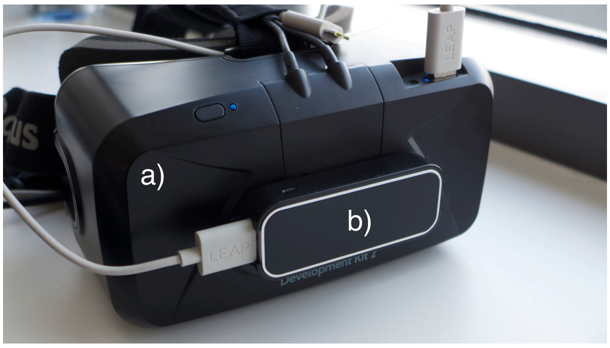

Figure 1: Oculus Rift (a) with mounted Leap Motion (b)

3.1.3 Additional Hardware Inputs

Using the Leap Motion, only hand (and tool) tracking is possible. Tracking the users legs may be beneficial for further immersion if he is looking down in the VE. Also, error correction for misalignment of the body tracking points and the virtual world may be countered by using multiple input sensors.

For obtaining whole body positions, we've been working on integrating Microsoft's new Kinect 2.0 [8] sensor. This also allows for spatial tracking inside a medium range and let's the users move around. As the Rift is not wireless, HDMI- and USB-extenders are needed to allow for freedom of movement. Larger location changes are only supported by controller inputs (e.g., by an analog stick of a game controller) or by gesture detections (e.g., flat palms up and fingers extended = move in this direction) in this approach.

Research and development on using only Inertial Measurement Unit (IMU) sensors for relative input is also done. One promising product to feature these capabilities in conjunction with low latency is "Control VR" [9], which lately went out of pre-order phase. Tracking sensors on the hand, down to the finger level, the lower and upper arm and chest are measuring movements and rotation. The system uses a calibration gesture to reset all stored positions to zero, by holding the arms straight down to the sides of the body. Relative changes to the individual sensors' position can then be measured and translated as VR input.

As an alternative to giving the user movement ability in a limited real world space, recent "treadmill" systems for consumers have been presented. They work by fixing the user on a stationary point and reducing friction on the feet. One widely known product using this technique is the Virtuix Omni [10]. Special shoes limit sideways slipping, but enable forward sliding movements, while sensors track the movements and convert them to regular game controller signals. This allows for a high compatibility with most 3D-engines and platforms.

3.2 Software Engine

While the Oculus Rift runtime can be attached to many graphics engine implementations, we decided to use an existing game engine at our disposal. Game engines, used to power computer games, are targeted at the highest performance (measured in frames per second) while providing the most possible realistic user experience, in contrast to scientific visualization. Rift support is enabled in the Unity game engine [11] and the Unreal Engine 4 (UE4) [12] through a plugin-based architecture. Unity is more mature, but the Unreal Engine allows for more detailed graphics and is completely open-source for subscribers. It also features a complete network stack for communication between servers and clients, usually intended for multiplayer gaming.

The UE4 comes included with an editor for creating and managing scenes (levels), materials, characters and animations, to name a few. The editor and engine are modifiable by plugins, written in C++, so that a developer can extend the functionality if needed.

Programming in UE4 can also be done via Blueprints, which are connectable event driven structured logical building blocks (cf. Figure 2). This enables modification of the application behaviour to be carried out even by non-C++ programmers.

Blueprints support context sensitive assists, suggesting several alternatives when drawing a connection from one component to another. Entities inside a scene can be referenced inside the Blueprint graph to change their attributes and they generate events that a programmer can subscribe to.

As the normal input methods are not in view to the user when wearing the Rift, menus should be integrated into the VE. Unlike traditional menu placement, which is overlaying the 3D view, menus in VE need to be placed in the rendered scene itself. A recent release from Leap Motion covers this possibility by attaching a menu on the user's virtual arm.

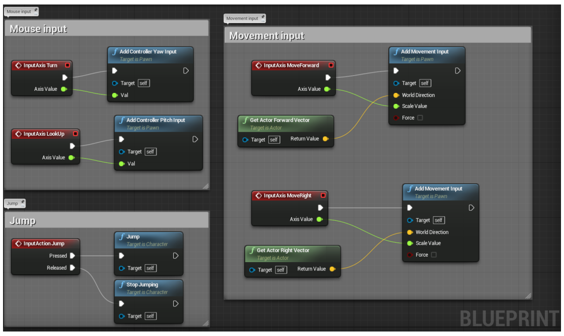

Figure 2: Example of blueprint logic modelling in UE4

Figure 3 shows the implementation inside a planetarium demo application. Users are able to change different settings of the software while keeping them immersed. Rotation of the wrist changes the displayed submenus and interaction with the second hand changes the displayed settings.

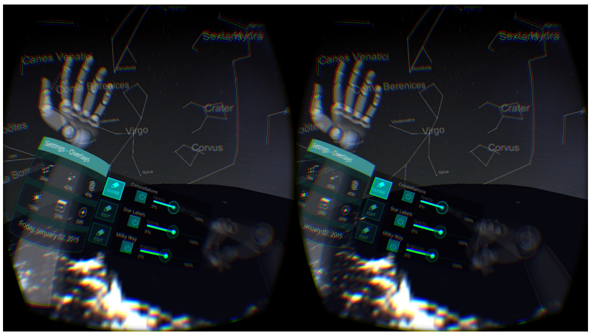
Figure 3: Example of a menu inside a VE. Source: [13]

The presented widget in [13] can be reproduced in multiple software engines, using the input provided by the Leap Motion controller. Alternatively, mapping controls to objects on surfaces inside the VE is also possible. This should be done when triggered actions result in modification of the VE's objects, mimicking a real world control circuit, such as a lift control.

UE4 allows also for very realistic rendering, supporting up-to-date graphics.

3.2.1 Application for architecture

UE4 uses global illumination (GI) for calculating static lighting in a level with an engine called Lightmass. Calculation results will be stored into a special texture channel during compilation time, taking load from the real time rending and resulting in very detailed light calculations for a scene.

Figure 4 shows a recent published engine example by Benoît Dereau [14], which demonstrates photorealistic rendering by Lightmass usable for architectural design. We were able to run this demo fluidly on an NVidia GeForce GTX980 with no noticeable lags at all. In the past, such quality was only possible with pre-rendered videos or stills. If a client wanted to view the scene differently, another offline rendering run would have been needed to produce additional video files. Using UE4 to navigate the scene allows interactive viewing and presentation of architectural designs.

Figure 4: "Paris" Demo by Benoît Dereau [14]

Pairing a scene like this with the Oculus Rift is possible, but it needs to be considered that the amount of render targets doubles. Each eye has to be rendered from it's own point of view, giving a moderate performance hit on slower machines. Networking support in the engine is helpful, if multiple users want to explore the space at the same time.

4 Implementation

For our development process, the Oculus Rift and Leap Motion are used as a base system to build upon to. UE4 will be used as the central point for simulating VEs. We've started developing an IFC plugin for the Unreal Engine that enables users to load existing IFC files into the Unreal Engine Editor. Attached metadata will be available to the user, such as the element ID of an IFC element. Geometry data will be passed during loading and available for placement inside the editor. Extending the editor interface is easy, as source code examples are readily available. To interface with existing IFC data, the ifcopenshell [15] (read the "IFC open shell") project is used.

Using the Oculus Rift with UE4 is simple due to the engine having a plugin available. This will be loaded automatically, as soon as the user has a Rift connected. Visual corrections for chromatic abbreviation and distortion caused by the lenses and input mapping to a virtual camera works out of the box.

For communication with the Leap Motion, an external plugin is needed. We found the event driven Leap Motion plugin [16] to be simple to setup and reliable to use. Convenience content, such as already rigged character models, is available and is tuned to usage with the Rift. We had to modify the physical collision shapes in order to enable full ten-finger support for interactions. Unreal usually uses only a capsule shaped collision model on characters to check for interactions with enabled actors in the environment. Therefore, each finger needs a tuned capsule shape for the physics system (Figure 5).

Grabbing objects in 3D space can't be achieved with regular collision modelling, as the object would bounce back and forth between fingers. Gestures or special finger combinations can be used to attach objects to certain slots on the user's character. A Leap Motion gesture could be moving certain parts of the hand in a circle, swiping in a direction or pressing a finger forward / downward. Gesture based input can be modelled with Blueprints only, giving non-programmers a possibility to extend the logic.

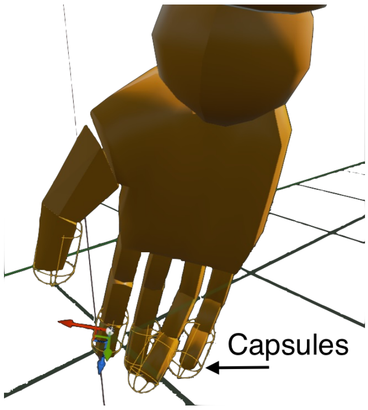

Figure 5: Capsule collision shapes

5 Scenarios

In the following segment we present some possible applications of virtual reality in conjunction with the Unreal Engine and the aforementioned VR hardware.

5.1 Evacuation Testing

Behaviour in case of an emergency differs from person to person. While legal requirements to provide emergency escape routes in construction are always fulfilled, they may be not necessarily be the optimal solution for a safe escape. With immersive VR environments it is possible to test multiple escape scenarios safely and realistically with many types of users. The usage of first person view using an HMD and integrated sound effects enable a more realistic impression to the user than otherwise possible. Test operators can monitor the subjective perception of escape route signs and set up the scene accordingly for another run. Figure 6 shows an example of escape sign placements and visibility to the user in case of a fire outbreak. The fire objects can be placed throughout the scene and even be spawned by the engine's logic using Blueprints.

Event scripting, such as the user triggering another fire sequence upon reaching a certain position within the environment, is also possible. This technique can also be used to measure the time needed to exit the building safely. When setting the character's movement speed to that of a running or fast walking person, the measured time should be similar to that of a real world scenario.

Figure 6: Immersive testing of escape route visibility in UE4

The lighting calculations of the UE4 in combination with sound effects lead to a very realistic scene. Ideally, this testing environment can be implemented using existing IFC models and the conversion plugin. Additionally, the UE4 interfaces natively with the FBX file format. Therefore, using Autodesk Revit, which is a software solution for creating and modifying BIM models, as a direct source of geometry is also possible (cf. [3]). Additions to the scene are always needed in this case, as escape signs and fire outbreak locations are not automatically generated, depending on furniture and materials.

5.2 Expert Training

Training to control special and / or heavy machinery is a key qualification in multiple professions. However, at the beginning of such training, accidents may happen due to major mistakes. Costs of using real machinery are also not negligible. While hands-on experience is the most important part of education, certain parts can be accelerated if the user knows the control scheme beforehand. Figure 7 shows the stereoscopic image of the user's view with the Oculus Rift and hand detection using the Leap Motion.

Collisions of hands and controls inside the cockpit can be detected and translated to scripted events for controlling movement of the machinery. The user gets a first person impression of environmental visibility around him. This scenario requires more scripting work when applied inside the UE4, as control input and resulting actions have to be modelled. However, reusing of Blueprint components is possible, resulting in a more rapid development enabling further test cases.

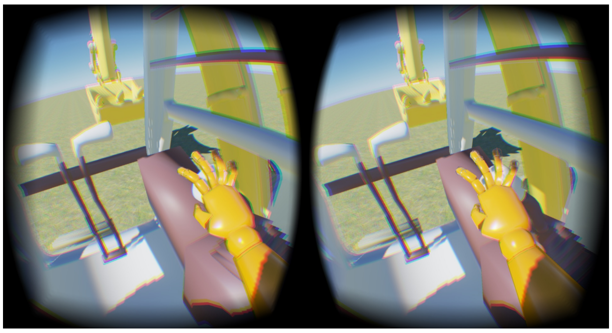

Figure 7: Example first person view of hand interactions with Oculus Rift and Leap Motion

Additional use cases include safety considerations when operating cranes at a construction site. The planned construction site layout can be used to model the environment. A user would take place in the cockpit of a crane, giving him the impression of limited visibility and movement range of the actual machine. Attaching objects to prepared slots on the crane supports the lifting and dropping of objects. Even the basic simulation of waypoint navigating non-player characters (NPCs) as workers on the site is possible. If a load hits any modelled object in the level, physics simulation would lead to a swinging motion. NPCs can be equipped with a defined number of hit points (health), which will drop if exposed to certain forces. A message to the user would then be presented to notify of the error and restart the scenario, again a great improvement over reality.

5.3 Accessibility validation

Accessibility planning for sidewalks or buildings needs to be accurate and mistakes at this stage may lead to expensive adjustments later on. First person testing, for example of wheel chair accessibility, is possible with the proposed VR environment.

The user can be placed into a wheel chair model and will see the designed level around him from a handicapped perspective (Figure 8).

Using the additional hand detection and resulting arm placement enables a check for unreachable controls, such as light switches or fire alarm buttons. The physics engine prevents entering rooms that are too small to drive into.

The UE4's physics engine allows for the definition of wheeled character blueprints. While these are usually intended for simulating cars, they can also be adapted to the wheelchair. Each tire of the vehicle supports friction settings and the total centre of mass will be taken into consideration by the physics engine. Acceleration and deceleration parameters are also supported.

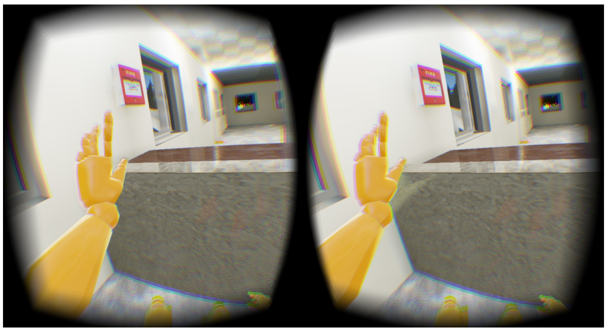

Figure 8: Simulated wheelchair inside a building

6 Conclusion / Outlook

HMDs today are getting more useful for a wide range of applications in construction and engineering, while costing less than in the past. Modern game engines, such as the UE4, enable even non-programmers to generate logic procedures and levels for presentation. VEs enable the users to experience complex models or control schemes instead of having to comprehend a complex explanation or offline rendered 2D/3D images.

However, initial development for such a system is still needed and no “out of the box” solutions exist. Pairing all the components mentioned in this paper gives users a high level of immersion and interaction into the virtual world, due to mapping hand gestures and movements. Complex and high level architectural scenes, such as seen in figure 4, still require a capable 3D artist for content creation or a developer experienced with the game engine. As immersion inside a scene does not solely depend on the realism of graphics, professional developers without deep understanding of modelling can also do fast creation of environments. Several free 3D model sites are available on the Internet, which can be used as a source of models for a scene.

Full body joint detection can be implemented using Microsoft’s Kinect or similar hardware. Users are then able to move every part of their body and see the results in the VE. Using IFC models for importing geometry and product data is possible and beneficial. We will continue development on this matter and further try to integrate new technologies to gain the maximum immersion possible for the users.

Table 1 shows the starting costs of each individual component. Costs for obtaining the starting package (Rift, Leap Motion and mount) are around 413€ (without considering the computer system) as of Feb. 2015. Additionally, $19 per month has to be paid for continuous access to the UE4 source code, but it is free for academic use. If the Unreal Engine is used with this licensing model, then 5% of gross revenue per product per year has to be paid to the creators of the UE4. However, this only applies to product sales after $3000 is exceeded per product per year.

Table 1 Costs of components as of February 2015 (w/o shipping)

Name	Category	Price
Rift DK2	HMD	$350
Leap Motion	Hand detection	89.00€
Leap Mount for Rift	Accessory	14.99€
Microsoft Kinect v2	Body / hand tracking	199,98€ (sensor + adapter)
Control VR	Body / hand tracking	$600 (two arm package)
Virtuix Omni	“Treadmill”	$699
Unreal Engine 4	Game Engine	$19/mo.

We have shown different use cases for utilizing a VE, namely evacuation plan testing, expert training and accessibility validation of environments. The supplied scenarios are only a small selection of what can be done using a VE for construction and engineering.

Hand detection using the Leap Motion is good, but tends to give false positives when not having free space in front of the user. Also, misdetection of the left and right hand are possible. This is an issue that is currently being worked on by the manufacturer and developers.

Regarding the fire example, we propose to further elaborate the scripting qualities of the UE4. Depending on material properties, a fire spread rate could be calculated, minimizing the set up time for the test operators and giving a more realistic environment.

Also, we are looking forward to future HMDs that may include eye-tracking mechanisms. Rendering from two different point-of-views is computationally expensive, even more, when considering increasing resolutions for HMD displays in the coming years. Eye-tracking can limit this impact, by only calculating high-resolution parts of the VE where the user is looking at, and giving more approximate representations at the peripheral vision.

7 References

[1] Sampaio, A. and Martins, O. The application of virtual reality technology in the construction of bridge: The cantilever and incremental launching methods. *Automation in Construction*, 37:58-67, 2013.

[2] Grabowski, A. and Jankowski, J. Virtual Reality-based pilot training for underground coal miners. *Safety Science,* 72:310-314, 2015

[3] Rüppel, U. and Schatz, K. Designing a BIM-based serious game for fire safety evacuation simulations. *Advanced Engineering Informatics,* 25:600-611, 2011

[4] Merchant, Z., Goetz, E., Cifuentes L., Keeney-Kennicutt, W. and Davis, T. Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A meta-analysis. *Computers & Education,* 70:29-40, 2014

[5] Roupé, M., Bosch-Sijtsema, P. and Johansson, M. Interactive navigation interface for Virtual Reality using the human body. *Computers, Environment and Urban Systems*, 43:42-50, 2014

[6] Oculus, Oculus Rift. Online: https://www.oculus.com, Accessed: 30.01.2015

[7] Leap Motion, Leap Motion. Online: https://www.leapmotion.com, Accessed: 30.01.2015

[8] Microsoft, Kinect SDK v2. Online: http://www.microsoft.com/en-us/kinectforwindows/develop, Accessed: 30.01.2015

[9] Control VR, Control VR. Online: http://controlvr.com, Accessed: 30.01.2015

[10] Virtuix, Virtuix Omni. Online: http://www.virtuix.com/products, Accessed: 30.01.2015

[11] Unity Technologies, Unity Game-Engine. Online: http://unity3d.com, Accessed: 30.01.2015

[12] Epic Games, Unreal Engine 4. Online: https://www.unrealengine.com, Accessed: 30.01.2015

[13] Leap Motion, Arm HUD VR (Alpha). Online: https://developer.leapmotion.com/gallery/arm-hud-vr-alpha, Accessed: 30.01.2015

[14] Benoîs Dereau, Unreal Paris Virtual Tour. Online: http://www.benoitdereau.com, Accessed: 30.01.2015

[15] Ifcopenshell, open source ifc geometry engine. Online: http://ifcopenshell.org, Accessed: 30.01.2015

[16] getnamo, leap-ue4 – event driven Leap Motion plugin for Unreal Engine 4. Online: https://github.com/getnamo/leap-ue4, Accessed: 30.01.2015