jamk

Environment Design by using real-world location in Unity Engine

Juho Harju

Bachelor's thesis
November 2023
Business Information Technology

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Description

Juho Harju

Environment Design by using real-world location in Unity Engine

Jyväskylä: Jamk University of Applied Sciences, November 2023, 49 pages

Degree Programme in Business Information Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: English

Abstract

This thesis explores the dynamic landscape of game environment design using Unity Engine, addressing challenges and opportunities in crafting immersive environments. The industry's trajectory towards realism prompted an exploration of tools and techniques to integrate real-world data into Unity Engine, aiming to enhance the authenticity of virtual landscapes.

The primary task involved exploring and implementing tools for importing real-world geographical data into Unity Engine. Objectives included understanding the significance of tools like Maanmittauslaitos (NLS) data, Unity Terrain Tools, file format conversion, different assets in creating realistic game environments. The goal was to seamlessly blend geographical accuracy with creative design.

The implementation phase involved a systematic utilization of the identified tools and resources. Maanmittauslaitos data served as the foundational real-world dataset. Unity Terrain Tools enabled terrain sculpting and texture application, while file format conversion ensured data compatibility. Custom 3D assets, crafted in-house, contributed to the unique atmosphere of the game and gameplay controller facilitated scene performance testing.

The integration of geographical data and Unity tools resulted in a dynamic and visually compelling game environment. Real-world accuracy coexisted with creative elements, enhancing the overall gaming experience. The study revealed the synergy between the chosen tools, emphasizing their role in crafting an engaging and immersive digital world.

In conclusion, the study not only successfully demonstrates the efficacy of integrating real-world data into Unity Engine for game environment design but also sheds light on the inherent challenges. The sheer volume and intricacy of assets required for achieving realistic natural environments pose a formidable task. This finding contributes to the ongoing discourse on the delicate equilibrium between realism and creativity within the gaming industry, emphasizing the need for a nuanced and sustainable approach to asset creation.

Keywords/tags (subjects)

Level design, environment design, computer games, data import, heightmap data, action research

Miscellaneous (Confidential information)

–

Contents

Figures

1 Introduction

The objective of this project is to examine the use of Unity in environmental design, generating a realistic representation of a real-world location for an upcoming game set in the 80s in Lake Finland. This work strives to identify the key design elements that enhance the effectiveness of virtual environments and to develop a method for creating precise and immersive virtual environments for video game production, storytelling, and gameplay purposes.

We are living in an era where the implementation of real-world locations in video games has increasingly gained traction. The creation of lifelike and immersive environments has proven to be crucial for players to engage fully with the game world and storyline. With the introduction of Unity, game developers are now equipped with highly sophisticated tools to construct stunningly realistic and immersive environments.

The thesis will explore the potential benefits and challenges of utilizing real-world locations when building a game scene. Specifically, it will address the need to balance between replicating the real-world location and making adaptations that better suit the game's mechanics and setting.

An integral part of this project will be understanding the constraints and possibilities associated with stand-alone VR, multiplayer aspects, and FPS games. We will delve into how different art designs can impact the overall gaming experience and how to choose the best methods and tools to bring real-life locations into a game world, ensuring a high level of fidelity to the original locale.

This exploration will draw upon a range of resources, including official documentation, tutorials, and industry best practices. Once the project is completed, a comparative study will be conducted to determine whether the game environments accurately mirror their real-life counterparts, considering factors like art design, VR limitations, and the balance between real-world and game locations.

2 Project

2.1 Real-World Environment Design

Creating realistic virtual environments that represent real-world locations necessitates a blend of accurate geographical data, intricate 3D models, and progressive rendering techniques (Calleja, 2011; Totten, 2014). The use of real-world data, including satellite imagery and elevation maps, permits developers to construct precise terrain and landscapes. Additionally, photogrammetry and other modeling techniques facilitate the creation of highly detailed 3D assets (Unity Technologies, 2023a; 2023c).

As Calleja (2011) noted, "It was seen as the culmination of a long history of the creation of virtual environments that attempted to give viewers the sensation that they were actually there" (p. 17). Hence, merging geographical data, advanced modeling, and rendering techniques is crucial for achieving this sensation of real-world immersion in virtual environments.

2.2 Reasons for thesis.

Project “Sommerspel”

Developers of this game project include Juho Harju, Tero Korpela, Jani Seppälä and Ari-Pekka Oinonen. Game is set on Finland and simulates cabin life for player and players. FPS / VR game that is going to be published for PC and standalone VR platforms. Project started in May, 2023 and development cycle began on August 10. The team is invited to Pocket Gamer Helsinki, to do Big Indie Pitch.

Primary Object

The principal objective of this thesis is dual-faceted: firstly, it aims to delve into the capabilities and limitations of the Unity game engine in terms of environmental design; secondly, it seeks to develop a systematic methodology for building intricate and immersive virtual environments. These environments are designed to serve various purposes such as video game production, storytelling, and gameplay. By concentrating this study on a distinct real-world location, it enables a

more focused examination of the unique challenges and effective strategies associated with crafting realistic environments in Unity Engine.

3 Production Methods

In the context of this thesis project, action research will be employed to actively engage with the challenges and opportunities presented using Unity Engine in environment design with real-world locations. Action research aligns with the project-based approach, as it emphasizes problem-solving, reflection, and iterative improvement.

This study aims to address the following questions.

3.1 Research Questions

What tools and methodologies are available for the seamless integration of real-world environment data into Unity Engine, and how do these tools contribute to the overall design process of immersive virtual environments?

In the realm of real-world environment data, which primary sources and datasets emerge as pivotal for integration into Unity Engine? How do these sources enrich the virtual environment creation process, and what criteria are essential for selecting the most suitable datasets?

Exploring the landscape of utilizing real-world data in Unity Engine, what advantages and disadvantages characterize this approach in the design of virtual environments? How do these factors influence decision-making processes during development, and what considerations are paramount for achieving an optimal balance between realism and performance?

3.2 Research Design

As described by the Indeed Editorial Team, Action Research is typically structured into five phases, which include identifying a problem, gathering data, interpreting data, taking action, and evaluating outcomes (Indeed Editorial Team, 2023.). For the purposes of this thesis research, a modified

adaptation of these five phases will be employed, with the aim of optimizing learning, performance, and the success rate of the prototype build.

Plan In this phase, the research will begin by planning the integration of real-world data into Unity for environment design. This includes identifying tools, selecting data sources, and formulating a plan for the project.

Act The "Act" phase involves implementing the planned changes and actions, such as using new 3D models or importing real-world data into Unity and creating the game world.

Observe During the "Observe" phase, the effects and outcomes of the actions taken will be systematically observed and documented. This includes evaluating the performance, visual quality, and overall effectiveness of the game environment.

Reflect The "Reflect" phase involves critical reflection on the observations and outcomes. It aims to identify what worked well, what encountered challenges, and what improvements can be made.

Iterate based on the reflections, adjustments and refinements will be made to the environment design and the integration of real-world data. This iterative process ensures ongoing improvement.

4 Alternative or future research ideas.

While this study delves into the integration of real-world locations in Unity Engine for environment and level design, there remain several avenues for future research to contribute to the expanding field of virtual environment creation. The following ideas and suggestions could serve as stepping-stones for subsequent investigations.

4.1 Advanced Procedural Generation Techniques

Exploring advanced procedural generation techniques within Unity Engine could deepen our understanding of how dynamically generated environments, informed by real-world data, can enhance realism and immersion in games. Investigating algorithms for procedural content generation and their impact on resource efficiency could be a promising avenue. Unreal Engine 5 offers PCG tools (Unreal Engine 5, procedural content generation overview) while Unity Engine has tutorials available in Udemy. (Udemy 2022, Penny de Byl)

4.2 Comparative Analysis with Other Game Engines

Conducting a comparative analysis with other game engines, such as Unreal Engine (Epic Games. Unreal Engine 5. n.d.a) and Godot (Godot. n.d.a) could shed light on the unique strengths and weaknesses of Unity in the context of environment design. Exploring how different engines handle real-world data integration and the subsequent impact on game development could be a valuable contribution. In theory this could help future game developers choose best engine for their needs.

5 Methodology

The knowledge base was gathered through a systematic selection of data derived from scholarly authors, peer-reviewed articles, game developers, and pertinent literature within the realm of game production, with a specific emphasis on game design and level design. Calleja and Totten's works provided invaluable insights into various facets of game design, aligning with established consensus in other areas of game production-related research. Arnold's contribution to the knowledge base was realized through his examination of immersion within a game-based learning environment. Gregory offered insight into game engines and rendering techniques. Harmonious perspectives on immersion and aesthetics were discerned in the works of Brown and Cairns. Christopher W. Totten's book, "Level Design: Processes and Experiences," served as a comprehensive resource, cross-referencing findings, and experiences from a diverse array of game developers and authors.

The theoretical foundation was intentionally structured to concentrate on game, level, and game design topics, deliberately excluding subjects related to game mechanics. This emphasis aimed to

foster the development of immersive and realistic environments. The selection of source materials was guided by their relevance to the overarching goal.

Regarding the choice of tools for the development and production phase of the prototype environment, existing tools, notably Unity Engine and Blender, were deemed sufficient and were not replaced. The decision to stick with these tools was rooted in the game development team's prior experience with Unity Engine and their proficiency in virtual reality game production. The team's familiarity and expertise with these tools were pivotal, as there was no need to invest time in acquiring new skill sets. Notably, new data required for the project was meticulously constructed and seamlessly integrated. Additionally, the geographical locations in Finland were selected based on easy access and their status as remote cabins, prerequisites essential for advancing the production process.

Various methods of geographic data gathering were explored, with the most common methods chosen. Most suitable for a game set in Finland was, unsurprisingly, the National Land Survey of Finland.

6 Game Design

The impact of game design is substantial as it profoundly affects the player's experience and overall enjoyment of a game. One fundamental component of game design is the creation of engaging environments and level design, which significantly influence the player's immersion, engagement, and interaction with the game world (Calleja, 2011; Totten, 2014).

According to Bleszinski (2000), meticulously designed environments can considerably enhance the overall atmosphere and believability of a game world, enabling players to feel truly immersed in the experience. Realistic environments, built with attention to detail and precise representations of real-world locations, can trigger powerful emotional responses and create unforgettable moments for players.

Level design focuses on the layout, structure, and flow of gameplay within an environment. A well-designed level is not just visually appealing, but also provides engaging challenges, clear objectives, and a sense of progression for the player. The creation of a harmonious and enjoyable experience necessitates a delicate balance of aesthetics, functionality, and gameplay elements (Juego Studio, 2022).

In contrast to the traditional approach, Juegoadmin from Juego Studio insists that effective level design plays a crucial role in maintaining player interest and immersion by shaping player progression and subtly guiding the player through intuitive design and pacing (Juegoadmin, 2022).

Both environment and level design contribute immensely to the success of a game, and their importance should not be neglected during the development process. By utilizing advanced tools and techniques, such as those provided by Unity, game designers can create immersive, realistic, and engaging virtual worlds that leave a lasting impression on players (Unity Technologies, 2023b).

6.1 Environment Design

Environment design is a vital aspect of game development, which involves creating the physical world where the game occurs. This includes designing and modeling 3D objects, terrain, buildings, and other items, as well as establishing the atmosphere, aesthetics, functionality, narrative, lighting, and weather conditions to make the game world feel alive and immersive (Arnold, 2017). In this section, we will discuss several key aspects of environment de-sign in greater detail.

In Red Dead Redemption 2, Rockstar’s game. (Rockstar Games, n.d.) The world of the American Wild West is painstakingly detailed. It encompasses vast plains, snowy mountains, bustling towns, and dense forests. The careful crafting of each area contributes to the game's immersion, offering players a sense of a living, breathing world.

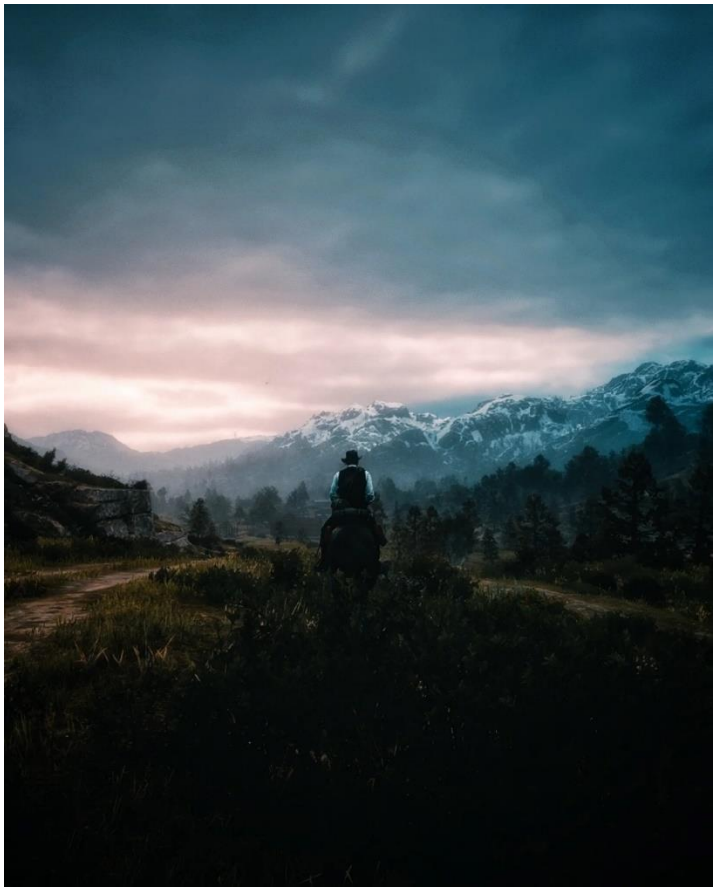

Figure 1: Red Dead Redemption 2, (jshot21_Nov.2021)

Figure 2: Red Dead Redemption 2 (reddeadgem_Aug.2022)

Aesthetics

Aesthetics in environment design refers to the visual appeal and artistic direction of a game world. Designers must consider a variety of factors to create a visually engaging environment. These include color schemes, which as Totten (2014, Chapter 4, p. 176) suggests can significantly influence a player's emotions and mood. However, unlike Totten's focus on color schemes, Brown and Cairns (2004, pp. 1297-1300) delve more into the psychological aspects of gameplay, identifying three levels of game-based immersion: engagement, engrossment, and total immersion. Contrary to previously published studies that focused solely on the visuals, Brown and Cairns present a broader perspective, asserting that the choice of aesthetics should bolster these levels of immersion, thereby enhancing player engagement and emotional connection to the game's narrative and setting. Contrasting with Totten's and Brown and Cairns' viewpoints, Arnold (2017) emphasizes the impact of immersive environments in game-based learning, suggesting that carefully crafted aesthetics can also facilitate player learning and engagement. Thus, a well-considered aesthetic design in a game world not only supports player immersion but can also contribute to game-based learning experiences.

Functionality

Functionality in environment design involves creating spaces that facilitate gameplay and enable players to interact with the game world in a meaningful way. This includes designing levels with clear objectives, providing navigational cues, and ensuring that the environment supports the game's mechanics and systems. Designers must strike a balance between aesthetics and functionality, creating environments that are visually appealing while also promoting enjoyable and engaging gameplay (Totten, 2014; Arnold, 2017).

Narrative

Environment design can contribute to the narrative of a game by providing visual storytelling elements and creating a sense of place. Designers can use environmental details, such as graffiti, signage, or architectural styles, to convey information about the game world and its inhabitants. These

elements can help build the backstory of the game, establish the setting, and deepen the player's connection to the world (Totten, 2014; Arnold, 2017).

Player Interaction

A well-designed game environment encourages meaningful player interaction, allowing players to explore, discover, and engage with the world. Designers must consider player agency, creating environments that offer choices and opportunities for various playstyles. This may include providing multiple paths through a level, designing interactive objects, or encouraging exploration with hidden areas or collectibles (Totten, 2014; Arnold, 2017).

In conclusion, environment design in video games is a multifaceted discipline that requires a careful balance of aesthetics, functionality, narrative, and player interaction. By considering these elements, designers can create immersive and engaging game worlds that enhance the overall experience for the player.

6.2 Game Engines

A game engine is a software framework that facilitates the creation and operation of video games, providing the necessary tools and features. The choice of an engine can significantly impact the outcome of a game as each engine has its strengths and weaknesses (Gregory, 2014; Koster, 2013). Among these, Unity and Unreal Engine 5 stand as some of the most popular game engines for creating high-quality 3D games, with other engines such as Godot and CryEngine also being widely used.

Game engines have significantly evolved over the past few decades, revolutionizing the game development process and enabling developers to create increasingly complex and immersive experiences. The history of game engines dates back to the 1980s and 1990s, with pioneering engines like the Doom engine and the Quake engine paving the way for modern game engines (Wolf, 2008)

Unity stands as a robust engine, known for its user-friendly interface, flexibility, and broad platform support. It is particularly praised for its suitability for indie developers and those new to

game development due to its simplified scripting languages and comprehensive tutorial support. However, compared to Unreal Engine 5, Unity does not have latest technological tools or variations. Nanite is a new way of doing distant object rendering (Unreal Engine, Nanite, Documentation). terms of graphic fidelity and built-in features for larger, more complex games. On the other hand, Unreal Engine 5, while requiring a steeper learning curve, excels in offering advanced lighting and rendering capabilities and a visual scripting system, beneficial for more graphically intensive projects.

Today, game engines like Unity, Unreal Engine 5, Godot, and CryEngine offer a comprehensive suite of tools and features that allow developers to create high-quality games across various platforms. These engines streamline the development process, enabling developers to focus on designing and implementing gameplay mechanics, environments, and assets without worrying about low-level programming and optimization tasks (Gregory, 2014; Koster, 2013).

The accessibility of game engines has improved greatly compared to 20 years ago. Many modern game engines offer free versions or affordable licensing options, lowering the financial barriers for indie developers and hobbyists. Additionally, extensive documentation, tutorials, and community support are available for these engines, simplifying the learning process for new developers (Unity Technologies, n.d.; Epic Games, n.d.; Godot Engine, n.d.; Crytek, n.d.).

While pre-built game engines offer numerous advantages, some developers might choose to create custom game engines tailored to their specific needs. Custom game engines can provide greater flexibility and control over the development process, potentially leading to better performance and optimization. However, building a custom game engine requires a significant investment of time and resources and extensive knowledge of programming, rendering, and other technical aspects of game development (Gregory, 2014).

In conclusion, advancements in game engines over the years have democratized the game development process, making it more accessible to developers of all skill levels and backgrounds. The availability of powerful and easy-to-use game engines has opened new opportunities for creative expression and innovation in the gaming industry.

7 Contents

7.1 Unity Engine

Unity, a versatile game engine developed by Unity Technologies, is recognized for its compatibility across various platforms and its widespread use by indie developers and established studios alike (Unity Technologies, 2023a). Although Unity may not be as visually advanced as some other engines, it provides developers with an array of tools and features that allow them to create detailed and immersive environments.

7.2 Environment Design in Unity

Unity provides a comprehensive suite of tools and features for designing realistic environments, including terrain sculpting, material creation, and advanced lighting systems. Notably, Unity supports the integration of real-world data like satellite imagery and elevation maps, enabling accurate terrain and landscape creation. Compatibility with photogrammetry software such as Agisoft Metashape and RealityCapture further allows developers to generate detailed 3D models based on real-world objects (Agisoft, 2023a; Capturing Reality, 2023a).

Creating immersive and accurate environments in a game engine like Unity requires meticulous attention to detail, rigorous research, and the effective use of various tools and techniques. Designing an environment based on a real-world location, like a cabin by a lake in Finland, involves considering unique contextual elements and gathering relevant reference materials for accuracy and authenticity.

For a game set in the 1980s, developers face the challenge of recreating a time period that can't be directly observed or photographed. Overcoming this obstacle requires thorough research into archival materials, historical photos, and local records, which are vital for gaining insights into the location's appearance, culture, and atmosphere during that time.

Designing a realistic environment based on a lakeside cabin in Finland during the 1980s within Unity presents both opportunities and challenges. The authenticity of a real-life setting can enhance player immersion and emotional connection to the game world (Calleja, 2011a; Totten, 2014a). Additionally, setting the game in a specific historical period can provide unique storytelling and gameplay opportunities, potentially educating players about the past.

However, balancing the environment's accuracy with gameplay considerations remains a primary concern (Totten, 2014). While fidelity to the original setting is important, developers also need to ensure that the environment supports engaging gameplay experiences, potentially leading to alterations or compromises in the environment's layout, scale, or other elements.

Potential limitations of available reference materials and the time and effort needed for detailed research also present challenges. In some cases, developers might need to rely on artistic interpretation and creative problem-solving to fill in gaps in available information.

In conclusion, designing a realistic environment based on a lakeside cabin in Finland during the 1980s within Unity is a complex process. This process requires careful research, planning, and execution. Developers need to leverage the capabilities of modern game engines and balance the demands of historical accuracy and gameplay to create immersive and engaging virtual worlds.

7.3 Comparison of Environment design in 2D / 3D

While both 2D and 3D game environments can be engaging and immersive, 3D environments generally offer a higher level of realism and interactivity. Designing 3D environments requires a deeper understanding of spatial relationships, lighting, and materials, as well as more advanced tools and techniques for modeling and rendering (Calleja, 2011; Totten, 2014e).

2D Design

2D environment design focuses on creating game worlds using two-dimensional graphics, typically employing a combination of sprites, tiles, and background images. 2D environments can range from simple, stylized designs to highly detailed and intricate worlds, depending on the artistic direction and intended gameplay experience (Rogers, 2014).

Some advantages of 2D design include lower production costs and reduced technical complexity, making it more accessible for smaller development teams or those with limited resources. 2D environments often rely on a strong visual style and effective use of color, composition, and animation to create engaging and immersive game worlds (Daley, 2012).

3D Design

3D environment design involves the creation of game worlds using three-dimensional graphics, which allows for greater depth, detail, and realism. 3D environments are constructed using a combination of models, textures, and materials, as well as advanced techniques such as lighting, shadows, and reflections to create a convincing and immersive game world (Totten, 2014d).

The increased complexity and technical requirements of 3D design can present challenges for developers, including higher production costs and the need for specialized tools and expertise. However, the advantages of 3D environments include a greater sense of immersion and interactivity, as players can navigate and explore the game world from multiple angles and perspectives (Calleja, 2011).

In conclusion, both 2D and 3D environment designs have their unique strengths and challenges. The choice between 2D and 3D design largely depends on the intended gameplay experience, artistic direction, and available resources for the development team.

8 Tools

8.1 Unity Engine Tools

Unity Engine equips developers with a suite of resources essential for crafting a dynamic and realistic game environment. Leveraging diverse tools and data sources, this section explores the tools employed in the project, emphasizing the synergy between geographical data, terrain sculpting, file format conversion, asset management, and gameplay controllers. That were used to craft the gameplay location.

Maanmittauslaitos (NLS, National Land Survey of Finland) data stands as the primary source of real-world geographical information. Offering elevation details, terrain data, and geographical coordinates, it forms the foundational dataset for the in-game environment (NLS, n.d).

Unity's Terrain Tools are pivotal for shaping the game world. These tools enable terrain sculpting, texture application, and the integration of real-world geographical data, fostering the creation of immersive and visually compelling landscapes (Unity Technologies, Terrain 2023).

File Format Conversion is a critical process in the pipeline, ensuring seamless integration of geographical data into Unity. Converting data into compatible formats, such as ASC files, facilitates the incorporation of real-world features into the digital realm (FileFormat.com. ASC File Format 2023).

The Unity Asset Store emerges as a versatile resource, offering a diverse array of 3D models, textures, and assets. These elements enhance the visual quality of the game environment, contributing to a more immersive player experience (Unity Technologies, Asset Store).

Gameplay Controllers play a pivotal role in testing and refining the scene's performance. Ensuring smooth interactions within the game world, these components are integral for crafting an engaging player experience. In Unity Engine, CharacterController is the component that is either built of modified for each game project. (Unity Technologies, CharacterController. API)

Custom 3D Assets, contribute to the unique atmosphere of the game. These assets are carefully designed to personalize the game world and enhance its overall aesthetic. Ari-Pekka Oinonen, professional 3D modeller has crafter models for development. And some external assets got used for testing purposes.

By seamlessly integrating these tools, custom and external. The development process ensures a successful blend of real-world features and creative elements, fostering a captivating and immersive gaming experience.

8.2 External / Other Tools

External

In addition to the built-in tools provided by UE5, there are several external tools and resources that can be used to enhance environmental design. These include heightmap and satellite data from sources like the USGS National Map, NASA's SRTM and ASTER Global Digital Elevation Model, terrain.party, heightmap.skydark.pl, and Tangrams Heightmapper (USGS, 2021; NASA, 2021; Terrain.party, n.d.; Skydark, n.d.; Tangram, n.d.). Photogrammetry software such as Agisoft Metashape and RealityCapture can also be used to create highly detailed 3D models based on real-world objects (Agisoft, 2021; Capturing Reality, 2021). Furthermore, data used by Land surveying institute of Finland (Maanmittauslaitos, 2023) offers relevant data for this thesis and project.

While Unity Engine provides a comprehensive set of tools for environment design, external tools can also play a crucial role in creating high-quality assets and textures. Adobe Photoshop and other image editors, as well as additional specialized tools, can enhance the overall quality and efficiency of the environment design process.

8.3 Image and texture editing.

Adobe Photoshop is a widely-used image editing software that can be utilized to create, modify, and optimize textures for 3D models in game environments (Adobe, n.d.). By working with layers, blending modes, and a variety of filters and adjustments, designers can achieve realistic and visually appealing textures for their models. Photoshop also supports various file formats, including those compatible with Unity Enginge, ensuring seamless integration between the two tools. Icons and animations are perfect example. (Rogers, 2014a).

Other Image Editors

Alternative image editors, such as GIMP and Krita, can also be used for creating and editing textures. These open-source options offer a range of features and capabilities similar to Adobe Photoshop, enabling designers to produce high-quality textures while working within their preferred software environment (GIMP, n.d.; Krita, n.d.).

Substance Suite

The Substance Suite, which includes Substance Painter, Substance Designer, and Substance Alchemist, is a powerful set of tools for creating procedural materials and textures for 3D models (Adobe, n.d.a). Substance tools enable artists to create realistic and complex materials with control over various parameters, making it easier to achieve the desired look and feel for game environments (Adobe, n.d.a). The Substance Suite also integrates with Unity Enginge, allowing for efficient asset creation and iteration (Adobe, n.d.a).

In conclusion, external tools like Adobe Photoshop, GIMP, Krita, and the Substance Suite can play a significant role in environment design by providing additional capabilities for creating and editing textures and materials. The integration of these tools with Unity Engine allows designers to streamline their workflow and achieve visually stunning and realistic game environments.

8.4 External / 3D Modelling and Blender

Blender is a powerful, open-source 3D modeling and animation software that can be used in conjunction with Unity Engine to create custom assets for game environments (Blender Foundation, n.d.). With a wide range of features, such as modeling, sculpting, texturing, rigging, and rendering, Blender offers a comprehensive toolset for creating high-quality 3D models and animations (Blender Foundation, n.d.).

Custom Models for Finnish Aesthetics

Blender can be utilized to create custom models that represent Finnish aesthetics, capturing the unique architectural styles, design elements, and cultural aspects found in Finland. By creating models that accurately represent the local culture and environment, game developers can enhance the authenticity and immersion of their game set in Finland (Totten, 2014a). Research into Finnish architecture and design trends, as well as reference photos and historical data, can inform the creation of these custom models, ensuring they accurately represent the intended setting (Wiebe, 2018).

Performance and Graphical Details

When creating custom models in Blender, it's essential to consider performance and graphical details. Designers must balance the level of detail in their models with the need for efficient rendering and optimization in the game engine (Rogers, 2014). By using techniques such as level of detail (LOD) models, designers can create assets that maintain visual quality while minimizing performance impact (Unity Technologies, n.d.a). Additionally, Blender's powerful texturing and materials tools can be used to create visually appealing models without relying solely on high polygon counts (Blender Foundation, n.d.).

In conclusion, Blender is a valuable external tool for creating custom 3D models and assets for game environments, particularly when capturing specific cultural aesthetics, such as those found in Finland. By considering performance and graphical details, designers can create high-quality assets that enhance the visual appeal and immersion of their game while optimizing for efficient rendering in Unity Engine

9 Research

During the research phase before commencing game world development, location scouting was crucial. It involved exploring various landscapes, structures, natural elements, and seasonal conditions. In August, the team visited Central Finland, specifically "Jani's family mökki”, a cabin nestled by a picturesque lake with inspiring small islands. At that point, we made the decision to use this real-life location as the foundation for our game. Subsequently, in October, our team ventured to an off-grid cabin in Central Ostrobothnia. This log cabin seamlessly aligned with the game's overarching theme, and the late autumn timing provided varying atmospheres with both rainy and snowy weather. In contrast to the Central Finland location, this cabin was situated next to a dry lake, now transformed into a swamp, offering a solitary and people-free environment.

9.1 Locations explored.

Kolonjärvi

A lake in Keuruu (Keuruu 2023), lansdape is said to be “a hilly and forested municipality in terms of its terrain, and its nature reflects the presence of the Suomenselkä ridge. Keuruu has many lakes

and wetlands. The main village of Keuruu is also situated by the water. Of Keuruu's total land area, slightly over a tenth is covered by water. The largest lake is Keurusselkä, including Ukonselkä” The decision to select Kolonjärvi as the primary location was motivated by the team's easy access to explore the site whenever necessary.

Figure 3: Kolonjärvi, (GoogleMaps)

Toholampi Wildreness

Toholampi Wilderness offers the team a unique opportunity to experience off-grid cabin life next to an old lake that has transformed into a swamp. In the Toholampi region, over half of the land consists of swampland, with very few lakes. This location provides valuable insights for future reference and accommodates different project requirements in the future (Toholampi, 2023).

Figure 4: Toholampi WIldreness, (Google Maps)

9.2 Heightmap Data transfer process.

Before the game location can be built, and specific location has been chosen. Real-world data must be acquired. For this project we will be using terrain data, Elevation Model 2m that depicts the height of the terrain above sea level. And the height values are in accordance with the system N2000, that is Finnish implementation of the common Europe Vertical Reference System. (FMI, Finnish Meteorological Institute, 2023)

As stated on NLS (National Land Survey of Finland, 2023) Elevation model 2 m is a model depicting the elevation of the ground surface in relation to sea level. Its grid size is 2 m x 2 m. The dataset is based on laser scanning data, the point density of which is at least 0.5 points per square meter.

Elevation model 2m.

The product's coverage is based on nationwide laser scanning. In some parts of the outer archipelago or the eastern border, the elevation model is not available. Elevation model 2 m is produced in two quality classes: the elevation accuracy in class I is on average 0.3 metres and the elevation accuracy in class II varies between 0.3 metres and one metre. NLS (National Land Survey of Finland, 2023)

Acquiring the data from NLS.

According to NLS Through the service you can order various printed map products. Besides topographic and basic maps, you can order map printouts. You can choose the area you want to print out as well as the size and scale of it. (NLS, n.d)

Thus, the team ordered information on the exact locations.

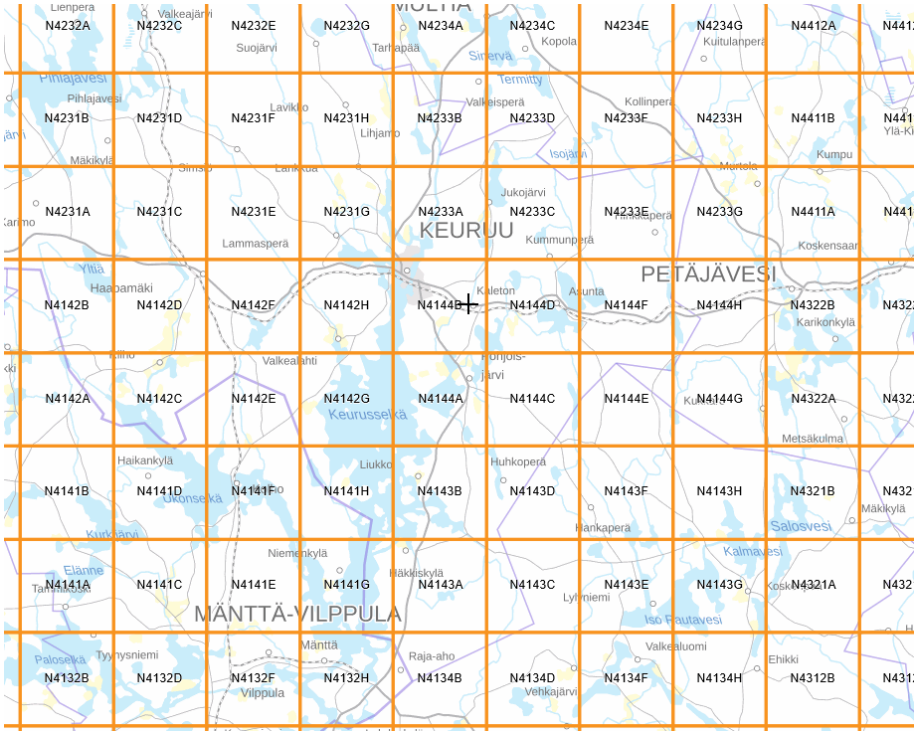

Figure 5: Grid Data, (NLS)

9.3 Planning

Realworld Data into Unity

The next step involved integrating real-world data into Unity. To achieve this, the team utilized Land surveying institute of Finland data. Tero, leveraging his programming expertise, proceed to transfer the data transfer to Unity, a process pivotal for generating heightmaps and terrain. A Zip file (Microsoft, n.d) with ASC file contains all the height data in text format, which was acquired previously from NLS (National Land Survey of Finland). For more information about the ASC file format, you can refer to the documentation available on FileFormat.com (FileFormat.com, n.d.).

```
ncols         3000
nrows         3000
xllcorner     392000.000000000000
yllcorner     6900000.000000000000
cellsize      2.000000000000
NODATA_value  -9999.000
 149.057 148.779 148.529 148.326 148.170 148
 148.966 148.761 148.592 148.403 148.234 148
 148.939 148.742 148.576 148.424 148.275 148
 148.921 148.749 148.579 148.368 148.237 148
 148.965 148.796 148.635 148.409 148.211 148
 148.968 148.852 148.730 148.503 148.268 148
```

Figure 6: ASC, (Tero Korpela)

The tool Tero Korpela made for Unity and “Sommarspel” project, reads the Zip file, and uses the ASC data to build the terrain according to the cell size, using *Elevation model 2m.* Height Data Importer

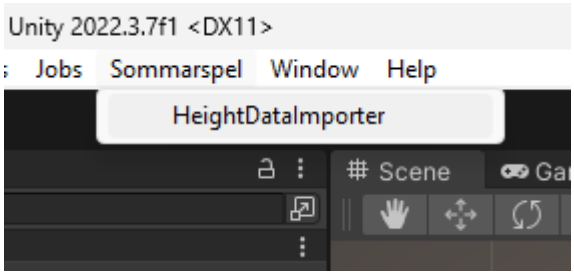

Figure 7: HeightDataImporter, (Tero Korpela)

Difference with Realworld Data and Gamification.

However, for optimal player enjoyment, the real island had to undergo significant transformation, including scaling it up. In addition to this, AP constructed billboards around the lake, a strategic move aimed at mitigating performance issues. Simulating each tree individually was deemed too resource-intensive for the engine. The primary objective here was to ensure that the game remained lightweight, especially to accommodate standalone VR.

9.4 Implementation

Heightmap Data files

In the context of our research, we are using Tero Korpela's Unity Engine's tool. When researcher opens the window, one will find a 5x5 grid of buttons designed for selecting the relative positions concerning other specific grid squares. For example, if we intend to import data for a 2x3 grid area from a map, we select the corresponding grid square within the importer. Then, we proceed to select the 'Select heightmap file' button to designate the height data file for that chosen grid square.

Figure 8: Height Data Importer, grid sections (Tero Korpela)

Files are downloaded in the background, eliminating the need to wait for the previous import to finish before starting the next one. Data transfer can be monitored on the progress bar in Unity's lower-right corner.

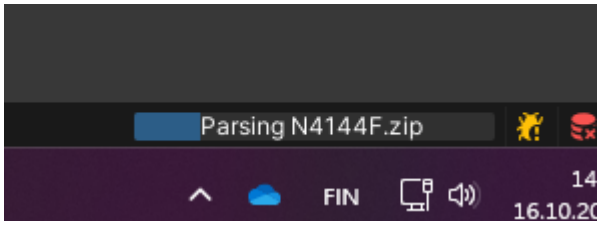

Figure 9: Data Transfer, (Tero Korpela)

Transfer of data

Upon the selection of preferred grids or cells contained within the zip files, activation of the 'Import and generate' button initiates the creation of Unity's terrain. The generated terrains adhere to Unity terrain's default settings as a default configuration. For this project, grids N4144D and N4144F were selected. Refer to Figure 1 for further details.

Completion of Data Transfer

The tools functioned as intended; however, there was an issue with the height data, which was mistakenly set to a Y scale of 600, instead of the actual values. This led to the creation of artificial mountains. After several attempts with the import process, a bug was identified and resolved by Tero Korpela. (See Figure 10 and Figure 11 for visual representations).

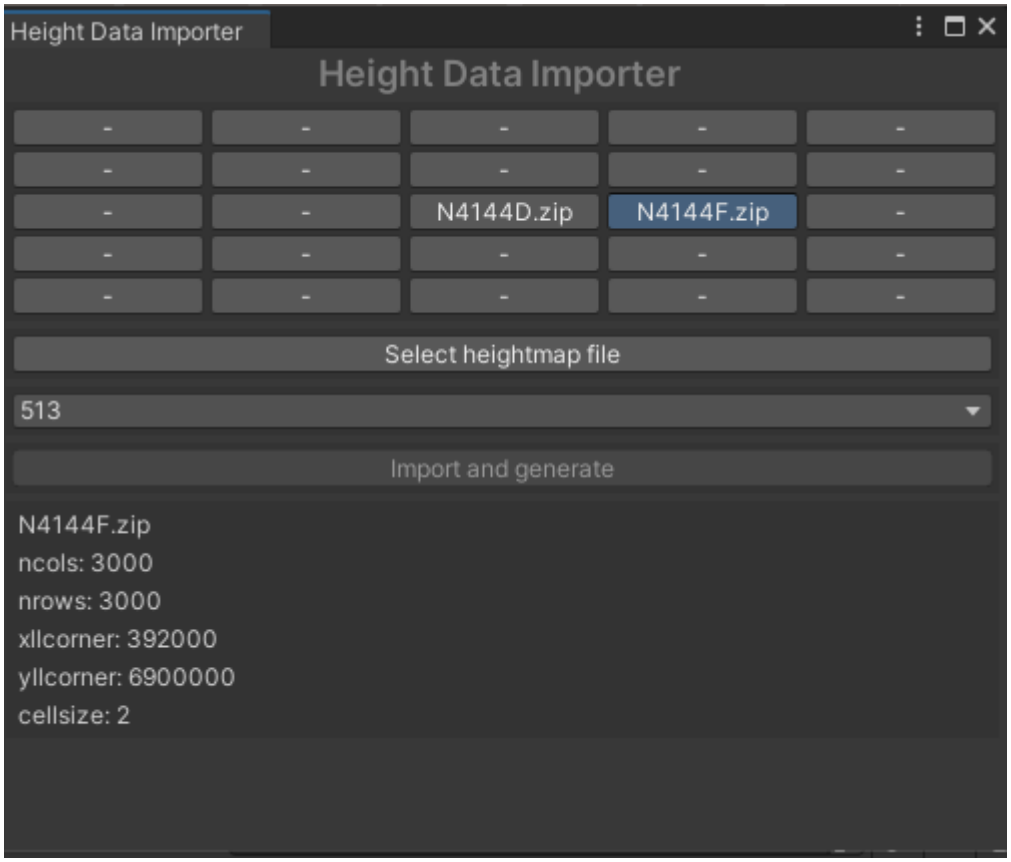

Figure 10: N4144D, N4144F Grids (Juho Harju)

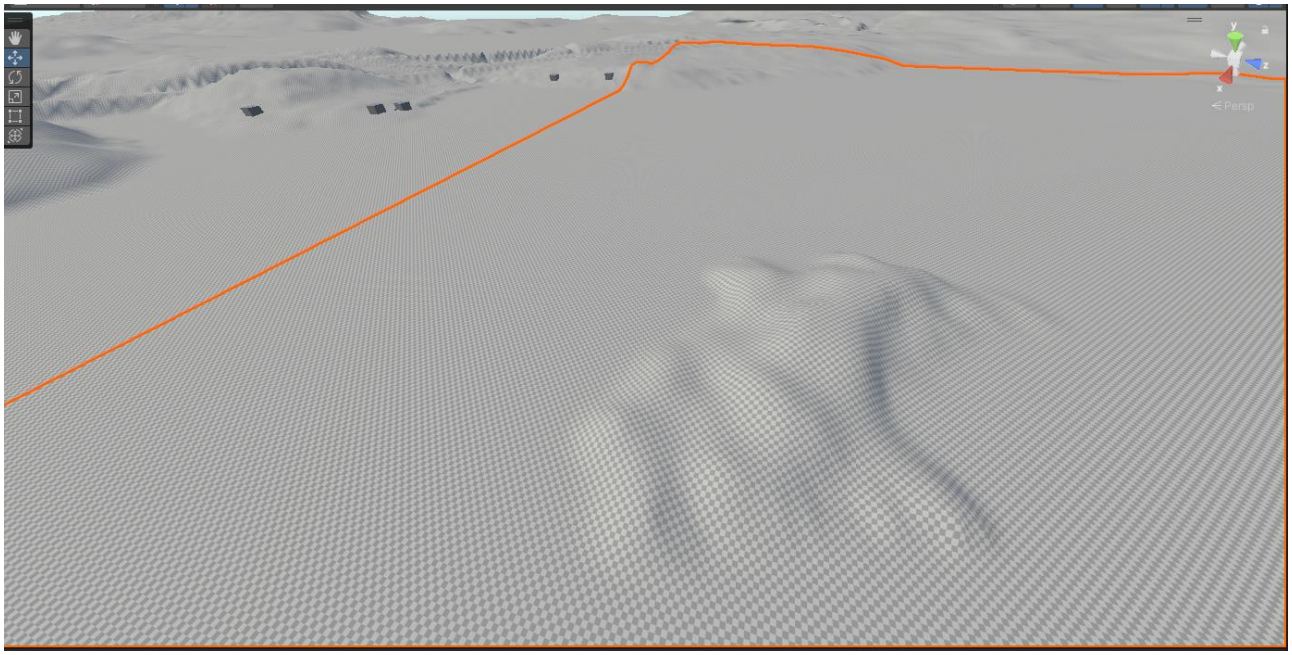

Figure 11: Blockouts and terrain generated, (Juho Harju)

9.5 Reflection

Successful Implementation

The tool employed in this process proved to be user-friendly and effectively transferred real data into Unity Engine's terrain generation. This approach offers a viable option for developing games with real-world map data, providing a swift and lightweight alternative. Notably, lakes are rendered distinctly in the terrain generation.

Challenges Encountered

The data, as indicated by the National Land Survey of Finland (NLS, 2023), is not without its imperfections. The elevation accuracy within Class I averages 0.3 meters, and Class II's elevation accuracy varies between 0.3 meters and one meter. These discrepancies may become noticeable to game developers and players when exploring the in-game terrain. For example, certain areas contain numerous trees that may compromise the terrain data's accuracy, resulting in the creation of

nonexistent mountains and ridges. Furthermore, NLS does not provide data for the generation of lakes or rivers.

Additionally, manually sculpting lakes or other specific features within the game world is a time-consuming process and may lead to these areas standing out from the rest of the game locations. To mitigate these challenges, developers may need to rely on their skills and employ more advanced terrain tools.

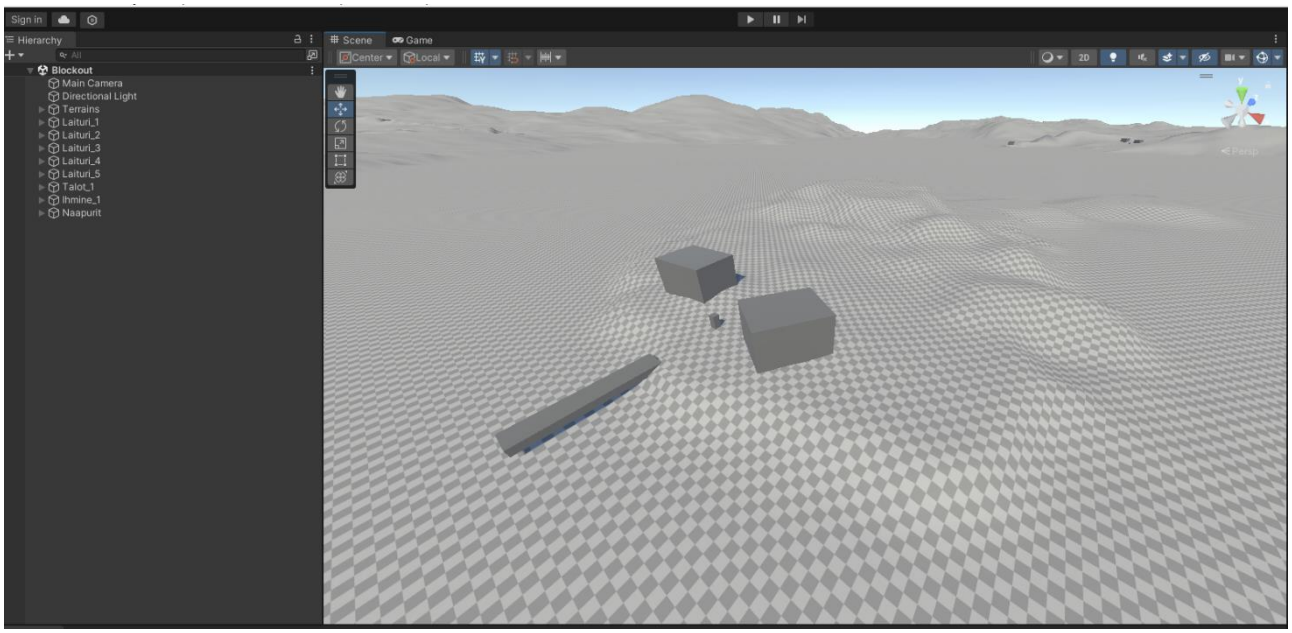

Figure 12: Island, Lake & Blockout, (Juho Harju)

9.6 Iterate

Enhanced Location Development

While the employed tool serves as a solution for swift and realistic terrain generation, further sculpting and editing of the terrain using Unity Engine tools is essential for achieving a perfectly realistic game location. Experienced developers may consider crafting custom tools to facilitate the terrain editing process. In the case of Tero Korpela's tools, the addition of automatic texture generation with layers would be an ideal enhancement.

Enhancements for Future Projects

In future endeavors, opting for a real-world location or an island within a lake is preferred over manual location crafting. Although this may limit some creative and artistic liberties, it aligns with the primary objective of prioritizing immersive, real-world game locations.

10 Concepting

10.1 Kolonjärvi

Real life location.

Location itself is located in Centra-Finland, Keuruu. Some data on the lake area.

- Surface height: 129,7m
- Coastline: 12,007 km
- Surface Area: 2,43149 km²
- Volume: 0,0105051 km³
- Medium depth: 4,32 m
- Maximum depth: 14,83 m

In the lake, it is estimated that there are 13 islands. Their total area is 5 hectares, which is approximately 2% of the total lake area. The Largest one is about three hectares in size, rest of them are either one or under one hectare (Kolonjärvi (Keuruu), 2023)

Game world location

As the original locations differ a lot from the ideal prototype location. Combination of the data, and level design techniques must be used to craft perfect gamified location for players to explore. The small island was made larger with the landscaping and terrain modification tools. And placeholder assets were used to decide locations of interest.

As the island is very small in real world. Sculpted island was built on its place. And this island is very different from generated data which is hard to mimic by just using standard Unity Engine tools.

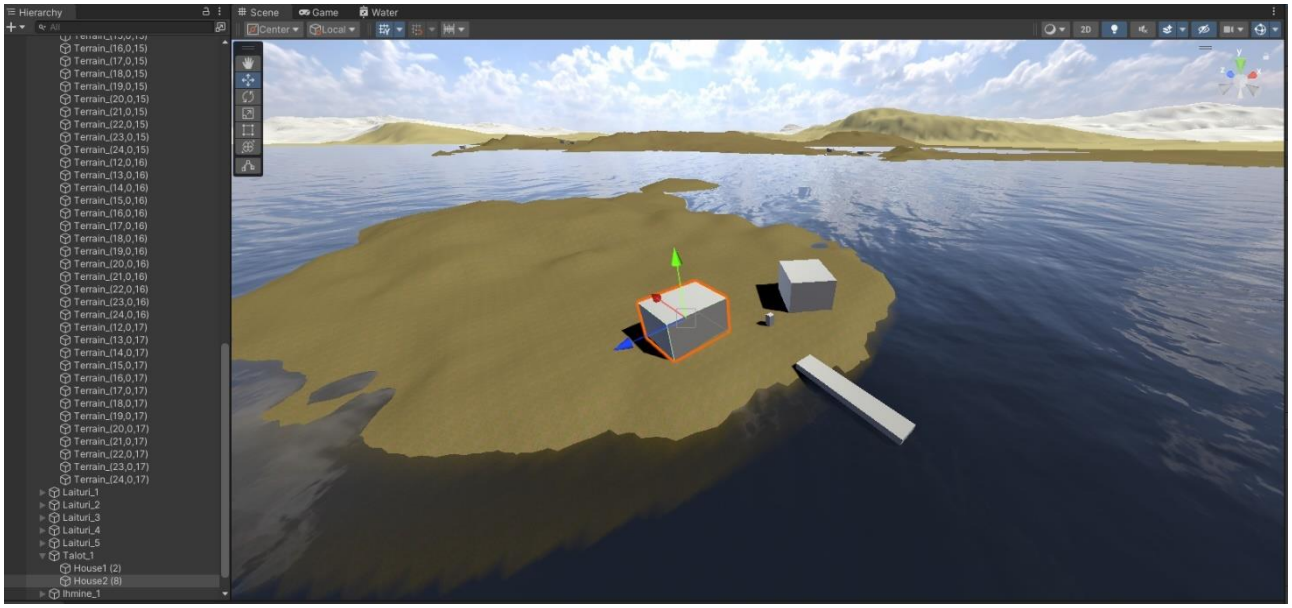

Figure 13: Island, Water shaders (Juho Harju)

10.2 Implementation

Initial Blockout

In the pursuit of developing a comprehensive understanding of the game world, a blockout of the environment is generated. This blockout comprises cubes designed to resemble a 2-meter-tall individual and buildings measuring 4 to 5 meters in height. Each box represents either a cabin or a house situated near a lake. Additionally, docks and water elements are incorporated into the scene to enhance the perception of spatial relationships within the game world.

Scene Modification for Design Adherence

Integrating the island base into the environment is an engaging and creative process. Utilizing Unity's terrain tools, basic shapes and terrain alterations are executed with efficiency. However, as previously mentioned in the "Heightmap Data Transfer" section, this island does not exist in real-

ity. Consequently, its appearance deviates significantly from the other islands within the lake. Furthermore, the neighboring island, which rivals the primary location for the prototype build in size, is situated too closely to the shore.

10.3 Blockout

The concept of blockouts in level design is discussed in depth by authors, Christopher W. Totten, Patrick Sotiriou, David Jagneaux, and Roberto Dillon in the "Level Design Processes" section of the book "Level Design: Processes and Experiences." The authors highlight how blockouts are used as a foundational step in level creation, providing insights into the role of blockouts in shaping the gameplay experience.

Blockouts, also known as grayboxing or whiteboxing, are a fundamental phase in level design that serves as the foundation for creating immersive and functional game environments. During this phase, designers construct the basic structure and layout of a game level before adding detailed visuals and interactive elements. This critical stage is informed by the principles of spatial design, player flow, and gameplay mechanics. The goal is to build a playable prototype that allows developers to test and refine their concepts.

By employing blockouts, level designers are empowered to build the foundation of a game environment that not only functions seamlessly but also offer an engaging and immersive experience for players. It is a testament to the meticulous planning and iterative design process that lies at the heart of every well-constructed game level.

11 Adding assets to the scene

11.1 Nature assets and Cabins.

Utilization of Nature Assets

For the prototype build, Ari-Pekka, a seasoned 3D artist, contributed nature and cabin assets. This marked his venture into crafting vegetation and natural assets for a gaming project. Nonetheless, these assets are tailor-made to optimize performance for projects VR production. Although there may exist more visually striking vegetation assets in the market, they are not tailored to the specifications of this project. It is possible, however, that third-party assets could be leveraged to explore methods for enhancing the visual appeal of vegetation in the game project.

Assets made by Ari-Pekka

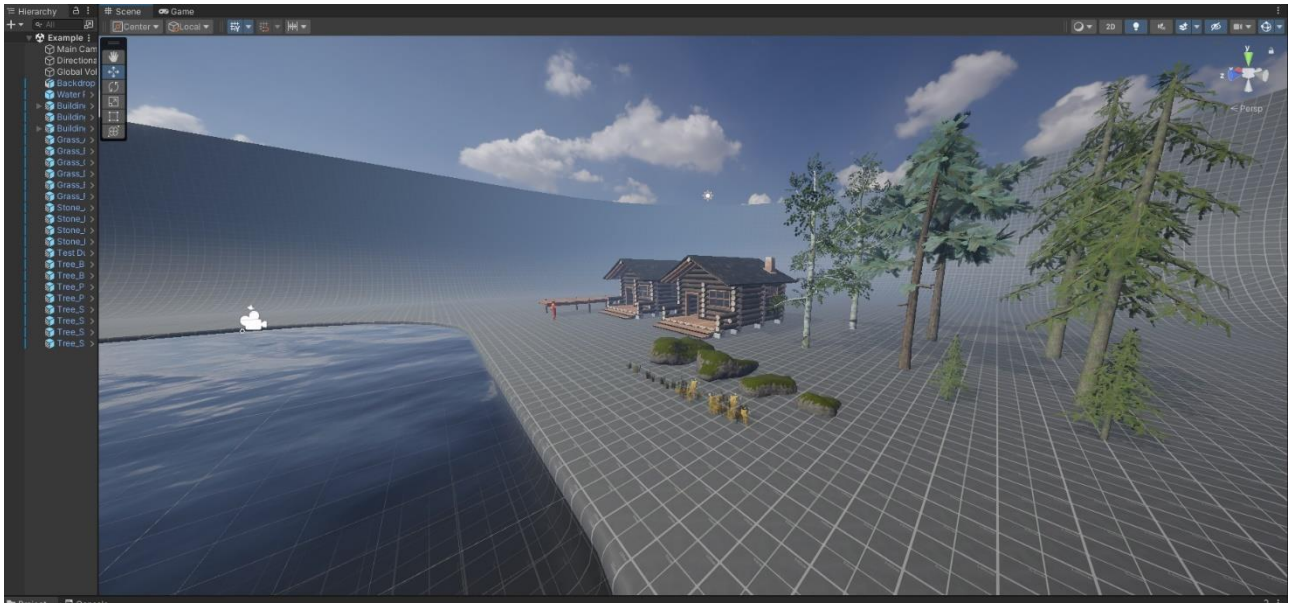

Figure 14: Cabin, Stones, Water Shader, Skybox, Trees, (Ari-Pekka Oinonen)

Textures Utilized

The textures employed for the game project originate from Poly Haven, and these assets are distributed under the CC0 license. This licensing framework grants us the freedom to utilize these

textures for any purpose, including commercial projects. For further details, please visit the web-site at Poly Haven. (2023). https://polyhaven.com/

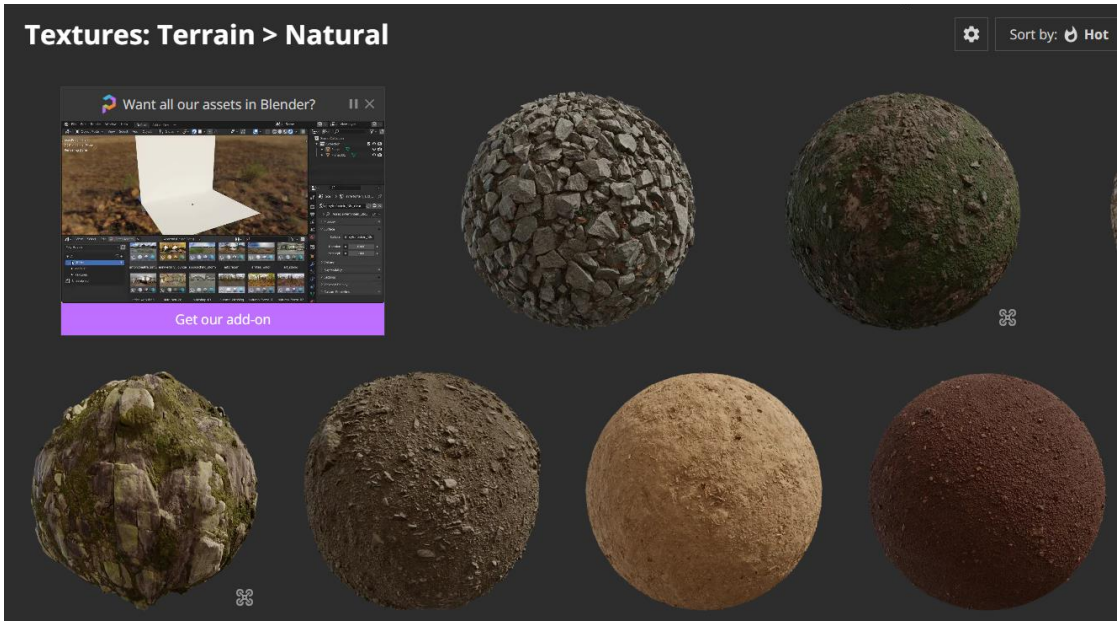

Figure 15: Textures (PolyHaven)

Figure 16: Textures in use (Juho Harju)

Textures project needed were forest ground, mud, coastal sand, stone texture. Ari-pekka provided moss texture. And of course, textured the assets we are using in the project. The moss texture captures Finnish forests immersion and blends the terrain together seamlessly with stones that are also covered with moss.

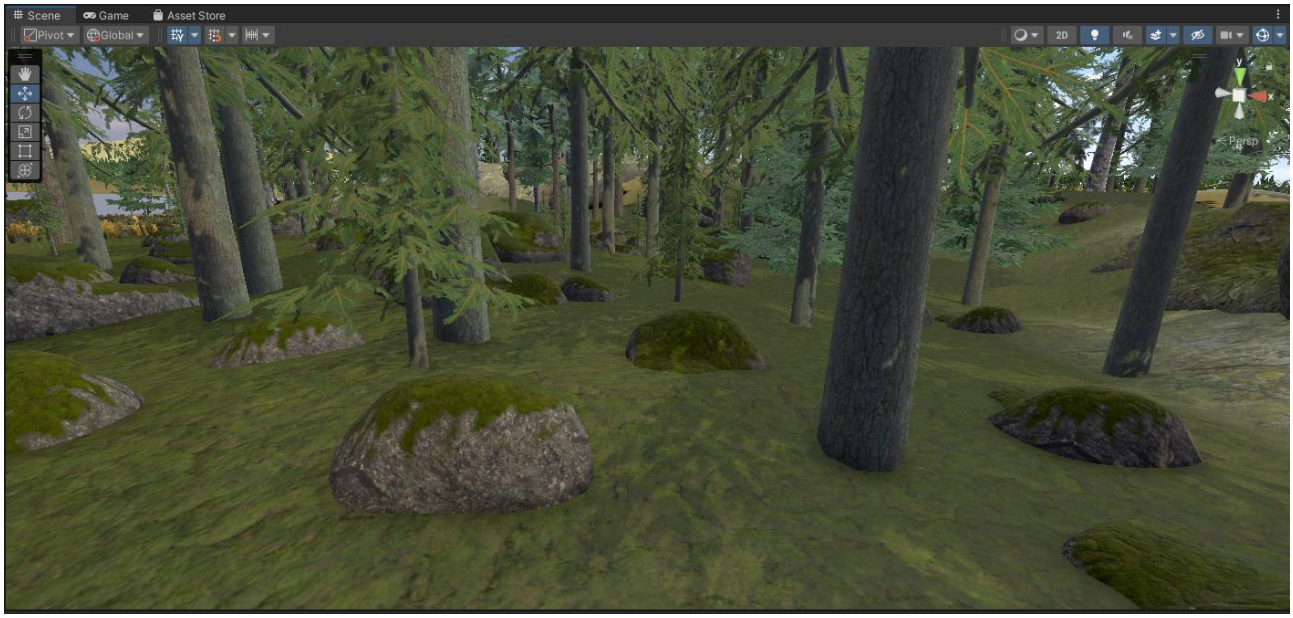

Figure 17: Moss forest, (Juho Harju)

During testing, we incorporated high-quality tree assets from NatureManufacture into our project. These assets, while visually impressive, are notably more resource-intensive compared to our in-house assets. However, they are designed with six different levels of detail (LOD), enabling their utilization as distant trees to enhance the overall vitality of the in-game world.

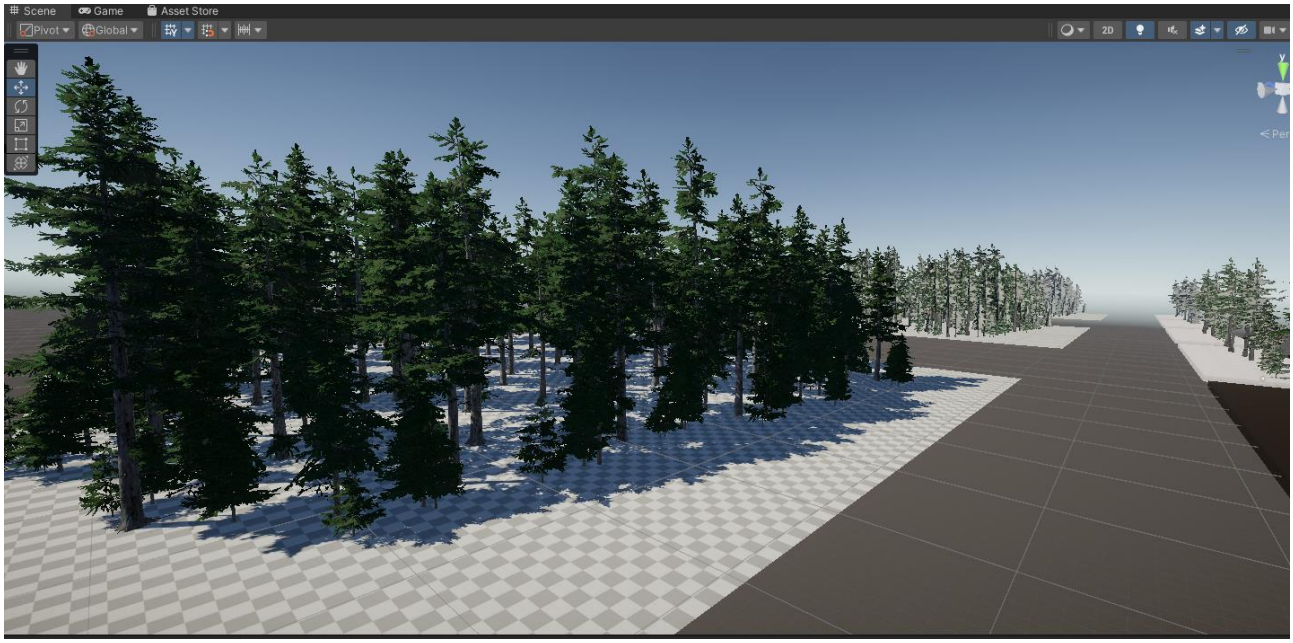

Figure 18: Mountain Tree Pack, (NatureManufacture)

11.2 Asset Transfer and Terrain Painting

With the concept phase completed, the focus shifted towards implementing assets into the prototype build. Ari-Pekka, our skilled 3D artist, created the necessary 3D assets and nature objects for the scene. Simultaneously, the terrain was painted using textures sourced from Poly Haven.

This phase marked the transition from conceptualization to the practical implementation of assets, bringing the virtual environment one step closer to reality.

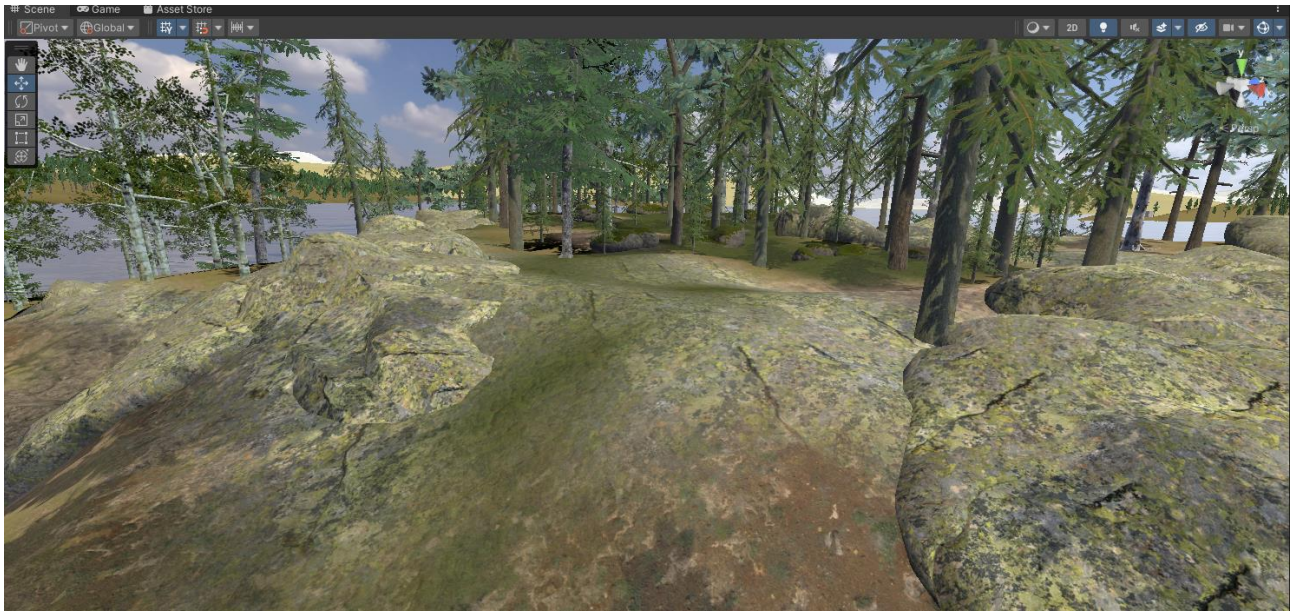

Figure 19: Terrain surface & cliff formations, (Juho Harju)

11.3 Asset Implementation

3D Assets Implementation

To integrate the 3D assets into the project, Ari-Pekka generously provided a Unity package containing all his professionally crafted assets, including the skybox texture and water shaders. These assets, created specifically for this project, operated seamlessly within the game and did not impose a heavy performance burden.

Texture Implementation

Custom terrain layers were created using .PNG file format textures, encompassing both diffusion and normal maps. Additionally, a rock model was enhanced using a Poly Haven texture to introduce graphical variety, particularly along the coastlines and cliffs. This phase involved the practical integration of assets into the project, enhancing the visual and environmental elements for the prototype build.

Figure 20: Cabin, player housing. (Juho Harju)

11.4 Reflection

Custom 3D Assets: Pros and Cons

Utilizing custom 3D assets in a game contributes to its unique character and realism. Professional 3D modelers can create assets that closely resemble real-world objects while ensuring they are optimized for smooth gaming experience. These custom assets offer artists the flexibility to tailor the game's aesthetics to precise design specifications. However, a challenge arises when considering the sheer volume of assets required for a game, even when the main game area is relatively small. Moreover, if the 3D modeler lacks prior experience with vegetation assets, extensive fine-tuning is necessary to replicate real-life locations. Capturing the essence of Finnish nature within Unity Engine presents a complex task for a single 3D and 2D artist.

Textures: Sourcing vs. Self-Creation

To alleviate the workload on artists, it is often advantageous to incorporate pre-existing high-quality assets such as textures. However, the use of intricate, high-quality nature assets may have drawbacks, particularly in mobile VR applications, where complex models can strain performance. Nonetheless, for prototype builds and presentations to potential funding partners or publishing studios, these high-quality assets can help convey an impressive visual quality, making them suitable for pitching purposes.

11.5 Gameplay controller.

Testing the scene's performance necessitates the incorporation of a gameplay controller. Notably, the project does not utilize Steam VR or Oculus default controllers. This choice is driven by the need for the controller to seamlessly switch between desktop mode and VR as required. An initial version, crafted by Jani Seppälä, was employed to assess the scene's frame rates. Promisingly, at this graphical level, the desktop build consistently achieved 100 frames per second (fps).

The next phase involves implementing a basic management system and attempting to create an Oculus Quest 3 build, with the aim of running the game on a standalone VR headset. However, this phase demands additional time for implementation due to some internal build challenges associated with the headset.

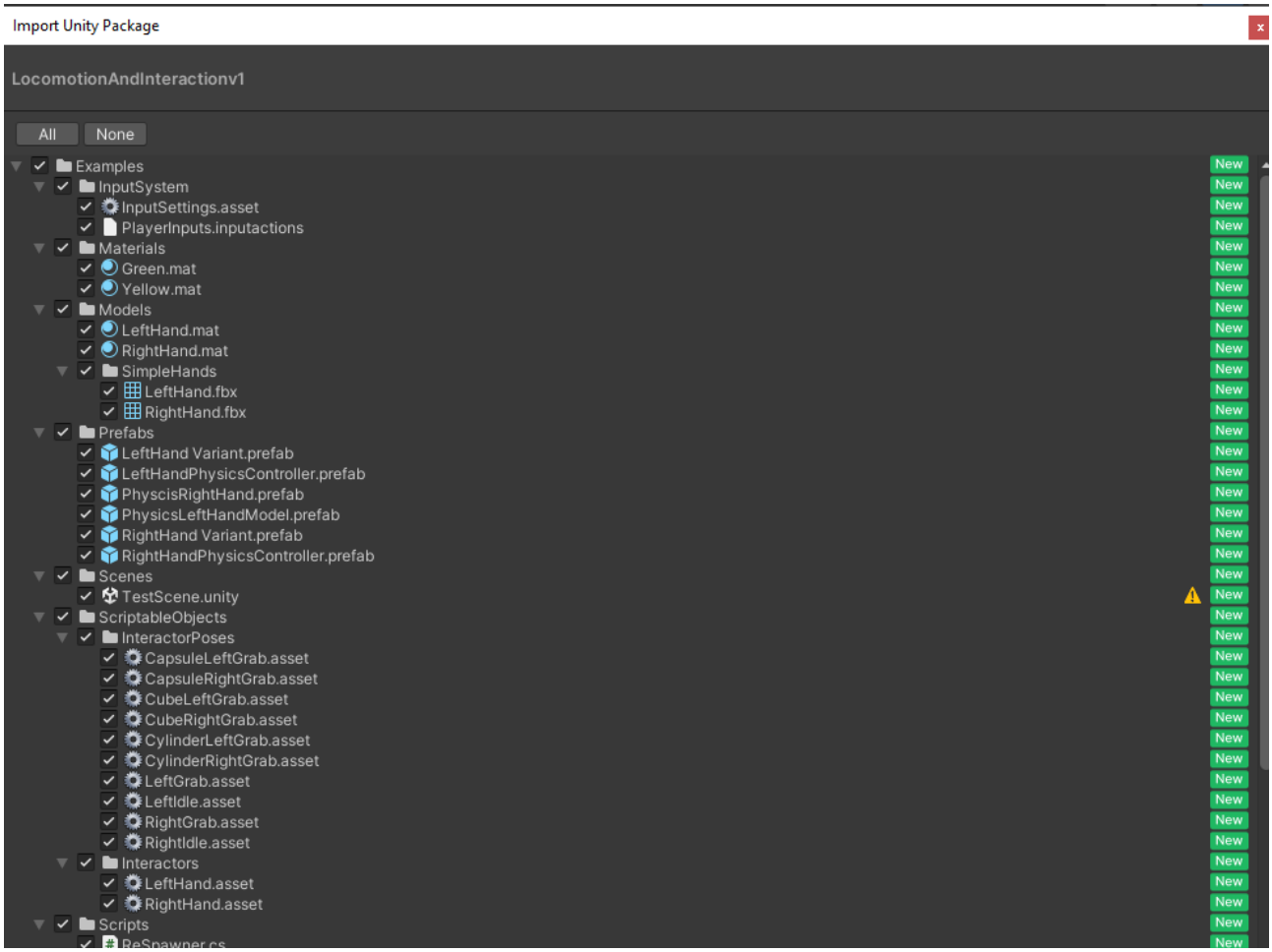

Figure 21: Locomotion&Interaction system, (Jani Seppälä)

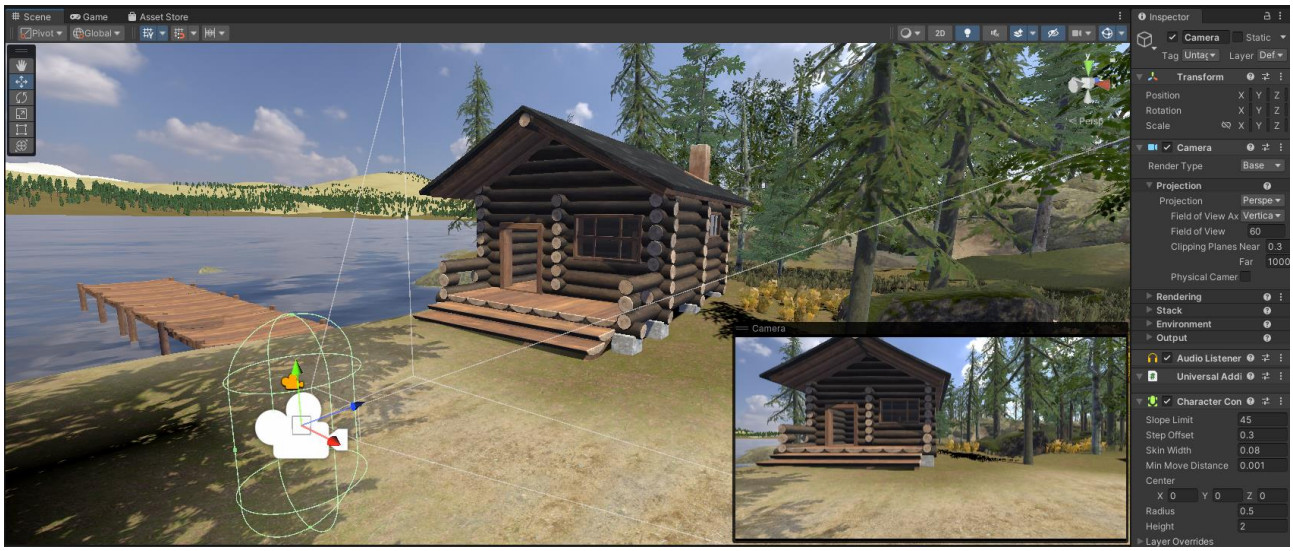

Figure 22: Character controller, Camera testing, (Juho Harju)

11.6 Iterate

Planning for the future

In the journey toward a minimum viable product, the project recognizes the need for additional assets. However, the current assets are already sufficient for thoroughly assessing the game's atmosphere and immersion levels. Once the scene has been rigorously tested with these existing features and assets, the project can progress to enhance the graphical detail and augment the number of assets within the game.

The terrain generation tool is invaluable, providing flexibility to experiment with different locations and the feel of the game world. Nevertheless, for a more comprehensive exploration of various natural environments and real-world locations, a more extensive library of models, textures, and post-processing settings is essential.

This forward-looking approach ensures that the project can evolve, both in terms of asset acquisition and the ability to replicate a variety of rich and authentic in-game environments.

12 Art Direction over Graphics

In the realm of game development, the concept of art direction transcends mere graphical fidelity. It emerges as a paramount aspect that shapes the game's personality and its connection with the players. While having ultra-realistic graphics is undoubtedly appealing, the approach of choosing art direction over raw graphics performance can often yield remarkable results.

Art direction encompasses the careful selection of visual styles, color palettes, and the overall aesthetic that harmoniously binds the game's elements together. A well-executed art direction can have a profound impact on players, ingraining a distinctive ambiance and mood, and enhancing the overall gaming experience. It allows developers to leverage creative freedom in delivering a unique identity to their games, even when working within technical constraints.

This approach proves especially beneficial for indie game developers or smaller studios, where resource limitations may necessitate a creative and strategic approach to game design. Instead of striving for raw graphical horsepower, they can achieve impressive results by leveraging astute art direction. For instance, the use of stylized visuals or other artistic techniques can lead to a visually striking and memorable game world.

Developers who adopt a strong art direction approach prioritize the visual coherence and narrative synergy of their games. This focus on artistic cohesion, combined with a well-considered visual direction, can often lead to more compelling and memorable gaming experiences.

As the thesis project continues to evolve, the appreciation for the balance between art direction and graphical power serves as a valuable lesson. It underscores the importance of a distinct and captivating visual style in crafting immersive game environments, offering developers an avenue to make a lasting impression on players.

This commitment to art direction embodies the essence of storytelling and emotion, ensuring that, in the world of gaming, art remains as significant as technology, and sometimes, even more so.

The game project we are currently working on provides a pertinent illustration of the significance of art direction over raw graphical capabilities. While our project may not possess the immense resources and graphical power of some triple-A titles, it excels in conveying a unique identity through meticulous art direction.

Our commitment to art direction is evident in the thoughtfully crafted 3D assets, the choice of textures, and the nature elements implemented within the game. Despite potential limitations, the art direction approach has allowed us to establish a distinct visual style that harmonizes with the game's immersive narrative and ambiance.

By making strategic decisions in line with the game's artistic vision, our project has the potential to create a lasting impact on players. It emphasizes that a well-conceived art direction not only complements but also enhances the gaming experience, contributing to the overall cohesion of the game world.

In essence, our project embodies the principle that art direction is more than just graphical prowess it's the bridge that connects players to the soul of the game. It demonstrates that creativity, artistic ingenuity, and a well-thought-out visual direction can often leave a more profound and memorable impression than sheer graphical might.

13 Conclusion

In my capacity, researcher focused on refining and testing the environment tools and assets sourced from Unity, as well as contributions from Tero Korpela, Ari-Pekka Oinonen, and Jani Seppälä. The primary objective was to evaluate the potential of transforming realistic locations into a captivating video game environment. The emphasis lay in creating an immersive space using the provided assets, all while ensuring compatibility with the target hardware and maintaining a balance between realism and the constraints of the platform.

While the default Unity Engine tools present challenges in seamlessly crafting real-world environments, a nuanced perspective reveals that expertise and hands-on experience can overcome these obstacles. Creating locations that closely mirror real-world settings, and thereby enhancing the player experience, is achievable with a certain level of familiarity with the engine. This resemblance is amplified when leveraging real-world data transfer or automated generation tools to mimic authentic locations.

The tools and methods selected for the workflow and development cycle demonstrated reliability, validated by the successful outcomes of previous game projects undertaken by the developers involved in this project. Theoretically, individuals possessing proficiency in Unity and programming skills should be capable of reproducing the environment and level creation process, at least up to the blockout version of the environment. This approach allows for flexibility in choosing between a focus on a realistic environment or the creation of entirely fictional locations for games in production.

Embarking on the manual creation of such environments demands a high degree of familiarity with the engine and the use of custom tools, whether self-created or sourced from third-party developers affiliated with larger game studios. It is, therefore, understandable that many indie game studios opt for simpler graphics and mechanics, given the extensive and time-consuming nature of game development. Even in seemingly simple games, the meticulous creation of every species of flora and fauna contributes to the cohesion of assets, textures, and locations, giving the entire game a unique character and the ability to stand out among more graphically intricate games.

In the contemporary gaming landscape, Unreal Engine, developed by Epic Games, provides a vast library of highly detailed assets, contributing to a triple-A aesthetic, especially on PC platforms. However, this has led to a commonality in assets and art among these games, emphasizing the critical role of art direction. The evolution of in-house assets, combined with online asset libraries, results in a harmonious unity that allows developers to bring their initial vision to life.

The investment in crafting the majority of a game's assets, and occasionally using effective tools, whether purchased or developed in-house, proves to be worthwhile. Developers' ability to create their own tools significantly impacts the final product, highlighting the importance of this capability. As the thesis project unfolds, these insights into the intricate process of creating and fine-tuning immersive game environments will inform future endeavours, fostering a deeper understanding of the evolving field of game design.

References

Adobe. (n.d.). Adobe Photoshop. https://www.adobe.com/products/photoshop.html

Adobe. (n.d.a). Substance 3D. https://www.adobe.com/products/substance3d.html

Agisoft. (2023). Agisoft Metashape. https://www.agisoft.com/

Awesome Technologies. (n.d.). *Vegetation Studio Pro*. https://www.awesometech.no/index.php/home/vegetation-studio/

Blender Foundation. (n.d.). Blender. https://www.blender.org/

Calleja, G. (2011). *In-Game: From Immersion to Incorporation*. MIT Press.

Capturing Reality. (2023). *RealityCapture*. https://www.capturingreality.com/

City of Jyväskylä. (n.d.). Jyväskylä - Human Technology City. https://www.jyvaskyla.fi/en/frontpage

Co, P. (2004). Level Design for Games: Creating Compelling Game Experiences. New Riders.

Crytek. (2023). *CryEngine Documentation*. https://docs.cryengine.com/display/CEMANUAL/CRYENGINE+Manual

Crytek. (n.d.). CryEngine. https://www.cryengine.com/

Daley, J. (2012). 2D Game Art for Programmers. https://2dgameartguru.com/

Epic Games. (2023). *Unreal Engine 5*. https://www.unrealengine.com/en-US/unreal-engine-5

Epic Games. (2023) *Unreal Engine 5*. Nanite. Documentation. https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/

Epic Games. (2023) *Unreal Engine 5*. The Procedural Content Generation Framework (PCG). https://docs.unrealengine.com/5.2/en-US/procedural-content-generation-overview/

FMI. Finnish Meteorological Institute. (2023) *"Sea level height systems in Finland".* https://en.il-matieteenlaitos.fi/water-level-height-systems

FileFormat.com. (n.d.). ASC File Format. https://docs.fileformat.com/cad/asc/

Godot Engine. (n.d.). Godot. https://godotengine.org/

Godot Engine Documentation Team. (2023). *Godot Engine Documentation*. https://docs.godoten-gine.org/en/stable/

GIMP. (n.d.). GNU Image Manipulation Program. https://www.gimp.org/

Gregory, J. (2014). *Game Engine Architecture*. CRC Press.

Hoefer, M. (2003). *The Importance of Level Design in Video Games*. https://www.gama-sutra.com/view/feature/131261/the_importance_of_level_design_in_.php

Indeed Editorial Team. (2023). Action Research: Definition, Benefits, and Examples. Indeed. https://www.indeed.com/career-advice/career-development/action-research.

Koster, R. (2013). *Theory of Fun for Game Design*. O'Reilly Media, Inc

Microsoft. (n.d.). *Zip and unzip files*. https://support.microsoft.com/en-us/windows/zip-and-unzip-files-8d28fa72-f2f9-712f-67df-f80cf89fd4e5

NASA. (2023). ASTER Global Digital Elevation Model. https://asterweb.jpl.nasa.gov/gdem.asp

NASA. (2023). Shuttle Radar Topography Mission (SRTM). https://www2.jpl.nasa.gov/srtm/

NLS. (2023). National Land Survey of Finland. "*Elevation model 2m."* https://www.maanmittauslaitos.fi/en/maps-and-spatial-data/expert-users/product-descriptions/elevation-model-2-m

NLS. (n.d). National Land Survey of Finland. "*E-Services" https://www.maanmittauslaitos.fi/en/e-services/mapsite*

Quixel. (n.d.). Megascans. https://quixel.com/megascans

Rogers, S. (2014). *Level Up! The Guide to Great Video Game Design*, Second Edition. Wiley.

Rockstar Games. (n.d.). Red Dead Redemption 2. Retrieved from https://www.rockstargames.com/reddeadredemption2/.

Skydark. (n.d.). Heightmap.skydark.pl. https://heightmap.skydark.pl/

Tangram. (n.d.). Tangrams Heightmapper. https://tangrams.github.io/heightmapper/

Terrain.party. (n.d.). Terrain.party. https://terrain.party/

Totten, C. W. (2014). *An Architectural Approach to Level Design*. CRC Press.

Totten, C. W., Sotiriou, P., Jagneaux, D., & Dillon, R. (n.d.). *Blockout*. *Level Design: Processes and Experiences*. https://book.leveldesignbook.com/process/blockout

Udemy. (2022). *Procedural Terrain Generation*. Penny de Byl. https://www.udemy.com/course/procedural-terrain-generation-with-unity/

Unity Technologies. (n.d.). *Unity*. https://unity.com/

Unity Technologies. (2023). *AssetStore*. https://assetstore.unity.com/

Unity Technologies. (2023). Terrain. Unity Documentation. Retrieved from https://docs.unity3d.com/Manual/class-Terrain.html

Unity Technologies. (n.d.a). *Level of Detail* (LOD). https://docs.unity.com/en/latest/Manual/class-LODGroup.html

Unity Technologies. (n.d.a). *Cinemachine Documentation*. https://docs.unity3d.com/Packages/com.unity.cinemachine@2.3/manual/index.html

Unity Technologies. (n.d.a). *CharacterController*. https://docs.unity3d.com/ScriptReference/CharacterController.html

Unity Technologies. (n.d.a). *ProBuilder Documentation*. https://docs.unity3d.com/Packages/com.unity.probuilder@4.0/manual/index.html

Unity Technologies. (n.d.a). *Lighting Overview Documentation*. https://docs.unity3d.com/Manual/LightingOverview.html

Unity Technologies. (n.d.a). *World Building Documentation*. https://docs.unity3d.com/Manual/CreatingEnvironments.html

Wiebe, E. N. (2018). *Designing Immersive Video Game Environments*. In D. Ifenthaler, D. Eseryel, & X. Ge (Eds.), Game-Based Learning: Theory, Strategies and Performance Outcomes (pp. 1-24). Nova Science Publishers.

Wolf, M. J. P. (2008). *The Video Game Explosion: A History from PONG to PlayStation and Beyond*. Greenwood Press.

Bleszinski, C. (2000). *The Art and Science of Level Design*. In Game Developer's Conference Proceedings.

Brown, E., & Cairns, P. (2004). *A grounded investigation of game immersion*. In Proceedings of the conference on human factors in computing systems (pp. 1297–1300). New York, NY: ACM.

USGS. (2023). The National Map. https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map

Maanmittauslaitos, Land surveying institute of Finland (2023) https://www.maanmittauslaitos.fi/kartat-ja-paikkatieto/asiantuntevalle-kayttajalle/tuotekuvaukset/korkeusmalli-2-m

Juegoadmin. (2022, October 12). *Game level design: What it is and why is it important*. Juego Studio. Retrieved from https://www.juegostudio.com/blog/game-level-design-what-it-is-and-why-is-it-important

Keuruu (Keuruu). (2023, November 16). In *Wikipedia*. https://fi.wikipedia.org/wiki/Keuruu

Kolonjärvi (Keuruu). (2022, December 11). In *Wikipedia*. https://fi.wikipedia.org/wiki/Kolonj%C3%A4rvi_(Keuruu)

Arnold, B. J. (2017). Sustaining immersive game-based learning environments. In Y. Baek, (Ed), Game-Based Learning: Theory, Strategies and Performance Outcomes. Nova Science Publishers.

Toholampi, Wildreness (2023, October 27) In *Wikipedia*. https://fi.wikipedia.org/wiki/Toholampi#Luonto