Tampere University of Applied Sciences

Creating Story-Driven Animations for Mobile AR Applications

Implementing Working Mobile Augmented Reality on a Near-Zero Budget

Jónasson Nikolai

BACHELOR'S THESIS
September 2020

Media and Arts
Interactive Media

ABSTRACT

Tampereen Ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media and Arts
Interactive Media

JÓNASSON, NIKOLAI:
Creating Story-Driven Animations for Mobile AR Applications
Implementing Working Mobile Augmented Reality on a Near-Zero Budget

Bachelor's thesis 77 pages, appendices 7 pages
September 2020

The purpose of this thesis was to explore the process of producing and integrating 2D and 3D animated storylines into working mobile augmented reality implementations that bring three stairwell murals to life. Given that the barriers to entry within augmented reality technology have been reduced, the aim was to complete this project on a near-zero budget.

The study was carried out in theoretical and practical sections. The theoretical section familiarizes the reader with the definition, history, and the current state of AR. It was further explored through two exemplary Mobile AR toolsets later utilized in practice. The practical section, which starts with the outlined concept and illustration phases for the mural artworks, establishes a visual and thematic foundation for the production of all three animated storylines. The resulting audiovisual content is integrated into a working Android Mobile AR app prototype, based on ARCore technology, and an existing AR platform Artivive.

These practical deliverables suggest that the available toolsets allow for Mobile AR implementations to be centered around animated storylines and meet the near-zero budget condition. The key aspects of visual storytelling were successfully integrated with the A Grid Aalto startup community in mind. Potential playtesting sessions would determine the preferred implementation method and feasible publishing options.

Key words: mobile augmented reality, 2D animation, 3D animation, mural, app

CONTENTS

ABBREVIATIONS AND TERMS

2D	Two dimensional
3D	Three dimensional
After Effects	A motion graphics and visual effects software used in film, video game and television productions
Android	A mobile operating system owned by Google
API	An application programming interface defines software interactions
APK	Android application package is the package file format used by the Android operating system for distribution and installation of mobile apps and games
AR	Augmented Reality
ARCore	A software development kit developed by Google for AR applications to be built
Asset	Anything that exists in a digital format and comes with the right to use
Blender	An open-source application for creating 3D models
Bridge	Artivive’s online content management tool
C#	C Sharp is a programming language made by Microsoft. It is used to develop web, desktop, and mobile apps
Cycles	A physically based path tracer for production rendering in Blender.
Eevee	A new physically based render engine added to Blender starting with version 2.80.
Image Tracking	The process of recognizing images automatically
Interior Design	The art of designing the interior of a room or building
IOS	A mobile operating system created and developed by Apple
Keyframe	A start, middle and end point of any animated motion
Lip sync	A term for matching lip movements to text or speech

Mesh	A 3D shape
Mobile AR	Mobile augmented reality
Mural	An artwork applied directly on a wall, ceiling, or other permanent surfaces
Orthographic view	A view with no perspective
Path Tracing	A computer graphics method of rendering 3D scenes such that the global illumination is faithful to reality.
Photoshop	A powerful image-editing software
Parallax scrolling	When background images move past the camera more slowly than foreground images, creating an illusion of depth in a 2D scene
QR Code	A quick response code is a type of barcode
Render engine	Software that forms the text and images for display
Rigging	A technique in computer animation in which skeletal joints are used to animate the mesh
Script	A programming language that automates the execution of specific tasks
SDK	Software Development Kit is a programming package to develop applications for a specific platform
Shader	A type of computer program originally used for shading in 3D scenes
Sprite	A computer graphic or image
Sprite sheet	An image that encompasses many smaller images
Squash and stretch	A principle of animation showcasing a contrasting change of shape or volume
Storyline	A plot of a story that encompass a sequence of events
UI	User Interface
Unity	A powerful open source cross-platform 3D engine
UX	User Experience

1 INTRODUCTION

For the past few years, the reduced barriers to entry within the sphere of augmented reality led to a massive overhaul by groups and companies anticipating its use cases. The resulting Mobile AR implementations and developed toolsets utilize said handheld technology of bridging the real and virtual worlds.

Considering the level of accessibility that Mobile AR brings about, potentially on a near-zero budget, would it be possible to apply the available toolsets to my project-based endeavors? Could Mobile AR be developed alongside complex 2D and 3D animations that constitute an overarching storyline in relation to a given interior space?

This objective was a follow-up initiative to an interior design client project for A Grid Aalto University campus. I was a part of it as a trainee under the supervision of Sebastian Sandelin, CEO of Sisätila Design & Interior Architecture. My main task centered around creating three digital mural artworks. These were printed out on stretch walls and installed by FinStudio Systems OY, in March 2020, inside the I-wing stairwell, thus marking the end of the Aalto Project.

The descriptive exploration format of this research paper aimed to streamline the near-zero budget development process centered around graphic design, 2D animation, 3D animation, and Mobile AR app implementation methods. It's accessible to readers that are interested in the thesis subject alongside those following their creative endeavors in this sphere. The structure of my thesis supports this purpose by presenting said deliverables in the theoretical and practical sections accordingly.

The theoretical section begins with chapter 2 in which I explore the defining characteristics of AR in its different shapes and forms, for instance, as mobile applications or eyewear products. These encompass the first or second waves of this technology. Chapter 3, on the other, hand explores Mobile AR through

Artivive and ARCore. The decision to feature these toolsets, based on my standpoint as an individual developer, is supported by clear argumentation for this preference available to the reader.

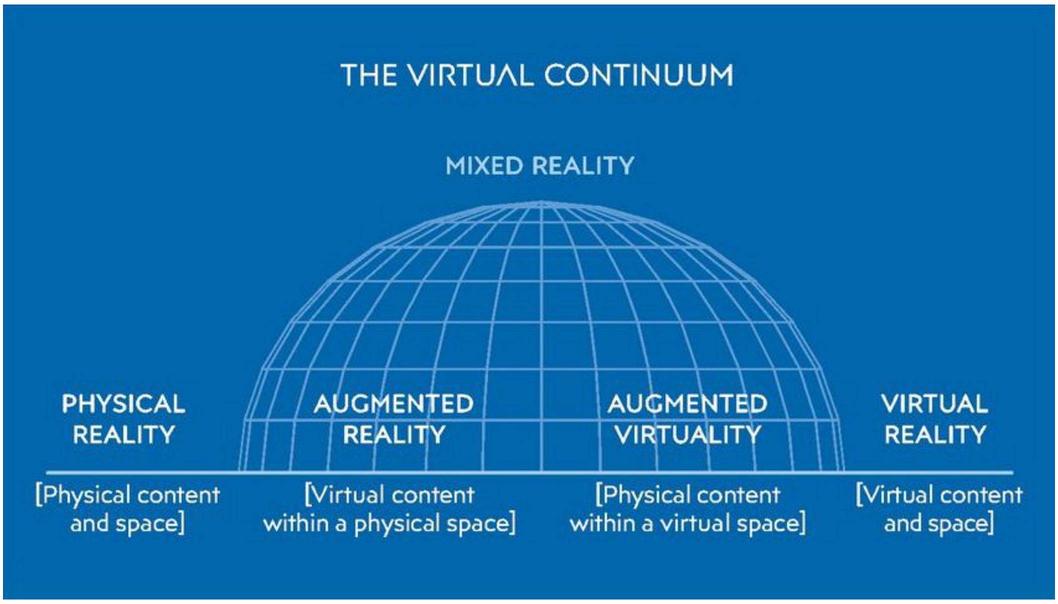
Chapter 4 marks the transitional step into the practical section. Key factors, within the areas of interior design, product design, and project management, influenced my decisions in realizing the objective. They played a vital role in the subsequent practical approaches outlined in chapters 5 through 6. All creative and technical challenges relating to concept art, digital illustration, 2D and 3D animation, as well as Mobile AR implementations, were investigated and solved through detailed documentation of the techniques or approaches I employed.

In chapter 7, I evaluate and discuss the practical deliverables in the context of all initial uncertainties at the beginning of this research. Potential playtesting sessions, developments, and other research topics conclude this thesis.

2 AUGMENTED REALITY

2.1 The Definition of Augmented Reality

Although AR is a digital layer, built on top of the existing physical environment, it is important to bring this definition into perspective. The virtual continuum model is a spectrum highlighting four core immersive states: physical reality, augmented reality, augmented virtuality, and virtual reality (PICTURE 1).



PICTURE 1. The virtual continuum model featuring four core immersive states (Ariel 2017)

There is a glaring difference between two of the most well-known terms within this spectrum, namely augmented reality and virtual reality, that can often be mistaken for one another outside of this context due to their virtual nature. The difference comes in the conditions under which these technologies achieved their immersion levels, for instance, virtual content experience within the physical space or virtual content experience within a digital space.

A less-known method of virtual immersion, arguably another opposite to augmented reality, is augmented virtuality. It aims to bring physical content into the digital space. Currently, augmented reality has the closest link to physical reality within this spectrum. Therefore, it’s safe to say that it serves as an extension of the real-world environment and any of its separate physical contents. The enhancement takes place through video and sound (Ariel 2017).

The real and virtual contents could coexist in an indistinguishable manner, through further developments, thus potentially sharing a spatial relation. One could argue that AR is steadily incorporating key features once exclusive to a more sophisticated and costly mixed reality method. As a result, AR's position within the sphere of spatial computing is appropriate.

2.2 History

When looking back on the history of AR, similarly to the previous chapter with the contextualized definition, the overview would center around the Four Paradigms. The cornerstone of a paradigm is the evolution of the user experience, which leads to technological developments as seen throughout history (Cronin & Scoble 2020).

The first paradigm marked the beginning of the personal computer, which turned into a mass-market consumer electronic device available to most users in the early 1980s by streamlining the technology and employing a user-centered approach, as seen in products such as Apple II and the IBM PC. The second paradigm involved the next logical step from text-based interfaces to ones with graphics and color capabilities. This graphical approach would once more center around the end-user as they could now much easier access all virtual features with just a click on an icon, especially after Douglas Engelbart's historic invention- the computer mouse (Cronin & Scoble 2020). Although the earliest examples of graphics-based personal computers could be traced back to 1973, namely the Xerox Alto, a general availability came with the launch of the Apple Macintosh in 1984 and Windows 95 (O'Regan 2018, 62). The third paradigm allowed for improved mobility, in a way bringing features from the desktop operating systems onto mobile devices. Companies like Blackberry, Nokia, and Treo were at the forefronts of bringing these features to our pockets until the historic shift into touchscreen technology popularized by Apple with its first iPhone in 2007 (Cronin & Scoble 2020).

Suddenly the user had a more intimate experience by being able to feel the interface through a simple touch or gesture. Although smartphones indeed laid the basis for an improved user interface within each mobile application, they also set up conditions for a realization of spatial computing, most notably by introducing the general public to a seemingly obscure term, augmented reality. Until the arrival of Pokémon GO back in 2016 (FIGURE 1), it was not nearly as widely known or accessible to the general public.

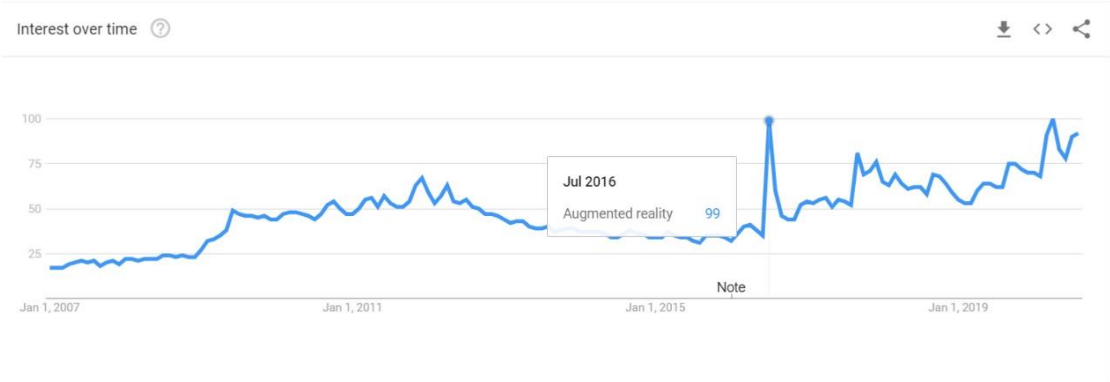


FIGURE 1. Google Trends for augmented reality encompassing the original iPhone's release date in January 2007 and Pokémon GO's release date in July 2016 (Google Trends 2020)

The first written account of a type of augmented reality could trace back to the history of the moving image and early cinema. The pioneering filmmaker Dziga Vertov called attention to the power of a camera in its depicting of a new reality, as seen in his very own film Man with a Movie Camera in 1929 (Papagiannis, 2017). The camera-centered approach to the extended reality sets Vertov's work and today's AR technology in the same evolutionary timeline. For the better or for worse, it lets the human become one with it. Vertov once famously stated in his written works, *"I am a mechanical eye. I, a machine, show you the world as only I can see it."* (Papagiannis, 2017). One could argue that this perspective is also one of the earliest depictions of the next step into spatial computing- the fourth paradigm.

According to AR expert Dr. Helen Papagiannis (2017), instead of imagining what it would be like to see like a camera, the opposite is being currently realized- to make the camera and computer see like a human.

2.3 Realizing Spatial Computing

There is a level of uncertainty in terms of a definitive realization of a technology that would work more as humans do. Arguably, the most realistic pathway towards such implementation stems from a combination of different technologies, including augmented reality, augmented virtuality, and virtual reality. The aforementioned Virtual Continuum model popularized back in 1994 by Milgram and Kishino, hinted towards such a method concerning mixed reality. Judging by current developments in this area, optics and displays are potentially closest in that regard. Early versions of such products have existed on the market since 2013 but unfortunately failed to encompass all usability breakthroughs, which led to general inaccessibility for the end-user.

Even though Google Glass, Microsoft HoloLens, and Magic Leap ML1 have demonstrated passthrough designs that seamlessly brought these devices closer to the human eyes, they have failed to be commercially successful. Unlike the case with smartphones, the high prices, small or low-quality screens, and low developer support constituted the high barriers to entry (Cronin & Scoble 2020).

3 MOBILE AR

Smartphones, with their lightweight designs, affordability, advanced camera sensors, and processing power, had finally caught up with more exclusive AR technologies. Smart glasses and headsets are expected to reach their full potential by 2024 or later (Cronin & Scoble 2020).

One could argue that Mobile AR has currently surpassed these devices. The ongoing improvements to software development kits, through Apple's ARKit and Google's ARCore, already in 2020 allowed for spatial awareness by introducing depth-sensing algorithms (Saini 2020).

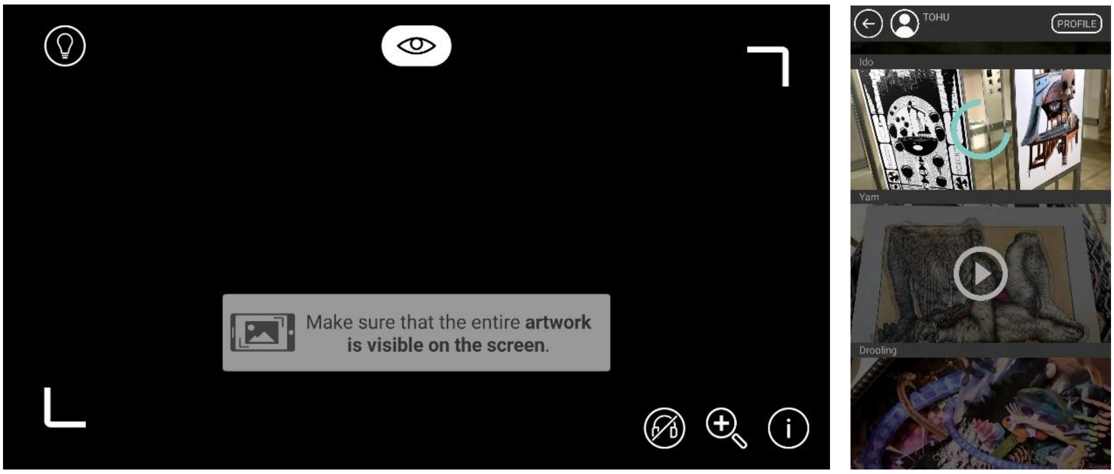
3.1 Artivive

Artivive is an augmented reality application, available for Android and iOS, with a vision to change the way art is created and experienced by the audience, leading to the formation of a community around this immersive approach. Founded in 2017, this Austrian company won in the Best Austrian Startup category at the Pioneers conference in Vienna, for garnering such overwhelming support for their Mobile AR app. It had surpassed 60,000 downloads and was used by 2000 artists in 65 countries during Artivive's first year of being active (Forbes 2017).

The app uses a technological implementation that originates from the first wave of augmented reality. Helen Papagiannis (2017) refers to it as the *"Overlay"* and rightfully so, for it is indeed a case in which virtual content appears fixed as a layer on top of a real-life object that acts as an image target.

Although it is true that from a technical standpoint, dependability on a physical image target disconnects the user from the real-life environment and reduces the level of interaction, there's something to be said about the appropriateness of this approach in the area of the arts. The physical artwork acting as an image target is a permanent element of the interior, rather than a disposable object meant to demonstrate AR technology. Furthermore, given the art-oriented nature of Artivive, it's difficult to argue for the necessity in placing virtual content exclusively outside of the given artwork image target as that would go against the whole point of bringing art to life.

Although a bold statement, the same could be said about Artivive’s streamlined approach in delivering all the necessary instructions, print material, toolsets to organize and deliver these augmented reality experiences, thus changing the way art is created and consumed.



PICTURE 2. Screenshots of the Artivive UI on the left and the community page on the right

The free Artivive mobile application has a minimal user interface (PICTURE 2) with all functions needed to improve the conditions of the AR experience. For instance, the lightbulb icon on the upper left-hand corner that switches the camera flash on and off or the magnifying glass icon near the bottom right-hand corner, that prompts a two-step digital zoom, either 2x or 3x.

The banner in the bottom center instructs the user on how to use the application through text and an appropriate minimalist icon representing the described action as well as the landscape orientation of the device itself that’s best suited for this user interface. The eye icon in the top center opens up, on a completely different page, in portrait view, filled with augmented reality video captures shared by the Artivive community from all around the world.

As soon as the app identifies the image target, indicated by a simple loading bar, the appropriate video content plays automatically thanks to the online content management tool Bridge, developed by Artivive. It lets the artist assign the right video content and store it on an online server. The user can record their experience, and after reaching the time limit, mainly in the interest of an artist's copyright, share it through social media, download it onto the mobile device itself, or share the experience with the Artivive community.

The permanent collection Monet to Picasso at the Albertina Museum in Vienna integrated the Artivive app (Artivive 2020).

3.2 ARCore

Founded in 2018, as a follow-up to a far less successful Project Tango, ARCore is Google's software development kit allowing developers to build AR experiences into their projects in a more advanced, interactive, and customizable manner (Kastrenakes 2017; ARCore 2020). In a bit over a year, ARCore had reached 400 million devices globally, which, together with Apple's leading ARKit platform, resulted in a total of 1.05 billion AR compatible devices (Boland 2019).

To this day, ARCore lags behind ARKit in terms of its user base and active developers utilizing the augmented reality technology. The reason for this stems from the fragmented nature of Android smartphones as they are developed by different companies, with the resulting hardware missing some key features, e.g., advanced camera sensors. Otherwise, this would allow for ARCore to fully function across Android's leading market share (Boland 2019; Makarov 2020; Casserly 2019).

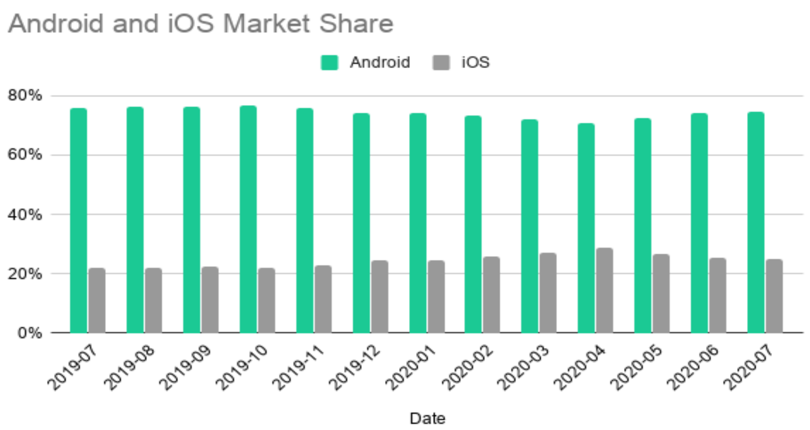
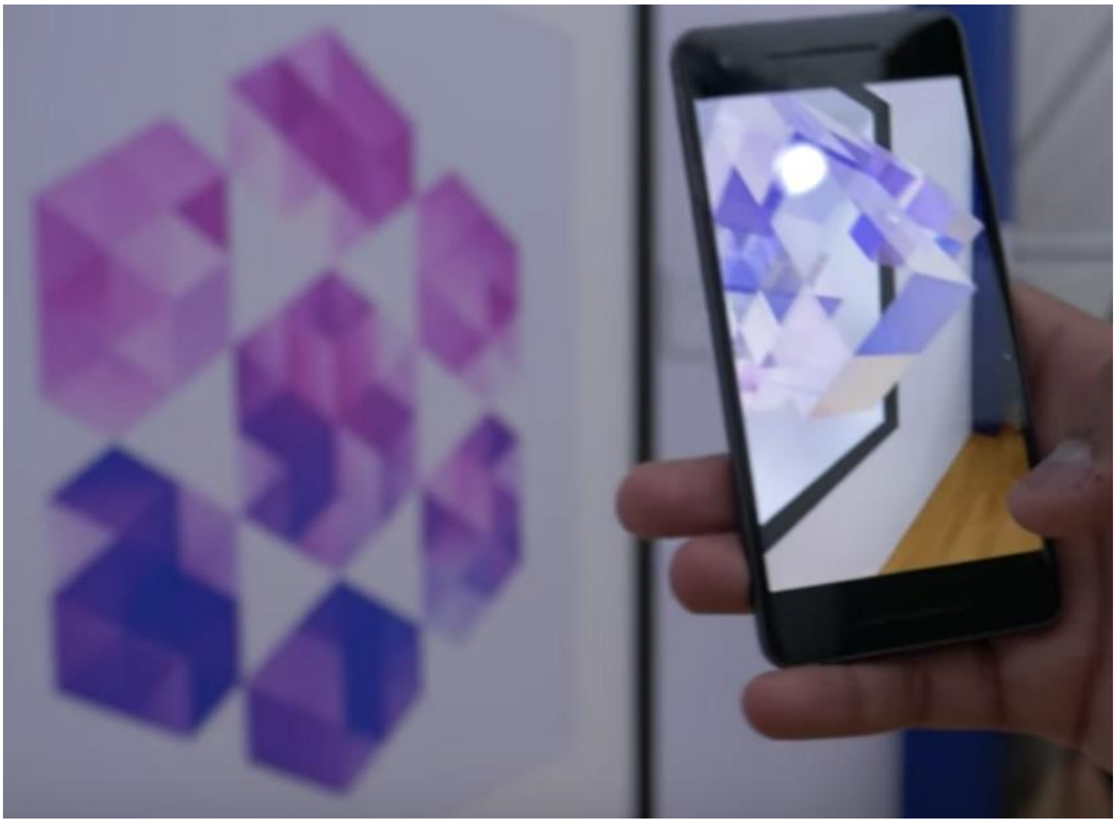


FIGURE 2. Google column chart showcasing the Android and iOS market share since July 2019 to July 2020 based on data gathered from Statcounter (writer's own picture)

It's this Android market share (FIGURE 2) that, from a developer standpoint, statistically shows the potential for an entryway into AR for such a large mobile operating system user base (Statcounter 2020).



PICTURE 3. The Transmogrification artwork poster and the ARCore mobile app demo in action (Simpson 2018)

In practice, the Mixed Mediums gallery showcase by Novel at the annual Google I/O event for developers, back in 2018, paved the way for the first wave of AR on Android smartphones. It did so through its three artwork demos, each exploring a separate use case of the ARCore Augmented Image feature. The Arboreal artwork poster prompted the complex branching and formation capability, similar to a tree. The Transmogrification artwork poster gained depth inside of the canvas and popped out in 3D at certain angles (PICTURE 3). Lastly, the Chrysalis artwork cube installation demonstrated the spreading virtual enclosure of the real-life environment, for instance, the actual cube installation (Novel 2018).

It is possible to draw a parallel between the Google I/O event, featuring a visual approach to AR technology, and Google's latest depth-sensing feature set titled Depth API, as they both bring a dramatic change to the status quo.

This second wave of augmented reality is referred to as the "Entryway" by Helen Papagiannis (2017), who goes on to express great expectations from this new interactive approach that turns the whole world into a trackable target, leading to the formation of one's contextualized augmented reality experience (PICTURE 4).

PICTURE 4. The Lines of Play mobile application utilizing the Depth API to occlude people and place the dominos in any part of the world (Damiani 2020)

However, it is one thing to be inspired by any of these Mobile AR implementations, but to appropriate them within a particular project might pose a challenge due to the associated factors.

4 KEY FACTORS

4.1 Target Audience

A Grid serves as a start-up community for 150 start-ups and partners. United Nations Technology Innovation Labs, European Space Agency's Business Incubation Centre, and Aalto's very own start-up center are some of the largest tenants. This information would determine my approaches to developing all three digital murals, animations, and the Mobile AR app.

It became self-evident that the target audience should be new and current members within A Grid. This notion would gain support in the official A Grid news blog post. Aalto University Campus and Real Estate's Managing Director Ville Jokela states that *"The spirit of A Grid lies in the member companies and the community they create. It's about working together to create new solutions for the future."* (Agrid 2020).

There was a level of ambiguity in the age group target as anyone could apply for either a private office or coworking space membership. Taking this into account, I employed folklore elements and set up different atmospheres for each floor. I sought to appropriate the visual storytelling aspect of my work by representing a range of subjects that adhere to the start-up community.

4.2 The Interior

Establishing a consistent placement of the murals on each floor and getting the precise measurements would help to determine the scale and time required to complete the entirety of the work. The 6,1 by 1,8-meter canvas in the stairwell provides plenty of space for ambitious work and exposure potential. However, this placement also carries with it some challenges. For instance, the attention span of people passing by and the inside corner wall. I would need to add engagement and functionality to the practical deliverables.

The stairwell connects floors 2-4, which, as a result of the renovation, are color-themed into red, green, and blue colors. In such a case, it would be appropriate to transfer this information through my work.

4.3 The Near-Zero Budget

The primary prerequisite that I placed on myself was to do my follow-up work at no extra cost. I would not exclude paid software such as Photoshop and After Effects, as I use them daily. Backing up my case is the practicality of this software concerning the project, although I do bring up free alternatives.

Having the near-zero budget in mind would lead to thoughtful approaches and practical solutions, thus setting realistic limits on developing the Mobile AR app prototype, especially with programming in mind.

4.4 Software and Hardware Used

I would use Blender 2.81 to produce 3D models and animations alongside the previously mentioned software. Unity 2019.3 (personal license) with a free version of Visual Studio 2019 for scripting made building the Mobile AR app prototype possible.

The mobile device I own, from 2016, works with ARCore but is not compatible with the latest Depth API. It has adequate specifications even now with Android version 9, Snapdragon 821, and 6 GB RAM, so this wouldn't pose an issue for the marker-based approach. One of the main problems is the lack of AR functionality on some budget smartphones. I recognized the possible accessibility issue early on, and so should any developer. In practice, a user with an unsupported device can only access the app menu. One possible solution worth considering revolves around two virtual content experience modes: one appearing in AR on the inside corner wall and the other on the mobile device itself.

5 CREATING THE MURALS AND ANIMATIONS

5.1 Second Life

PICTURE 5. 2nd LIFE mural illustration (writer's own picture)

2nd LIFE is the name for the second-floor level and the mural artwork leading to it in the stairwell (PICTURE 5). It presents the viewer to a dramatic showdown between a neon Aalto knight and a fire breathing dragon on a cosmic scale.

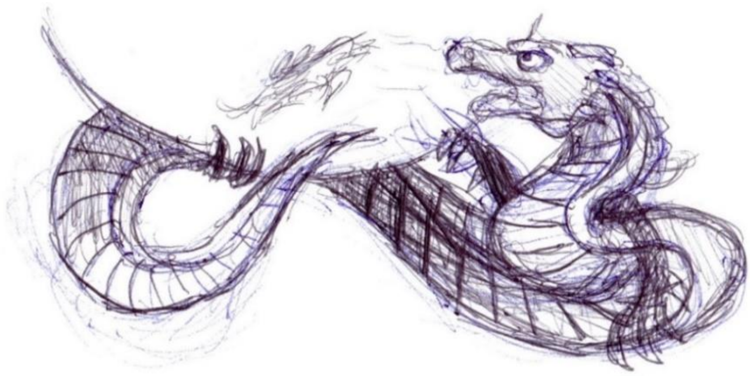
5.1.1 Concept

Myths and classic video games featured lives through different visual means. Yet, the storytelling aspect would always be central, even if it would take a while, as was the case with the gaming industry, in the early 1980s, when themes and stories finally happened (Byrne 2016).

The Hero's Journey formula by Joseph Campbell describes our ability to overcome any fears of the future and harkens back to an earlier time. For instance, based on fond memories of trying to win a boss battle, one life after another. As Mark Collington puts it: *"Recognizing these hopes and fears, as well as 'the tests, enemies and allies' we face in real life, in many of the fairy tales and films we enjoy reading or watching provides a means of reflection on, but also escape from our everyday lives."* (Collington 2016, 54).

A mythical beast would adhere to the aggressive nature of the red color and serve as an amalgamation of all project-based issues and the resulting anxiety.

Norse mythology and Chinese mythology were primary visual influences for my dragon design, as seen in the first concept sketch (PICTURE 6).

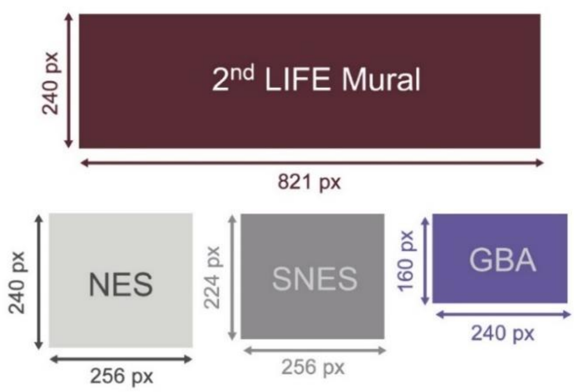


PICTURE 6. 2nd LIFE mural concept sketch (writer's own picture)

A specific sea creature depiction, found in Carta Marina by Olaus Magnus from 1539, was reminiscent of Níðhöggr, which is a serpent that gnaws the roots of Yggdrasill- the tree of life (Taylor 2016). One of its earliest depictions is in the 17th-century Icelandic manuscript (Handrit n.d.). The Edda Oblongata also features Jörmungandr, which is the sea serpent large enough to eat its tail and encircle Earth. Both depictions would influence the facial features of my dragon character design and the surrounding composition structure. The resulting body shape originates from a generalized Chinese dragon depiction.

5.1.2 Illustration

Looking back on old video gaming systems (PICTURE 7), I chose an optimal pixel height of 240, featured in the NES. This way, the pixel art style would be viewable up close, on the phone display in AR, and from a distance.



PICTURE 7. 2nd LIFE mural canvas resolution compared to the Nintendo Entertainment System, Super Nintendo Entertainment System, and the Game Boy Advance resolutions (writer's own picture)

Setting up the image interpolation to nearest neighbor and exporting through the save for web export method in Photoshop allowed me to save the finalized artwork in any desirable scale.

It was essential to use a color palette that would limit the amount of shading. Such an approach would help replicate the real-life limitations of the NES, so I only used eight colors for the whole composition. The dragon sprite, however, used all eight simultaneously (PICTURE 8).

PICTURE 8. The dragon character sprite using the 8-color palette (writer’s own picture)

Since the NES could only display four colors per sprite the character design would need to use the sprite layering approach or have sub-palette assignments (Oriz and Geig, 2019). Besides adding light and texture to the character sprite, this highlights the floor number indicator. To take advantage of the inside corner, I chose an anamorphic illusion approach popularized by artists such as Odeith (Stewart 2019). This approach involves distorting a design element in the corner section of the mural. As a result, the dragon even further pops out of the mural from certain viewpoints (PICTURE 9).

PICTURE 9. Digital renders of the dragon character from different angles (writer’s own picture)

William Stewart Watson's collection of sketches from 1840 was the primary reference for the ambiguous Aalto knight character design that aims to be relatable regardless of nationality or gender (New York Public Library n.d.). The helmet design would feature the letter A, which stands for Aalto. Having the Aalto knight on horseback would support a more heroic image adhering to the story and theme (Collington 2016, 23). The horse also creates a visual balance (PICTURE 10). The blue light ray, which rises from the first-floor mural by Teo Georgiev (PICTURE 11), is used to establish continuity within the Aalto space.

PICTURE 10. 2nd LIFE mural in the stairwell (writer's own picture)

PICTURE 11. Unicorn mural by Teo Georgiev (Georgiev 2018)

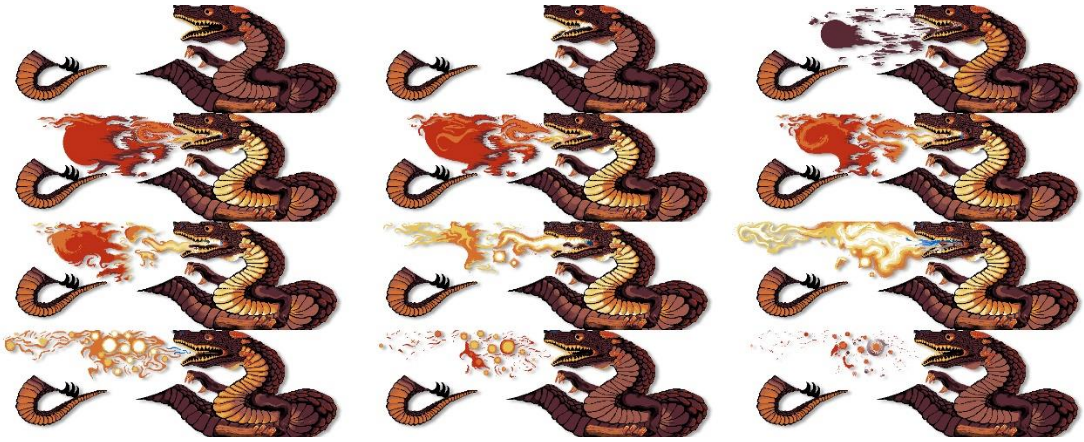
5.1.3 Creating the Animated Sequence

The 2nd LIFE mural artwork plays a vital role in creating the foundation. All animated sequences would follow a self-contained or episodic approach that centers around enriching and clarifying the visual storytelling. Through the globe and RGB color model motifs, the animations would still allow for overarching continuity.

Preparatory work involved optimizing all required pixel art sprites. I would make the frame by frame animations in Photoshop then export them as image sequences. The reason for this is the timeline animation feature in Photoshop, which is limiting. Free alternatives for pixel art creation and animation do exist, for instance, Piskel and Pixilart. However, I needed to replicate a smooth in-game player movement, which is why I favored After Effects to realize this task efficiently. I should note that free software such as Da Vinci Resolve work similarly. When creating the pixel art animations, I aimed to place them in a sprite sheet. Thus, all character states where limited to a certain number of frames as old video game animations.

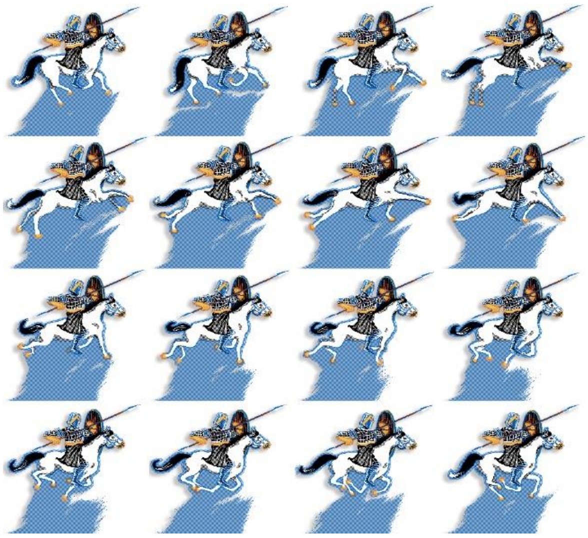
I divided this animated storyline into six scenes to replicate a classic video game boss battle. Working in chronological order would prove to be beneficial as I would complete all base animations and reuse them.

Scene one brought with it the biggest animation challenges- the fire and dragon character movement. A color swapping technique would speed up the process of redrawing each frame of the fire lifespan. The stiff movement of the dragon accentuates the scale and weight of the dragon. There is dependability on the fire breath, hence why the most dynamic movement takes place in the upper torso. The secondary dynamic movement is in the tail when there is a chance to hit the Aalto knight from behind. I took into account smaller aspects of this base dragon animation, for instance, arm movement, periodic blinking, and flaring nostrils (PICTURE 12).



PICTURE 12. The fire breathing section of the base dragon animation in a sprite sheet and the full animation demonstrated in action: https://youtu.be/ej4IairT48Y (writer's own image)

The animation for the Aalto knight was more manageable as I already had all limbs, the shield, and weaponry on separate layers. The horse movement, on the other hand, required a lot more work. The main challenge was to create a loop for the horse's leg movement (PICTURE 13). To do this, I referenced the famous Muybridge racehorse gallop sequence from 1887 (Muybridge 1887).

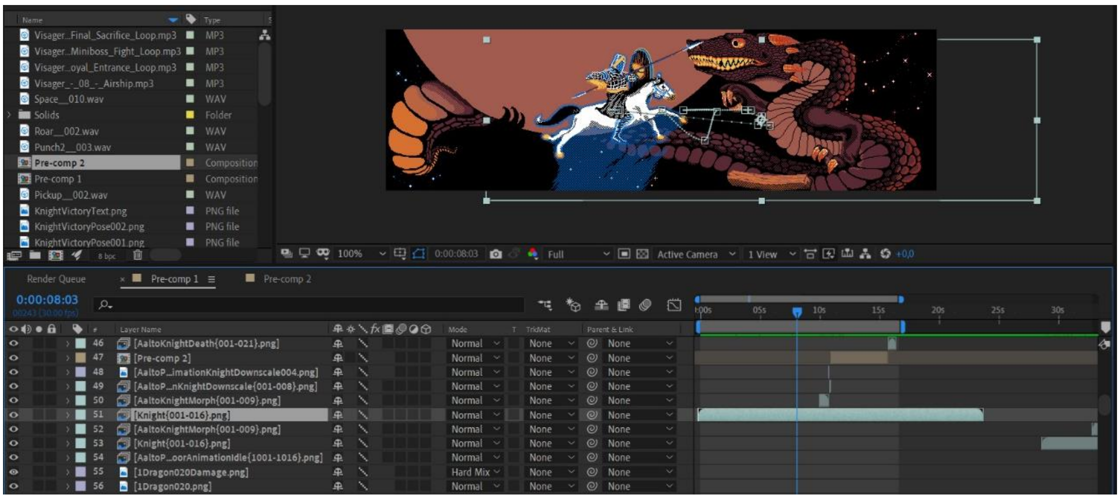


PICTURE 13. The base animation loop for the Aalto knight on horseback presented in a sprite sheet and the animation in action: https://youtu.be/660ANkOuews (writer's own image)

Attempting to cope with all project-based issues, with no help from fellow teammates, is destructive. Through such carelessness, one could run into a shattered state of self-esteem. The act of the character shrinking in size (PICTURE 14) symbolizes this process. It is also a standardized visual indicator of the lost health, as seen in Super Mario Bros. The now small Aalto knight is at the most vulnerable state, right before the inevitable failure.

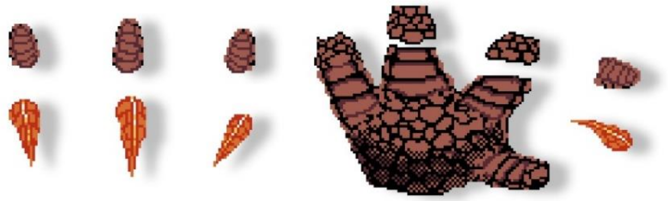
PICTURE 14. The base morphing into a smaller version presented in a sprite sheet and demonstrated in action: https://youtu.be/VQ0fQ1fKaJU (writer's own image)

When moving the image sequences in-between keyframe points, I would use the on-screen motion path feature in After Effects (PICTURE 15). Having the layer sampling quality switch set to draft allows for the crisp nature of pixel art to be retained in After Effects. The same goes for the render output in the built-in renderer or Adobe Media Encoder.



PICTURE 15. A preview of how the Aalto knight sprite player movement was constructed utilizing the on-screen motion path feature in the After Effects project

Scene two acts as a dramatic death screen as it involves an enormous dragon hand capturing the lost life or heart. Both dissolve into the darkness from which the 2nd LIFE title screen emerges. This scene utilizes two previously discussed approaches. The small Aalto knight morphs into a heart shape, and the dragon hand sprite consists of different layers (PICTURE 16).



PICTURE 16. The deconstructed dragon hand (writer's own image)

Scene three is the 2nd LIFE title screen, which resembles a game over screen. Even this scene had some level of dynamicity in the form of a second heart, in red color. This playful character is ready for action and prompts the protagonist to continue through the speech bubble and simple lip sync (PICTURE 17).

PICTURE 17. Title screen in action: https://youtu.be/3Yy_oZf4Bf8 (writer's own image)

The heart character acts in a determined and hopeful manner, thus underlining the importance of not letting go of one's endeavors, let alone giving up when faced with failures.

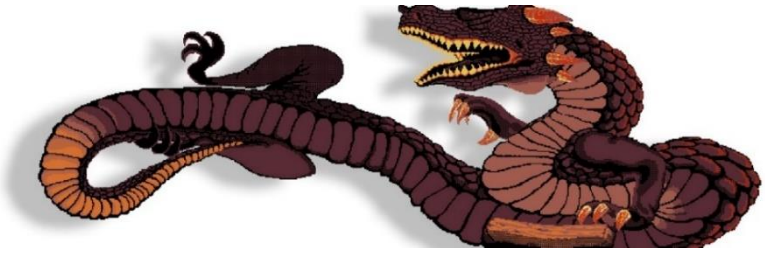
Scene four returns the viewer to the beginning of the boss battle. The Aalto knight learns from prior mistakes and delivers combo levels of damage to the dragon, but this is not done without a helping hand, or in this case, hove.

PICTURE 18. Unicorns in action: https://youtu.be/kP8MafVPO6M (writer's own image)

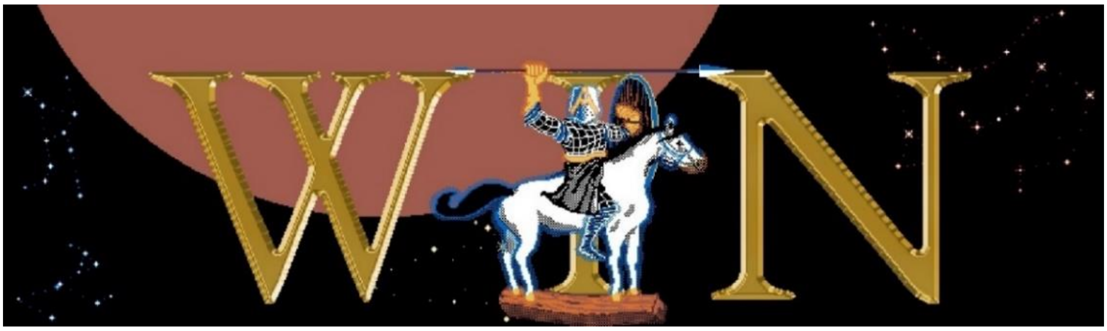
Scene five reintroduces the cartoony and personality-driven unicorn characters from the first mural by Teo Georgiev. This time around, they symbolize the individual team members ready to help return the Aalto knight to the initial powerup state (PICTURE 18). Their animations are based on the squash and stretch principle. The emitted beam of light stands for the act of helping the Aalto knight. It also references one of the ongoing motifs, the RGB color model.

Scene six is a direct continuation of the boss battle. Upon defeating a boss in classic video games, it either explodes or slides away out of frame. Since there is enough fire, I made the dragon lose its grip around the sphere and get hurled out of frame. Thus, I constructed a full dragon body sprite. It would encompass a pose based on general depictions of a Chinese dragon. This sprite (PICTURE 19) would then move between two keyframe points in After Effects.

PICTURE 19. The full dragon body sprite which is its last state (writer's own image)



The battle ends with the Aalto knight's winning pose (PICTURE 20). The beveled text, made using Photoshop, resembles pre-rendered 3D graphics, which is a nod to the progression in years to come after the NES. The singular expression of *"a win"* references real-life working experience that doesn't end in general, more so helps everyone progress in life through each new project.



PICTURE 20. The last frame of the 2nd LIFE animated sequence as seen in the completed product: https://youtu.be/ktD4siWWQXA (writer's own image)

5.2 Third Element

PICTURE 21. 3rd ELEMENT mural illustration (writer's own picture)

3rd ELEMENT is the name for the third-floor level and the mural artwork leading to it in the stairwell (PICTURE 21). It presents a condensed overview of the multifaceted Finish natural landscape, the center of which lies within an isolated island that brings about the digitalization of the surrounding environment.

5.2.1 Concept

The idea to create the mural artwork in a low poly style originated at the earliest stages of the Aalto Project. Much like the color green, the perception of the angular nature of low poly can be villainous. This way, allowing me to reflect on the technological risks to working life. On the other hand, the use of this simplified representation of an evergreen forest landscape adds to the ambiance of the scene. This calmer outlook on working life would allow the observer to find peace of mind.

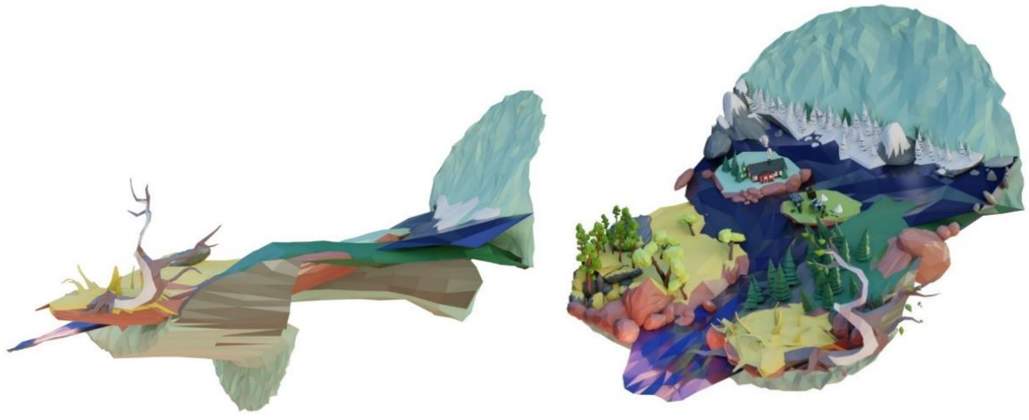
PICTURE 22. 3rd ELEMENT mural concept sketch (writer's own picture)

As seen in the concept sketch (PICTURE 22), a mythical association was still drawn, in line with the 2nd floor. The Yggdrasil tree serves as an allegory for one's personal growth. A healthy outlook rises above a self-inflicted state of isolation through the eyes of the digital world, which is a condensed reflection of the real world. This realization takes place after broadening the working experience and seeing technology as a tool rather than an alternative to reality.

5.2.2 Illustration

Contrary to the previous mural artwork, there were no prerequisites other than the need for original low poly 3D assets. I would model and reuse them efficiently throughout the scene in Blender 2.81.

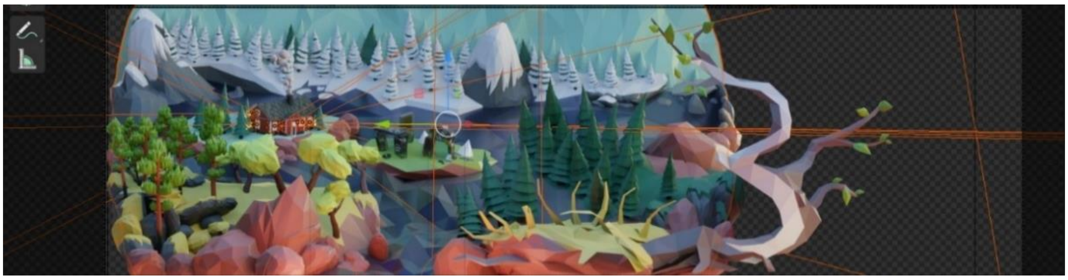
Since the natural landscape is in Finland, I was able to use real-life as a visual reference. To keep the combined consistency of the globe motif, I chose a surrealist approach in the orthographic view. This way, I could segment the composition into a foreground, middle ground, and background a lot easier in front view. In practice, this involved extending specific areas of the separate meshes and larger landmasses, which in turn form the entirety of the perceived natural landscape (PICTURE 23). The whole composition fits into place from a precise angle, turning this into a type of anamorphic illusion.



PICTURE 23. The stretched landscape and the same mesh group with added elements to form the complete scene. (writer's own picture)

I achieved a sense of structural stability of the composition through a collection of rocks, boulders, and mountains. The main concern was the possible risk of these heavy elements being destructive. I avoided the risk by applying appropriate colors, thus blending them with existing landmasses. The beige tone represents the foreground and matches the placement of the globe segment on the second-floor mural, in a way making such heavy objects appropriate even in the context of all murals combined.

Within the Blender Eevee engine, I used a built-in HDR and a considerable variety of light sources such as point light, sunlight, spotlight, and area light. I did this to replicate real-life lighting conditions (PICTURE 24).

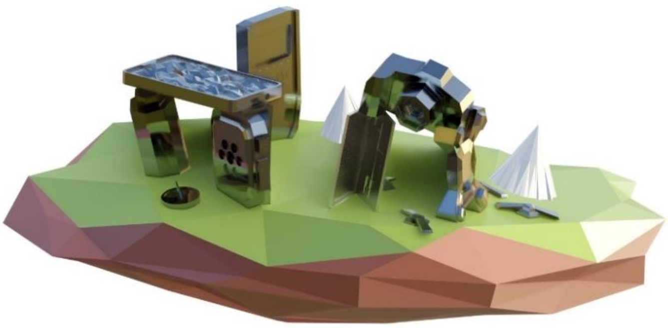


PICTURE 24. All light sources used within the Blender 2.81 interface

A crucial element of the foreground, which extends naturally towards the observer, is a large tree acting as the third-floor indicator. The shapes of its trunk and top branches allowed for native integration of anamorphic illusion (PICTURE 25).

PICTURE 25. Digital renders of the Yggdrasil tree from different angles (writer's own picture)

The isolated island done in the same low poly style seems out of place in context to the rich natural scenery. This Stonehenge- like formation is a type of monument worshipping artificial icons. Cone-shaped white deposits scattered throughout the island are the leftover natural resources used for mass-production, the third element- lithium. Products such as a smartwatch, a smartphone, a laptop, a drone, a digital camera, and a calculator depend on it (National Geographic 2019). Even though these products establish constant virtual connectivity within the team, the risk of this turning into an isolationist dependency remains. The reflective metallic surface finish applied to all products (PICTURE 26) amplifies the notion that they are merely a simplified reflection of the real world.



PICTURE 26. The isolated island in the 3rd ELEMENT 3D scene featuring different types of products based on lithium and lithium-ion batteries (writer's own picture)

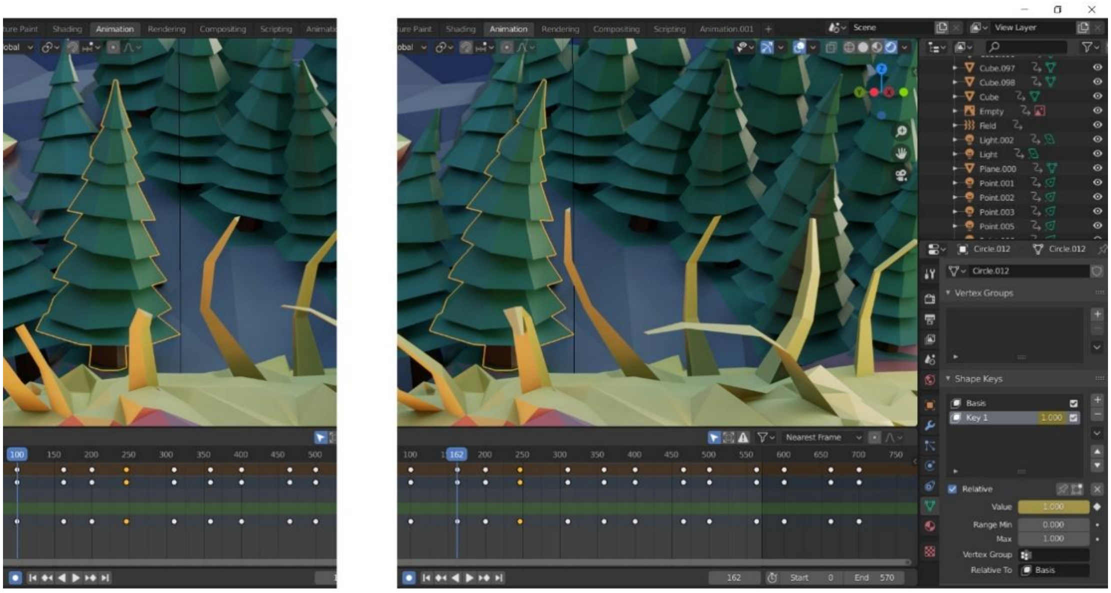
The 3rd ELEMENT mural artwork (PICTURE 27) acts as a window into the mind of an individual amidst one's search for their peace of mind and reflection on prior and current working experiences within a team or in a startup community.

PICTURE 27. 3rd ELEMENT mural in the stairwell (writer's own picture)

5.2.3 Creating the Animated Sequence

I divided the animated storyline into three scenes. The first scene brings life to the mural artwork and establishes the presence of technology in our daily life. The second scene presents the title and result of such dependency. The third scene shows a different outlook on technology through a passage of time.

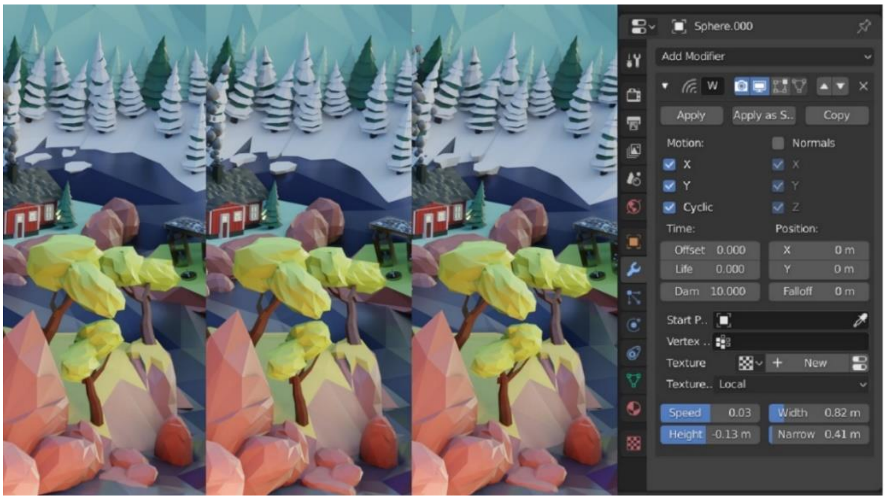
The lack of characterization and dynamic movement establishes a case in which every element needs to be in subtle motion at all times. One way is to rig and weight paint each element of the scene. The other way is to add different blend shapes, also known as morph targets. I chose the latter approach, as it would ensure an efficient workaround.



PICTURE 28. The fir mesh in its original state on the left at frame 100 and the fir mesh toggled fully towards the swaying target on the right at frame 162

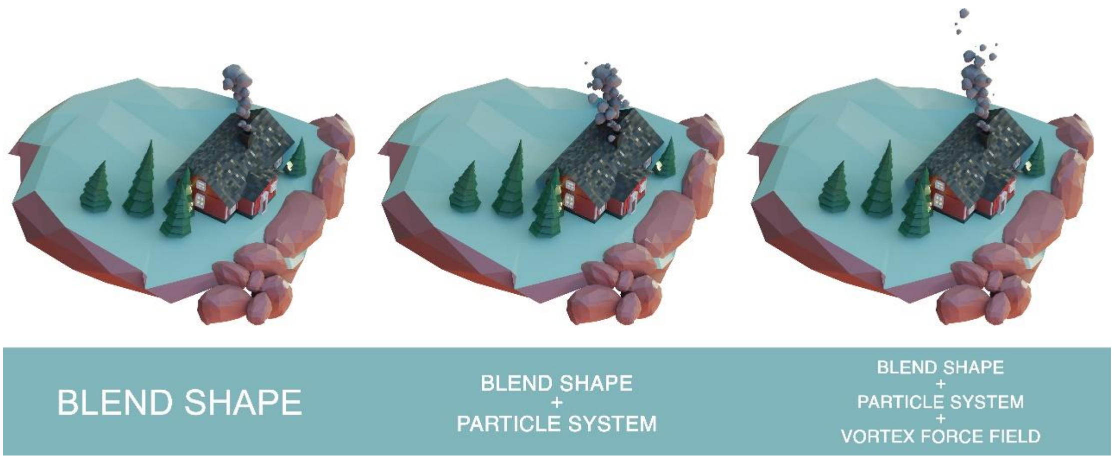
By applying a blend shape to each natural element in the first scene, I was able to replicate a summer wind. I would differentiate keyframe spacing and blend shape values to create a unique rhythm for the motion of each element (PICTURE 28). Thus, the forests and grassland would gain a sense of realism and physicality. The animation would coincide with audio of a windy forest ambiance in summer.

I used a feature called Wave modifier to animate the water as it allows for a modifiable fulfillment of the water flow motion (PICTURE 29).



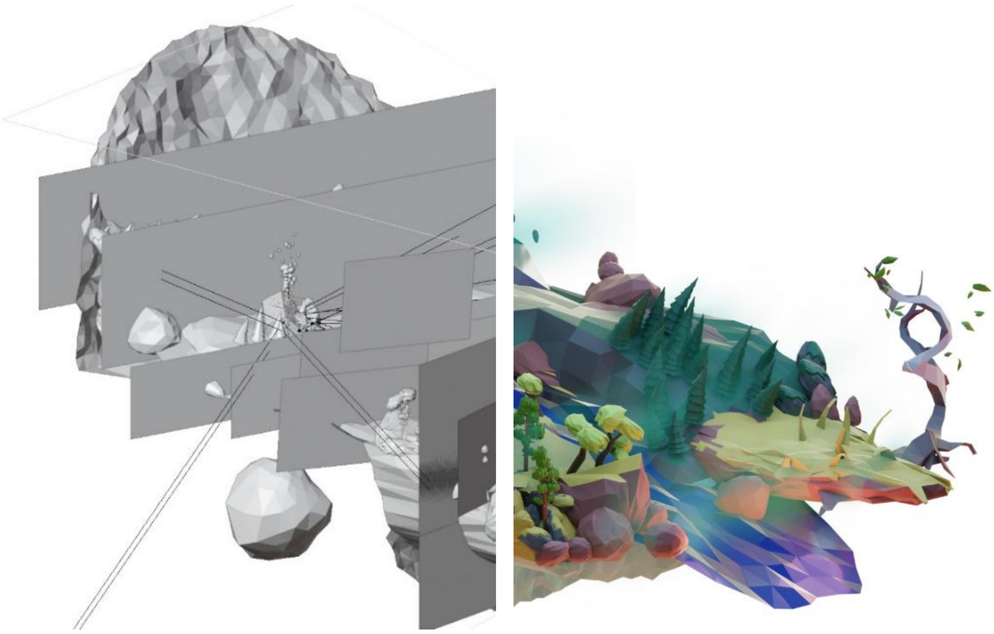
PICTURE 29. Screenshots of the water level difference between frames 1, 20 and 40

The last touch in bringing the scene to life was the animation of the smoke, which comes from the summer cottage chimney. I used a combination of different approaches involving blend shapes, a particle system, and a vortex force field (PICTURE 30). The base motion of the smoke involved changing the blend shape values and keyframe distances. To better replicate the dissolving effect of the smoke, a separate smoke particle model was created and put into action by a particle system. Courtesy of the vortex force field, the smoke could flow in a randomized upward motion.



PICTURE 30. Visual steps towards accomplishing the sought for randomized upward flow of the chimney smoke (writer's own picture)

The light green fog that engulfs the natural landscape would symbolize a clouded or misdirected mind. As a precursor to a digitalized world, the fog would fill the scene in its entirety. I considered using Blender smoke simulations, which turned out unfeasible due to the increased rendering time. The solution involved plane meshes, layered in a row within the 3D scene. I would apply fog textures with transparency to them (PICTURE 31). As a result, the animation render would not be lengthy or need a lot of post-processing in After Effects.



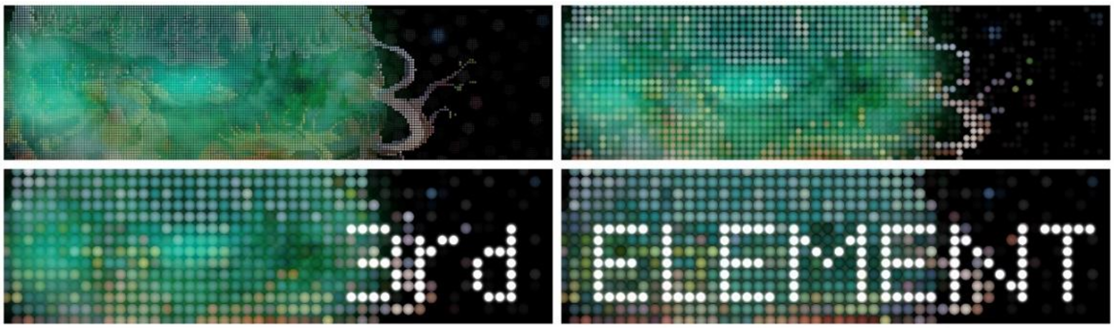
PICTURE 31. The plane mesh alignment screenshots in solid mode and with the fog texture applied as seen in the render preview mode

PICTURE 32. Technological products emitting a white ominous glow (writer’s own picture)

This fog increases the visual isolation of the central island, at the cost of the remaining natural landscape, which sets the stage for technology's leading role in a complete digitalization of the world. All technological products act as indexical signifiers through an ominous white glow (PICTURE 32). Indexical signifiers are indicators that infer on a turn of events, for instance, a warning (Collington 2016, 19). These could also be phonetic identifiers, in my case, the added computer sound effects.

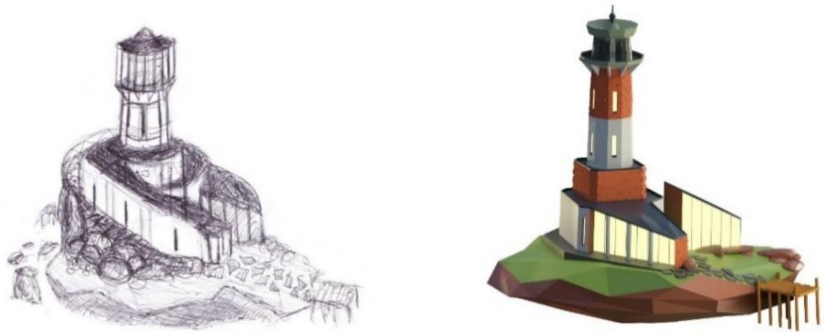
To create the glow effect, I drew over specific areas of these products in Photoshop. I imported this as one PNG file into After Effects to add the glow and gaussian blur effects. So far, both visual and phonetic signifiers have prompted the mental concept of the signified, being the complete digitalization of the scene, but it was time to visualize this transition. The main reference point for my approach was the RGB color mode motif, which coincides with LED technology. LED is used in a wide variety of digital screens. To replicate an LED screen, I used the CC Ball Action effect on top of the footage. This effect creates a smart circular grid and, by adjusting the grid spacing, I could change the scale of the LED screen. I used this feature to signify the change and transition into the next scene.

Scene two is the digitalized environment, represented by a scrolling 3rd ELEMENT title, created in Photoshop (PICTURE 33). A sense of misdirection is further emphasized by using all prior hardware and software sounds.



PICTURE 33. The transition into the 3rd ELEMENT LED title (writer’s own picture)

Scene three presents the last stage of the animated storyline. It delves into a completely different outlook, one in which technology is a source of inspiration and clarity. I imagined a guiding light that would shine warm and bright across an ambient natural landscape amidst the night. Since the real-life location of the Aalto University campus area is so near the sea, a lighthouse seemed most reasonable for this purpose. The resulting concept sketch and 3D model (PICTURE 34) encompass architectural elements from the main building of Aalto University's Undergraduate Centre and the Harmaja lighthouse.



PICTURE 34. Lighthouse sketch on the left and 3D model on the right (writer's own picture)

I would repurpose the existing Blender scene into the final animation with some adjustments. First and foremost, I replaced the models on the isolated island with the lighthouse, and the nighttime scene came to be by reducing the intensity of the HDR. The removal of the light-green fog provides a different outlook. This also formed the RGB color model motif through the beige foreground, the evergreen forest, and the now clear blue sky at the horizon.

PICTURE 35. The last frame of the 3rd ELEMENT animated sequence as seen in the completed product: https://youtu.be/-phMWB_9NyE (writer's own image)

I then imported the resulting image sequence into After Effects, where the only post-processing actions involved adjustments to the glow of the rotating 3D light beam and the lens flare of the rotating lighthouse lamps. With that, I completed the final scene of the 3rd ELEMENT animated sequence (PICTURE 35).

5.3 Fourth Dimension

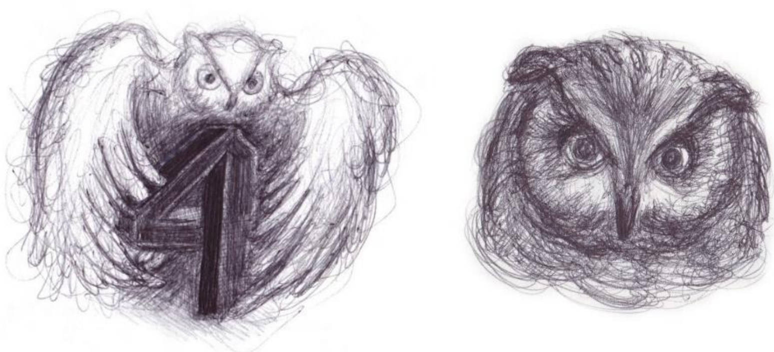
PICTURE 36. 4th DIMENSION mural illustration (writer's own picture)

4th DIMENSION is the name for the fourth-floor level and the mural artwork leading to it in the stairwell (PICTURE 36). It shows the expanding space filled with planets, stars, galaxies, and the owl nebula with its wings enveloping the clearest floor level indicator.

5.3.1 Concept

So far, the surrounding space has remained a backdrop for the foreground. To build up space as a character within itself I would focus on an element within the cosmos. Nebulae are often familiarized from the Earth's perspective as animals and the owl nebula is no exception (National Geographic 2018). The owl itself symbolizes knowledge, wisdom, wealth, and extrasensory perception (Macrameowl 2017). The concept sketches of an owl nebula follow a realistic approach (PICTURE 37).

PICTURE 37. 4th DIMENSION owl nebula concept sketches (writer's own picture)



The concept of the fourth dimension is ambiguous for what it represents. In the context of natural science, one could describe it as space where past, present, and future meet, thus rendering these terms as being useless, which is relatively paradoxical. The floor level indicator would reflect the paradoxical nature of the fourth dimension. On the other hand, one could also interpret it as an added physical depth. I would highlight the latter interpretation by simulating a model of the solar system. Accurate placements of each planet, from the perspective of Earth, would add a sense of depth.

5.3.2 Illustration

Beyond the prerequisite of the 4th DIMENSION mural artwork featuring space, there were no stylistic recommendations, which allowed for experimentation within the sphere of digital painting. Unlike the previous 3D approach, I worked in a 9870 by 2880 pixel resolution at 400 pixels per inch in Photoshop, which ensured an optimal workflow and printing quality. I would retain the digital painting aesthetic, meaning that I would not replicate the traditional painting look. Keeping this illustration within the merits of built-in Photoshop brushes reinforces the notion of a functional symbiosis between the real and the virtual without one or the other having to take over.

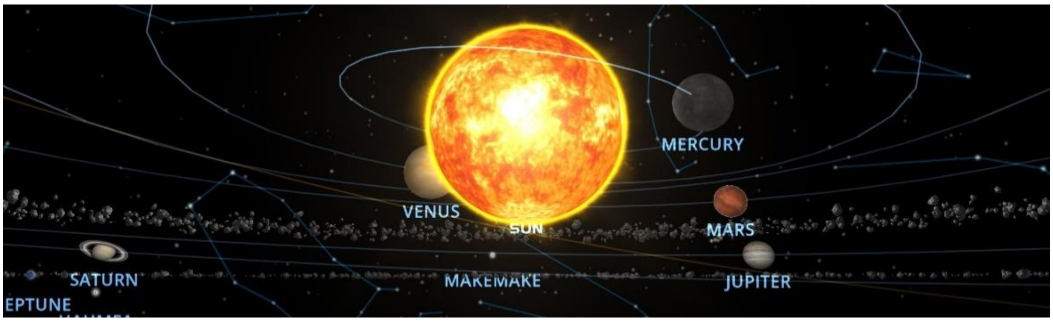
PICTURE 38. A 3D preview of all Earth layers, based on the Photoshop layer list alignment (writer's own picture)

I started with the upper hemisphere of planet Earth, which I divided into six layers (PICTURE 38). On the very top of the alignment sits the sixth layer, which emphasizes the glow of the sunset at the horizon by utilizing the screen blend mode. A blend mode gives the ability to manipulate the pixels of images within specific layer combinations or blends. The screen blend mode projects a lighter color, based on the top layer, onto all layers below (Adobe 2020). In practice, the orange glow affects both the Sun and Earth. In the very middle, four layers feature various aspects of the Earth's landscape and atmosphere. The first layer, at the very bottom, serves as the fill layer for the Earth globe.

Generally, it is a good idea to keep the layer-based approach to its minimum, which goes for Photoshop or any other free alternative program, including Krita. Having the potential for change or adjustment in mind, by working in layers, ensures a non-destructive workflow.

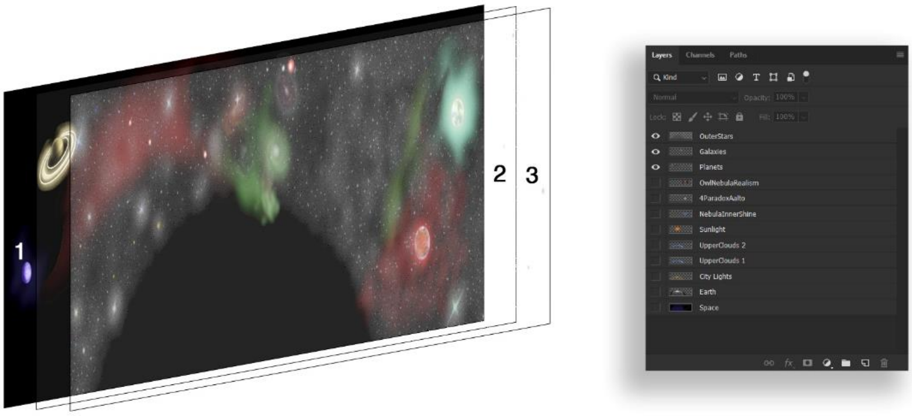
PICTURE 39. A 3D preview of all owl nebula layers, based on the Photoshop layer list alignment (writer’s own picture)

I only used three layers to build up the owl nebula illustration (PICTURE 39). The third layer includes the general features of a real owl, including the head, body, and large wings. The second layer features the floor level indicator in line with the Penrose Triangle model, which is a type of impossible optical illusion supporting the paradoxical nature of the fourth dimension. The blue glow effect on the first layer completes the nebula representation.



PICTURE 40. Screenshot of the Solar System Scope app in action

I approached the accurate placement of visible planets from creative and contextual standpoints. Photographic material by Nasa and a real-time interactive solar system model by Solar System Scope (PICTURE 40) were primary references. This way, I could better replicate individual planet textures and pinpoint planetary positions. I completed the mural artwork by drawing all visible planets, stars, and galaxies on three layers as well (PICTURE 41).



PICTURE 41. A 3D preview of all space layers, based on the Photoshop layer list alignment (writer’s own picture)

When illustrating the owl nebula and the floor level indicator, I added minor distortions for the anamorphic illusion to take place. As a result, the owl nebula seems to gaze upon the observer from all angles (PICTURE 42). In the case of the indicator, it retained three-dimensionality within the interior (PICTURE 43).

PICTURE 42. Digital renders of the owl nebula from different angles (writer's own picture)

PICTURE 43. 4th DIMENSION mural in the stairwell (writer's own picture)

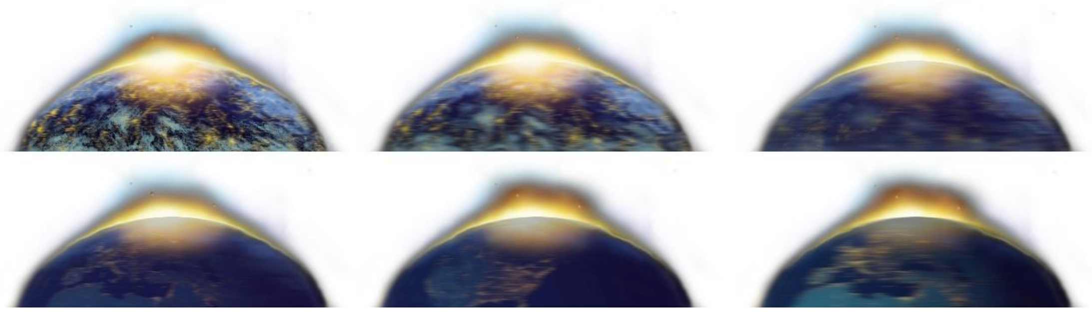
5.3.3 Creating the Animated Sequence

A conclusive act in a series of story-driven animations requires closure and a sense of grandeur. I approached this by working within 2D and 3D, through Photoshop, After Effects, and Blender. This way, I would bring life, depth, and resolution to the 4th DIMENSION. Unlike two previous animated storylines, this one takes place in one continuous scene. To bring life to the mural artwork, all objects such as the Earth, all remaining planets, owl nebula, and galaxies, were set into motion.

Work began on subtle animations, such as the heat waves at the horizon caused by the intense sunlight. There is noticeable distortion on the right side of the hemisphere that replicates this phenomenon in a still image format, as seen in chapter 4.4. To contextualize this distortion, I duplicated the Earth layer and created two identical Ellipse masks, one inverted and the top one unchecked. Feathering on these masks ensured that the Turbulent Displace effect would only affect the curved edges of the hemisphere (PICTURE 44).

PICTURE 44. Turbulent Displace effect in action within After Effects

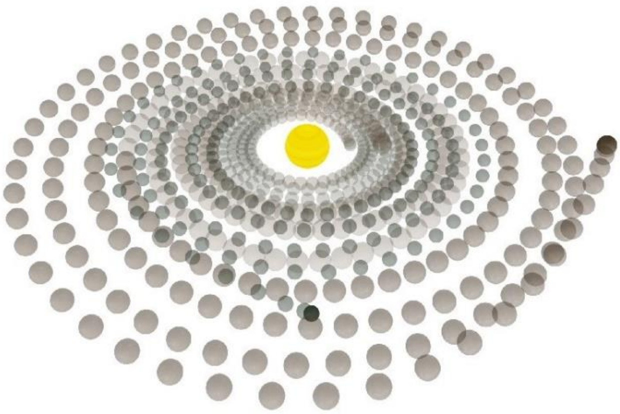
The more dynamic aspects of the animation process involved Earth spinning counterclockwise. The issue revolved around the inability to use the original Earth illustration for this exact purpose. Creating additional digital paintings from different sides would be inefficient, hence why I resorted to a functional effect within After Effects called CC Sphere. It allowed me to turn a 2D map of the world at night into a 3D Earth globe with rotation controls. An additional directional blur effect emphasized the speed of the rotation and created a smooth transition between the illustration and the CC Sphere (PICTURE 45).



PICTURE 45. Six transitional steps showcasing the directional blur effect working alongside the CC Sphere effect. (writer's own picture)

Moving beyond a rotational loop of planet Earth would involve setting the entirety of the solar system into motion. At first, I used base transform tools within After Effects to animate Saturn, Neptune, Mars, and Mercury. I moved beyond into 3D planetary movement through Blender. Analyzing the Solar System Scope interactive model led to the realization of two planetary movement groups. In my case, the first one focuses on the Sun. The second one focuses on Earth as it is from this view that the audience explores the solar system. Both groups contained Moon, Mercury, Venus, and Sun 3D models, courtesy of Nasa Solar System Exploration Resources (2019).

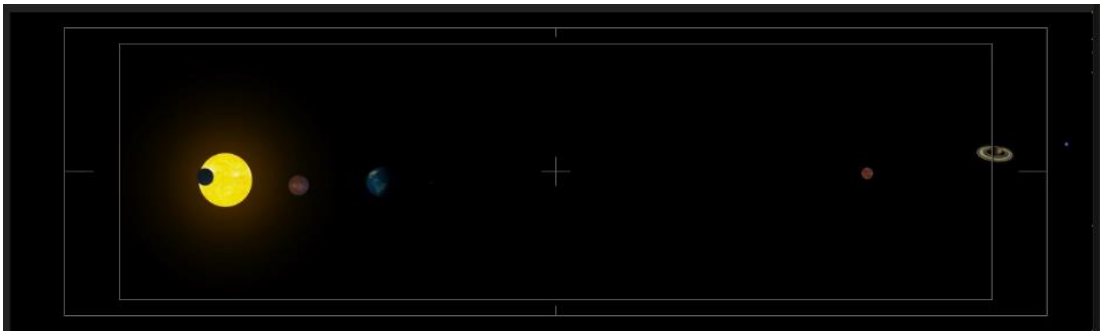
PICTURE 46. All 350 frames of the first planetary movement group shown simultaneously. (writer's own picture)



I wouldn't replicate the exact scale of each planet for viewing purposes. Instead, I adjusted the exponential distance in-between, as seen in the first planetary movement group (PICTURE 46). This way, I approached a more accurate solar system representation that references the fact that the universe is expanding.

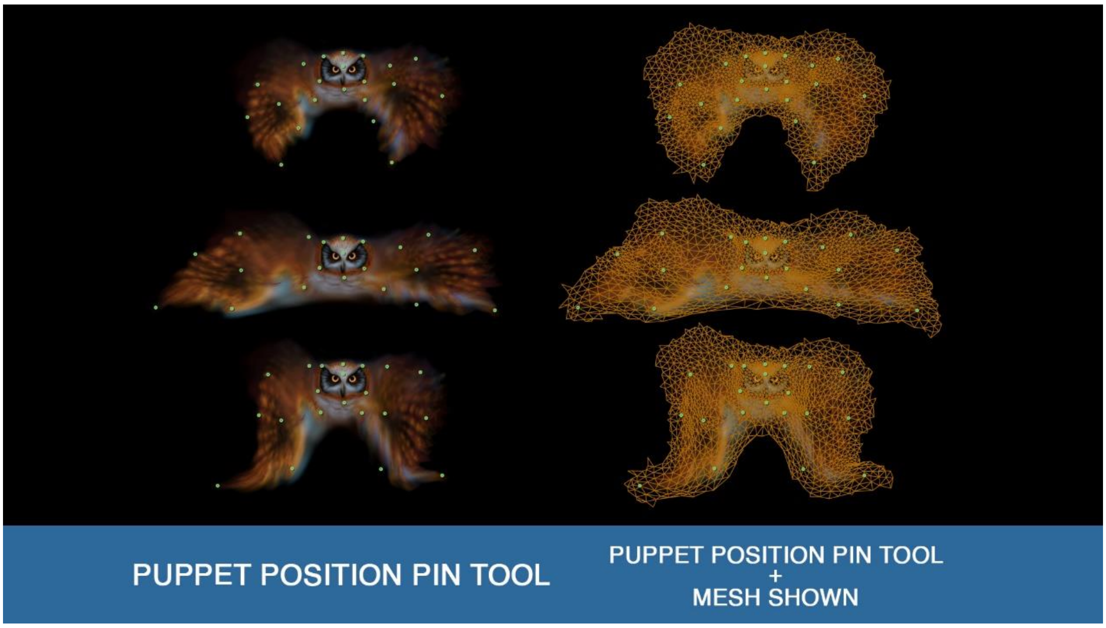
For the Sun texture to emit its light onto Mercury and Venus, I rendered this animated sequence using the Cycles render engine. The second planetary movement group did not require more realistic lighting, hence why I used the faster Eevee render engine. The second planetary movement group contains Earth and its Moon. Due to a type of 3D Earth model already present within After Effects, I would only render the Moon in Blender.

Having done all the necessary work in Blender, it was time to incorporate these image sequences into the After Effects scene. I came up with a simple solution for a smooth transition into the third dimension. The sunset would be an indicator for the appearance of all the planets from behind Earth, which in turn would prompt an expansion into the cosmos, finally closing in on a more accurate representation of the solar system and its scale (PICTURE 47).



PICTURE 47. The solar system (writer's own picture)

Animating the owl nebula would be the next logical step when going up the ascending cosmic scale. The Puppet Position Pin Tool, found in After Effects, allowed me to turn the owl nebula illustration into a mesh. The twenty-one advanced puppet pins would deform the area of the assigned mesh. This way, I was able to create a realistic owl flight animation cycle (PICTURE 48).

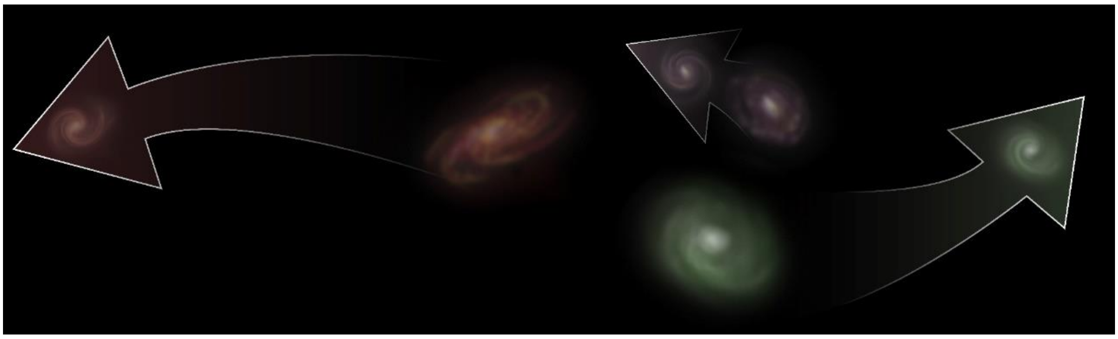


PICTURE 48. Key poses of the owl flight animation cycle with a preview of all twenty-one pins used to deform the generated mesh as seen on the right in After Effects

The Puppet Position Pin Tool is a notable feature in After Effects unmatched by a free video editing software such as Da Vinci Resolve. Unity's 2D Animation toolsets are the closest free alternatives that I had in mind. The advantage of using After Effects has to do with the reduced dependency on a gaming engine to handle one animation cycle. This way, I could export it as an animated sequence without resorting to screen recording.

By illustrating a living owl in the nebula representation, I was able to add an animated blink sequence done in Photoshop. It would match the animated owl flight cycle by copying the puppet effect parameters onto the new owl blink sequence layer within After Effects.

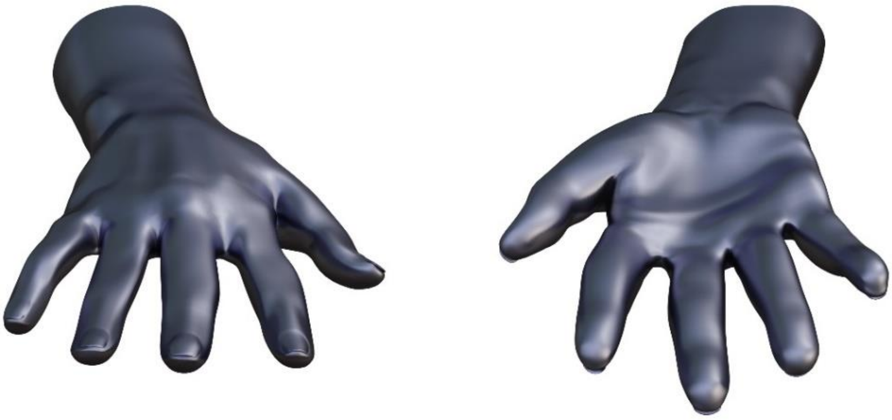
Even though nebulae are stargazing spectacles, they are merely subsets to galaxies. The way I would bring these unspecified galaxies to life was through the Puppet Position Pin Tool and the Twirl effect. One would deform an angled galaxy mesh so that the latter would generate spiral arms from its center. I represented the idea of an expanding universe through the increasing distance between said cosmic objects (PICTURE 49).



PICTURE 49. Directional arrows showing the path of each individual galaxy, the first and last keyframes of the sequence. (writer's own picture)

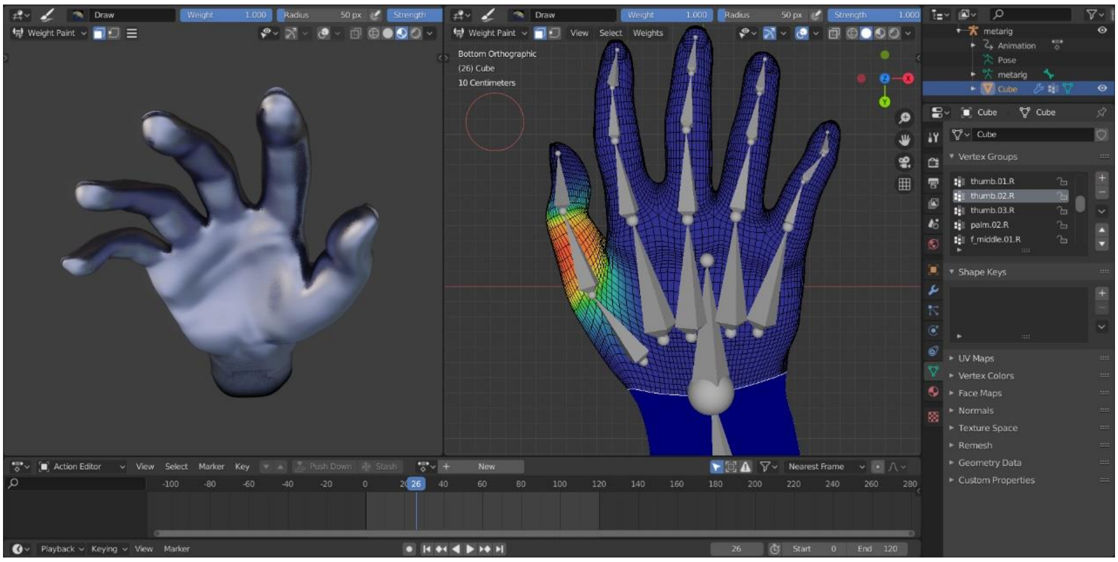
In chapters 5.1 and 5.2, an individual experienced fear and misdirection that clouded their mind. But now, clouds of the past are at the grasp of the creator moving beyond. To reinforce this notion, in visual form, I created a 3D hand model in Blender, based on my hand (PICTURE 50). To highlight an allinclusive aspect, I applied a metallic PBR texture, which also establishes a functional symbiosis between people and technology (PICTURE 51).

PICTURE 50. Four stages in creating the hand model in Blender 2.81



PICTURE 51. A preview of the completed metallic hand model rendered using Eevee render engine in Blender 2.81 (writer's own picture)

Although chapter 5.2.3 introduced blend shapes as an animation tool within Blender 2.81, the required movement for this realistic hand model is anything but subtle or linear. I utilized Blender's built-in human meta-rig for this very cause. After aligning the bones to the given model, I applied automatic weights to simulate real bone movement. These weights, highlighted in red, orange, yellow, green, and blue, in the order of influence, act on specific parts of the mesh (PICTURE 52).



PICTURE 52. A hand pose made possible after applying the meta-rig and automatic weights as seen within the Blender 2.8.1 interface

Now that the hand model was ready and rigged, it was time to add a unifying factor for all mural artworks. Merging the existing motifs into a tricolor globe would emphasize that connection. The hand animation needed to reflect a sense of confidence within a given sphere, hence the creation of the tossed tricolor globe animation (PICTURE 53).

PICTURE 53. Key poses of the tossed tricolor globe animation (writer's own picture)

To achieve the overarching resolution, planet Earth dissolves into the cosmic void, re-emerging as the tossed tricolor globe. During the final seconds (PICTURE 54), it transitions into the combined A Grid Aalto University mural composition (PICTURE 55). Thus, I present the viewer to the project-specific reinterpretation of Leonardo da Vinci's quote: *"Learn how to see. Realize that everything connects to everything else"* (Atalay & Wamsey 2008, 96).

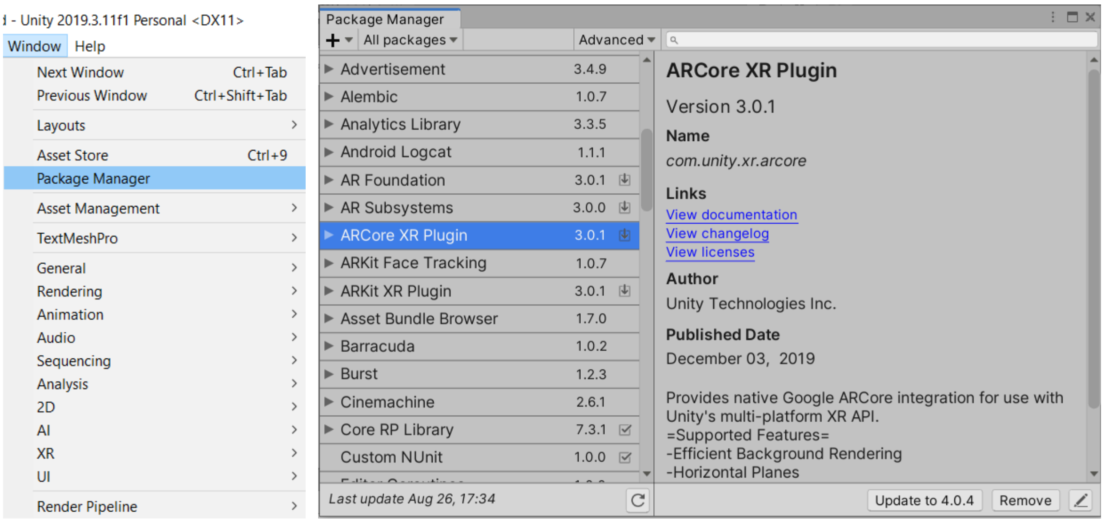
PICTURE 54. The last frame of the 4th DIMENSION animated sequence as seen in the completed product: https://youtu.be/cogv2bqF4qY (writer's own image)

PICTURE 55. All mural artworks combined into one (writer's own picture)

6 MOBILE AR APP IMPLEMENTATIONS

6.1 Using Unity 3D to Develop a Mobile AR App Prototype

First and foremost, it is important to choose the right project type when launching Unity. Thus, I created a new 3D project in Unity 2019.3.11f1. I installed a few free packages by accessing the Package Manager through the Window tab (PICTURE 56). A package contains features, both internal and external, that fit the various needs of any project (Unity 3D 2020). The ARCore XR Plugin, version 3.0.1, alongside the all-encompassing AR Foundation Plugin, version 3.0.1, allowed a potential transition onto iOS. The 2D Sprite Plugin, version 1.0.0, allowed me to turn custom designs into UI elements.

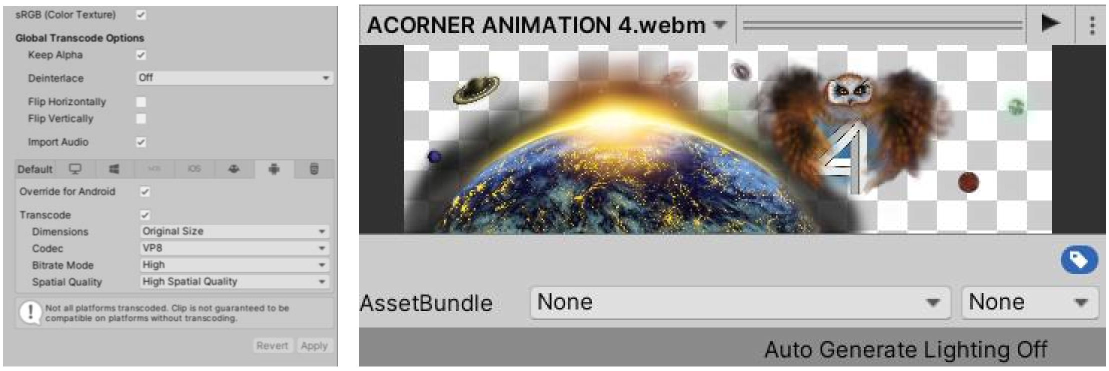


PICTURE 56. Screenshots showing how to navigate to the Packages

6.1.1 Optimizing and Importing Video Assets

I exported most of the animated sequences in the appropriate MP4 file format and 2560 by 752 px resolution. The H265 video codec wouldn't result in increased file sizes, as proven by the first two animated sequences being 34,1 and 32,6 megabytes. There was, however, the need to switch to a different video format for the 4th DIMENSION. It relied on background transparency, which is not supported by MP4. A feasible video file format, which is compatible with Unity, is the open-source WebM along with its VP8 video codec (Unity 3D 2020). Adobe Media Encoder that comes bundled with After Effects worked with the free external WebM open video plugin by Fnord. Thus, I resolved this initial issue, and the resulting video file size for such dynamic one-minute animation was just 16,4 megabytes.

When dropping these video assets into the Unity project, the recommendation is to transcode all files in the inspector. This way, allowing for proper optimization depending on the video format and target build platform (PICTURE 57).



PICTURE 57. Screenshots showing transcoding settings for 4th DIMENSION animated sequence with transparent background and result of doing so

As these three video files needed to coincide with the interior, I applied them as textures to simple 3D planes, with an inside corner. Furthermore, considering the accessibility factor brought up in chapter 4.4, I also imported a flat 3D plane model. Consequently, the videos would be viewable on the inside corner wall in AR and on the mobile device itself. This accessibility measure targets users with old or budget Android smartphones that do not work with ARCore.

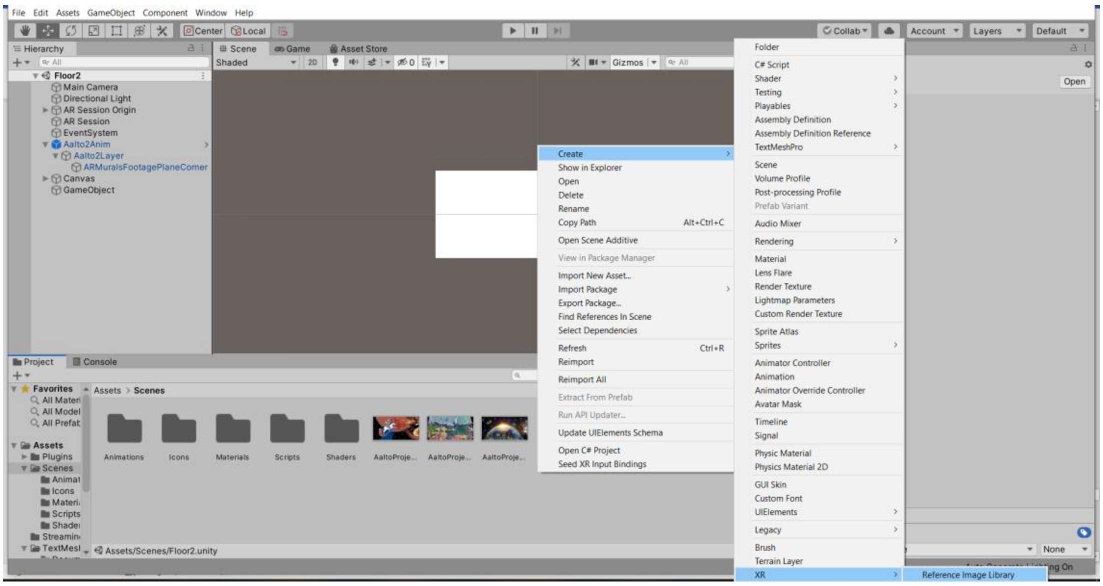
In practice applying video files as textures onto 3D objects involves adding a Video Player component to the model. I should note that if a video clip has transparency, only the basic materials in the Sprites category allow it to show up, although with no lighting. My intension was to present the contents of each video without any obstructions, for instance, gloss, so this was not an issue.

By adding the 3D model into the scene or hierarchy, I could play the video. At the same, doing so turns the model into a Prefab. Prefabs are systems that can store multiple child GameObjects, which was helpful for the 4th DIMENSION animation that is layer-based.

6.1.2 Developing a Working Prototype

In chapter 4.2, I brought up a potential image tracking obstruction caused by the 90 degrees inside corner walls in the stairwell. An easy solution to this issue involved only using the left side, which encompasses two-thirds of the canvas.

I would add all three cropped image trackers, featuring the mural artworks, into the Reference Image Library. It is accessed by right-clicking in the project amongst all assets, then Create, XR, and Reference Image Library (PICTURE 58). The AR Session Origin and AR Session need to be in the scene hierarchy for augmented reality to function. I accessed these the same way but within the scene hierarchy.



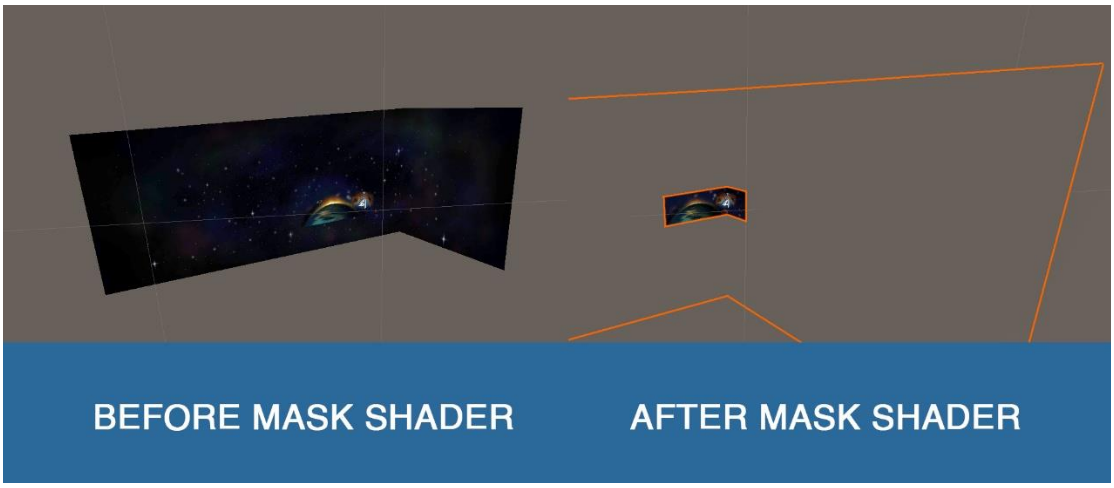
PICTURE 58. A screenshot within the Unity 2019.3.11f1 interface showcasing the process of accessing the Reference Image Library

The main issue revolved around the fact that a library of multiple image references only adhered to a single Prefab. For all intents and purposes, it wouldn’t matter which image target got scanned and recognized as only one animated sequence would be overlaid in augmented reality.

The only solution I could find, courtesy of Unity developer and programmer H. Ibrahim Penekli (2020), involved a Tracked Image Manager script. It allowed me to directly assign a Prefab, within a separate scene, to a specific image target found in the Reference Image Library (Appendix 1).

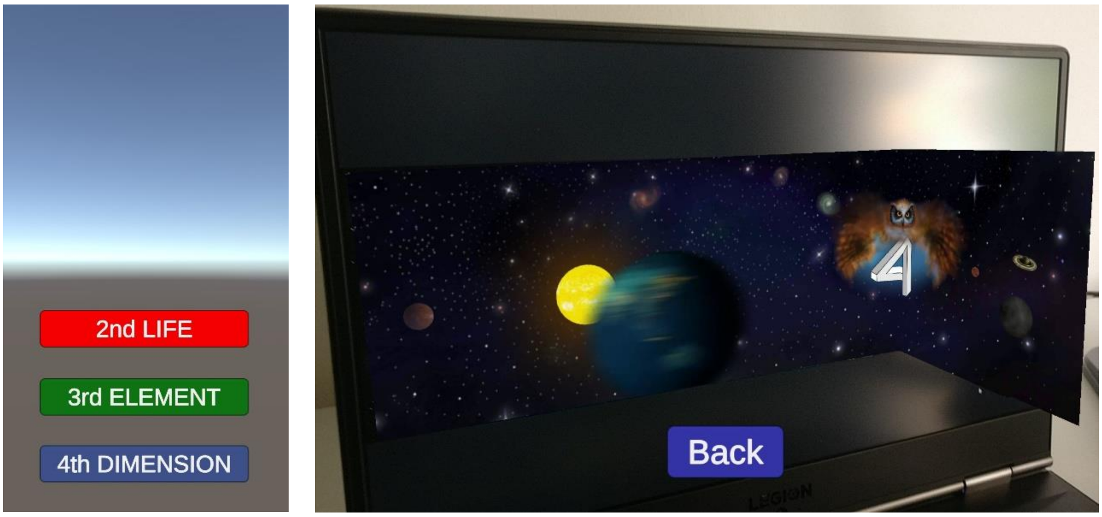
It was time for me to work on adding depth to the 4th DIMENSION. The resulting Prefab would consist of four layers, the video clip with transparency and three upscaled image layers aligned in 3D space, based on the three space layers featured by the end of chapter 5.3.2. To establish a parallax effect everything outside of the mural canvas boundaries had to be obstructed and masked.

First of all, I created an additional 3D plane, with a cutout for the canvas, that would obstruct all subsequent layers from the user’s field of view. Secondly, I would apply a mask shader, courtesy of Augmented Startups founder Ritesh Kanjee (2018). This mask shader (PICTURE 59) would mask out the walls of the obstruction along with anything behind them (Appendix 2).



PICTURE 59. Screenshots within the Unity scene showcasing the 4th DIMENSION Prefab before and after using the mask shader

To fully test out the functionality of the initial Mobile AR app prototype, a placeholder menu was created alongside a back button to manage all the separate scenes for each mural artwork (PICTURE 60).



PICTURE 60. Screenshot featuring the placeholder menu on the left and the screenshot of the 4th DIMENSION scene in augmented reality on the right

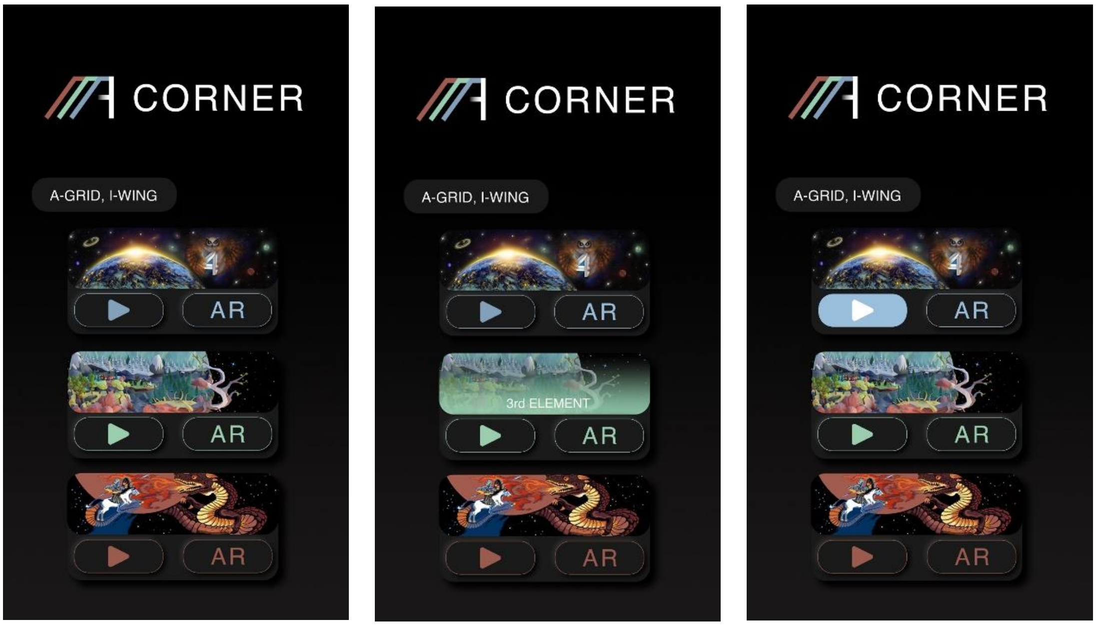
I employed two Scene Manager scripts, courtesy of Brackeys CEO and founder Asbjorn Thirslund (2017). These scripts allowed the user to click on any of the three buttons corresponding to each separate animated sequence in AR (Appendix 3).

6.1.3 Designing the UI and UX

When coming up with a minimal design for the user interface, on top of the ready-made placeholder, I referenced the Aalto Space mobile app. With it, one could find and book different facilities for Aalto University students and staff (Aalto 2019).

The resulting UI design (PICTURE 61) would emphasize the idea of this mobile application showing the contrasting virtual side of Aalto University facilities. In a way, it would expand the given Aalto space, starting from A Grid and continuing with other animated works within the Aalto University interiors in the future. As a way to establish an identity for my Mobile AR app prototype, in concept alongside or within the existing Aalto Space app, I named it A

CORNER. The logomark, featuring a triple-A, is an abbreviation for A Grid Aalto AR and a color-coded reference to the existing murals with AR functionality. The supporting tagline *"Energizing your experience"* highlights the entertainment aspect of the app, its ability to power one's day, and inspire through audiovisual means.



PICTURE 61. Screenshots of my Mobile AR app UI in action with animated mural title highlight option and clear sprite swap for each pressed button

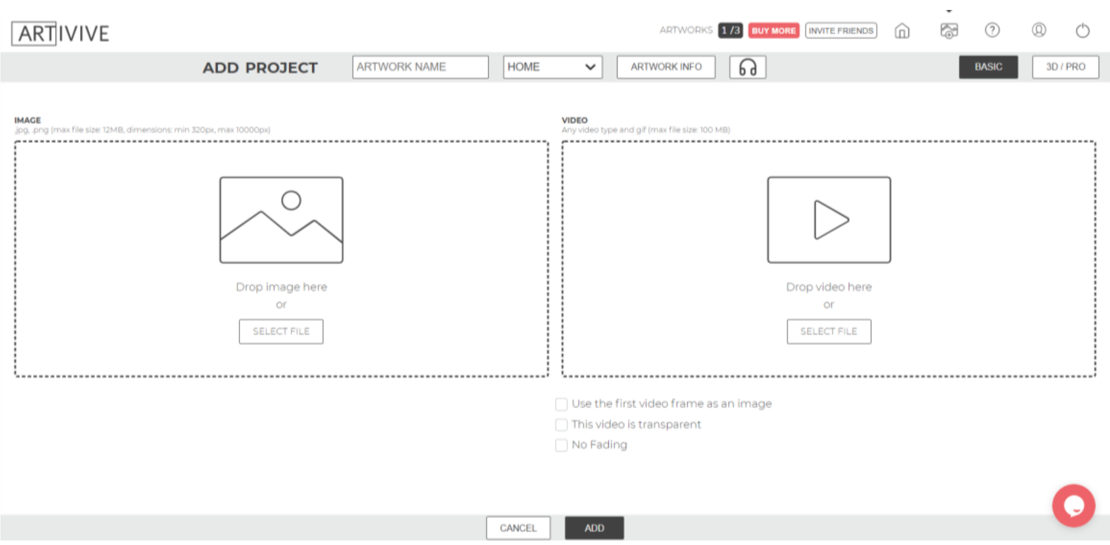
PICTURE 62. Pinch to zoom feature in action with the 3rd ELEMENT animated storyline playing (writer's own picture)

User experience could not be limited to high-end smartphones, which is why I applied the same video clips onto flat 3D planes within three additional Unity scenes that don't rely on AR functionality (PICTURE 62). The content could also be viewed interactively on a user's smartphone display (Press Start 2018). The PanZoom script allowed the user to zoom in and out two of the animated sequences in an orthographic view. The PerspectivePan script allowed the user to see the parallax effect in action. As a result, I was able to utilize this interactive approach in delivering a similar experience, albeit condensed, accessible to everyone (Appendix 4).

My A CORNER Mobile AR app prototype successfully delivers both experience modes. The completed Android build was 119 megabytes (Appendix 5).

6.2 Using the Artivive Bridge Tool

To gain access to the Artivive online content management tool Bridge, I created a free artist account that was pre-packed with three project spaces. I should note that unlike more established augmented reality companies such as Eyejack and Blippar, there is no set view limit. This fact alone provided a great deal of relief in the longevity of this implementation.



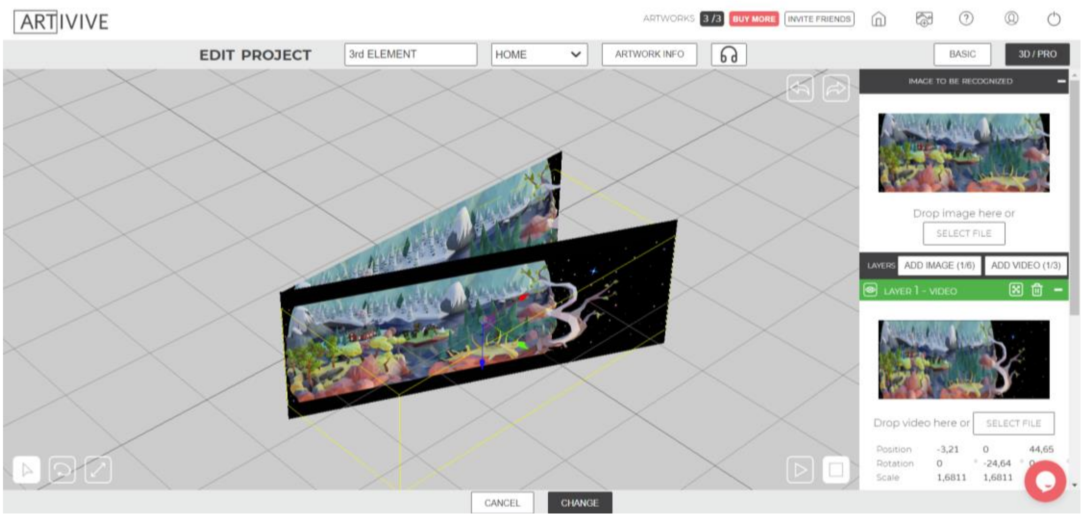
PICTURE 63. The minimal file import interface as seen within Bridge

The file importing process takes place in a minimal and intuitive interface (PICTURE 63) with the image tracker drop area on the left and video content drop area on the right. Fortunately, little to no optimization needed to be done when importing all the animated sequences, as 100 megabytes per video was more than enough.

6.2.1 Organizing the Virtual Content

Switching to the 3D/ PRO content managing mode, I initially hoped to replicate the physical corner wall by combining two parts of each video clip, as Bridge does not support 3D models in their full-scale content manager. This approach turned out ineffective as there was a significant latency of the video clips due to the video content streaming from a server at different internet speeds.

I realized that the most efficient approach would be to place a singular video clip diagonally to the tracked mural artwork (PICTURE 64). This way, there is a natural progression that still stems from the corner wall.



PICTURE 64. A whole video clip placed diagonally in Bridge 3D/ PRO

By downloading the Artivive app and using my mural artworks as image targets, at any scale, anyone could experience these animations.

7 CONCLUSIONS AND DISCUSSION

The two project deliverables provide a great deal of evidence that supports the idea of open availability of toolsets. All of which are relevant to augmented reality and mobile app development. These toolsets allowed me to implement three animations with a story-driven progression appropriate for the A Grid Aalto startup community, even on a near-zero budget in Mobile AR.

The permanent installation of the mural artworks and the confined space of the stairwell excluded the need for a type of augmented reality implementation that would be anything else than tracker based. In this particular case, I was able to retain a fundamental level of spatial awareness through practical approaches in dealing with the initial corner wall tracking issue. I did this by replicating the physical shape of the installed mural artwork in 3D or by having it appear diagonal, thus turning the corner wall into sides of a right-angle triangle shape.

I was successful in bringing the initial concepts to life and carrying out the main objective of this thesis. The working A CORNER Mobile AR app prototype obtains a level of exclusivity to the A Grid Aalto campus. It delivers a more believable Mobile AR experience alongside an interactive video experience that does not rely on AR functionality for accessibility reasons, without the need for an internet connection. However, there's room for improvement. Currently, all assets depend on the local storage, which may halt the app performance.

In my case, paid for cloud services or servers act as the sole technical barrier. They could reduce the size of the app and stabilize the performance by streaming these video clips remotely. The resulting stability can be seen, for instance, in the Artivive app. Even though it is far more limiting from a technical aspect, it delivers a more stable augmented reality experience as the online content manager tool Bridge handles most of the heavy lifting.

The potential post-production developments go beyond the scope of this research project. They depend on a possible playtesting session at A Grid Aalto campus, which would determine the preferred Mobile AR implementation method. Furthermore, evaluate the feasibility of my app deployment suggestions revolving around the integration of the A CORNER app prototype into the existing Aalto Space app, or as a published standalone. In both cases, the existing servers employed by the client to store and stream the animations would be utilized, thus resolving the performance issue. As a means to clarify and redirect the user to the real and virtual contents of the mural artworks, there would be an installation of three wall plaques. They would summarize the overarching storyline and feature a unique QR code with a download link to the chosen Mobile AR implementation.

In the long run, the thought of my work helping to inspire and empower the A Grid startup community as well as establish a foundation for subsequent creations by others, in vain with a more compelling visual storytelling approach accessible to anyone, within the Aalto University campus area, is unreal and I'm thankful for this opportunity.

REFERENCES

Aalto. 2019. Aalto Space mobile application. Read on 20.07.2020. https://www.aalto.fi/en/services/aalto-space-mobile-application

Agrid. 2020. Startup hub A Grid grew to a community of 150 startups and partners. Read on 11.06.2020. https://agrid.fi/startup-hub-a-grid-grew-to-a-community-of-150-startups-and-partners

Ariel, G. 2017. Augmenting Alice: The Future of Identity, Experience and Reality. Amsterdam: BIS Publishers. Read on 16.06.2020. Requires access right.http://libproxy.tuni.fi/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,uid&db=e000xww&AN=2002028&site=ehost-live&scope=site

ARCore. 2020. Discover: ARCore overview. Read on 09.07.2020. https://developers.google.com/ar/discover

Artivive. 2020. Albertina masterpieces at your home - Artivive App. Read on 24.08.2020. https://artivive.com/albertina/

Atalay, B., Wamsey, K. 2008. Leonardo's Universe: The Renaissance World of Leonardo Da Vinci. 1st Edition. Washingon: National Geographic Books.

Boland, M. 2019. ARCore Reaches 400 Million Devices. Read on 24.08.2020. https://arinsider.co/2019/05/13/arcore-reaches-400-million-devices/

Byrne, S. 2016. Play me a story: How video game storytelling has evolved. Read on 07.12.2019. https://www.cnet.com/news/video-game-storytelling-evolution-bushnell-schafer-morhaime/

Casserly, M. 2019. iPhone vs Android market share. Read on 25.08.2020. https://www.macworld.co.uk/feature/iphone/iphone-vs-android-market-share-3691861/

Collington, M. 2016. Animation in Context: A Practical Guide to Theory and Making. London; Bloomsbury Publishing Plc.

Cronin, I., Scoble, R. 2020. The the Infinite Retina: Spatial Computing, Augmented Reality, and How a Collision of New Technologies Are Bringing about the Next Tech Revolution. Birmingham, England; Mumbai: Packt Publishing, Limited. Read on 16.08.2020. Requires access right. https://learning.oreilly.com/library/view/the-infinite-retina/9781838824044/?ar

Forbes. 2018. Augmented Reality Will Reinvent How We Experience Art. Read on 23.08.2020. https://www.forbes.com/sites/alicebonasio/2018/06/20/augmented-reality-will-reinvent-how-we-experience-art/#25e5646e636c

Handrit. n.d. AM 738 4to. Read on 05.09.2020. https://handrit.is/en/manuscript/imaging/is/AM04-0738#page/42v++(95+of+293)/mode/2up

Kanjee, R. 2018. Augmented Reality Portal using ARCore and Unity - AR Portal Tutorial. Watched on 15.04.2020. https://youtu.be/BeoY5mRm1UY

Kastrenakes, J. 2017. Google's Project Tango is shutting down because ARCore is already here. Read on 24.08.2020. https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality

Macrameowl. 2017. Owl Symbolism. Read on 04.08.2020. http://macrameowl.com/owl_symbolism.html

Makarov, A. 2020. ARKit vs ARCore: Image Detection and Tracking. Read on 25.08.2020. https://arpost.co/2020/06/12/arkit-vs-arcore-image-detection-and-tracking/

Muybridge, E. 1887. Sequence of a race horse galloping. Read on 21.04.2020. https://commons.wikimedia.org/wiki/File:Muybridge_race_horse_gallop.jpg

National Geographic. 2018. The Giant Squid Nebula and More Animals of the Cosmos. Read on 04.08.2020. https://www.nationalgeographic.com/news/2018/03/galaxies-space-animals-astronomy-cosmos/

National Geographic. 2019. This metal is powering today's technology—at what price? Read on 30.06.2020. https://www.nationalgeographic.com/magazine/2019/02/lithium-is-fueling-technology-today-at-what-cost/

New York Public Library. n.d. Knight, 1200. Read on 07.11.2019. https://nypl.getarchive.net/media/knight-1200-b5e28e

Novel. 2018. GOOGLE I/O 2018. Read on 25.08.2020. https://novel.af/project/io2018

O'Regan, G. 2018. World of Computing: A Primer Companion for the Digital Age. Cham: Springer International Publishing AG.

Oriz, E., Geig, M. 2019. 2D Pixel Perfect: How to set up your Unity project for retro 8-bit games. Read on 26.06.2020. https://blogs.unity3d.com/2019/03/13/2d-pixel-perfect-how-to-set-up-your-unity-project-for-retro-8-bits-games/

Papagiannis, H. 2017. Augmented human: how technology is shaping the new reality. 1st edition. Sebastopol, California: O'Reilly Media. Read on 14.04.2020. Requires access right. https://learning.oreilly.com/library/view/augmented-human/9781491928363/?ar

Press Start. 2018. Unity tutorials: Free Unity tutorials on cameras. Watched on 29.08.2020. https://pressstart.vip/tutorials/camera

Saini, R. 2020. Catching up with AR. Read on 23.08.2020. https://gfxspeak.com/2020/07/09/catching-up-with-ar/

Statcounter. n.d. Mobile Operating System Market Share Worldwide. Read on 25.08.2020. https://gs.statcounter.com/os-market-share/mobile/worldwide

Stewart, J. 2019. This Street Artist Transforms Cement Blocks into Mind-Blowing Illusions. Read on 09.10.2020. https://mymodernmet.com/odeith-illusion-art/

Taylor, C. 2016. Literature of the Middle Ages in Translation: Mysteries of the Grail. Read on 05.09.2020. https://scalar.usc.edu/works/mysteries-of-the-grail/the-tree-of-life

Thirslund, A. 2017. START MENU in Unity. Watched on 11.06.2020. https://youtu.be/zc8ac_qUXQY

Tyson, J. 2000. How Video Game Systems Work. Read on 07.12.2019. https://electronics.howstuffworks.com/video-game2.htm

Unity 3D. 2020. Unity Manual: Packages. Read on 26.08.2020. https://docs.unity3d.com/2019.3/Documentation/Manual/PackagesList.html

Unity 3D. 2020. Unity Manual: Video file compatibility. Read on 25.06.2020. https://docs.unity3d.com/2019.3/Documentation/Manual/VideoSources-FileCompatibility.html

Unity Forum. 2020. AR Multiple Image Tracking: Multiple Objects. Read on 03.08.2020.https://forum.unity.com/threads/ar-multiple-image-tracking-multiple-objects.846901/

FIGURES

FIGURE 1. Google Trends. 2020. Augmented reality. Read on 21.08.2020. https://trends.google.com/trends/explore?date=2007-01-01%202020-07-21&q=%2Fm%2F0lqtr

FIGURE 2. Writer’s own image. 2020. Android and iOS market share.

PICTURE 1. The Virtual Continuum. 2017. Ariel, G. 2017. Augmenting Alice: The Future of Identity, Experience and Reality. Amsterdam: BIS Publishers. Read on 16.06.2020. Requires access right. http://libproxy.tuni.fi/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,uid&db=e000xww&AN=2002028&site=ehost-live&scope=site

PICTURE 2. Screenshots from the Artivive app

PICTURE 3. Simpson, L. 2018. Poster triggers a 3D virtual object using Augmented Image in ARcore. Read on 24.08.2020. https://medium.com/@thereallukesimpson/how-machine-learning-augmented-reality-crushed-it-at-google-i-o-2018-ce2827985404

PICTURE 4. Damiani, J. 2020. Google Creative Lab has officially released 'Lines of Play,' an open-source AR experiment that taps the functionality of the ARCore Depth API. Read on 23.08.2020. https://www.forbes.com/sites/jessedamiani/2020/06/28/google-creative-lab-launches-lines-of-play-an-open-source-experiment-to-showcase-arcore-depth-api/#557cbd11705f

PICTURE 5. Writer's own picture. 2019. 2nd LIFE mural illustration.

PICTURE 6. Writer's own picture. 2019. 2nd LIFE mural concept sketch.

PICTURE 7. Writer's own picture. 2020. 2nd LIFE mural canvas resolution compared to the Nintendo Entertainment System, Super Nintendo Entertainment System, and the Game Boy Advance resolutions.

PICTURE 8. Writer's own picture. 2020. The dragon character sprite using the 8-color palette.

PICTURE 9. Writer's own picture. 2020. Digital renders of the dragon character from different angles.

PICTURE 10. Writer's own picture. 2020. 2nd LIFE mural in the stairwell.

PICTURE 11. Teo Georgiev, T. 2018. A GRID'S UNICORN MURAL. Read on 28.06.2020. http://teogeorgiev.com/project/a-grid

PICTURE 12. Writer's own picture. 2020. The fire breathing section of the base dragon animation in a sprite sheet and the full animation demonstrated in action: https://youtu.be/ej4IairT48Y

PICTURE 13. Writer's own picture. 2020. The base animation loop for the Aalto knight on horseback presented in a sprite sheet and the animation in action: https://youtu.be/660ANkOuews

PICTURE 14. Writer's own picture. 2020. The base morphing into a smaller version presented in a sprite sheet and demonstrated in action: https://youtu.be/VQ0fQ1fKaJU

PICTURE 25. Writer's own picture. 2020. Digital renders of the Yggdrasil tree from different angles.

PICTURE 26. Writer's own picture. 2020. The isolated island in the 3rd ELEMENT 3D scene featuring different types of products based on lithium and lithium-ion batteries.

PICTURE 27. Writer's own picture. 2020. 3rd ELEMENT mural in the stairwell.

PICTURE 28. Screenshots from Blender 2.81.

PICTURE 29. Screenshots from Blender 2.81.

PICTURE 30. Writer's own picture. 2020. Visual steps towards accomplishing the sought for randomized upward flow of the chimney smoke.

PICTURE 31. Screenshots from Blender 2.81.

PICTURE 32. Writer's own picture. 2020. Technological products emitting a white ominous glow.

PICTURE 33. Writer's own picture. 2020. The transition into the 3rd ELEMENT LED title.

PICTURE 34. Writer's own picture. 2020. Lighthouse sketch on the left and 3D model on the right.

PICTURE 35. Writer's own picture. 2020. The last frame of the 3rd ELEMENT animated sequence as seen in the completed product: https://youtu.be/-phMWB_9NyE

PICTURE 36. Writer's own picture. 2019. 4th DIMENSION mural illustration.

PICTURE 37. Writer's own picture. 2019. 4th DIMENSION owl nebula concept sketches.

PICTURE 38. Writer's own picture. 2020. A 3D preview of all Earth layers, based on the Photoshop layer list alignment.

PICTURE 39. Writer's own picture. 2020. A three-dimensional preview of all owl nebula layers, based on the Photoshop layer list alignment.

PICTURE 40. Screenshot from the Solar System Scope app.

PICTURE 41. Writer's own picture. 2020. A 3D preview of all space layers, based on the Photoshop layer list alignment.

PICTURE 42. Writer's own picture. 2020. Digital renders of the owl nebula from different angles.

PICTURE 43. Writer's own picture. 2020. 4th DIMENSION mural in the stairwell.

PICTURE 44. Screenshot from After Effects.

PICTURE 45. Writer's own picture. 2020. Six transitional steps showcasing the directional blur effect working alongside the CC Sphere effect.

PICTURE 46. Writer's own picture. 2020. All 350 frames of the first planetary movement group shown simultaneously.

PICTURE 47. Writer's own picture. 2020. The solar system.

PICTURE 48. Screenshots from After Effects.

PICTURE 49. Writer's own picture. 2020. Directional arrows showing the path of each individual galaxy, the first and last keyframes of the sequence.

PICTURE 50. Screenshots from Blender 2.81.

PICTURE 51. Writer's own picture. 2020. A preview of the completed metallic hand model rendered using Eevee render engine in Blender 2.81.

PICTURE 52. Screenshot from Blender 2.81.

PICTURE 53. Writer's own picture. 2020. Key poses of the tossed tricolor globe animation.

PICTURE 54. Writer's own picture. 2020. The last frame of the 4th DIMENSION animated sequence as seen in the completed product: https://youtu.be/cogv2bqF4qY

PICTURE 55. Writer's own picture. 2019. All mural artworks combined into one.

PICTURE 56. Screenshots from Unity 3D.

PICTURE 57. Screenshots from Unity 3D.

PICTURE 58. Screenshot from Unity 3D.

PICTURE 59. Screenshots from Unity 3D.

PICTURE 60. Screenshots from the working Mobile AR app prototype.

PICTURE 61. Screenshots from Unity 3D.

PICTURE 62. Writer's own picture. 2020. Pinch to zoom feature in action with the 3rd ELEMENT animated storyline playing.

PICTURE 63. Screenshot from online content manager tool Bridge.

PICTURE 64. Screenshot from online content manager tool Bridge.

THESIS APPENDICES

Appendix 1. Multiple Tracked Image Manager Script


```
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

public class ARImageAnchor : MonoBehaviour
{
    public string ReferenceImageName;
    private ARTrackedImageManager _TrackedImageManager;

    private void Awake()
    {
        _TrackedImageManager = FindObjectOfType<ARTrackedImageManager>();
    }

    private void OnEnable()
    {
        if (_TrackedImageManager != null)
        {
            _TrackedImageManager.trackedImagesChanged +=
OnTrackedImagesChanged;
        }
    }

    private void OnDisable()
    {
        if (_TrackedImageManager != null)
        {
            _TrackedImageManager.trackedImagesChanged -=
OnTrackedImagesChanged;
        }
    }

    private void OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs e)
    {
        foreach (var trackedImage in e.added)
        {
            Debug.Log($"Tracked image detected:
{trackedImage.referenceImage.name} with size: {trackedImage.size}");
        }

        UpdateTrackedImages(e.added);
        UpdateTrackedImages(e.updated);
    }
```

```
    private void UpdateTrackedImages(IEnumerable<ARTrackedImage> trackedImages)
    {
        // If the same image (ReferenceImageName)
        var trackedImage =
            trackedImages.FirstOrDefault(x => x.referenceImage.name ==
ReferenceImageName);
        if (trackedImage == null)
        {
            return;
        }

        if (trackedImage.trackingState != TrackingState.None)
        {
            var trackedImageTransform = trackedImage.transform;
            transform.SetPositionAndRotation(trackedImageTransform.position,
trackedImageTransform.rotation);
        }
    }
       }
```

Appendix 2. Mask Shader

```
Shader "MASK/MaskShader"
{
	SubShader
	{
		Tags{ "RenderType" = "Opaque" "Queue" = "Geometry-3" }

		ColorMask 0
		ZWrite on

		CGINCLUDE
#include "UnityCG.cginc"

		struct appdata
	{
		float4 vertex : POSITION;
	};

	struct v2f
	{
		float4 pos : SV_POSITION;
	};

	v2f vert(appdata v)
	{
		v2f o;
		o.pos = UnityObjectToClipPos(v.vertex);
		return o;
	}

	half4 frag(v2f i) : SV_Target
	{
		return half4(0,0,0,0);
	}
		ENDCG

		Pass
	{
		Stencil
		{
			Ref 1
			Comp always
			Pass replace
		}

			Cull Front

			CGPROGRAM
#pragma vertex vert
#pragma fragment frag
			ENDCG
	}
	}
}
```

Appendix 3. Scene Manager Scripts

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class ButtonManager : MonoBehaviour
{
    public void ButtonMoveScene (string Floor)
    {
        SceneManager.LoadScene(Floor);
    }

       }
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class BackButtonManager : MonoBehaviour
{
    public void BackButtonMoveScene(string Menu)
    {
        SceneManager.LoadScene(Menu);
    }

}
```

Appendix 4. Pan and Zoom Camera Scripts


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PanZoom : MonoBehaviour
{
    Vector3 touchStart;
    public float zoomOutMin = 1;
    public float zoomOutMax = 8;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            touchStart = Camera.main.ScreenToWorldPoint(Input.mousePosition);
        }
        if (Input.touchCount == 2)
        {
            Touch touchZero = Input.GetTouch(0);
            Touch touchOne = Input.GetTouch(1);

            Vector2 touchZeroPrevPos = touchZero.position -
touchZero.deltaPosition;
            Vector2 touchOnePrevPos = touchOne.position -
touchOne.deltaPosition;

            float prevMagnitude = (touchZeroPrevPos -
touchOnePrevPos).magnitude;
            float currentMagnitude = (touchZero.position -
touchOne.position).magnitude;

            float difference = currentMagnitude - prevMagnitude;

            zoom(difference * 0.01f);
        }
        else if (Input.GetMouseButton(0))
        {
            Vector3 direction = touchStart -
Camera.main.ScreenToWorldPoint(Input.mousePosition);
            Camera.main.transform.position += direction;
        }
        zoom(Input.GetAxis("Mouse ScrollWheel"));
    }

    void zoom(float increment)
    {
        Camera.main.orthographicSize = Mathf.Clamp(Camera.main.orthographicSize
- increment, zoomOutMin, zoomOutMax);
    }
      }
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PerspectivePan : MonoBehaviour
{
    private Vector3 touchStart;
    public Camera cam;
    public float groundZ = 0;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            touchStart = GetWorldPosition(groundZ);
        }
        if (Input.GetMouseButton(0))
        {
            Vector3 direction = touchStart - GetWorldPosition(groundZ);
            cam.transform.position += direction;
        }
    }
    private Vector3 GetWorldPosition(float z)
    {
        Ray mousePos = cam.ScreenPointToRay(Input.mousePosition);
        Plane ground = new Plane(Vector3.forward, new Vector3(0, 0, z));
        float distance;
        ground.Raycast(mousePos, out distance);
        return mousePos.GetPoint(distance);
    }
       }
```

Appendix 5. The Mobile AR app prototype

Jónasson, N. 13.09.2020. APK download link to the Mobile AR app prototype.

https://drive.google.com/drive/folders/18khFRcfchiztuSOD6lRUHHUBK_oDHBXZ?usp=sharing