University for the Common Good

MSc Project

Research Studies for Computing

and Creative Technologies
(MMG526752)

Student ID:	S2437574
Student Name:	Laura Robbie
Degree:	MSc 3D Design for Virtual Environments
Project Title:	What is stopping runtime virtual textures from becoming the norm in game development?
Keywords:	Runtime Virtual Textures, Unreal Engine, Environment Art
Supervisor:	Martin Woodburn

Contents

1. Introduction

1.1 Aim of Project

In the world of 3D artistry, textures are a pivotal factor in the visual quality of a scene. They influence not only the look of a scene, but they impact the performance of the game or interactive art piece. Runtime Virtual Textures (RVTs) are a relatively new technique in the games industry, it was introduced to Unreal Engine to optimise texture streaming by utilising the GPU instead of relying on traditional methods. RVTs stream only the texel data necessary for the screen, it is a tile based rendering approach and provides an efficient use of the GPU while also providing advanced ability of blending materials between landscape and non-landscape assets.

With the rise in popularity of procedural generation, modular assets and large scale open-world environments, RVTs offer a powerful, significant improvements in maintaining visual quality and performance. They reduce the reliance on high-resolution texture maps, simplify the management of texture LODs (Levels Of Detail), and resulting in a smoother gameplay experience. Along with these benefits RVTs offer many more advantages. Despite the benefits, using RVTs in games are still lacking across the industry, this is due to technical complexity, lack of documentation and consumers not having access to better hardware. The current gap between RVTs potential and it's full integration into the games industry presents an ideal slot for investigation.

The aim of this project will be to create a scene in Unreal Engine 5 that demonstrates RVTs and how they can be built, and specifically highlighting the blending of assets into landscapes. It will involve other progressive texturing techniques such as vertex painting and procedural generation, RVTs will be active showing how the techniques interact with each other. Work will be carried out to understand the effectiveness of this technique, and information will clearly be displayed across the documents to understand how this technique is shaped. The industry is currently showing a lack of this technique across all genres, the goal is to identify where and how this technology can be more widely adopted in professional pipelines, particularly in areas of game development that benefit from high visual quality and efficient GPU usage.

2. Context and Technical Background

2.1 Evolution

Applying textures to models evolved from research created in 1974 from Edwin Catmull – computer scientist and cofounder of Pixar from the United States of America. Many have reported on this research and labelled Catmull a pioneer in the 3D graphics industry through his contribution of texturing. Catmull was responsible for texture mapping and is shown through his 1974 PHD thesis documentation “A Subdivision Algorithm for Computer Display of Curved Surfaces” (Catmull, 1974). From this the world of 3D graphics changed and we were able to adapt Catmull’s technique to what it is today.

Loading textures was once a long tedious process where textures were loaded individually one by one, this was a hugely inefficient way of processing 3D environments. This is where Multiple Render Targets (MRTs) were introduced which essentially gave the GPU the ability to render multiple textures at once which rapidly dropped the render time. MRTs focus on rendering all textures within a scene at the same time, this is a limiting process as there would need to be a heavy reduce in quality of textures to keep the game running. In the never ending progressive rendering pipeline, creating a norm for rendering multiple textures at once became apparent when Mip Map rendering was formed. They only render what is required of the viewport however they store this in memory, although limits can be set on this it is not optimal. RVTs were developed to work to solve this memory issue, while still loading what is necessary, this process eliminates what is no longer required at runtime.

2.2 Runtime Virtual Texture Definition

Runtime Virtual Textures (RVTs) is a process which uses texel data to render textures on demand from the GPU at runtime, they only load the relative textures rather than loading entire high resolution textures into memory. RVTs are similar to Streaming virtual textures (SVTs) according to (Epic Games, 2025), however they work from the disk space and are “Well suited for texture data that takes time to generate, such as lightmaps or large, detailed artist created textures.”

2.3 How Runtime Virtual Textures Work

To put it in simplistic terms instead of keeping single separate textures for each object rendered in the scene, all relative textures needed are stored in a memory pool referred to as “Virtual Texture”. Obert et al. (2012) describes the RVT saving technique as “The size of the virtual texture is on the order of billions of texels, and each object is assigned unique virtual-texture coordinates from the virtual texture.”

To explain this further RVTs work by saving one tile of virtual texture as a page into a memory pool, the pool acts as a recently used cache. As the system requests a tile it is rendered into an available page in the memory pool, should all of these pages be taken the least recently used cache is destroyed and the new page will take its place. Figure 1 shows an example of the memory pool.

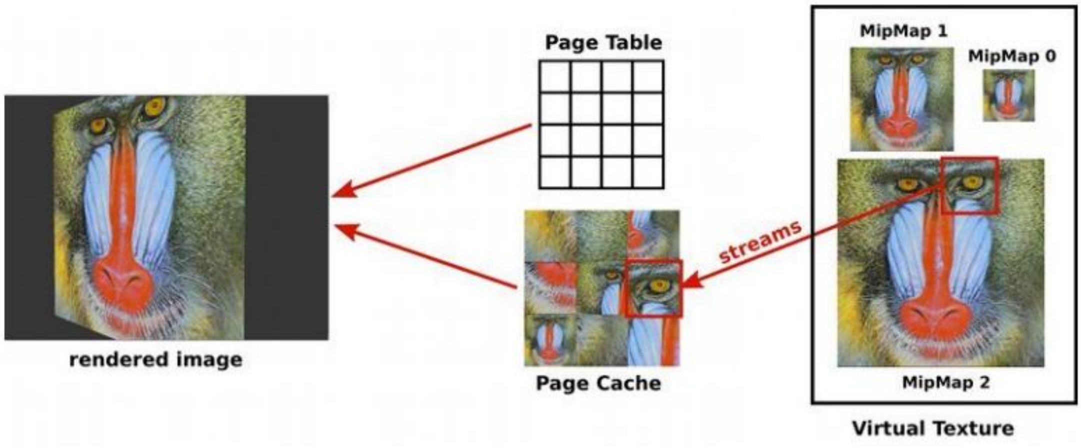

Figure 1 - Example of RVT system saving technique (Neu, 2010)

As shown the pages collected from the main texture are all saves as the same size in the memory pool, although this is a small example it shows that there is hierarchy in the system and that pages aren't necessarily saved in order by object. RVTs are split up to make these pages by evenly dividing the texture. As the scene updates the pages currently loaded into the memory pool are reevaluated, if textures are needing to be written and some are no longer in use these pages will be over written.

It is a valuable process as it works effectively for rendering large complex landscapes, it also gives the ability to texture blend between assets to camouflage the clipping of them, including decals and splines. It is a relatively new and useful process that gives high quality visuals and improved workflows. RVTs are important to the developing game industry due to hardware becoming more accessible with better specifications it provides developers the ability to create projects working with the GPU rather than relying solely on memory usage.

2.4 Comparison of Similar Processes

MipMap streaming is still at the forefront of the way developers are loading their games, game engines like Unreal Engine automatically choose MipMap by default. It is a clever system as it creates a hierarchy of textures to pull from, depending on how far away the asset is from the viewport, this is shown in figure 2. Each MipMap level is divided and connected with the original divide, keeping that link allows the system to easily up the

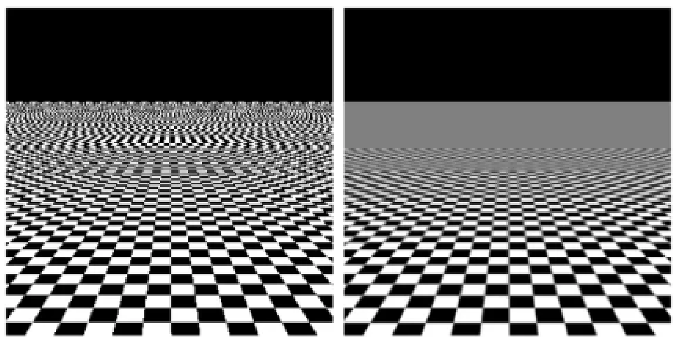

Figure 2 - Left no MipMap, Right Use of Mipmap to prevent aliasing (Sagar, 2022)

resolution as the user comes closer. An example showing this hierarchy is shown in figures 3 & 4. They are loaded in as needed much like RVTs, however their main drawback is that they get drawn to memory and then will sit there using up storage. According to Sagar (2022) MipMaps use around 33% more storage than just loading the textures regularly.

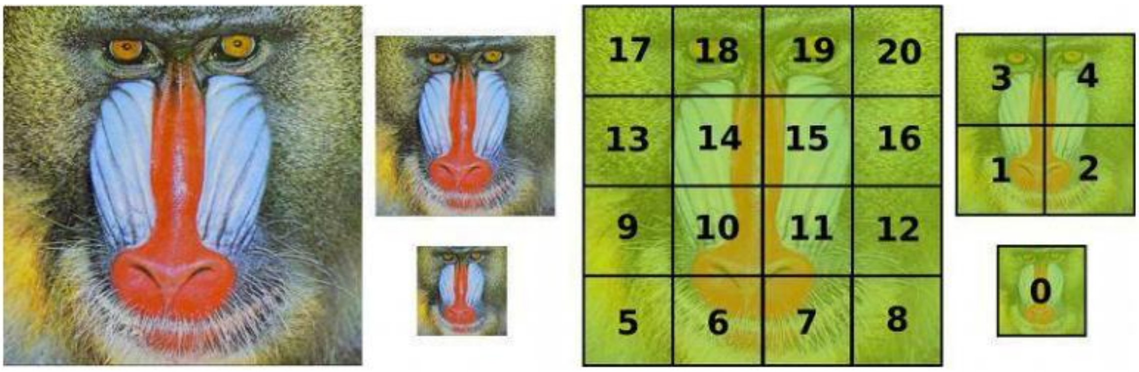

Figure 3 - Example of memory pool system. (Neu, 2010)

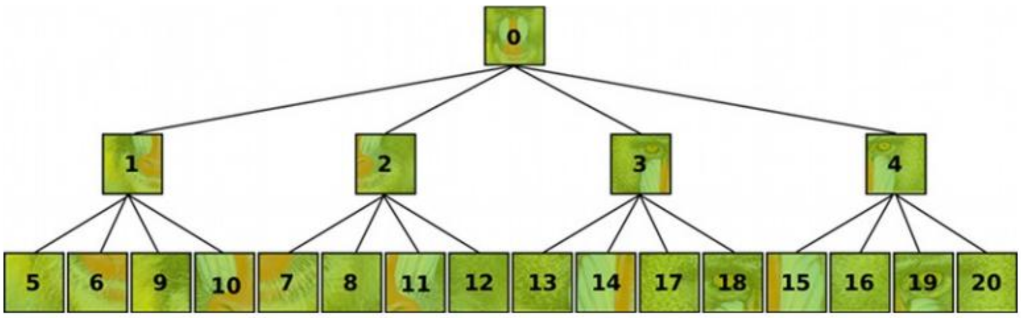

Figure 4 - Example of memory pool system's Hierarchy. (Neu, 2010)

2.5 Benefits or Runtime Virtual Textures

As RVTs are using the GPU to run and manage textures it makes them will suited for rendering large complex landscapes. This makes them the ideal candidate for working with procedural workflows, it enables developers to create layered materials without having to rely on static textures. For example, materials can be dynamically blended with the landscape, with this method it camouflages the clipping of assets through the ground by integrating them into the environment.

Another advantage of RVTs is simplifying the rendering pipeline by eliminating the need for blueprints to project or draw materials onto surfaces, they can project directly onto the material using the landscape as reference. This gives developers the opportunity to use different decals across multiple instances without having to run the cost of each instances modification. Similarly spline-based tools also benefit from RVTs by enabling decal like effects which will conform to the landscape terrain, this will work well when using this for roads, paths and rivers as they will naturally blend with the environment. This boosts the visual realism and performance as it lowers the need for a static mesh. In general RVTs improve the visual cohesion of the environment.

2.6 Challenges and Limitations

RVTs offer many advantages, in contrast with this they also come with limitations, the main factor in this is that they rely heavily on the GPU. This is apparent specifically in scenes crammed with a high number of RVT samples, or the pages in the memory pool are too large in size. This reliance on the GPU is putting a strain on people's hardware, warranting in a need for a higher specification of system, this limits the accessibility of older / lower-end hardware.

The additional complexity of new workflows follows a steep learning curve, but without tools and documentation this is drastically slowed. Running a newer process that has had a lack of use due to accessibility and documentation will result in a slower start to being integrated. The smaller studios and individual developers will be impacted harder with their limited resources.

3. Evaluating Additional Techniques

3.1 Procedural generation

As RTVs are a newly developed process, it is important to show how they are reacting with other techniques. Procedural generation is an algorithm, and parameter-based workflow automating the process of texture generation. Texturing and Modelling: A Procedural Approach (Ebert and Al, 2003) provides an accurate description of what the process is "Procedural techniques are code segments or algorithms that specify some characteristic of a computer-generated model or effect." This book provides significant insight into the efficiency of procedural generation, it explains that due to the lack of reliance on large image files it reduces the memory usage needed.

With procedural generation being a new technique that has become increasingly more popular in recent games it was an ideal technique to test RVTs with and understand how they interact with each other. It is key to show the demonstration of this as it can be a valuable tool for new people learning how to use these techniques together in industry.

3.2 Vertex Painting

Vertex painting is a powerful technique that interacts with the 3D mesh itself, as Inspiration Tuts (2024) explains "It allows artists to paint directly on top of 3d models instead of using hardware power to process the textures." This meaning that the setup of this process can enable smooth transitions between materials without needing additional textures or UV maps, unique data can be applied per asset, dirt, moss or snow are great examples of this. Vertex painting is another important technique that needed to be tested with RVTs, as it heavily involves material blending. As it is a widely used technique and many are aware of how it works, it would be ideal to show how RVTs interact with it.

4. Industry Integration and Case Study

4.1 Unreal Engine Implementation

According to Unreal Engine's 4.23 update manifesto (Wilson, 2019) they had released in 2019 they brought RVTs with this update and was available for artists/ game developers in its beta form. They introduced RVT Beta as a support to create and use large textures at a lower memory cost. They did state that it would be an increase of performance at runtime, but also an expensive option from sampling textures regularly. Despite its large claim RVTs were often unfortunately glazed over or as AskNK (2019) described it as "much more expensive" than the previously used MIP streaming process. With many people reviewing the update focussing on the other features this sent RVTs onto the back burner with many discarding this feature.

4.2 Integration with Hardware

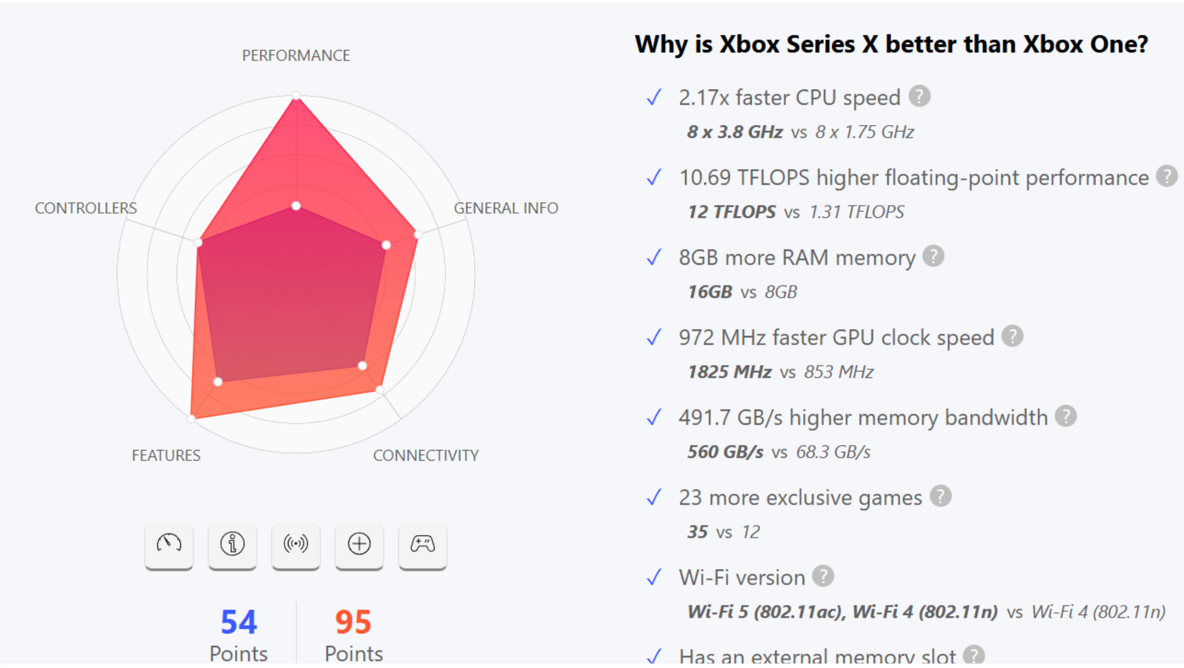

Figure 5 – Graph showing the specification between Xbox generations (Versus, 2025)

RVTs came with great promise to the developing games industry, but the cost of running this technique is only just started becoming accessible. The strongest comparison located is the difference between the Xbox One and Xbox series X, figure 5 displays the comparison between the consoles. The progress made between the generations have advanced enough to allow for RVTs on the newer generation of console.

Although the hardware has recently become available there are a few drawbacks to bringing RVTs into mainstream. Firstly, the sales from the newer console appear on a similar trajectory as the previous, but are now starting to fall off. With these sales taking place around 7 years apart, there would be an expectation that more consoles would be sold. According to Shewale (2023) there are 200 million active users a month but only 27.68 million people own an Xbox series X. Despite having the opportunity to get newer hardware, people are not buying them, this raises the question of there being a point to develop something for hardware that isn't selling. This is something developers will have to consider when creating products, as people will not have the accessibility to run their games without buying a newer generation of console.

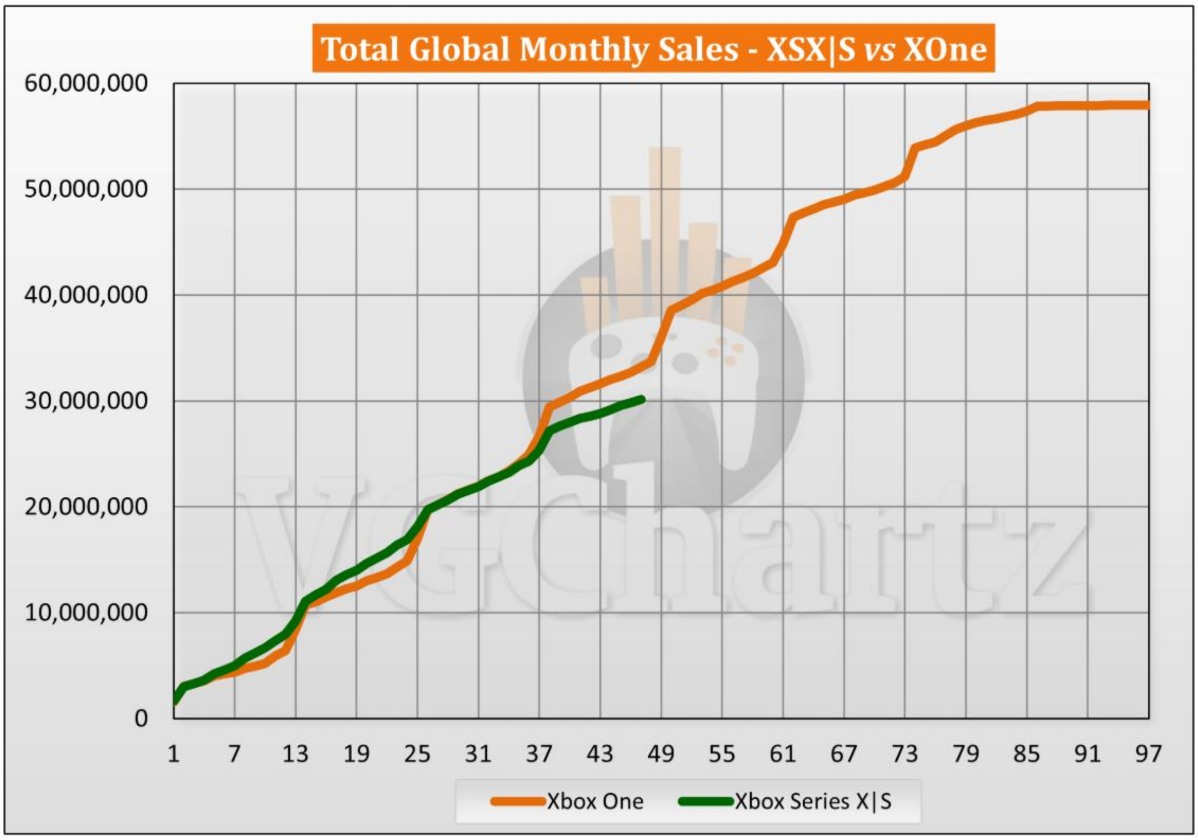

Figure 6 - Graph of sales (D'Angelo, 2024)

4.3 Case Study – Hogwarts Legacy (2023)

As stated many games aren't clear whether they use RVTs or not, so extensive research was carried out to try find an example of where RVTs were used compared to not. The one that was most prominent for the was the open world 2023 game Hogwarts Legacy, which was built in Unreal Engine 4. From observing the game through gameplay both in person and online clear comparisons were made for the two consoles. Figure 7 & 8 show the same piece of the map in Hogwarts Legacy, figure 7 displays ground blending well with the terrain clearly showing its use of RVTs. Figure 8 shows the same scene clearly lacking in the capacity to run RVTs and have been removed entirely.

Figure 7 - Asset Blending in Hogwarts Legacy (SkyCaptain, 2023)

Figure 8 - Lack of asset blending in Hogwarts Legacy (Robbie, 2025)

There is may bug fixing videos for Hogwarts Legacy, one of them by SystemDLL Tech (2023) is in regard to fixing textures. He demonstrates that there is an active memory pool that is used for rendering in Hogwarts, this further confirms that RVTs were in use with this game. In fact in the official website for the game (Avalanche, et al, 2024) goes into detail on the specifications needed to run the game at different levels of quality. Using this technique while making versions available for lower compatible hardware was intensive but inclusive. It shows that although RVTs are only compatible for some it is worth starting to include it when developing something new.

5. Untapped Potential and Future Directions

5.1 RVTs Current Downfall – Documentation and Tools

Unreal Engine's update has been available for 6 years yet still RVTs remain underutilised. Selective studios are starting to integrate this technique for terrain blending and rendering large worlds, many smaller studios and indie developers still avoid them. They tend to stick with what has accumulated more documentation due to the smaller budgets for these studios. With the lack of information provided on how to integrate this technique with others, and a smaller group of people exploring and experimenting, RVTs will not progress as quickly as intended into the mainstream workflow. This suggests there is much more to be explored with RVTs and will flourish with more documentation.

Even with this information, most RVT workflows are still manual, developers will need to have an in-depth understanding of UV mapping, tile streaming and the RVT node systems. For RVTs to become the norm an automated workflow or plugin will need to be implemented, even with increased documentation there will always be a favourable choice of automation over manual. Improving the technique in this fashion will likely increase adoption, especially as GPU-powered workflows could become standard.

5.2 Procedural Worlds and Dynamic Environments

Open/Procedural worlds have become more popular according to (Nabeel, 2025) "release of *Skyrim* and *Minecraft* in 2011 ushered in a golden age of open-world gaming – the ensuing decade saw many seminal open-world games that would take the genre to new heights". With RVTs supporting real-time updates this gives the possibility of using dynamic effects on the environments, this could take the form of footprints, vehicle tracks or even explosion and weathering effects without relying on conventional methods. This proposed idea was discussed with PrismaticaDev's (2025) video, and suggested that it would be a very hard system to implement as it could clog up the rendering process as each page is updated with these moving elements it could cause "hitching … and general performance issues". This was based on there being multiple actors spawned to simulate these footsteps, however this is not completely the end of this idea, as there is the option to destroy these actors. Therefore with more investigation into this technique it could be possible to design a system using RVTs to create these tracks.

6. Practical Implementation

6.1 Scene Justification

A scene was selected previous to the final outcome, this was chosen without knowledge of RVTs and the way that they work, this was based a market corner on old Turkish town. However once extensive research into the technique was carried out the scene was quickly abandoned as it lacked the ability to show RVTs effectively. It was important to mention this as there are still elements and screenshots taken when practicing techniques with this scene in Unreal Engine. The project now takes the form of a forest log cabin set in a specific memorable location. With a natural overgrown scene it will be the perfect habitat to experiment with runtime virtual textures, as they can work well with organic materials.

6.2 Asset Production Pipeline

The artefact of this project has taken form of a rural scene built in UE5, it takes shape of a moss ridden cabin in Scotland, the specific area was based on The Hermitage, located in Dunkeld, Perthshire. This place previously held the UK's record of tallest tree, and due to the independent significance it held for the developer of this project it was an ideal location to gather inspiration for the project. Multiple trips were carried out here across the period of the project to collect images for inspiration and to experience the lighting under different weather conditions to get an accurate depiction of how the light behaves in a forest.

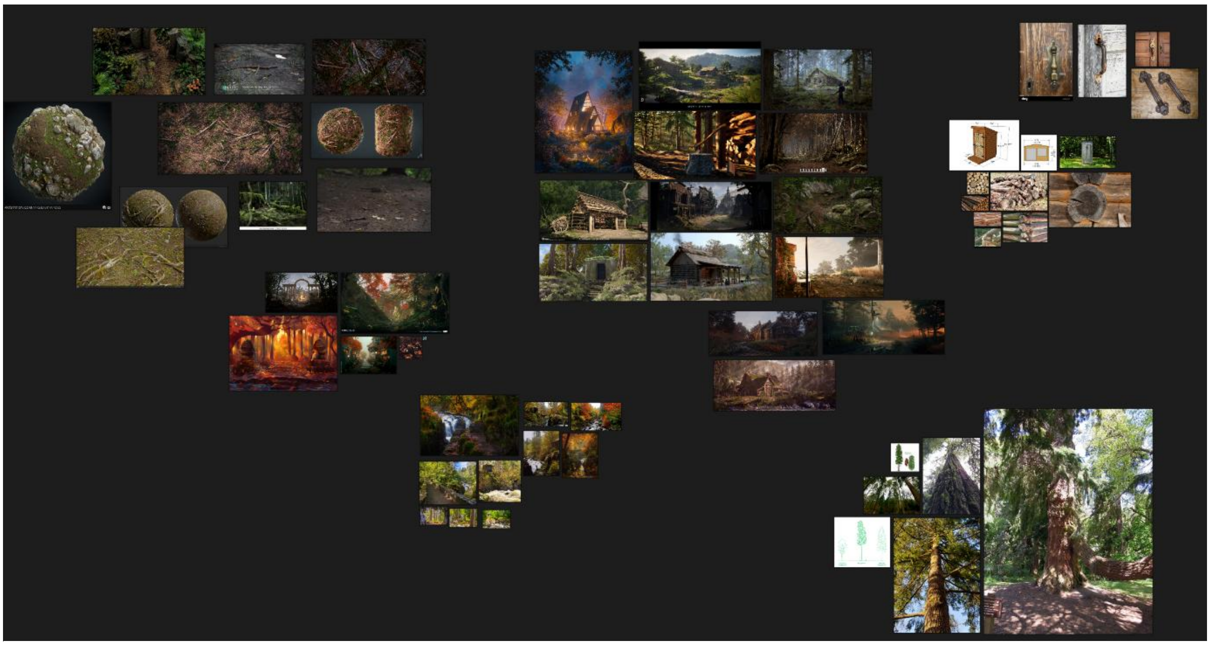

Figure 9 - Screenshot of PureRef File (PureRef, 2025)

Starting with the project references were taken from multiple sources online to ascertain sizes and materials for the assets, they were collected in PureRef and split into sections so specific references can be easily located, shown in figure 9. After a block out of the project was carried out in Sims 4 (2014) due to the familiarity of this game with the artist of this project this made sense for the project as materials could be applied quickly giving an understanding of scale and where the textures can be replicated across the scene reigning in the project.

Assets created:

- Log Cabin (Walls, Floor, Roof, Roof Log, Columns, Porch
- Out House (Outhouse, Roof)
- Tree
- 3 different Rocks
- Fence (one model split into parts)

The scene is formed largely with original assets that were created by the artist for this project, and the rest were outsourced online for set dressing purposes. A list of external assets and their creators will be attached to the bottom of this document. Each asset followed similar workflows starting in 3Ds Max to create low poly versions of the models, then taken into Zbrush to create a high poly model for the future baking process. This was crucial to stick to assets suitable for games as it keeps most of the detail in the textures rather than loading a high poly model saving processing time. The next step was baking in Substance Painter to successfully capture the high poly details onto the texture, many custom smart materials were created and used to keep consistency throughout the project, further upping the realism. Everything was then collated and brought into Unreal engine for a polished look where the landscape and lighting were built up to reflect the real world. Joe Taylor (2021) provided great insight on how to build up lighting in an Unreal scene.

6.3 Integrating Techniques

Creating vertex painting came easily as it had been set up by the artist in a previous project, however a YouTube tutorial from Procedural Minds (2023) acted as a reference to reflect towards to make sure things were kept on track. With the other technique – procedural generation – this was a new technique for the artist and many tutorials were referred to when creating this part of the project, fortunately there was a tutorial from Gorka Games (2024) which used the Procedural Content Generation (PCG) that was very helpful.

6.4 RVT Implementation

Proceeding with the RVTs were originally troublesome and intricate and tutorials were followed step by step to understand how they are crafted. Referring to the video accompanying the process document shows the steps taken to carry out this implementation. At first there was a learning curve to understand what each node is instructing, however after careful analysis it became apparent to the developer what everything was. From knowing what each node was used for when it came to combining the techniques it was easily manipulated in the exact way that the developer intended. Connecting the popular techniques with RVTs came relatively intuitive even for the new developer.

Figure 10 - Images taken before and after RVT implementation in UE5 (Robbie, 2025)

Despite the setup going fairly smoothly with the tutorials, there was a lot of machine struggle when trying to load everything in the scene. The computer used to process this scene crashed regularly while trying to load more intricate detailing. Although a machine capable of producing this technique was used it still produced problems. This could cause another drawback of RVTs being used in industry.

7. Critical Analysis and Reflection

7.1 Practical Reflection

As expected asset blending with environments worked well in the final result, it has clearly given the natural blending look that was intended. Although perpendicular faces showed a bit of struggle with the first initial scene this fell down to the structure of UVs when being read by RVTs. However an asset with a surface at an angle towards the surface blends with ease producing great quality blends as shown in figure 11. It also visually shows how RVTs interact with assets spread through a procedurally generated content (PCG) volume. As RVTs are applicable to the material itself it can be applied directly to single assets spread through the PCG process. This is further solidifies how effective this process is in connecting with techniques that are already widely implemented in industry today.

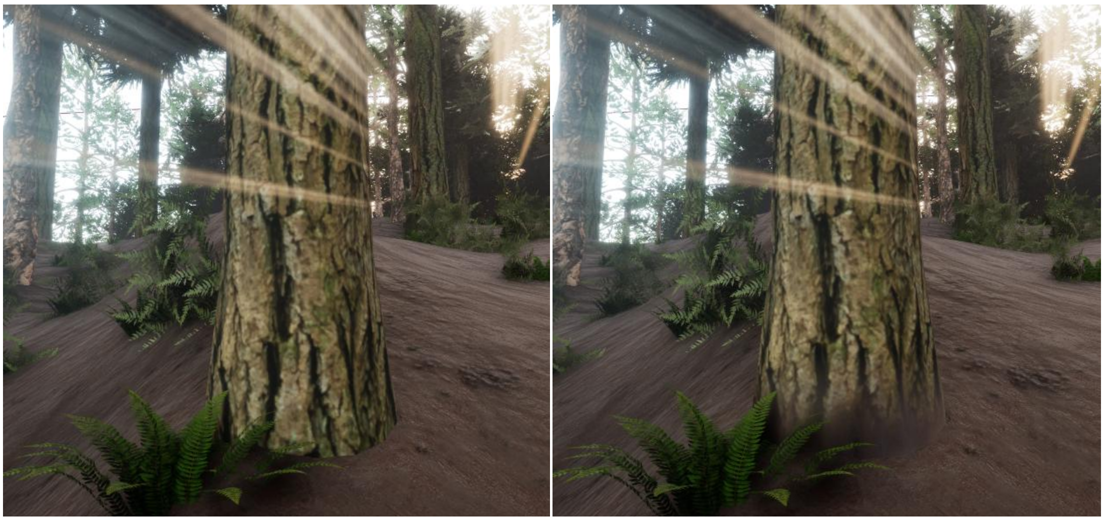

Figure 11 - Blending shown with procedurally generated assets. (Robbie, 2023)

7.2 Lessons Learned

Although most assets worked well with the scene and RVTs implementation with other techniques was fairly intuitive not every asset was perfectly blended. Next time with the project more investigation into the technique will be carried out before assets would be made for the project. This is one downfall where it limited time on the final practical project as the previous one was started in preparation for the technique, it would have been ideal to practice the technique with a scene already in existence before creating something new without full knowledge of what was needed.

7.2 Runtime Virtual Textures in Industry

Due to RVTs heavy reliance on the GPU is its largest downfall when deciding to implement into games. There is still a large market without the hardware to run games with this technique implemented, developers are reluctant to include something in their games when a large portion of the consumers are unable to use them. Companies will need to make a decision of creating something accessible or something GPU reliant.

8. Conclusion

Although the scene could have been more polished, the project clearly demonstrates where RVTs can be used practically in projects, as they can be used in conjunction with other techniques without having an impact on them and effectively enhance the scene by blending objects into the landscape. The project displays different instances of using RVTs across multiple assets further proving where they are useful in game development.

With the current lack of documentation on RVTs this project will be suitable to understand how RVTs interact with a scene built in Unreal Engine, as well as the structure on how to implement this through the process document. Therefore, should technology continue advancing runtime virtual textures be effectively integrated into modern procedural workflows in the near future, providing more documentation on implementation is provided.

9. Resources

9.1 Tech Used

Due to the nature of this project dedicated hardware and software were used. The minimum requirements needed – according to Maxon (2025) – for a PC to run the 3D modelling software have been investigated and the following is a comparison between what was required and what was used:

Required	Used
64-bit editions of Windows 10 or 11	64-bit editions of Windows 11
Intel 64-bit CPU or AMD 64-bit CPU.	Intel Core i9 16-Core processor
RAM: 4GB	RAM: 32GB (2X16GB)
20 GB of free hard drive space for ZBrush and its scratch disk	2TB
1280x1024p Monitor Resolution	3840 x 2160p Monitor Resolution
4GB VRAM	8GB VRAM

This is crucial as having a PC which is aimed towards supporting 3D graphics aided the production, the PCs provided by Glasgow Caledonian University (GCU) come equipped with the minimum requirements or higher and were ideal for this project. However as another PC with these specifications was available at a home residence the work was primarily carried out there. The software required was licensed through GCU and have been accessed through the PC, software used:

- Zbrush
- 3Ds Max
- Substance Painter
- Substance Designer
- Photoshop
- Unreal Engine

9.2 External Assets

Anwar, M.S. (2024). *ArtStation - 5 Wood Knots Digital Surface Brushes & Alphas Vol.15 - ZBrush 4R8+/Blender/Mudbox/3dcoat | Brushes*. [online] ArtStation. Available at: https://www.artstation.com/marketplace/p/lBg51/5-wood-knots-digital-surface-brushes-alphas-vol-15-zbrush-4r8-blender-mudbox-3dcoat [Accessed 23 Jul. 2025].

AspectStudios (2025). *Tree Stump*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/tree-stump-ba7a0d04459e4c8b81c02df1ac186170 [Accessed 5 Aug. 2025].

Cynerix (2019). *Game Ready Grass*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/gameready-grass-60a4f92d8226482ea32184f987ef0342 [Accessed 1 Aug. 2025].

Effiebop (2021). *Scattered Maple Leaves*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/scatted-maple-leaves-161fbc23a1234993a7d12006643b2621 [Accessed 4 Aug. 2025].

EFX (2017). *Pinecone*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/pinecone-photogrammetry-721359a5c0cd4cda8c5a378cd1e83494 [Accessed 4 Aug. 2025].

EvolvedUK (2022). *Pine-Tree*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/pine-tree-d45218a3fab349e5b1de040f29e7b6f9 [Accessed 1 Aug. 2025].

FenrixGT (2025a). *ArtStation - 100 Wood Brushes / Alphas - 1K JPG | Vol.3 | Brushes*. [Alpha] *ArtStation*. Available at: https://www.artstation.com/marketplace/p/jN9Dr/100-wood-brushes-alphas-1k-jpg-vol-3?utm_source=artstation&utm_medium=referral&utm_campaign=homepage&utm_term=marketplace [Accessed 6 Jul. 2025].

FenrixGT (2025b). *ArtStation - Vase Pack 1K/2K/4K – Game-Asset (FBX, OBJ, BLEND, Substance) | Resources*. [3D Model] *ArtStation*. Available at: https://www.artstation.com/marketplace/p/DNvdr/vase-pack-1k-2k-4k-game-asset-fbx-obj-blend-substance [Accessed 6 Jul. 2025].

GameDev Nick (2021). *Old Rusted Bucket v2*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/old-rusted-bucket-v2-267d9f0c78fe400fb315dd71e1b99852 [Accessed 4 Aug. 2025].

Pngaaa (2020). *Pine Tree Branches*. [Image] *Pngaaa.com*. Available at: https://www.pngaaa.com/detail/518770 [Accessed 31 Jul. 2025].

POLYSCAN (2023). *Fern Grass 01*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/fern-grass-01-b0a96490483a48d2877d2ed8abc9e436 [Accessed 31 Jul. 2025].

Sanforge Studio (2021). *3D Fern - 01*. *Sketchfab*. Available at: https://sketchfab.com/3d-models/3d-fern-01-165f3870237f488885faf406d9deddc0#download [Accessed 31 Jul. 2025].

TonyS (2024). *HDRi Space Spheremaps - Space Spheremaps*. [online] Space Spheremaps. Available at: https://www.spacespheremaps.com/hdr-spheremaps/ [Accessed 8 Aug. 2025].

Zheltkov, V. (2023). *Oil Lamp*. [3D Model] *Sketchfab*. Available at: https://sketchfab.com/3d-models/oil-lamp-67c0f2f7f83d44cf8d4b349bdbdf0aab [Accessed 4 Aug. 2025].

9.2 References

askNK (2019). *UNREAL 4.23 RELEASED - WHATS NEW?* [online] YouTube. Available at: https://www.youtube.com/watch?v=8Z0UFt9JtY8 [Accessed 6 Aug. 2025].

Avalanche, Portkey Games and Warner Brothers (2024). *PC SPECS*. [online] Hogwarts Legacy. Available at: https://www.hogwartslegacy.com/en-gb/pc-specs [Accessed 7 Aug. 2025].

Blender (2025). *Introduction - Blender 4.5 LTS Manual*. [online] Blender.org. Available at: https://docs.blender.org/manual/en/latest/sculpt_paint/vertex_paint/introduction.html?utm_source=chatgpt.com [Accessed 6 Aug. 2025].

Catmull, E. (1974). *A Subdivision Algorithm for Computer Display of Curved Surfaces*. [PDF] Available at: https://ohiostate.pressbooks.pub/app/uploads/sites/45/2017/09/catmull_thesis.pdf [Accessed 6 Aug. 2025].

D'Angelo, W. (2024). *Xbox Series X|S vs Xbox One Sales Comparison - September 2024*. [online] VGChartz. Available at: https://www.vgchartz.com/article/462859/xbox-series-xs-vs-xbox-one-sales-comparison-september-2024/ [Accessed 7 Aug. 2025].

Epic Games (2025). *Streaming Virtual Texturing in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community*. [online] Epic Games Developer. Available at: https://dev.epicgames.com/documentation/en-us/unreal-engine/streaming-virtual-texturing-in-unreal-engine?application_version=5.5 [Accessed 6 Aug. 2025].

Gorka Games (2024). *How to Make a Forest in Minutes with PCG in Unreal Engine 5*. [online] YouTube. Available at: https://www.youtube.com/watch?v=cD7Q-tu1DgU [Accessed 25 Jul. 2025].

InspirationTuts (2024). *Texture Painting Vs Vertex Painting: Which One Is the Best?* [online] YouTube. Available at: https://www.youtube.com/watch?v=oNysnAiVwHk [Accessed 6 May 2025].

Nabeel, I. (2025). *The Evolution of Open-World Games - GPWebsiteWIP*. [online] Gameopedia.com. Available at: https://gameopedia.com/blogs/the-evolution-of-open-world-games [Accessed 7 Aug. 2025].

Neu, A. (2010). *Virtual Texturing*. [PDF] pp.4–7. Available at: https://arxiv.org/pdf/1005.3163 [Accessed 6 Jul. 2025].

Obert, J., van Waveren, J.M.P. and Sellers, G. (2012). Virtual texturing in software and hardware. *ACM SIGGRAPH 2012 Courses*, [online] pp.1–29. doi:https://doi.org/10.1145/2343483.2343488.

PrismaticaDev (2025). *Runtime Virtual Textures: The Ultimate Starter Guide [UE5]*. [online] YouTube. Available at: https://www.youtube.com/watch?v=RLEPA16QDRw [Accessed 5 Jul. 2025].

PureRef (2025). *Moodboard*. [Image] *PureRef*.

Robbie, L. (2025). *Hogwarts Legacy Lack of RVTs on Xbox One Console*. [Image].

Sagar (2022). *MipMaps and Why They Are Useful - Sagar - Medium*. [online] Medium. Available at: https://medium.com/@rgbguy/mipmaps-and-why-they-are-useful-6d3e33c68b16 [Accessed 6 Aug. 2025].

Shewale, R. (2023). *39+ Xbox Statistics 2023 (Users, Market & Revenue)*. [online] Demandsage. Available at: https://www.demandsage.com/xbox-statistics/ [Accessed 7 Aug. 2025].

Sims 4. (2014). [Game] EA.

Sivaneswaran, T. (2021). *Runtime Virtual Texture (RVT) in Unreal Engine 4 - Arimac - Medium*. [online] Medium. Available at: https://medium.com/teamarimac/snow-surface-deformation-using-runtime-virtual-texture-rvt-in-unreal-engine-3f49ec4b3adb [Accessed 6 Aug. 2025].

Skycaptin5 (2023). *Hogwarts Legacy Xbox Series X Gameplay Review [Perfect] [Optimized] [120fps] [Ray Tracing]*. [online] YouTube. Available at: https://www.youtube.com/watch?v=gijUlz4Tp-Q [Accessed 7 Aug. 2025].

SystemDLL Tech (2023). *Hogwarts Legacy Textures Not Loading - FIXED *NEW**. [online] YouTube. Available at: https://www.youtube.com/watch?v=pWWCUJi3ys8 [Accessed 7 Aug. 2025].

Versus (2025). *Versus*. [online] Versus. Available at: https://versus.com/en/microsoft-xbox-one-vs-microsoft-xbox-series-x [Accessed 7 Aug. 2025].

Wilson, J. (2019). *Unreal Engine 4.23 released!* [online] Unreal Engine. Available at: https://www.unrealengine.com/en-US/blog/unreal-engine-4-23-released [Accessed 6 Aug. 2025].

9.2 Bibliography

Reference list

Bogdan, D. (2025). *Tropical Farmlands, Doru Bogdan*. *ArtStation*. Available at: https://www.artstation.com/artwork/a0XWG2 [Accessed 4 Jun. 2025].

Buhr, M., Pfeiffer, T., Reiners, D., Cruz-Neira, C. and Jung, B. (2022). *Real-Time Aspects of VR Systems*. *Real-Time Aspects of VR Systems*, Springer, pp.245–289. doi:https://doi.org/10.1007/978-3-030-79062-2_7.

Dong, T. (2025). *Page Restricted*. *Artstation.com*. Available at: https://www.artstation.com/artwork/K3QZko [Accessed 4 Jun. 2025].

Gamefromscratch (2019). *Unreal Engine 4.23 Released*. [online] YouTube. Available at: https://www.youtube.com/watch?v=KDh3hvJ_b24 [Accessed 6 Aug. 2025].

GSN Composer (2022). *What are Mipmaps? Texture Filtering in GLSL [Shaders Monthly #6]*. [online] YouTube. Available at: https://www.youtube.com/watch?v=qMCOX3m-R28 [Accessed 6 Aug. 2025].

Kennedy, R. (2025). *The Average Height of Humans Over Time*. [online] LIVESTRONG.COM. Available at: https://www.livestrong.com/article/542877-the-average-height-of-humans-over-time/ [Accessed 26 Jun. 2025].

Leyla Yvonne Ergil (2022). *The ins and outs of a Turkish village house*. [online] Daily Sabah. Available at: https://www.dailysabah.com/life/the-ins-and-outs-of-a-turkish-village-house/news [Accessed 6 Jul. 2025].

Mansour, M. (2025). *Scarlet Sunset*. [online] ArtStation. Available at: https://www.artstation.com/artwork/dy6m6x [Accessed 4 Jun. 2025].

Menzies, P. (2025). *Douglas fir (DF)*. [online] Forest Research. Available at: https://www.forestresearch.gov.uk/tools-and-resources/tree-species-database/131593-douglas-fir-df-2/ [Accessed 29 Jul. 2025].

MYP2 - Going Solo - Vocabulary List | Vocabulary.com. (2018). *Vocabulary.com*. [online] doi:https://doi.org/10.9624.96.

National Trust Scotland (2024). *Ossian's Hall*. [online] National Trust for Scotland. Available at: https://www.nts.org.uk/visit/places/the-hermitage/highlights/ossians-hall [Accessed 22 Jul. 2025].

Nevnihal Erdoğan and Izzet Yüksek (2013). *Traditional Window Designs of Kirklareli, Turkey*. [online] Bentham Science Publishers. Available at: https://ebookcentral.proquest.com/lib/gcal/reader.action?docID=1394959&c=UERG&ppg=1 [Accessed 6 Jul. 2025].

Ordnance Survey (2019). *Detailed maps & routes to explore across the UK | OS Maps*. [online] Osmaps.com. Available at: https://explore.osmaps.com/route/3994295/country-walking-the-hermitage--rumbling-bridge-perth--kinross?lat=56.55626&lon=-3.62108&zoom=14.0413&style=Standard&type=2d [Accessed 31 Jul. 2025].

SaveAs Yazılım (2025). *ISTANBUL CONSTRUCTION TECHNOLOGY | History of Istanbul*. [online] Istanbultarihi.ist. Available at: https://istanbultarihi.ist/703-istanbul-construction-technology [Accessed 6 Jun. 2025].

Taylor, J. (2021). *Dynamic Lighting & Environment Composition (Part 1)*. [online] Artstation.com. Available at: https://www.artstation.com/blogs/joetaylorart/X3mb/dynamic-lighting-environment-composition-part-1 [Accessed 22 Jul. 2025].

TJGM (2022). *What is Mipmapping in San Andreas?* [online] YouTube. Available at: https://www.youtube.com/watch?v=v0OVto8xv_k [Accessed 6 Aug. 2025].

Unity (2025a). *Introduction to mipmap streaming*. [online] Unity Documentation. Available at: https://docs.unity3d.com/6000.1/Documentation/Manual/TextureStreaming-introduction.html [Accessed 6 Aug. 2025].

Unity (2025b). *Texture Streaming*. [online] Unity Documentation. Available at: https://docs.unity3d.com/2020.1/Documentation/Manual/TextureStreaming.html [Accessed 6 Aug. 2025].

Unreal Engine (2025). *Runtime Virtual Texturing in Unreal Engine | Unreal Engine 5.4 Documentation | Epic Developer Community*. [online] Epic Games Developer. Available at: https://dev.epicgames.com/documentation/en-us/unreal-engine/runtime-virtual-texturing-in-unreal-engine?application_version=5.4 [Accessed 6 Aug. 2025].