EBU
OPERATING EUROVISION AND EURORADIO

TECHNICAL
REVIEW

# Review of software for the creation of CG-based videos and AR/VR apps

MAY 2018

Selajdin Bilali and Paola Sunna, EBU

# FOREWORD

The main purpose of EBU Technical Review articles is to critically examine new technologies or developments in media production or distribution. All Technical Review articles are reviewed by one or more technical expert(s), at the EBU or externally, and by the EBU Technical Editions Manager. Responsibility for the views expressed in this article rests solely with the author(s).

To access the full collection of our Technical Reviews, please see: [tech.ebu.ch/publications](https://tech.ebu.ch/publications)

If you are interested in submitting a topic for an EBU Technical Review, please contact: [tech@ebu.ch](mailto:tech@ebu.ch)

## ABSTRACT

Extended Reality (XR) applications such as Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR) often rely on the availability of Computer Graphics (CG) assets. Broadcasters have extensive experience with 2D graphics, but often less so with 3D graphics creation.

In this article we explore three software packages designed to aid in the creation of CG-based media applications: Muvizu for building CG clips, Sumerian for the creation of 3D interactive CG applications, and the Magic Leap SDK to test applications in an MR simulator.

# INTRODUCTION

Video has seen a strong rise in popularity and increasingly rivals text and images as a means of communication. Technology has also put content production for more immersive experiences – such as 180 or 360 degree videos – within easy reach.

Things get more complicated when it comes to content that relies on the availability of Computer Graphics (CG) assets. This is often the case with content for what might be called Extended Realities (XR): Augmented Reality (AR), Virtual Reality (VR) and their cousin Mixed Reality (MR).

Many broadcasters already have some experience with CG, especially with 2D applications such as on-screen graphics, and to lesser extent with 3D graphics, which are sometimes used in sports coverage, for virtual studios or for special effects. A typical 3D CG graphics workflow, as depicted in Figure 1, requires a number of skills, such as modelling, rigging, and texturing, which are more commonly found in the gaming industry.

To lower the threshold for content creation, market players have started releasing software tools that aim to simplify some of the processes needed to create a 3D scene with animated CG assets.

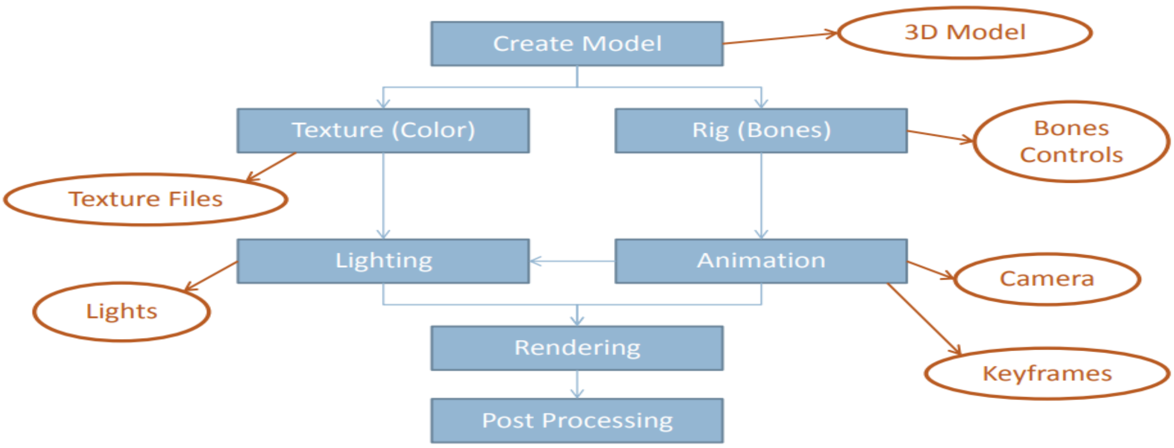



Figure 1 - 3D Graphics in a nutshell [1]

In this article we explore three software packages that promise to assist in the creation of CG-based immersive applications: Muvizu for creating CG-based clips, Sumerian for the creation of 3D interactive CG applications, and the Magic Leap SDK to test applications in an MR simulator.

## MUVIZU

Muvizu [2] aims to help create short 2D animated movies using 3D CG models. It is based on the Unreal Engine. The software reviewed here is the free beta version, with release date April 2013.

Muvizu comes equipped with a dataset of characters, scene sets and animations. The user is like a director in a movie set; he has to place the cameras, characters, lights and objects. With the help of a timeline the user can prepare events such as head and eye movements, walking and running. It is possible to use up to four cameras, to switch among them, move them, and to zoom in and zoom out.


Figure 2 – Screenshot illustrating some of the features of Muvizu: controlling of the eyes, a timeline with actions and monitoring of camera 1's point of view

### Key features

Muvizu is simple to use and quickly allows a user to create semi-professional 2D videos using pre-built 3D animated objects.

The software supports lip-synchronization: when an audio file is attached to a CG character, the software will automatically synchronize the movement of the lips with the audio.

The software allow a good amount of customization of it humanoid characters, including hairstyle, nose, mouth, eyes, ears, and clothes. The skin colour can be changed, too and user-provided textures may be used. It is also possible to add photos of real faces as textures, but the quality achieved is very low.

**Animated versus Rigged object**

***An animated 3D object*** comes with pre-programmed movements that can be triggered; examples include a soccer player walking and then running, an engine spinning, and a butterfly flying.

***A rigged 3D object*** is characterized by the presence of a skeleton. A rigged object can come with or without pre-programmed animations. Its main strength is that new animations can be created freely, either manually or by integrating motion capture data. Packages frequently used for this include Blender [3] and Maya [4]. The resulting animations can be highly realistic.

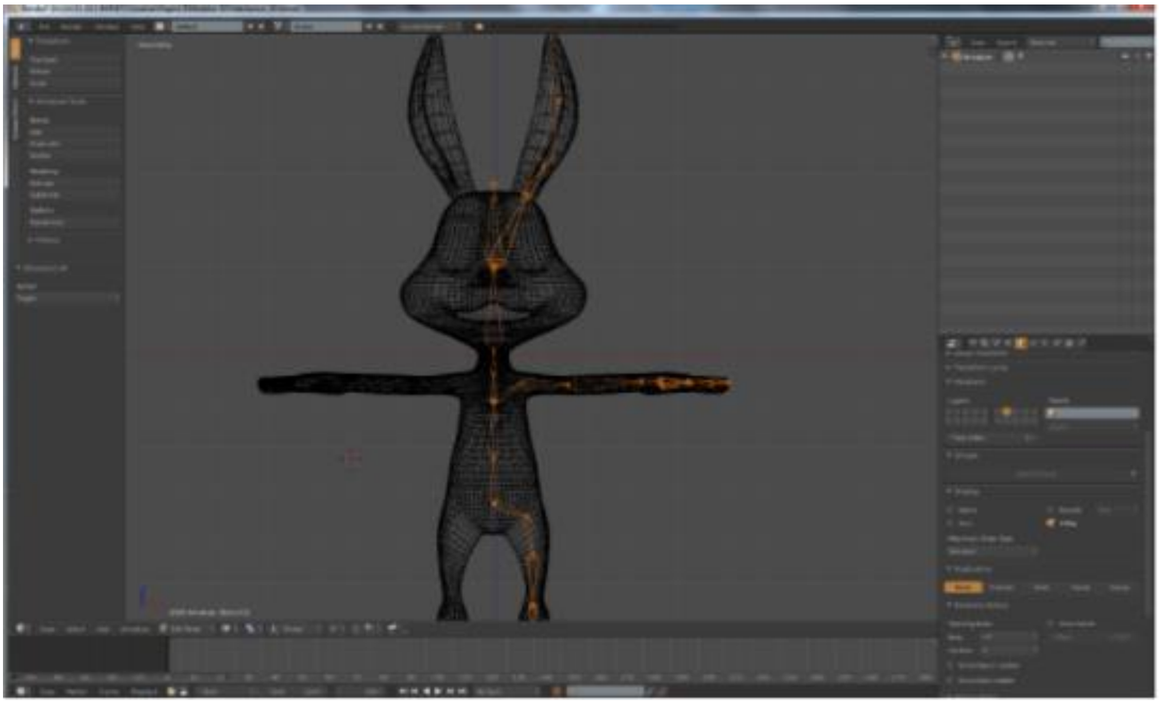


Figure 3 – Example of a rigged 3D object.

## Limitations

Muvizu does not provide advanced features such a coordinate system; instead objects have to be placed by hand.

A severe limitation is the fact that the software records sequences 'live'. If you want to make a character walk, for example, you would need to move the character manually while recording is enabled. It is also impossible to make objects appear or fall at a precise moment in time.

| Feature | Muvizu Play | Muvizu Play+ |
|---|---|---|
| Customisable character types | 2 | 17 |
| Character attachments | 75 | 990 |
| Objects | 79 | 570 |
| Special effects | 1 | 22 |
| Scenes | 1 | 77 |
| Auto lip-synch | ✓ | ✓ |
| Ability to add content / expansion packs | ✗ | ✓ |
| Import free user generated content (.set, .fbx, .ase) | ✗ | ✓ |
| Hand held attachments | ✗ | ✓ |
| Watermark free | ✗ | ✓ |
| Post process effects | ✗ | ✓ |
| Render layers | ✗ | ✓ |
| Video output | SD | HD |
| Price | Free to evaluate | £34.99 |

Table 1 – Muvizu's features overview [2]

### EBU Proof of Concept

To test Muvizu, the audio of the video available at https://tech.ebu.ch/aboutus was used in a simple proof of concept project [5]: an anchorman is followed by a camera while he walks out onto a set to present the different activities of the EBU T&I Department. More examples of what is possible to build are available on Muvizu's Youtube channel [6].

### Conclusions

Muvizu provides a low-threshold entry point into the world of 3D-model-based animation. It comes with a pre-built set of characters, animations, stages and the lip-sync feature. It may be worthwhile to consider if a customized version of such a framework, built with broadcast-typical applications in mind, could help public service media to create CG-based videos more efficiently.

## AMAZON SUMERIAN

Sumerian [7] is an Amazon Web Service designed for the creation of 3D applications. These applications can be viewed in a *WebVR* compatible browser, including headsets such as the HTC Vive and the Oculus Rift (see Table 2). Scenes can also be exported to be played natively on devices with AR capabilities, such as iPhones and Android devices. For this article we used the closed beta version of Sumerian, published by Amazon in November 2017.

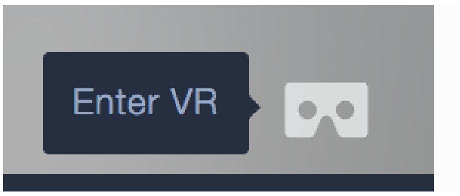
| **Device** | **Software** |
|---|---|
| Google Cardboard | Chrome |
| Daydream | Chrome |
| Samsung Gear VR | Oculus Carmel, Samsung Internet |
| Oculus Rift | Firefox Nightly on Windows |
| HTC Vive | Firefox Nightly on Windows |
| Playstation VR | No support |
| Windows Mixed Reality headset | Edge |

Table 2 – List of devices supporting Web VR [8]

**WebVR**

WebVR is a technology that allows web browsers to detect and use the virtual reality capabilities of a connected device. With WebVR a 3D scene/game/experience can be embedded in a traditional web page and the device can run that scene. It works similarly to a video embedded in a web page.

The user simply connects the VR-enabled device (e.g. Oculus, HTC Vive) to their PC, and after clicking the glasses icon (see the picture below) the application starts running on the device.




In case of standalone devices (e.g. Google Cardboard, Samsung Gear VR) the application needs to be launched from the WebVR browser available in the device.

### State machines

Sumerian's interface and the way of working is very similar to Unity's [9]. A priori knowledge of Unity can therefore reduce the learning time considerably.

**What is Unity?**

Unity is a multiplatform game engine. Content is built once only and then is deployable "with the click of a button" on all major mobile, VR, desktop, console, and TV platforms, including the Web.

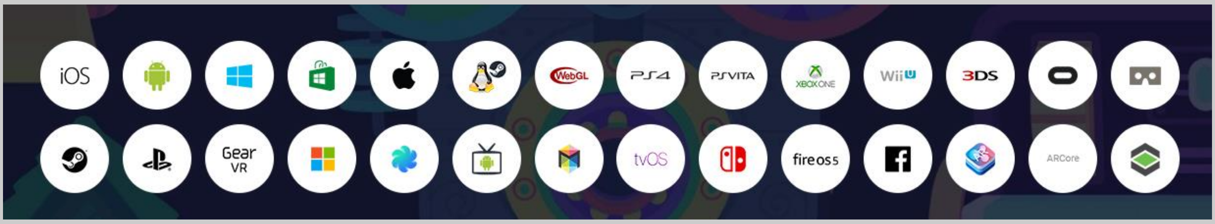



Figure 5 – Platforms supporting Unity applications [9]

Scripting in Sumerian is based on JavaScript, but there are also so-called *State Machines* that allow non-expert coders to use a GUI to manage events and actions. This is particularly handy to create simple scenes.

As Figure 6 illustrates, each *state* is presented as a rectangle. The text in white is the name of the state. The text in blue is the name of the *action* to be executed. Blue arrows symbolize a *transition* to another state. In addition, a state can transmit an *event* to another state machine.

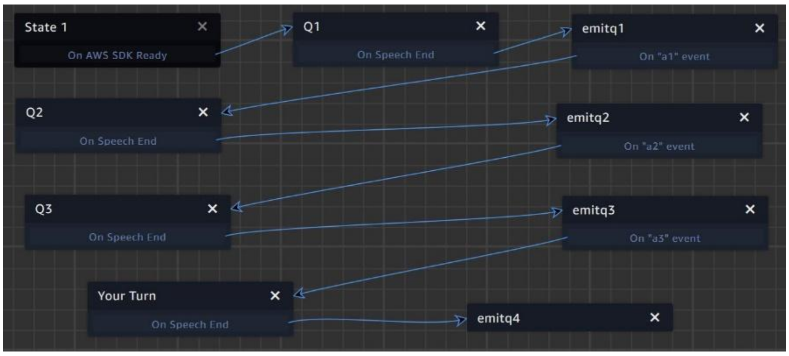



Figure 6 – Graphical representation of a state machine in Sumerian [10]

**Talking hosts**

Currently, Sumerian comes with a limited set of CG assets, such as simple furniture for example. But it also includes a more sophisticated concept: hosts. For now Sumerian provides two hosts: Amy and Preston (see Figure 11).

The hosts are especially interesting in terms of audio-integration. Sumerian supports text-to-speech out of the box. A text can be attached to a 3D CG asset (such as an anchorperson) and it will be vocally reproduced, maintaining lip-sync with the animated character. This feature is implemented via Amazon Polly, a web service that uses machine learning to recognize the text and reproduce it as audio in many languages. Configuration and usage of this service are well integrated in Sumerian. The text can also be synced with several gestures.

**What is a Host?**

Figure 11 – Amy and Preston, the current Sumerian hosts

A host is a 3D model that looks like a human. The difference between a Sumerian host and a classic humanoid 3D model is that the host adds lip-sync and gestures automatically. Sumerian does not currently allow the addition of a host component to a 3D model created or imported by users themselves.

Figure 12 – Text synced with the gesture "Hello"

**Amazon Lex**

Another service integrated in Sumerian is Lex. This uses a chatbot to answer questions. A chatbot is a programme that tries, by means of keywords, to identify the intent of the question, and then answers it with predefined sentences that have to be prepared manually. For example: for the questions “please book a room” or “book a hotel” the intent is similar and the answer may be “In which city?”

**Other features**

Users can also import their own 3D objects. Sumerian supports the .fbx and .obj file formats. Imported objects can be static, animated or rigged.

Sumerian is not a 3D modelling software like Blender or Maya, so objects cannot be animated to walk or run for example. However, simple aspects of the objects can be changed, e.g. their colours and textures.

There is no need to compile the final scenes for a specific device or to register them in a specific applications store, because the scene can be published simply by sharing a URL.

**Sumerian pricing**

At the time of writing Amazon provided the following pricing information [11]

***Scene Storage***
Charges depend on the total storage size of the 3D assets uploaded and stored in Sumerian. The rate is $0.06 per GB per month.

***Scene Traffic***
Charges depend on the total volume of traffic generated by a scene, at a rate of $0.38 per GB per month. The total cost is therefore the number of views received in a month, multiplied by the published project size and the cost of $0.38 per GB transferred.

***Sumerian Hosts (optional)***
Charges apply for the use of *Amazon Lex (chatbot)* and *Polly (text-to-speech).*

## EBU Proof of Concept

To demonstrate Sumerian's features, an interview scenario between the two hosts was implemented. Preston is a journalist who interviews Amy, the actress. He asks some questions, which she answers, and the end-user can step in by typing his questions via an input box. The demo is available here [12].

The scene is comprised of:

- 2 hosts: Preston and Amy
- 2 chairs and 1 table
- 2 cameras
- 1 light
- 1 background

A list of Questions (Q1, Q2 and Q3) and Answers (A1, A2 and A3) was attached to Preston and Amy's respective speech components. A specific voice and language can been associated with each host. An interesting automation tool provided by Sumerian enables the automatic generation of gestures based on the text. For example, if Amy pronounces "I", the pronoun will be recognized and an animation will be added for Amy to put her hand on her chest at that moment in time.

State machines are heavily used in this example, in particular:

- State 1: Access To AWS SDK (mandatory for the use of text-to-speech)
- State 2: Speak Q1 (the text inside the Q1 file is read by the host Preston)
- State 3: Broadcast message 'Q1 finished' (lets the other state machine know)
- State 4: Listen to a message 'A1 finished' (waits for a message called A1 to finish before continuing to the next state)
- State 5: Move camera and ask Q2
- …

A state machine can be very useful for non-programmers, but it takes time to build graphically (see Figure 6), as it requires a lot of clicking to add the states, the actions, etc. A script would be faster, but unfortunately Amazon's scripting documentation is currently not very detailed.

At the end of the interview, the end-user can also put a question to Amy the actress, by writing it in an input box. To implement this:

- the ‘Dialogue’ component is attached to Amy
- a state machine is used to read the text in the input box and to transmit it to Amazon Lex. For this example, a pre-made chatbot called BookTrip was used to simulate the booking of a hotel. When the answer of the chatbot is received, the text is read by Amy’s voice.

## Sumerian and Unity

It's important to bear in mind that Unity and Sumerian target different scenarios and skills. In particular, Sumerian:

- makes it easier for people with no programming skills to create simple 3D animated scenes.
- is de facto multiplatform, as it is web-based.
- supports WebVR.
- provides well-integrated features that are easy to use, such as text-to-speech and chatbots.

Unity requires advanced programming skills and the binaries need to be compiled for every target platform. External APIs such as text-to-speech can be integrated, but this requires heavy coding. However, once scenes become complex in terms of animations, interactions, etc., Unity can provide better performance, as the programme runs natively on the target device, and because the code is optimized to use all the available resources of the device.

## Conclusions

The web-based approach of Sumerian has several advantages, but it requires powerful computers to run on, as the rendering is performed locally. The performance can be slow and also depends on the internet connection speed, the number of assets present in the scene and the browser used. Sumerian currently contains only a few assets and does not allow for the creation of new hosts. The scripting language is JavaScript. Debugging is quite painful, as it is done through the browser's web console. For simple scenes the state machine approach can be used.

# MAGIC LEAP SDK

The company Magic Leap is developing a head-mounted display to superimpose 3D CG-images on real-world objects. This mixed reality headset is claimed to solve some of the issues that other current headsets suffer from, such as limited field of view, weight, and performance. The Magic Leap One headset has been announced for release in 2018 [13].

In March 2018, Magic Leap released an SDK to pre-test the headset's features and develop applications with the help of a simulator. It comes with Lumin OS, a customized operating system capable of supporting spatial computing based on Linux and Android. The SDK also embeds Unity and Unreal plugins along with the emulator used to simulate the upcoming real-world device. The SDK is available for download from Magic Leap’s website [14].

## Installation

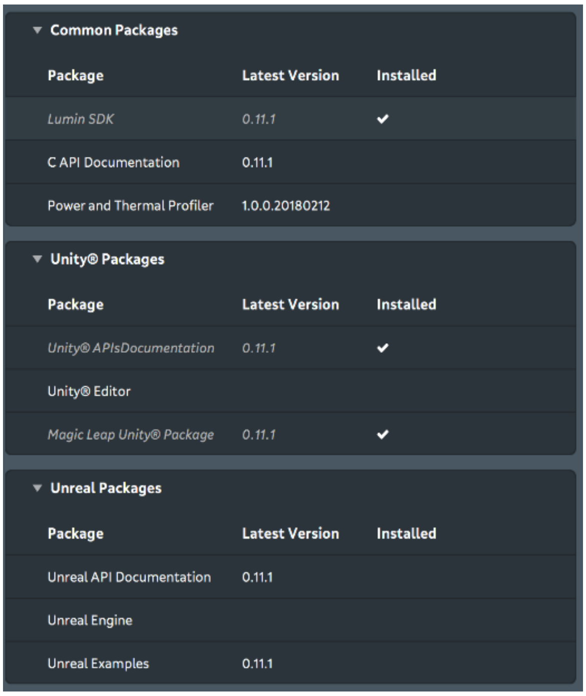



Figure 13 – The Magic Leap package manager

First, the Magic Leap Package Manager has to be installed, then the Lumin SDK (version 0.13.0 at the time of writing) can be added. This includes the Magic Leap Remote Simulator to mimick a “real” device. For this article, the latest version of Unity was used (Unity 2018.1), because it is the only version compatible with Magic Leap.

## The Magic Leap Simulator

The simulator uses three different main concepts: Peripheral, Simulator and Virtual Room.

The peripheral module in the software emulates the input that should eventually be provided by the headset. The image below indicates that the head, eye, gestures and the controller inputs are (virtually) ready.

| Status | Name | Audio | Head | Eye | Gestures | Lifecycle | Input | Room | Graphics | Mesh | Planes | Ray |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ● | Peripheral started | | ✓ | ✓ | ✓ | | ✓ | | | | | |

The simulator itself emulates the output (audio & CG) that will be displayed in the scene viewer window, as shown below.

| Status | Name | Audio | Head | Eye | Gestures | Lifecycle | Input | Room | Graphics | Mesh | Planes | Ray |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ● | Simulator started | ✓ | ✓ | ✓ | | ✓ | | | ✓ | | | |

The Virtual Room is a 3D scene that is used to simulate a “real” environment. Even if the device is not available, the developer can test his programme in a Virtual Room, as illustrated below.

| Status | Name | Audio | Head | Eye | Gestures | Lifecycle | Input | Room | Graphics | Mesh | Planes | Ray |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ● | Virtual Room started | | ✓ | | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ |

## Setup

The Magic Leap Remote must be launched first. Three Virtual Rooms are available "out of the box" and there is also a tool for users to create their own Virtual Rooms.

The Unity plugin needs to be configured to use Lumin SDK as a platform. The plugin creates a connection with the Magic Leap Simulator, and after a restart, the simulator is usable. Elements added in Unity can be seen in the simulator.

As a simple first step, we created a cube to check that the setup is working properly. In the screenshot below, the left window is Unity with the scene view on top, and the camera view below. The right window is the Magic Leap Simulator with the scene view on top and the eye view below. Note that the Play Button is clicked on Unity, so the scene is present in the simulator.

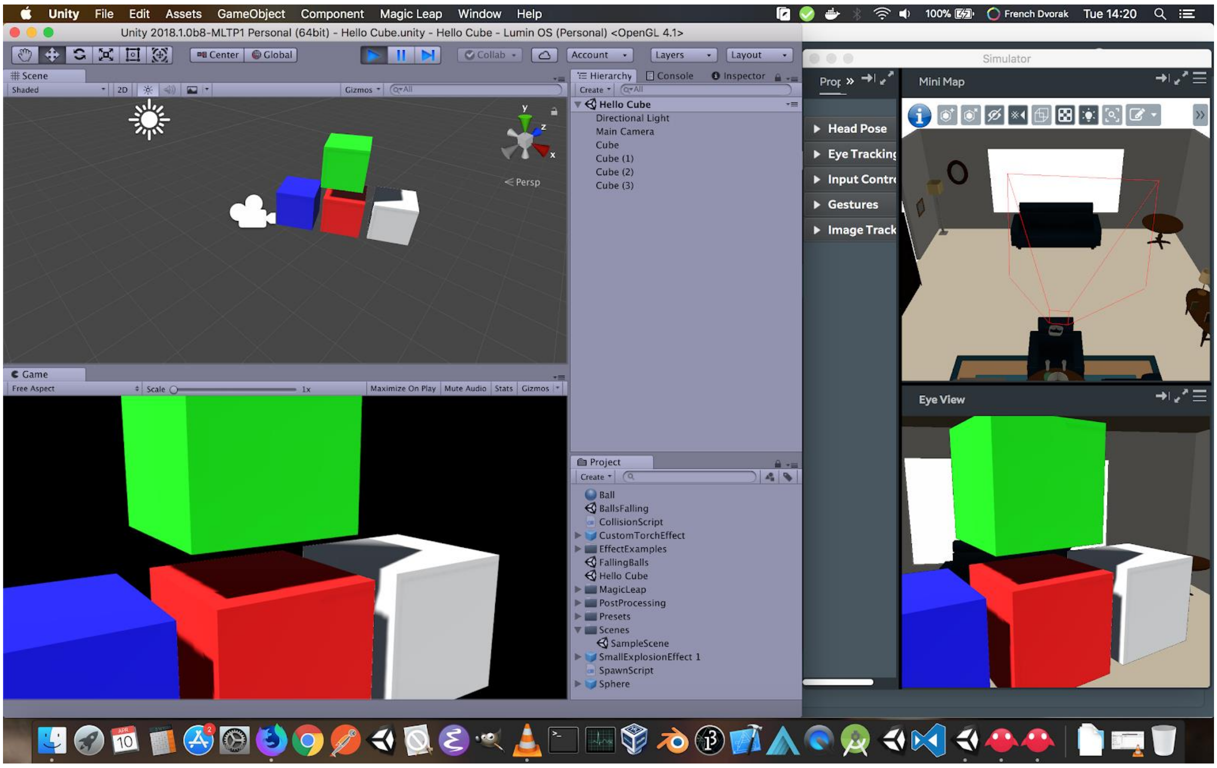



Figure 14 – The Simulator shows the composite scene created in Unity

In the simulator, it is possible to define:

- The head position
- The eye movement
- The input controller position
- The hand gestures to use
- The Image for tracking position

## Examples

The SDK Package Manager comes with a number of examples:

### *Control*

The control example visualizes how the Magic Leap controller will react. To do so, it is possible to bind keyboard keys or to use a compatible gamepad controller (e.g. Xbox, PS4).

### *Gesture Input*

Users of the Hololens already know that gestures are very important to navigate the user interface. The Magic Leap SDK makes more gestures available, which can use combinations of the left and right hand. This allows more natural movements, e.g. picking up a heavy object with two hands. Figure 15 depicts the gestures currently available.


Figure 15 – Gestures available in the Magic Leap SDK

### *Image Tracking*

Magic Leap also provides tools to perform image tracking. Similar to traditional marker-based AR applications, a set of predefined images can be used as a reference system to position a 3D object correctly in the user's field of view.

### *Meshing*

This example converts the real-world depth data into a connected triangle mesh that can be used for occlusion and physics.

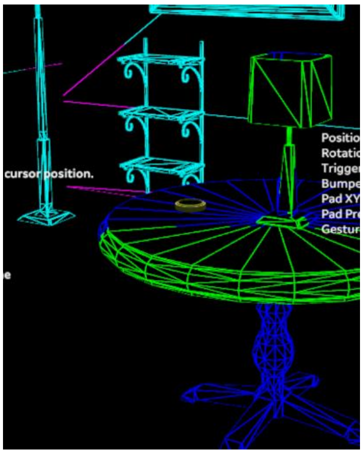



Figure 16 – Magic Leap Meshing example

The meshing scene demonstrates that the Magic Leap can scan the environment and interact with physical objects. It is useful to create occlusions and interactions between real and virtual objects.

***Planes***

This example (see Figure 17) shows that Magic Leap can detect the floor and flat surfaces, including the walls and even the ceiling. This information makes it possible to place, hang, bounce and hide objects in a room.

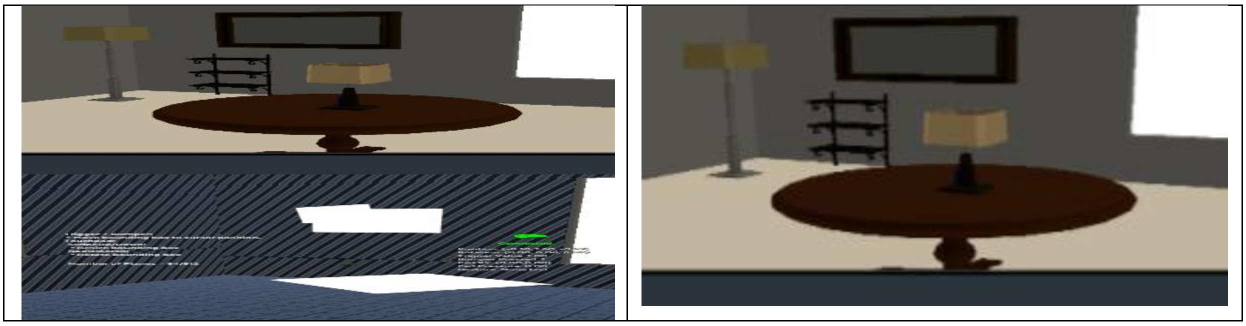


Figure 17 – Discrimination of different planes in the room (left) as detected by the Magic Leap software (right). Note the floor and walls have different textures and that flat surfaces are rendered in white.

***Raycast***

Raycast creates an invisible ray that starts at an arbitrary position in space and detects the objects that it traverses. It can be compared to a laser pointer. This example shows that it’s possible to point at objects by casting three rays: head, eye and controller. The raycasts are independent of each other, so it is possible to move the head around and select an object with a controller. This will provide a more natural feeling when wearing a physical headset.

***Selection***

This example allows the selection of a virtual object using the raycasts feature described above.

**Field of View**

User tests of the emulator's field of view (FoV) seem to indicate that the headset may achieve a horizontal FoV of between 40 and 60 degrees [15] – assuming the emulator accurately models the headset rendering. By way of contrast, the Hololens' FoV is around 30 to 35 degrees, while the next generation Hololens might target 70 degrees.

**EBU Proof of Concept**

The EBU Proof of Concept (see Figure 18) consists of an infinite number of fire balls falling in a room and exploding when they touch the ground or other surfaces. Two components are mandatory to make this scene work: the Camera provided by the Magic Leap plugin for Unity and the MLSurfaceDetection component.

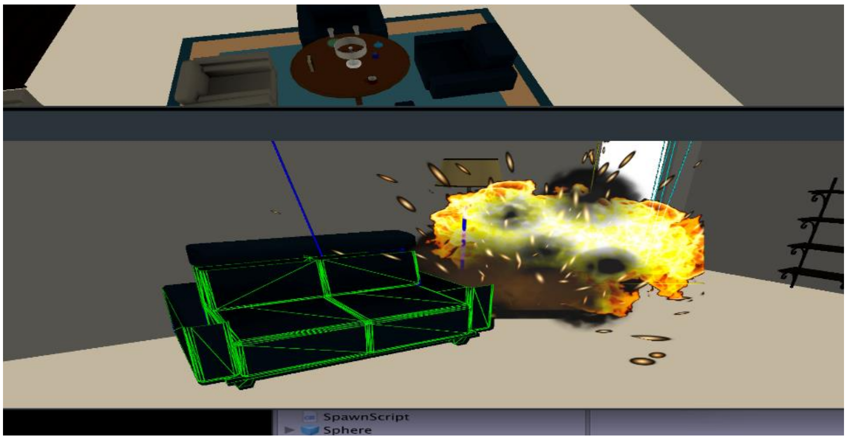



Figure 18 – Screenshot from the EBU Magic Leap Proof of Concept

Examples of what Magic Leap developers have already done with this SDK are available at the Magic Leap World apps store [16].

### Conclusions

The Magic Leap Simulator is an interesting platform for testing and simulations, and also offers clues about the feature set of future MR devices. The installation of the Magic Leap SDK was easy and the integration with Unity is quite good. The documentation still needs to be extended with more tutorials and examples. As the physical Magic Leap device is not yet commercially available, its actual feature set is not yet known. Based on the the simulator, the headset should be capable of:

- Eye tracking
- Room scanning and meshing
- 6DoF (6 Degree-of-Freedom) hand controller tracking
- Unreal/Unity engine support

Additional details can be found in [17].

## OVERALL CONCLUSIONS

Both Muvizu and Sumerian present interesting features that can facilitate the process of building 2D CG clips and 3D interactive CG applications, respectively. For testing the Magic Leap SDK provides a platform to emulate (upcoming) headset interaction.

Muvizu and Sumerian differ in complexity and feature-richness. Audio synchronization is supported in both, but only the Amazon software offers state-of-the art (cloud-based) text-to-speech conversion. But the limited number of Hosts currently provided (only two) and the lack of detailed scripting documentation indicate that Sumerian has not yet matured, yet.

Although relatively easy to use, it should not be forgotten that neither of these packages helps in the creation of the CG asset itself, so that part of the workflow will still require specific skills depending on the complexity of the scene (e.g., rigging and animation, texturing) and the degree of photorealism to achieve.

The Magic Leap SDK is of course very product-specific and for now the promise of the headset becoming available has not yet been fulfilled. One could argue the software currently is more a teaser for the announced product than a tool to develop applications with.

## REFERENCES


[1] http://www.cs.tau.ac.il/~dcor/Graphics/adv-slides/Introduction%20to%20rendering%20techniques.pdf

[2] Muvizu Product Details - www.muvizu.com/Get-Muvizu

[3] https://www.blender.org

[4] https://www.autodesk.com/products/maya/overview

[5] EBU PoC: Muvizu - https://git.ebu.io/ar/muvizu-demo/blob/master/video.avi (EBU Members only)

[6] Muvizu YouTube channel - https://www.youtube.com/user/MuvizuYT

[7] Sumerian - https://docs.aws.amazon.com/sumerian/latest/userguide/amazon-sumerian.html

[8] WebVR - https://webvr.info/

[9] Unity - https://unity3d.com/

[10] Sumerian state machines - https://docs.aws.amazon.com/sumerian/latest/userguide/sumerian-gettingstarted.html

[11] "Amazon Sumerian - Pricing." *Amazon Web Services, Inc.* – https://aws.amazon.com/sumerian/pricing/

[12] EBU PoC: Sumerian - https://bit.ly/2rh9bke

[13] https://www.rollingstone.com/glixel/features/lightwear-introducing-magic-leaps-mixed-reality-goggles-w514479

[14] Magic Leap SDK - https://creator.magicleap.com/downloads/lumin-sdk/overview

[15] Magic Leap One's field of view: making assumptions from the emulator - https://skarredghost.com/2018/03/23/magic-leap-ones-field-of-view-making-assumptions-from-the-emulator/

[16] http://creator.magicleap.com

[17] How Magic Leap Works: SDK Edition - https://gpuofthebrain.com/blog/2018/4/7/how-magic-leap-works-sdk-edition

# AUTHOR'S BIOGRAPHIES

| | |
|---|---|
|  | Selajdin Bilali got a Bachelor of Science from the University of Applied Science of Western Switzerland in 2016.<br><br>He started his career as a web developer and he is currently working at the EBU as software developer on AR/VR proof-of-concepts. |
|  | Paola Sunna studied Electronic Engineering at Turin Polytechnic and got a Master of Business Administration in 2005. Her expertise in media technology covers a wide range of domains: from video coding and image quality to middleware, multi-modal user interfaces, web technologies and immersive experiences. She began working at the RAI R&D centre in 1997. Over a 20-year period her work has brought her into contact with many EU R&D programmes as well as the DVB Project, MPEG, ITU and other standardization organizations. She started a new adventure at the EBU in 2017, focusing on cutting edge immersive audio and video technologies, from content production to user experience. |

Published by the European Broadcasting Union, Geneva, Switzerland

ISSN: 1609-1469

Editor-in-Chief: Patrick Wauthier

E-mail: wauthier@ebu.ch

*Responsibility for views expressed in this article rests solely with the author(s).*