Blender Eevee Render Engine in Indie Production

Using Blender's Eevee Render Engine for Art Projects

Mikael Seidler

Bachelor’s thesis
May 2018
Media and Art
Fine Art
This study was carried out as a project. Two render engines were compared for results. Artists are limited by funding when creating 3D animations. As rendering a 3D animation is the most costly part of creating a short animation video, cutting the rendering cost down by introducing a new engine opens up a freedom for an artist to pursue more ambitious videos. The introduction of Extra Easy Virtual Environment Engine render engine in Blender 2.8 allows for render animations in real-time. This new rendering option revolutionizes the approach for 3-5 minute short animations. As a still in development engine, there are features not yet implemented. The objective of this study was to demonstrate how to use this new engine in the full production of a short animation video, proving the viability of the engine as a full production ready engine that cuts costs dramatically.

Key words: principle bsdf, virtual gallery, sheepit, license
CONTENTS

1 INTRODUCTION ........................................................................................................... 5
2 BUDGET CONSIDERATIONS ...................................................................................... 6
  2.1 Cost of producing animation films ........................................................................ 6
    2.1.1 Complexity of film to cost correlation ......................................................... 6
  2.2 Cost Advantage .................................................................................................. 8
    2.2.1 Render farm use .......................................................................................... 8
    2.2.2 Licenses ..................................................................................................... 10
    2.2.3 Software .................................................................................................... 10
3 RENDER ENGINE ....................................................................................................... 12
  3.1 Choosing the render engine ............................................................................... 12
    3.1.1 Cycles render engine ................................................................................. 12
    3.1.2 Eevee render engine .................................................................................. 13
  3.2 Eevee .................................................................................................................. 13
    3.2.1 Why I chose Eevee .................................................................................... 13
    3.2.2 Principle BSDF .......................................................................................... 14
    3.2.3 Comparing Eevee and Cycles ................................................................... 15
  3.3 Eevee’s new features ......................................................................................... 28
    3.3.1 Screenspace reflection and light probes ....................................................... 28
    3.3.2 Development branch work around and complications .............................. 30
4 PROJECTS ................................................................................................................ 32
  4.1 Thesis Animation .............................................................................................. 32
    4.1.1 Presenting ideas .......................................................................................... 32
    4.1.2 Metaphorical representation of fears .......................................................... 32
    4.1.3 Lighting and color grading ................................................................. 33
  4.2 Virtual Gallery .................................................................................................. 33
    4.2.1 Creating a photo real replica .............................................................. 33
    4.2.2 Realism versus practicality ...................................................................... 38
    4.2.3 Results of the virtual gallery research and development ...................... 39
5 CONCLUSION AND DISCUSSIONS ....................................................................... 40
REFERENCES ........................................................................................................ 41
APPENDICES ........................................................................................................... 43
  Appendix 1. How Stigma Changed My World Artwork 2018 ....................... 43
### ABBREVIATIONS AND TERMS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSDF</td>
<td>Bidirectional scattering distribution function</td>
</tr>
<tr>
<td>Eevee</td>
<td>Extra Easy Virtual Environment Engine</td>
</tr>
<tr>
<td>GNU license</td>
<td>General Public License</td>
</tr>
<tr>
<td>HDRI</td>
<td>High Dynamic Range Image</td>
</tr>
<tr>
<td>IOR</td>
<td>Index of Refraction</td>
</tr>
<tr>
<td>PBR</td>
<td>Physically Based Rendering</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

The purpose of this study is to prove and provide a viable alternative to traditional 3D animation production. This paper will focus specifically on rendering projects made by an individual artists or small studios developing short animations. In this study I will be covering Cycles as a traditional rendering option and Eevee as the new alternative rendering process.

For the purpose of this thesis I decided to create a 3D animation video to demonstrate the capabilities and new possibilities that Eevee would offer to an artist. I chose to create an animation video addressing mental illness stigma. To do this I had to dive deep into metaphorical illustrations. This meant that I would be creating complex and time consuming renders for a short 2-3 minute animation video. After two years of studying Blender and the Cycles render engine I was able to create short animation video with the visual style and complexity of short videos such as Pixar's For the Birds 2001. However, as an innovative artist, I wanted to try to produce this on a low budget, yet keep the quality visually appealing to the audience.

I will discuss the cost factors that led to me pursue a cheaper alternative for artists. I will also show the capabilities of this new engine when compared to the traditional engine, Cycles. Finally, I will expand on how Eevee workflow limited and allowed me to create the final animation video. As a result this thesis will show the practicalities, limitations, and benefits of using Eevee in future artists’ projects. Also, in this thesis I will demonstrate the significant cost difference that would take place to render the final video in both engines.

During my research into practical and future uses of Eevee for artists I was presented with a research and development grant exploring a Virtual Gallery. I started to research the best way to take advantage of Eevee engine while analyzing and testing the practicality and usability of a Virtual Gallery space within Blender.
2 BUDGET CONSIDERATIONS

2.1 Cost of producing animation films

Independent creation of animation films for an artist has multiple different costs such as development, sound, music, and rendering, which is the most expensive part of it. According to Animation World Network (Winder 2001), budgets are split into two parts. The first half is the cost of salaries, license, music, and other creative cost. The second half is the actual production cost such as equipment, rendering, and subcontracting.

If the production of the animation film takes place in a small team, some factors to consider are facilities. As the production team grows larger facilities are required to house the production team. Then there is the consideration of in house or subcontracted rendering from a render farm. With the advancement in cloud rendering, having a private rendering farm is financially less beneficial. However, when the production team is only a one or two artist production, equipment and facility cost drop significantly, leaving the rendering of the final product as the most significant cost.

With all the different ways cost can fluctuate, I will focus on the reduction of cost in the rendering part of animation film creation. With the introduction of Eevee render engine, small studios and artists have the potential to use a real-time render engine to cut render cost by a significant amount.

2.1.1 Complexity of film to cost correlation

To go more in depth into the cost of producing my animation film I will take a look at all the factors. As a one man project I was creating every aspect of the film myself. This meant that the equipment and facility cost were low due to using my home as the production office. My production workflow was to build highpoly models, then texture them. I was aiming for realistic geometry to give my animation film a more photo-real visual style. This meant that not long into the process, my polycount was starting to bloat project files past some of the cheaper render farm options that have file size limits.
To send a scene to be rendered at one of these farms I would need to pack the textures into the project files to limit the amount of errors from missing data. The main scene in my video takes place on a narrow street with 10 buildings (picture 1). Each building has four materials with PBR texture sets each containing five different textures. This meant that the project file would need to contain around two hundred 2048x2048 textures and twenty five million polygons. Therefore, before sending my project files, I would need to implement heavy streamlining of both textures and models. Doing this would almost double my production time, which meant a direct increase in cost.

As my video was going to be dealing with heavy volumetric lighting, the render times per frame were going to be monumental, resulting in an expensive render. On a computer with GTX 1070 and a GTX 970 running in tandem, rendering a clean image in Cycles took just over 8 hours. The average render hour cost calculated from: Garagefarm, Render Street, Foxrenderfarm, and Qarnot Render is around 3 euros per render hour. At a cost of 3 euros per render hour on that computer, it comes down to 24 euro per frame. With the length of three minutes, the animation video would have had a total cost of 103680 euro. This is calculated by multiplying ((8h*3euro)*24fps)*180s). Of course, this hypothetical calculation is based on the final build designed for Eevee production pipeline which would not be optimized for Cycles engine resulting in this large of a render cost. With some
optimization and reduction, for example, fog render times could be improved vastly but would also change the final animation from its intended look. It is also important to note that final cost could vastly change as well as large studios such as Disney would have their own render farms and not rely on this type of service.

With the incredibly high Cycles’ render cost, previewing the final output mid production would not be viable. During the production stages, I would render out the real-time playback for test audiences to view between each iterations. Having virtually no render cost made this possible.

2.2 Cost Advantage

2.2.1 Render farm use

Cloud based rendering or render farms have enabled indies or even individuals to pursue small animation films. During the year 2017, I was working on a VR animation for the Finnish Embassy in Japan. During the production stages, our team had to deal with the upcoming rendering of the animation. The main concern was rendering the animation with zero budget. Rendering the animation required about a week’s worth of render time on multiple computers. This meant that we would have been using an entire row of computers from a classroom. As the school lacked any official rendering stations, this meant that the classrooms that were used for teaching would have lost a significant portion of computers while the rendering took place. The next obstacle was file security on storage. As there is no rendering setup and the computers are set to wipe on restart, this meant that power cuts could result in all data being lost. This unofficial rendering is a larger issue to both student creativity and time. With no real way to render larger longer projects, what students are able to create is restricted. This is where cloud based rendering and render farms step into the picture. At the rendering stage I had several options. Most low cost render farms operate at around 3 euros per render hour. For a short animation such as mine, the lowest price would be around 7500 euros. This is a significant sum to pay for a 4-minute animation video.
Paid render farm services that I looked into, with support for Blender, were Rebustfarm.net, Render.st, and Foxrenderfarm. There is a free alternative to this problem called Sheepit render farm.

The way Sheepit works is, it is a crowdsourced render farm where you render other people’s projects on your own computer, which in return provides you with render points that you can spend to have your own projects rendered. There are, however, a few things to keep in mind. As a free service that relies on other people offering to render your files, you come across certain restrictions, such as uncertainty of completion time or any guarantee of completion offered. In addition, you need to collect enough points to ensure you have the required funds to render your project.

A suggested solution for the school is to create a rendering group. The way groups work is that a group can have up to 100 members at any given time. The group collects points as a whole, and when a project is rendered, the group pool is used. Not only does a group allow for a greater number of points to spend, but as priority is also connected to the total point amount, this would give groups more priority thus guaranteeing completion rather than rendering on your own.

The group could be set up this way: a few of the teachers as managers and students can join the group during their 3D studies. If a student collects points for the group they get access to the rendering option. As years go by, the overall point pool will grow. This will make the group work on a similar basis that a company works. The longer it is active and actively operating, the more supply it will have. The whole setup would cost the school nothing and would free up the school computers from rendering. If at a later date the school would want to ensure rendering possibility with less student input, they could set up a single computer to constantly generate points.

Render farming can get expensive fast as shown in the cost calculations. For the past three months, I have been running a test of Sheepit render farm. I set up my notebook with Nvidia 540m card to render 24/7 to collect render tokens that can be used in Sheepit render farm. During the three months that I have had my low powered notebook rendering, I have accumulated over 1 million points. With this I could render a significant animation film, or so I thought.
Going back to the cost estimate for my animation video in a render farm being 103680 euros. According to the FAQ at Sheepit render farm (2007) if we calculate the point cost for this same animation video we get the result of 31104000 points. This shows us that during my three months of collecting points, I would still have only 1/31 of the points required to render my animation.

2.2.2 Licenses

The next big cost factor for a small animation studio is engine licensing. When buying a software such as Maya, you gain access to the software and license to use it in production. However, buying a single license or even 10 licensed versions of the software for your small company is not cheap. On the other hand, if you use software like Unreal Engine 4 for real-time animation, you need to deal with the license fees with the company where they will take a commission of your profits. Blender runs under the GNU license. This allows the creators to freely create, sell and profit without needing to worry about licensing fees.

2.2.3 Software

In the 3D world there are dozens of software designed for specific purposes. Whether it is Maya or 3Ds Max, Cinema 4D, Houdini, Zbrush, or Marvelous Designer, for an artist or small development studio acquiring the software is a big investment. For example, Zbrush is used for sculpting and modelling, Marvelous Designer is used for clothing and fabric based modelling. Maya is used for animating. These three programs are the base for an animation studio infrastructure. Following is the current pricing of the software from their respective websites.

A Maya license is 1984.40 euros a year. (Autodesk n.d.)
Zbrush license is 895 USD. (Pixologic n.d.)
Marvelous Designer is 490 USD. (Marvelous Designer n.d.)
Blender is free. (Blender foundation n.d.)

When deciding on the software, it is always important to analyse what is best for the job. In this case, the goal is to drop the price as much as possible while keeping the visuals
competitive. Blender removes the software fees and is introducing the new engine Eevee to reduce rendering fees as well.
3 RENDER ENGINE

3.1 Choosing the render engine

3.1.1 Cycles render engine

The Cycles render engine is a powerful photorealistic render engine. It can, however, be resource intensive and require powerful hardware to run. According to Andrew Price (2015) Cycles can compete in visuals with render engines such as Renderman, Luxrender, Maxell, Mitsuba, V-Ray, and Octane. To give an example of Cycles power, picture 2 shows an entirely computer generated scene I created.

PICTURE 2 Rustic Modern Interior

The render farms mentioned in chapter 2 support Cycles render engine. For visual fidelity, Cycles is the best natively integrated engine for Blender. Cycles is a physically based production renderer developed by the Blender project.
3.1.2 Eevee render engine

Eevee, nickname for Extra Easy Virtual Environment Engine, is a render engine that works similar to most game engines. (Felinto 2017) It uses OpenGL playback with PBR materials. Eevee allows PBR real-time playback eliminating the long multi-hour renders for each frame. This not only accelerates production but lowers the cost, as you do not need to render the same frames repeatedly after every tweak. Eevee is designed for easy showcasing and material preview of 3D objects.

3.2 Eevee

3.2.1 Why I chose Eevee

I decided to attempt producing my thesis animation using Eevee. After all the tests I ran, Eevee seemed the way to go: photorealistic materials and no additional render times. At the time of writing this, Eevee is still under heavy development. In the 2.8 Blender development build, certain important features relating to character and physic are disabled or hidden. The build gets updated almost every day and is incredibly unstable. The developers have stated that Eevee and Blender 2.8 should not be used in any large-scale production until it is complete. Here is a short list of features that are disabled that would reduce any animation to mostly showcase.

- There is no way to use bones and bone animation with deformation.
- Shape keys do not work.
- There are no physics simulations.
- All modifiers are hidden and need to be applied before being able to see the effect.
- Node animation can be done but must be adjusted via the graph editor.
- Animated textures are possible, but only one animated texture works at a time in viewport playback.
- Importing non blend files such as fbx or obj will crash the program.
- A save file from Eevee or Blender 2.8 will crash Blender 2.79.
- Large project files can easily crash during opening because loading a file in Blender 2.8 Eevee creates instances and has to load in objects, effects, and lighting all at once.
Volumetrics are some of the heaviest things to calculate in Cycles. Calculating light bounces inside a volume can take a long time and usually gives a noisy image in short renders. As seen in both the calculation and examples, later it will be shown that full environment volumetrics is very slow and costly. In Eevee volumetric lighting is fast and easy. You are able to have full environment volumetrics and real-time playback.

### 3.2.2 Principle BSDF

Blender materials use a node-based interface. A node performs an operation on the material changing the appearance of the material when applied to a mesh. Shader nodes in particular deal with visual properties of a material such as Glossy, Diffuse, Transparency, Emission, and Principle. (Blender manual n.d.) The Principle BSDF shader is a shader that was introduced in Blender 2.79 and can be used in both Cycles and Eevee. (Felinto 2018) This allows for workflow to be shared. It allows the model to be textured in Cycles and imported into Eevee. At of the time of writing this paper, Eevee and Blender 2.8 are still very unstable, which means that this easy import into Eevee from Cycles 2.79 is a key to getting anything done, because you spend less time on the unstable Blender version. The principle shader is a universal shader that has most shader nodes packed into one. It can be used to create a variety of materials: from skin with subsurface scattering, to glossy metals and even glass. Of course, as with any material, to get more specific looks, shaders can be combined and mixed. The best part about the principle shader is its easy use. For an artist working in game development, the principle shader works like most game materials. When using PBR material texture sets, you can plug in all the textures to the right locations in the principle shader and see pretty much how the material would look in a game engine or a site like Sketchfab. As a side note, if you are using properly named texture sets, such as textures exported from a substance painter or designer, you can use the built in feature of node wrangler add-on to simply select all textures, and have them automatically linked to the correct inputs.

When loading up a model in Eevee that has a principle shader, there are a few extra steps to creating the same looks for it. If the material is a transparent material you will need to select a transparency type. In addition to transparency, there is screenspace reflection and subsurface scattering. After enabling these, the material will start to represent the intended look.
15

Picture 3 from Blender Guru demonstrates all the properties of the Principal BSDF shader

3.2.3 Comparing Eevee and Cycles

To explain and demonstrate the differences and advantages of using Eevee engine over Cycles, I created a sample project using the Blender BMW M1 demo file. (Blender: Demo
files 2011) In this demo I am using a night time HDRI map from HDRI Haven to light up the scene. The animation of the car accelerating and decelerating is 1000 frames long. Since both Cycles and Eevee use the Principle BSDF shaders, I could just import my Cycles project into Eevee and enable all the post processing such as screen space reflection, bloom, and depth of field.

While running the animation in real-time playback on my GTX1070, I was getting around 18 to 19 frames a second. In real-time playback, I was running at 19 samples. The sample count is a numerical value that determines the time a light path is calculated. A higher sample rate results in a more accurate and less noisy image. An animation with 19 frames does not have enough frames for a smooth animation, but has plenty to see how the animation works. This simple scene of just the car would render at around 4-5 frames per second using OpenGl rendering at 32 samples. When rendering the animation out, I can increase the sample rate, shadow resolution and overall artefact reduction. By calculating the animation length of 1000 frames divided by 4 frames then dividing by 60 seconds, rendering out the entire animation will take just around 4 minutes.

At the same time, Cycles, rendering with 150 samples and denoising, was taking 20-25 seconds per frame. This would result in almost 7 hours to render the entire animation if taken into account the render time variable fluctuation of each frame.

In picture 4, 5, 6, and 7 there are side by side comparison of the Cycles render and Eevee render engines.

In picture 4 we take a close look at the car headlights. Here you can see the heavier Cycles producing a better result over its brother Eevee.
In picture 5 we look at the cars from the front. As you move further from the small details, Cycles and Eevee start looking very similar.
In picture 6, similar to picture 5, we see that the difference between the two images is noticeable especially in the lights and glass surfaces.

![Picture 6](image)

The next example I want to cover is about displacement and close up. Eevee is built to use PBR materials. PBR materials are standard in game engines and can achieve a realistic looking surface without the high level geometry. In picture 7, I then proceeded with a test which used a displacement modifier instead of a normal map to create geometry onto the object mesh. In Cycles, I enabled adaptive geometry that generates geometry based on how close to the camera the object is. This reduces the overall geometry and saves memory and render time as distant objects will look right with less detail.

Eevee, on the other hand, had to use a constant displacement as it works on instances to increase performance. Having all the geometry in place greatly reduced my frame rate during real-time playback, but hardly affected my render times. Cycles was rendering at 5s per frame while Eevee was rendering at 0.1s per frame.
In the case of a roof, unless the scene required the camera to be at this level of close up, the better option would be to use normals instead of real geometry. In Picture 8, I ran a comparison of Cycles and Eevee using normals for displacement. Here you can see that the difference is noticeable. Using this method, the render times stayed the same but the real-time playback got significantly better in Eevee.
Next we will look at Principle shader properties and how they change depending if we are using Eevee or Cycles. In picture 9 I demonstrate how the four levels of change in values within the Principle shader looks. In picture 10 we compare metal surfaces at four different roughness settings: 0, 0.25, 0.5 and 1. The only noticeable difference is the noise amount in Cycles. As with everything before, Eevee had instant render times while Cycles was rendering anywhere between 10-40 seconds at 250 samples.
In picture 11, I removed roughness and only adjusted the metallic value of the spheres. One of the main differences with Eevee and Cycles when dealing with the shaders is enabling the postprocessing for certain effects such as subsurface scattering, screen space reflection and transparency.

PICTURE 11

In picture 12, I show the difference of a metallic surface with and without screenspace reflections enabled.
Next we look at the comparison of glass. First, in picture 13, we take a look at the way IOR (Index of Refraction) works in both at the values: 1.5, 2.5, 5, and 10. Due to the more complex calculations in Cycles, the different IOR levels are much more varied. The Cycles version also works more like real glass rather than a mirror ball effect. Similarly, when adjusting the roughness of the glass, Cycles is able to keep more information, producing a better result as seen in picture 14.
The final material property to look at is subsurface scattering. Subsurface scattering is used to simulate skin and similar materials. Subsurface scattering is another highly resource intensive process to calculate. In Eevee, there is realtime subsurface scattering as seen in the picture 15.
When it comes to atmospheric lighting, the most photorealistic way to simulate full environment volumetrics is by adding a volume scatter or volume absorption to create a fog that all light travels through. In Cycles, a well-lit volumetric scene can be stunning but also incredibly resource intensive. Silent night by artist Reynante M. Martinez (picture 16) is a great example of volumetric lighting. On a GTX1070 and GTX 970 running side by side at 8888 samples it takes 4 hours to render.
The most noticeable part of volumetrics is the noisy image at lower samples. In picture 17, I render spheres in a Cycles volume at 5000 samples for 40 seconds while I had a realtime playback of Eevee volume with instant render times.
3.3  **Eevee’s new features**

The Eevee render engine is a new engine with new features that were not present in Cycles render engine. The most notable new feature is screenspace reflection and light probes, which I will elaborate on.

### 3.3.1 Screenspace reflection and light probes

In Eevee, there are four ways to create reflections. The first is material screenspace reflection. Material screenspace reflection is the most computationally resource intensive during rendering but also allows for moving reflections while the reflection probes create
reflection snapshots. Using screenspace reflections over light probes will increase render times and reduce real-time playback fps. A reflection plane is used for perfect flat reflections like mirrors or creating water. A reflection sphere creates a reflection on all surfaces inside the sphere. These reflections are great but as they are a single state reflection, they are not the best option for an animation due to the reflections not updating while the camera moves. The final is volume reflection that creates multiple instances of reflection within the volume of the reflection cube. (CG Master 2017)

When dealing with reflections there are different uses and needs, for example, transparency with reflection. First, consider the following questions: will there be a reflection, does the window have a texture, is the window transparent, and is there light emitting from inside. A metallic object with screenspace reflection will reflect perfectly, but what if we switch to a window. A window has a reflection at an angle but not a reflection when looking straight at it. When dealing with transparency in Eevee, there are a few options to choose from: Alpha Blend, Alpha Hashed, Alpha Clipped, Multiple, Additive, and Opaque.

None of the options above have a good fresnel layer weight for transparency to glossy reflective surface. To fix this, a reflection plane can be placed behind a window to create the reflection while the actual glass stays transparent. When a light source is introduced, reflections become less apparent, therefore, a reflection is not as important.

In a situation where a car is in a show room state, when the car is lit up in a center of a dark room with perfect reflections and the camera spins around the car, the spherical reflection map can be used (picture 18). The state of the car stays the same; only the camera moves. Using the spherical reflection map instead of screenspace reflections saves in render time.
3.3.2 Development branch work around and complications

Most particles, smoke, and physics do not work inside Eevee, yet this can be circumvented by using pre-rendered alpha sequence planes. By creating the wanted visual effect in Cycles, rendering it out, and adding it as a plane in Eevee, it is possible to create the illusion that the affect is happening inside Eevee. An example of this is the raindrops on the puddles used in my thesis animation. I simulated rain drop ripple effects using the physics engine in Blender 2.79. I then rendered out a grayscale tileable texture of the ripple animation. After this, I was able to convert the grayscale animation into a normal map sequence that could be input into a material for animated normal (picture 19). Before the animation will work, it is important to check that the image frame rate matches with the material frame rate. The settings for auto refresh and cyclic must be enabled.
Creating the Rain drops on the windows of my animation used a similar technique to the puddles. Instead of simulated ripples, I had video footage against a black backdrop that I used as the animated texture (picture 20).

Due to some technical development issues, keyframe adjusting the values of a nodes is not possible, but the graph editor circumvents this issue. The graph editor can be used to animate, for example, the change in power for a light source. The graph editing can be enabled once two keyframes have been created for the same object or value.
4 PROJECTS

4.1 Thesis Animation

For my thesis I chose to create an animation video about how stigma towards mental illness affects the people suffering from mental illness. I focused on creating a state of mind, a feeling visualized by motion, sound, and environment. (Appendix 1)

4.1.1 Presenting ideas

It was very important for me to be able to create the right atmosphere. The animation video deals with the feelings of a person suffering and having been diagnosed with a mental illness and this person having to face the stigma and perceptions of others because of this state. The video represents the outlook of an empty world where no one accepts you for what you have become, when in reality it is not something you have become but a condition in which you were born. The video attempts to show how this general negative view can make the person feel. They no longer feel like they are accepted, doors close, people vanish, and what is left is a gloomy world where they are the only person left.

4.1.2 Metaphorical representation of fears

One of the main elements needed was having a dense fog and rain. I wanted to not just represent the world as empty to the eye of the suffering person on the receiving end, but a way to also show the limitations of their ability to see hope and to find someone who accepts them for what and who they are. Fog is a natural occurrence that is well known to limit visibility. It hides things and makes the world around you feel small. I wanted to create this idea of the possibility that there is someone who accepts you right there in front of you but is just far enough away that the fog hides them. They exist in the world, but due to the situation, you cannot see them. The result is this idea of overwhelming negativity blinding you to believe that the entire world succumbs to the hateful stigma.
Another metaphor that I wanted to visualize is tears, a visual representation for sadness and sorrow. In a video where everything is in the first person, I had to use the environment to demonstrate this emotion. After consideration and testing I decided on using a window with water running down it. This fit the theme of foggy or rainy weather, and a cold gloomy environment. Just like on eyes and cheeks the water builds up until it can no longer be held in place and runs down with the force of gravity leaving trails of water.

### 4.1.3 Lighting and color grading

For lighting and color grading I took advantage of the natural bloom and tint the fog would generate. I used blue tinted environment texture to give a cold night feeling to the environment and used white lights for light sources. Due to the real-time rendering using OpenGL, there was no post production in the video. All color and lighting was identical to the in engine viewport. This made visualizing the final output very easy.

### 4.2 Virtual Gallery

As part of a research grant I started looking into the different uses of a Virtual Reality Gallery. As I was already researching Eevee I saw the virtual gallery as an opportunity to take advantage of this new render engine.

#### 4.2.1 Creating a photo real replica

As part of a research grant I started looking into the different uses of a Virtual Reality Gallery. As an exhibition planner tool, I looked into creating a virtual reality replica of a gallery for the purpose of planning out the exhibition in an environment that would give a realistic visualisation immediately. As game engines can be tricky to learn and require software installation, they rely on building the game for demonstrations. For this reason I theorised that Blender 2.8 with the new Eevee engine was a good solution. It allowed me to complete a portable gallery that used photorealistic lighting and was easy to use. It also had no licensing issues as it is under the GNU License, making the sale of the virtual gallery a possibility. (Blender foundation n.d.) With a portable virtual gallery this meant that the replica space could be taken and shown at any meeting. The only equipment
required was a computer and the usb thumb drive that contained the software and the project files. Implementing new art meant just adding new objects into the scene.

In the pictures 21, 22, 23, 24, 25, and 26 I demonstrate the realism we were going for in the virtual gallery.
During the early development stages of the virtual gallery, things were going smoothly with great progression. However, it became clear very fast that the plan had some holes. The most notable one was scale. Even though the gallery was built at 1 to 1 scale, it felt smaller than the real space. Artworks were hard to scale appropriately and the space felt cramped. This made test audience complain that there was no way the artists believed they could fit the artworks inside without making the space feel entirely full. The virtual gallery replica space can be seen in picture 27.
4.2.2 Realism versus practicality

The next issue was the complexity of the gallery and finding the line between realism and practicality. Although this virtual space could have been turned into a tool, no artist was really keen on using it. They were more focused on working on their artworks, believing they could get them to fit without trying to create small replicas of in development works for a planning tool. At the same time, as most works were not complete until the very last moment it meant that there was no artworks to truly test the practical uses.

It was at this point that we started to look into a more interactive game version of a virtual gallery. We transitioned from a tool to an experience: a virtual gallery that would be in virtual reality. Removing the wall to create open spaces was the first idea. As a result a lot of the structural details were not used as they added nothing to the experience. The virtual gallery became a re-visioning of the artwork inside the gallery’s physical space.

The practical use of Eevee, however, did not end although the development moved from Eevee back to the Unity game engine. With Eevee real-time PBR, all objects could easily be visualized and tested with PBR materials without the need to import and test them inside Unity. The workflow was faster and the 3D model turnaround was efficient. A new object idea was sketched, designed, PBR textured, and tested within an hour.
4.2.3 Results of the virtual gallery research and development

In conclusion, visualizing both the environment objects and the artwork replicas inside Eevee was great but due to still having both a technical barrier between the common artist and the software, as well as the lack of motivation to use the tool, the creation of a replica virtual gallery was not a great option. However, using Eevee as a visualizing step of models and assets for the virtual reality game ended up saving hours of time but is not an essential tool.
5 CONCLUSION AND DISCUSSIONS

The purpose of this study was to find an alternative method for rendering animation films. The study was conducted in the Eevee render engine coming out with Blender 2.8. The study was a success with concrete proof. I have successfully shown the quality comparisons and the advantages of using Eevee render engine. As all the functionality of the program is implemented into Eevee, it will become a viable render engine, but due to the still in development stage, animations are still limited. Currently, bones, rigs, shape keys and drivers are not supported, neither are the physics simulations, but with the official release these features will be re-enabled. Beyond animations Eevee can be extremely useful already in its current development stage for previewing PBR textures, checking models for mistakes, and creating display images. This can be seen in the practical use implemented during the virtual gallery research and development. Being able to review and watch the animation with all lighting and effects is a great time and money saver. The reduced time for rendering allows for fast production time and low cost rendering. As proven with the cost difference covered in the paper, artists such as myself are given the possibility to produce and envision animation videos that would otherwise be too expensive to render.

To conclude, this research Eevee and Blender 2.8 with all of its new features will not only push Blender to become more competitive in the 3D world, but also become an essential tool for students, artists, and small studios that lack the budget for traditional animation and rendering. Even free options such as Sheepit, although viable for some animations, are not a solution for artists as proven with the cost calculations done in this research paper. It is recommended for students to take advantage of Eevee. It will also be beneficial for universities that lack the facilities and resources to provide rendering options.
REFERENCES


https://www.blenderguru.com/articles/render-engine-comparison-cycles-vs-giants

Autodesk. Maya Subscription. n.d. Read on 12.4.2018
https://www.autodesk.eu/products/maya/subscribe?plc=MAYA&term=1-YEAR&support=ADVANCED&quantity=1

https://www.blender.org/about/license/

Blender Manual. n.d. Read on 9.5.2018

Catherine Wilder, AWN, Producing Animation: Building the Budget, 9.8.2001. Read 18.4.2018

https://www.youtube.com/watch?v=DR_uRoKuST8

https://www.blender.org/download/demo-files/

https://code.blender.org/2017/03/eevee-roadmap/

Dalai Felinto. Blender: Blender 2.8 Highlights. 22.4.2018. Read 6.4.2018
https://code.blender.org/2018/03/blender-2-8-highlights/

https://code.blender.org/2017/10/blender-2-8-design-document/

https://code.blender.org/2018/03/eevee-f-a-q/

Fox Renderfarm pricing. n.d. Read on 9.05.2018
http://www.foxrenderfarm.com/pricing.html

GarageFarm cost calculation. n.d. Read on 9.5.2018
https://garagefarm.net/cost-calculator/
https://www.marvelousdesigner.com/product/pricing

Pixologic. Zbrush single user license. n.d. Read on 12.4.2018
https://store.pixologic.com/zbrush-2018/single-user-license/

Quarnot Render Pricing. N.d. Read on 9.5.2018
https://render.qarnot.com/pricing

Render Street pricing. n.d. Read on 9.5.2018
https://render.st/plans-pricing/

Rynante M. Martinez. Rynantematrinez. Read 10.3.2018
http://www.reynantemartinez.com/silent-night.html

Sheepit: F.A.Q. 2007. Read on 5.2.2018
APPENDICES

Appendix 1. How Stigma Changed My World Artwork 2018

“In 2016 I was diagnosed with bipolar disorder. My world changed as I watched the fall-out. The world has a stigma towards mental illness and until you are on the other side you don't know what it feels like. I was getting better. I was able to resume normal life but the world around me felt different. I would catch myself staring into oblivion. The world seemed to be empty. People saw me as broken and distanced themselves from me. I wanted to create a short video to show what the world looks like every day as I look out the window.