The Volkspark Enschede Project

Jonas Bartsch | 471999

Saxion University of Applied Sciences

Creative Media and Game Technologies

Graduation Report

Company Supervisor - Bob Dickhoff

University Supervisor – Herman Statius Muller

Acknowledgments

I would like to thank all the people that made this project a reality. Either through professional knowledge, additional guidance, or emotional support.

Thank you to:

Herman Statius Muller – For guiding me through the project as a teacher and professional 3D artist, guiding me through the project and helping with any questions that arose during the many months.

Afra Willems – For providing a second professional opinion on my report.

Bob Dickhoff – For creating the chance to work with the municipality of Enschede and being the bridge between the students and the Volkspark committee.

Boris Gospodinov, Nikita Fleur Konings, and Stefan Daniel Spoiala – For creating the virtual environment scene together with me.

Raphael Baier – For proofreading my draft and final version of the report.

Thomas & Sabine Bartsch – For being my emotional support as parents during the whole project.

Quitte – For being my parents' dog who kept me company during difficult days.

Abstract

The goal of the Volkspark Enschede project is to recreate the park as a virtual environment. The municipality of Enschede aims to revitalize the vegetation and use the medium as concept art and an exemplary model for potential investors. This thesis will clarify the process of development of the virtual environment. It goes into detail for the description of the use of photogrammetry and the development cycle in the Unreal Engine.

The conclusion of the project is the success and understanding of photogrammetry, and the work with the newly released Unreal Engine 5.

The project was developed in collaboration with the XR-Lab in Enschede and additional students of the Saxion University of Applied Sciences.

Information in this report is gathered through desk- and field research, while underlined through various media, including papers, articles photographs, and screenshots.

Table of Contents

Glossary

Engine – An engine is a piece of software used in game development. It is generally used to combine art, code, and additional elements (e.g. lighting) to create an executable game for different platforms.

Level of Detail (LOD) – LODs are different versions of a single 3D model. Each version has a lower polygon count (and subsequently fewer details) than the original. These models can be switched out depending on the view distance of the player, to save the performance of the game.

Mesh – A different word for 3D model.

Nanite – A performance-saving feature used for 3D models in the Unreal Engine 5, developed by Epic Games.

Lumen – A lighting technique that can adjust to real-time changes in the Unreal Engine 5, developed by Epic Games.

Nodes – A non-destructive workflow that is used in many software for movie/television and videogame creation. It is mostly present for 3d modeling and visual effects.

UV Map – UV mapping is a technique being used for wrapping a 2D image around a 3D model.

Texture Map – The 2D image that gets wrapped around the 3D model.

Version Control – A system for saving and managing data. It allows multiple people to work on a single project at the same time.

1. Introduction

The dying of trees is an important part of nature. Not because it is pleasant, but because it is something that has to be controlled. The aftermath of dying trees can among other things cause ecohydrological problems through drought or bug infestation (Adams, et al., 2011). The municipality of Enschede faced this problem in the past and decided to revitalize the Volkspark in the city center.

Back in 1872, the Volkspark in Enschede was donated to the municipality by the van Heek family (Commissie van Beheer over het Volkspark, n.d -a.). For the last 150 years, the park thrived and was an essential part of the city. Due to the dying of the trees, the municipality of Enschede tasked the Volkspark Committee in combination with the XR-Lab to create a virtual environment of a new park.

The goal is to create a futuristic version of the park that is leaning visually toward the past version from 150 years ago. Concept drawings are created by an external concept artist, which in turn are being developed into a 3D environment. The final objective for the municipality is to have a park that is pleasant for visitors to walk through and in which they can observe nature and historical monuments.

The general idea for the project is to create 3D assets with tools/techniques that allow high-quality environment assets. Methods like photogrammetry are especially useful to recreate architectural objects and foliage at a certain level of detail.

2. Company Outline

The project was initiated by the municipality of Enschede, the Volkspark Committee, and the XR-Lab in Enschede. Enschede is a city in the east of the Netherlands. The city inhabits about 160.000 citizens (Centraal Bureau voor de Statistiek, 2022), many of those being students.

The Volkspark Committee is a subsidiary of the municipality and was established to further develop the park, care for nature and present it in the best possible way to the people of Enschede.

The XR-Lab in Enschede offers a wide variety of equipment for students and a place to research projects together (Hogeschool Saxion, 2019). Its focus is on the development of virtual- or augmented reality software. The lab is run by Mathijs Van Veen and is supported by a wide array of teachers from Saxion (Saxion Hogeschool, 2019).

3. Theory

3.1 The Volkspark in Enschede

The Volkspark in Enschede is a 15-hectare big park in the middle of the city (Volkspark, n.d.). Established by Hendrik Jan van Heek, it was designed by Dirk Wattez and opened to the public in 1874 (Stichting Cultureel Erfgoed Enschede, 2018). The Van Heek family is considered to be the founder of the textile industry in Enschede. Together with his brothers Herman and Gerrit, Hendrik founded the Van Heek & Co company (Snuif, 1915). Due to his legacy to the people of Enschede, Van Heek was honored with a monument in the Volkspark.

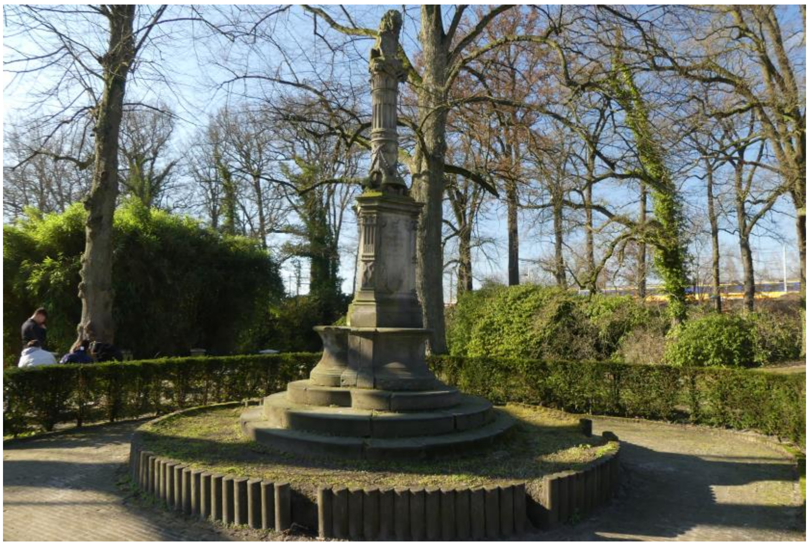


Fig. 1 – Van Heek Monument (2022)

Other monuments in the park are dedicated to the victims of World War II or are art installations.

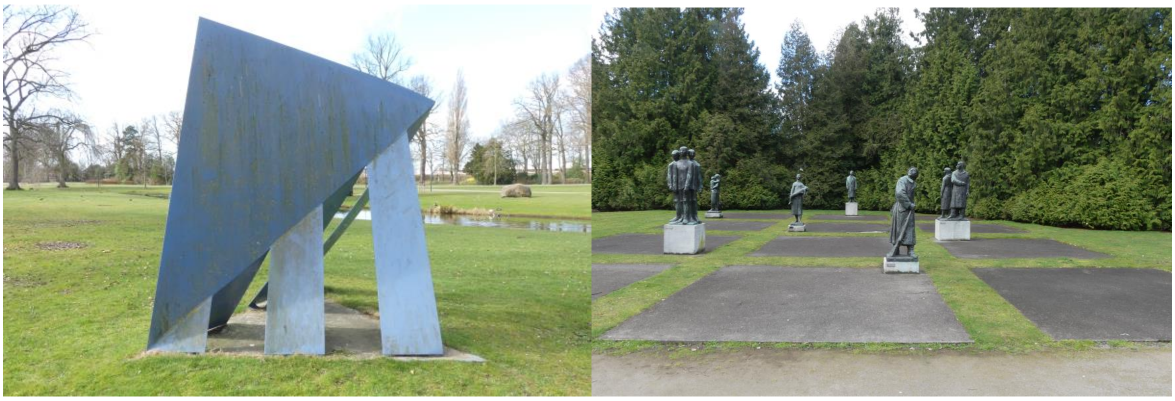


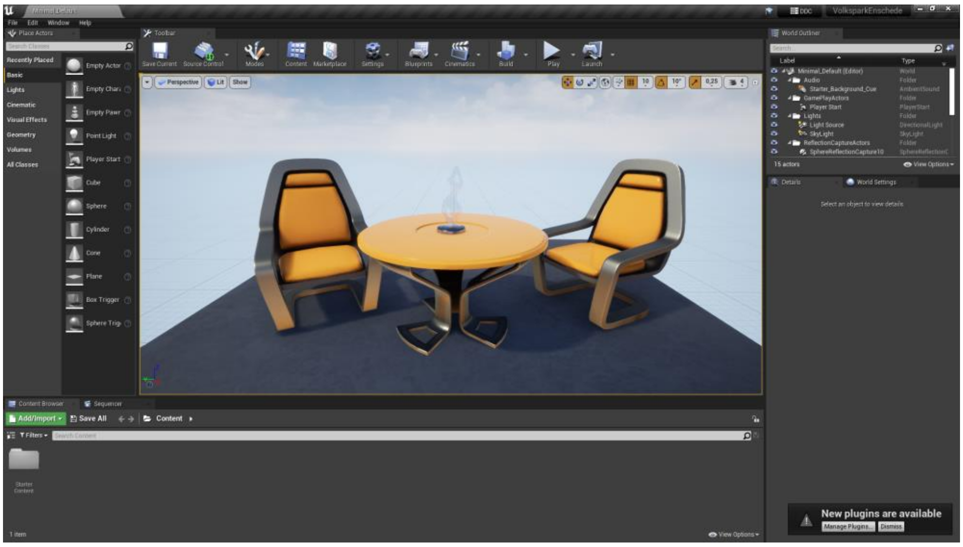
Fig. 2 – “Kanteling” statue by Kyrill Lixenberg and War Monument by Mari Adriessen (2022)

Additionally, park visitors can enjoy their time at the petting zoo, or the newly built playground (Commissie van Beheer over het Volkspark, n.d. -b)

3.2 Unreal Engine

The Unreal Engine is a game engine developed by Epic Games. The engine allows the development of multimedia, such as games, movies, and other software outside of entertainment. It enables working with real-time technology for multiple platforms.

Thanks to a user-friendly business model, Unreal Engine is available to everyone for free. Only if the created product revenue exceeds over 1 million USD, Epic Games will demand royalties of 5% of the generated product income. Thanks to the documentation and the countless tutorials, gathering knowledge is easier than ever and makes the engine a prominent choice for many game developers (Epic Games, n.d. -b).



Label	Type
Minimal_Default (Editor)	World
Audio	Folder
Starter_Background_Cue	AmbientSound
GamePlayActors	Folder
Player Start	PlayerStart
Lights	Folder
Light Source	DirectionalLight
SkyLight	SkyLight
ReflectionCaptureActors	Folder
SphereReflectionCapture10	SphereReflectionC

15 actors

Select an object to view details

New plugins are available

Fig. 3 – Unreal Engine 4 user interface (Screenshot Unreal Engine 4 2022)

3.3 Photogrammetry

“Photogrammetry is the science and technology of obtaining reliable information about physical objects and the environment through the process of recording, measuring and interpreting photographic images and patterns of electromagnetic radiant imagery and other phenomena” (Wolf, Dewitt, & Wilkinson, 2014).

The term “photogrammetry” was used since the 19th century and was first established in 1867, when it was published in the “Wochenblatt des Architecktenvereins zu Berlin” (Grim, 2007). Initially, the idea of photogrammetry was solely to map out buildings or landscapes. To get the best possible pictures, scientists used aerial traversal options like balloons. With the invention of the plane, photogrammetry advanced even further (Burtch, 2008). With modern technology, it is now possible to scan models or architecture through photographs and create a digital 3D model, which then in turn can be printed out. The models can be used in many ways, e.g. game development, archeology, or like it was envisioned to be a hundred years ago land mapping.

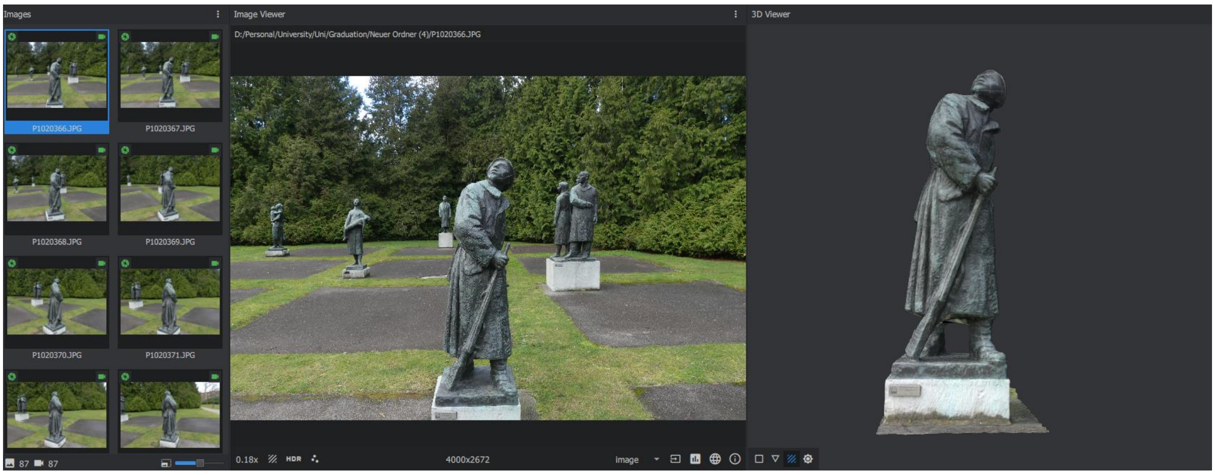


Fig. 4 – Photogrammetry example (Screenshot Meshroom 2022)

4. Concepting

Before the start of the actual development, it is important to look at the concept art and photos to visualize the idea of the upcoming 3D model. During this project, existing and newly created concept art is going to be used. The Volkspark committee already commissioned an additional concept artist, who made multiple art pieces to display a possible vision of the park.

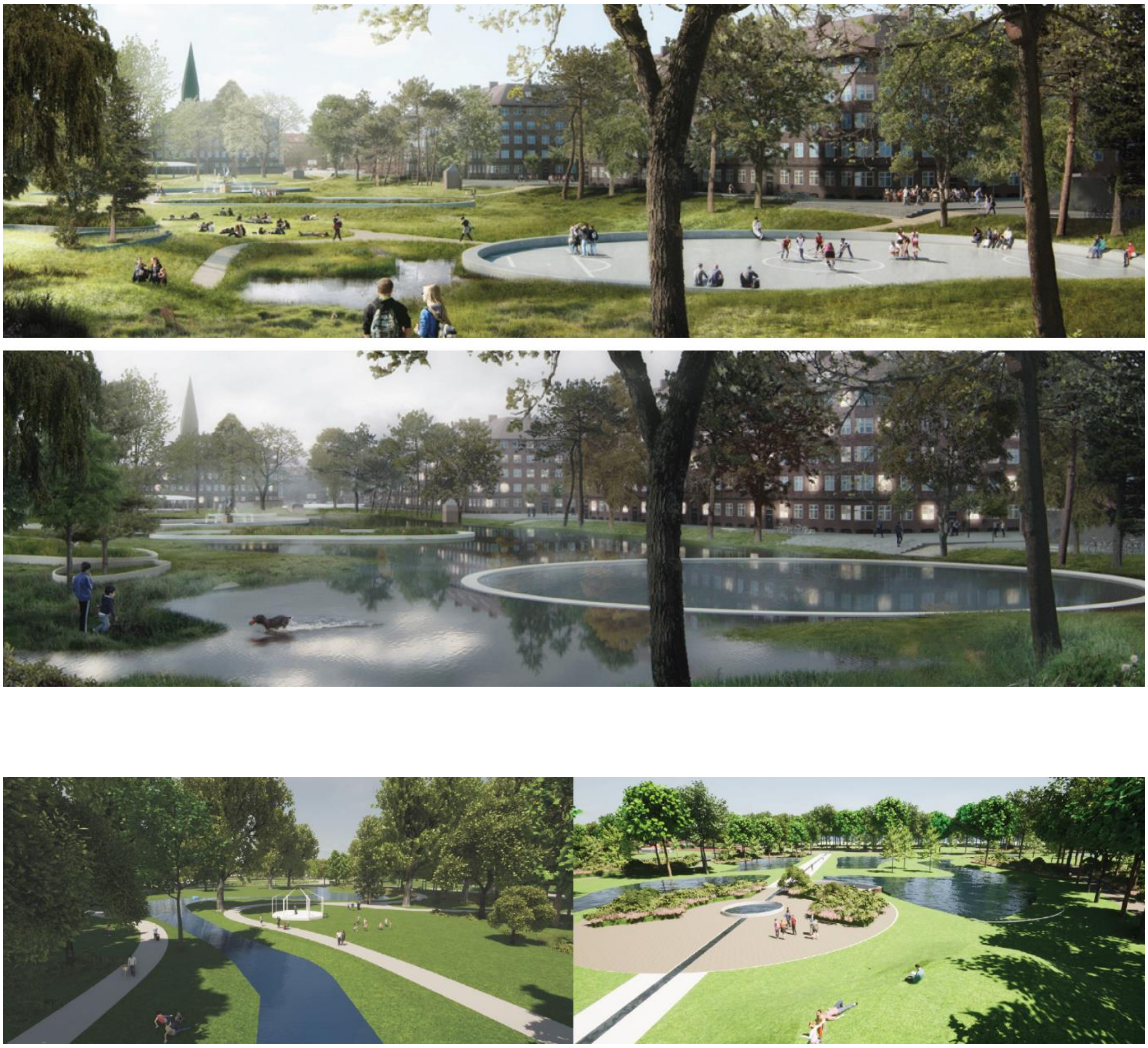


Fig. 5 – Volkspark Concept Art (Marina Eenschoten 2022)

Since the concept art only depicted certain angles of the park, the team decided to create additional concept art. This helped with the development of the layout and placement of trees.

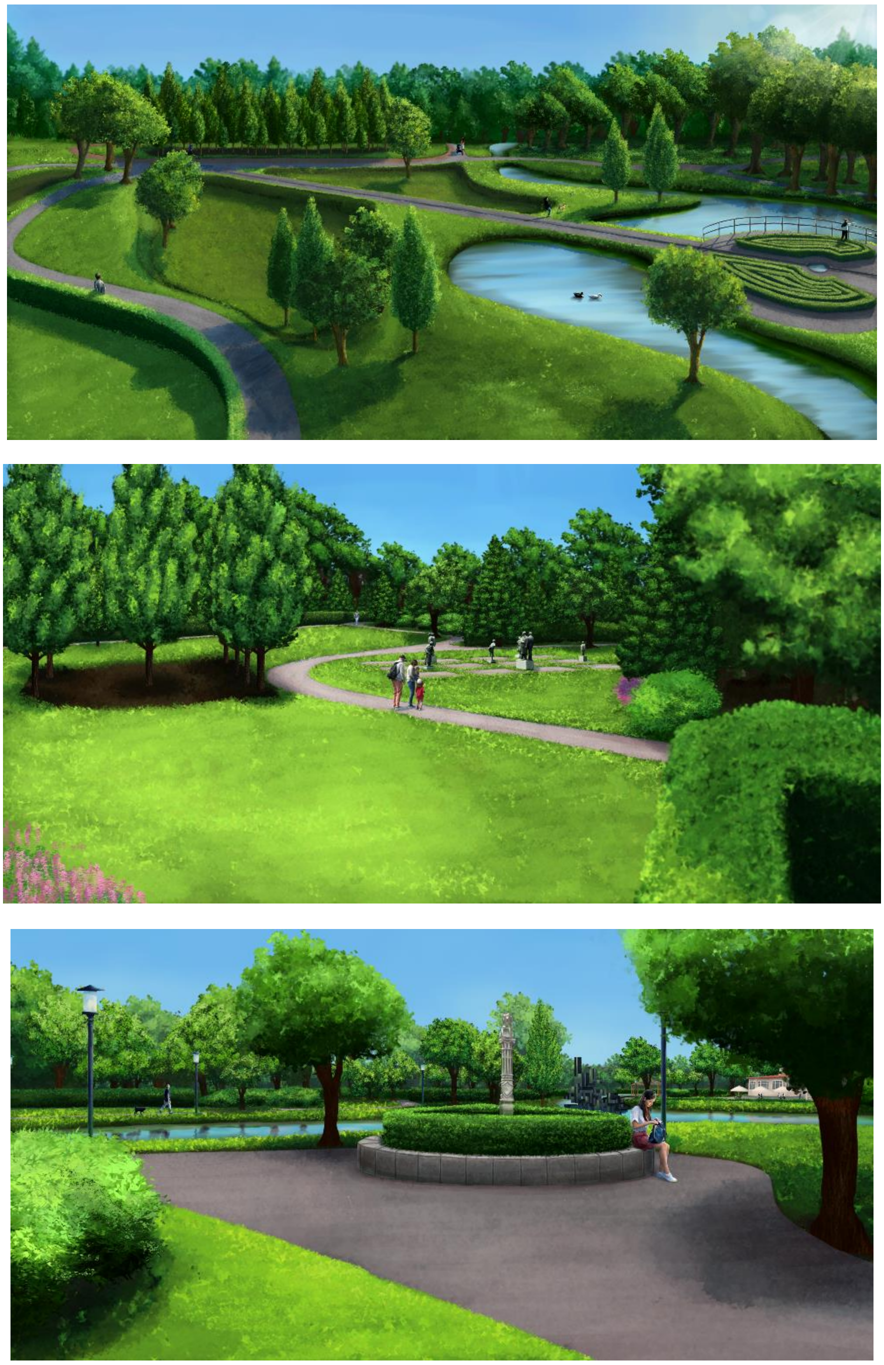


Fig. 6 – Volkspark Concept Art (Nikita Fleur Konings 2022)

Furthermore, pictures of the statues in the Volkspark were made, so that the team could decide on how to realize them in 3D.

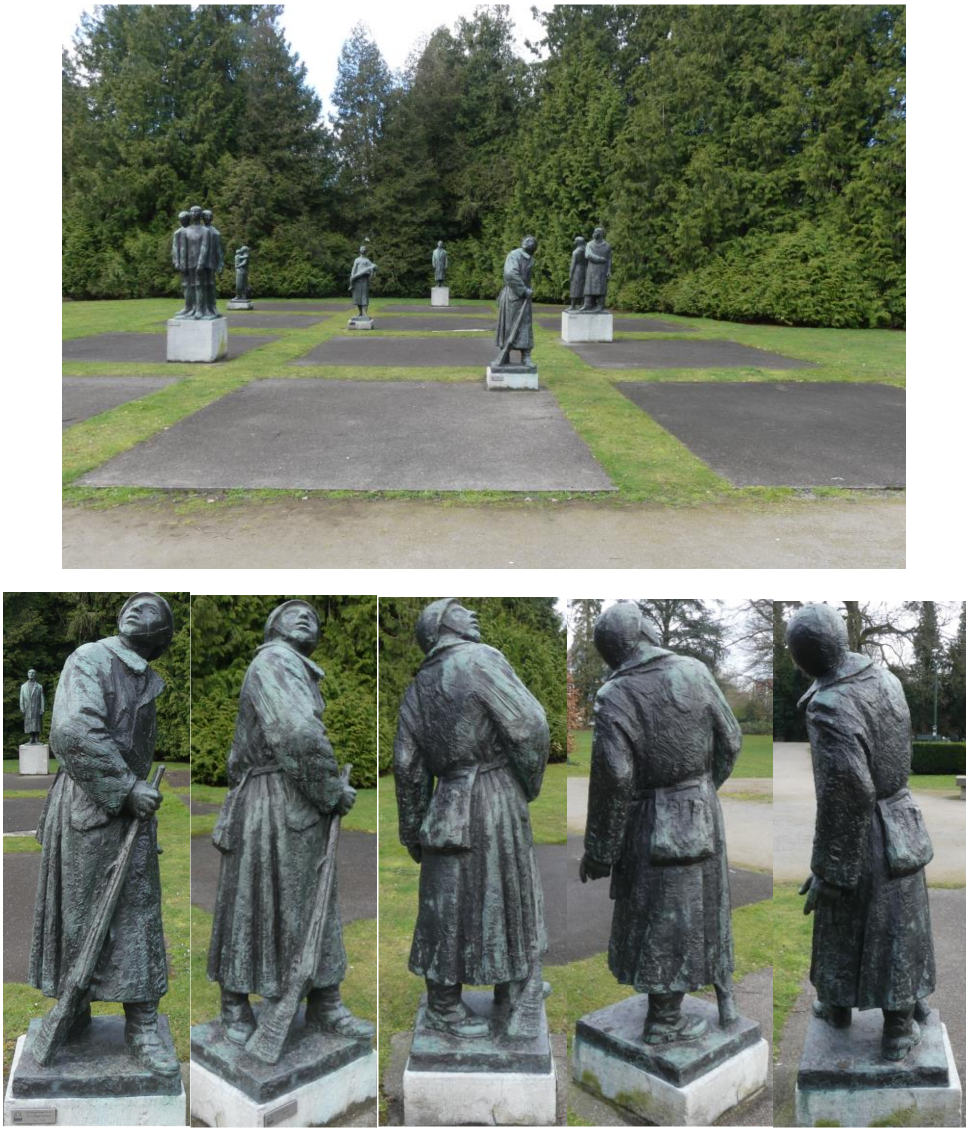


Fig. 7 – Volkspark World War II Monuments (2022)

The texture, color, and complexity of the statues were highly detailed, which is why the team quickly realized that it was impossible to recreate the monuments in a short time frame with the desired quality through modeling. After many discussions, the team decided to settle on photogrammetry for the statues, since it was a chance to recreate them in 3D as detailed as possible.

5. Problem Definition

The client wants to revitalize the park due to dying trees. The goal of the committee is to use a realistic virtual environment of the park for different purposes. That means that photogrammetry is a viable option to recreate the monuments.

While photogrammetry allows the creation of realistic-looking 3D models, it requires certain knowledge of how to execute the workflow. Questions that have to be answered are the following:

- How important is the right camera?
- Which software can be used to create 3D models out of photos?
- What is the process of creating a 3D model out of photos?
- How can the photogrammetry model be used efficiently in the Unreal Engine?

6. Main/Sub Questions

How can realistic 3D models be created through the means of photogrammetry for the Unreal Engine?

Sub-Questions:

In what way does the photo camera affect the final model?

For the correct use of photogrammetry, it is important to use a camera that fits the requirements for a good photo. Through testing and research, different cameras can be compared with each other. What kind of camera values affect the photos used for a 3D model?

Which software can be used to create photogrammetry models, and how does it work?

There exist a lot of different photogrammetry software solutions on the market. Which ones are a suitable choice for this project? How does the workflow for creating a 3D model out of photos function?

What methods can be used to ensure a stable framerate in the Unreal Engine?

Trying to build a large environment while aiming for a realistic art style is going to be performance-heavy. In order to achieve the desired style, research on the realistic depiction of architecture has to be done. This includes the measures that can be taken to improve performance (e.g., LOD).

7. Development

7.1 Choosing the engine

The development begins with choosing the right engine for the job. An engine allows proper development for real-time experiences. To ensure a smooth development phase, the decision between the Unreal Engine and Unity had to be made. They are the most approachable game engines for larger development cycles, which are publicly available.

Unity:

Unity was created in 2005 by Nicolas Francis, Joachim Ante, and David Helgason, three Danish developers that wanted to make game development available to a broader audience. While first only being developed for Macintosh, a version for Windows was released in 2009 (Haas, 2014).

Over the years, Unity established itself as a game engine that is often used for small-scale projects, due to its user-friendly UI and big community (Paul E. Dickson, 2017). The engine supports mainly C# as a scripting language.

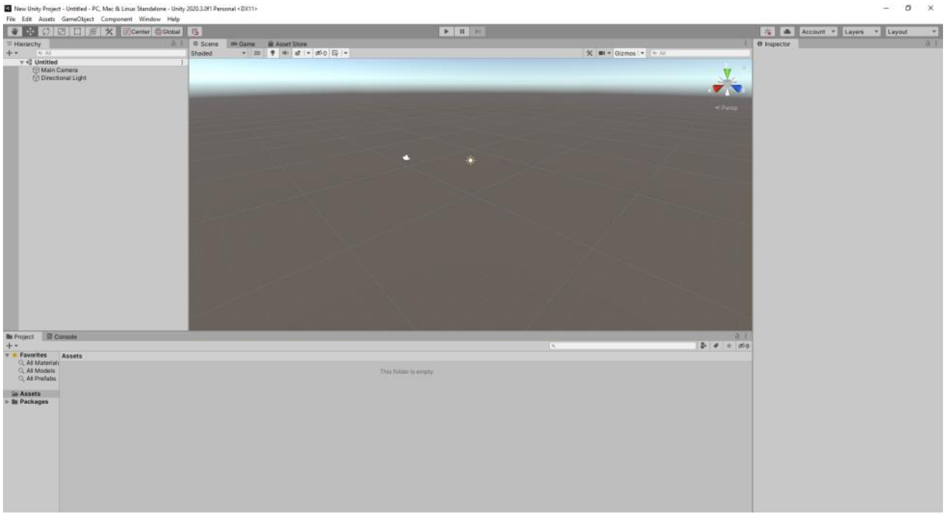


Fig. 8 – Unity Engine (Screenshot Unity 2020)

Unreal Engine:

The Unreal Engine was developed in 1994 by Tim Sweeney, with a public release in 1998 (Sweeney T. , 2009). It was developed for the videogame "Unreal", which was released simultaneously to the engine. Similar to Unity, the Unreal Engine got multiple major releases, with the newest version, "Unreal Engine 5" being released in April 2022 (Rein, 2012) (Epic Games, 2022 -a).

The software uses so-called "Blueprints" as a visual scripting tool. This allows developers with little to no programming skills, to create complex functions through a node-based system, based on the C++ coding language (Epic Games, n.d. -a).

Comparing the Engines

While both engines had some major differences during their beginnings, at this point, the line between them starts to blur a little bit. Both engines offer a great variety of options to create a game. At this point, both Unity and the Unreal Engine can create games with high graphical fidelity. The biggest differences are scripting languages and the user interface.

The deciding factor for the choice of the engine, in this case, first and foremost the previous experience with the game engines. To achieve an immersive experience, based on existing, real-life property, it is important to capture a realistic-looking art style. Since most team members had more familiarity with the Unreal Engine than Unity, the decision was made to use the Unreal Engine.

Unreal Engine 5

Halfway through the development of the experience, Epic Games released Unreal Engine 5 to the public. While an early access version was released in 2021 (Epic Games, 2021), the development of the Volkspark environment started with version 4, since version 5 was still unstable and filled with bugs. Features like Nanite and Lumen are advantageous to the development process, which is why the decision was made to convert the project.

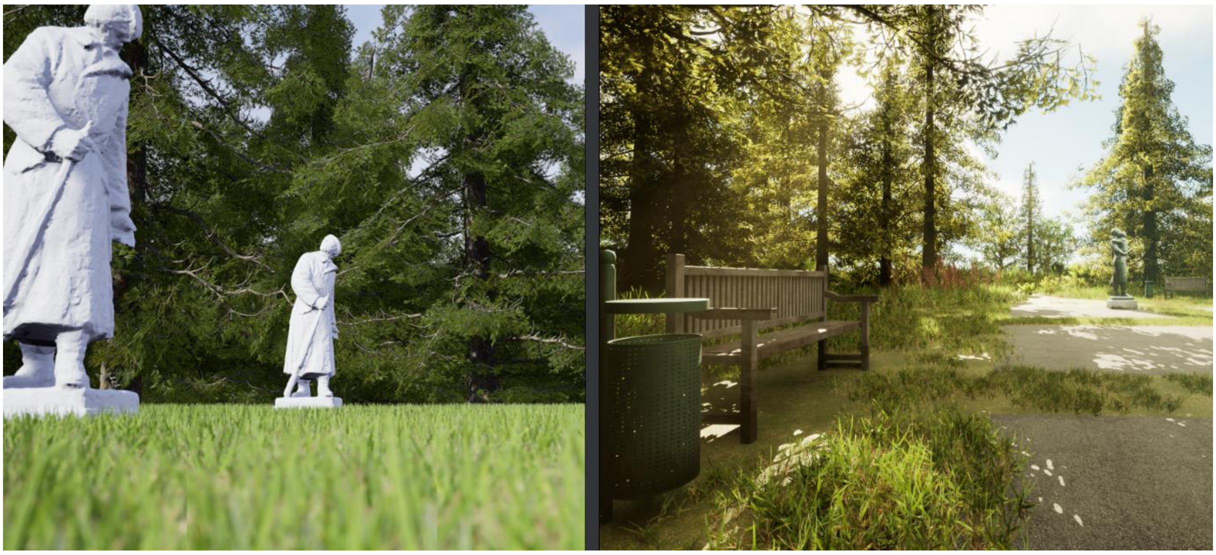


Fig. 9 – Unreal Engine 4 (left) - Unreal Engine 5 (right) (Screenshot Unreal Engine 5 2022)

7.2 Recreating Monuments with Photogrammetry

Creative Process

Photogrammetry is a technique that, as previously mentioned, was created as a tool to map out different environments. Over roughly 150 years, photogrammetry evolved into a distinctive method to scan real objects and create a 3D model out of them. To accurately recreate the monuments of the Volkspark, the decision was made to use photogrammetry to achieve the most detail. The creative process of recreating an object through photogrammetry is complex, and can be broken down into the following steps:

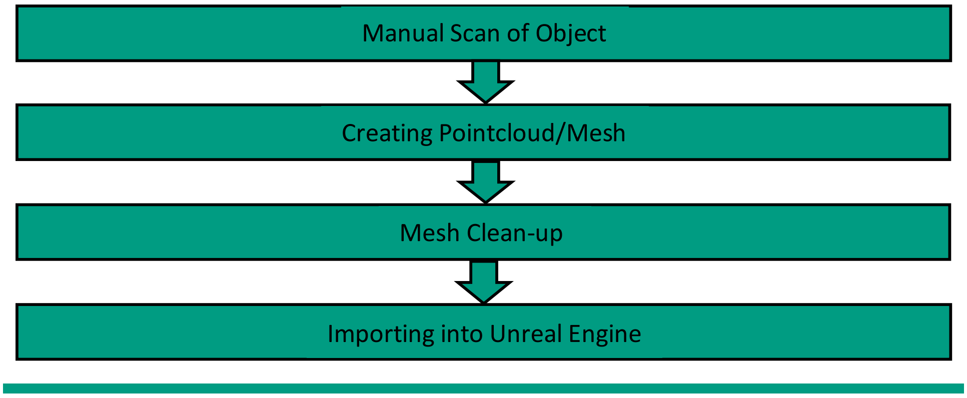


Fig. 10 – Photogrammetry Process

Creating the Photographs

To capture the best possible photos for a 3D model, multiple factors have to be taken into consideration. It is important to know what kind of camera can be used and what the deciding part's features of a photo are.

- Resolution:
  The resolution is the width and length of a single photograph. It is the number of single points/pixels, which comprise the full picture.
- Pixel Quality/Effective Resolution:
  It is not only important for the picture quality to get a great number of pixels, but also high quality of them. Factors like grain, contrast, and other aberrations can greatly improve or reduce the characteristics of an image.
- Raw Data:
  Every picture taken by a camera is compressed, depending on the file format the photo is saved in. The general idea is to get the least amount of compression so that the quality stays as sharp as possible. Ideally, the file gets saved as a .raw file, which leads to no compression whatsoever, but takes a great toll on the memory media. (Alexandrov, 2021)

The capturing process of the photos follows the same procedure almost every time. The best weather condition for taking photographs is a clear sky, to avoid having hard shadows on the object. This makes it easier for the software that is later used to reconstruct the model evenly. It also avoids potential problems down the road, when a different light source is used to light the model. A color grading chart can help with getting the correct colors of an object later during color correction.

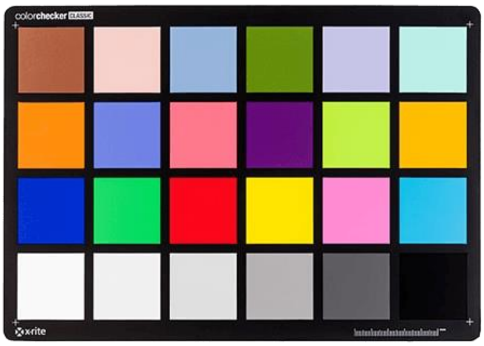


Fig. 11 – ColorChecker (Xrite)

The method of taking the photos is circling the object in small steps and taking as many images as possible. This allows the 3D software to create more details. While doing a circular motion, it is not advised to change the height of the camera, since this will harm the quality of the pictures. Only after a circular movement has been completed, height adjustments can be done.

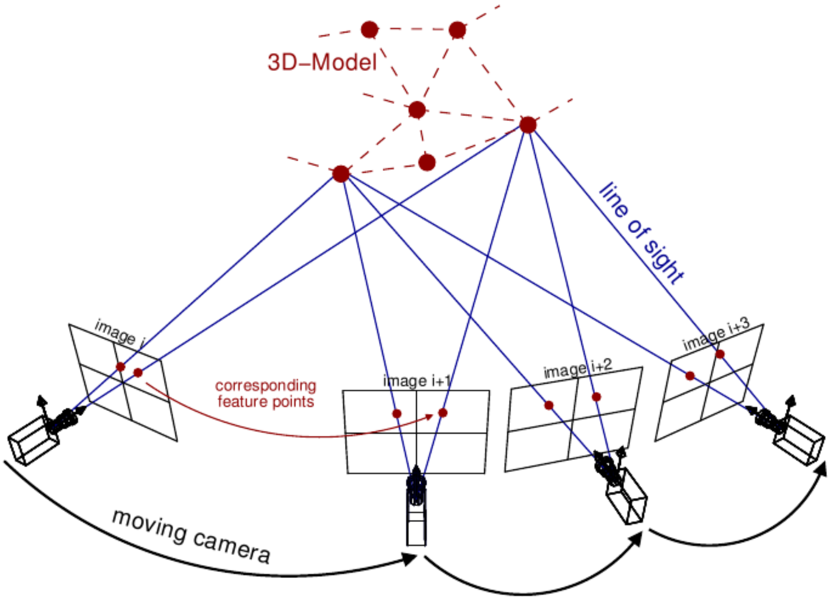


Fig. 12 – Photogrammetric Principle (Sweeney C.)

For this project, the first tests were done with a Smartphone that had the following specifications:

- Samsung Galaxy S6
  32 GB
  Camera (16 Megapixels, f/1.9, 28mm, 1/2.6 inch sensor, 1.12 µm, Autofocus, Optical Image Stabilizer)
  (Ho, 2015)

After testing the camera quality with a Playstation 5 controller as a model, it was clear that the quality of the smartphone was not enough to create a completed model. That's why the camera was switched out against a better model created by Panasonic:

- Panasonic Lumix DMC-FZ300
- 12 Megapixels, f/2.8, 4.5 – 108mm, 1/2.3 inch sensor

Each picture was taken with the following settings:

- 4000 x 2672 pixel
- 180 dpi
- sRGB
- f/5.6
- 1/60 sec
- ISO – 100
- 4 mm
- .jpg
- Ca. 90 pictures per statue

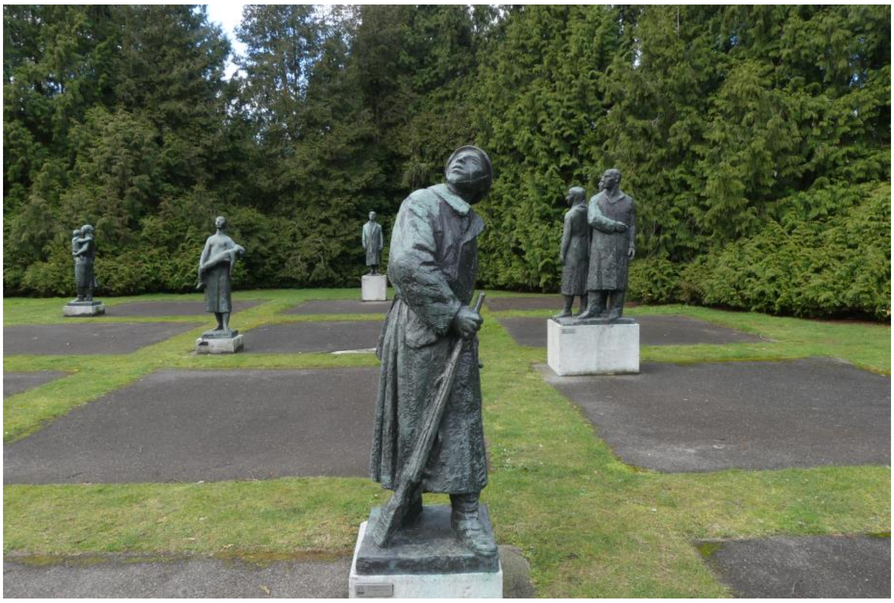


Fig. 13 – Volkspark Monument (2022)

Photogrammetry Software

After all the photos were taken, it was time to decide which photogrammetry software had to be used. While Reality Capture is another tool used in the professional field, the choice came down to Meshroom and 3DF Zephyr, due to them being (partially) free to use.

Meshroom

Meshroom is an open-source project, initiated through a joint research group of the Ecole des Ponts ParisTech and Centre Scientifique, together with Mikros Image (Carsten Griwodz, 2021). The software has been used since 2014 and is now one of the most popular tools for photogrammetry. Meshroom uses a node-based system, which allows changes in the workflow at any time, without having to restart the whole process from scratch.

3DF Zephyr

3DF Zephyr was released in 2014 and created completely in-house by 3DFLOW. Multiple packages have been released at some point, though two of them were merged with the 5.0 release (3DFLOW, 2020). The software is free to use, with the limitation of 50 pictures per 3D model. The software is widely available through additional platforms like Steam.

Comparison

Similar to the engine comparison, the photogrammetry software is very comparable. Both of them offer functions that are necessary for creating models. The tipping point for this project was the 50 picture limitation on 3D Zephyr. The comparison of the created meshes gave a clear indication of a better model. Both models were created with the same settings, with the only difference being the photo number.

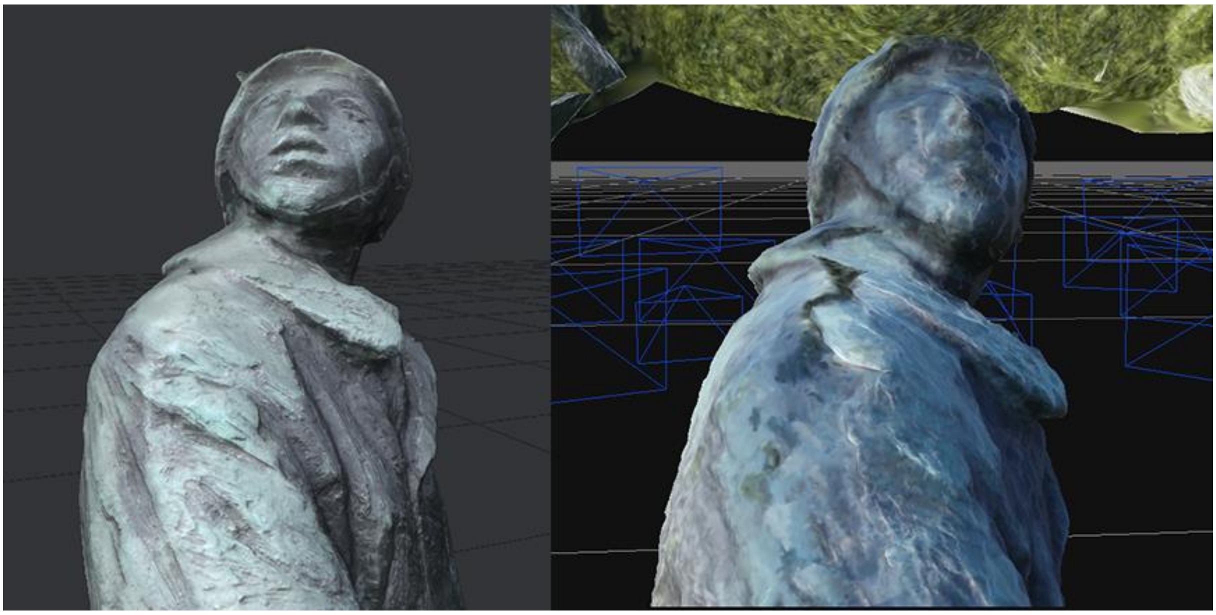
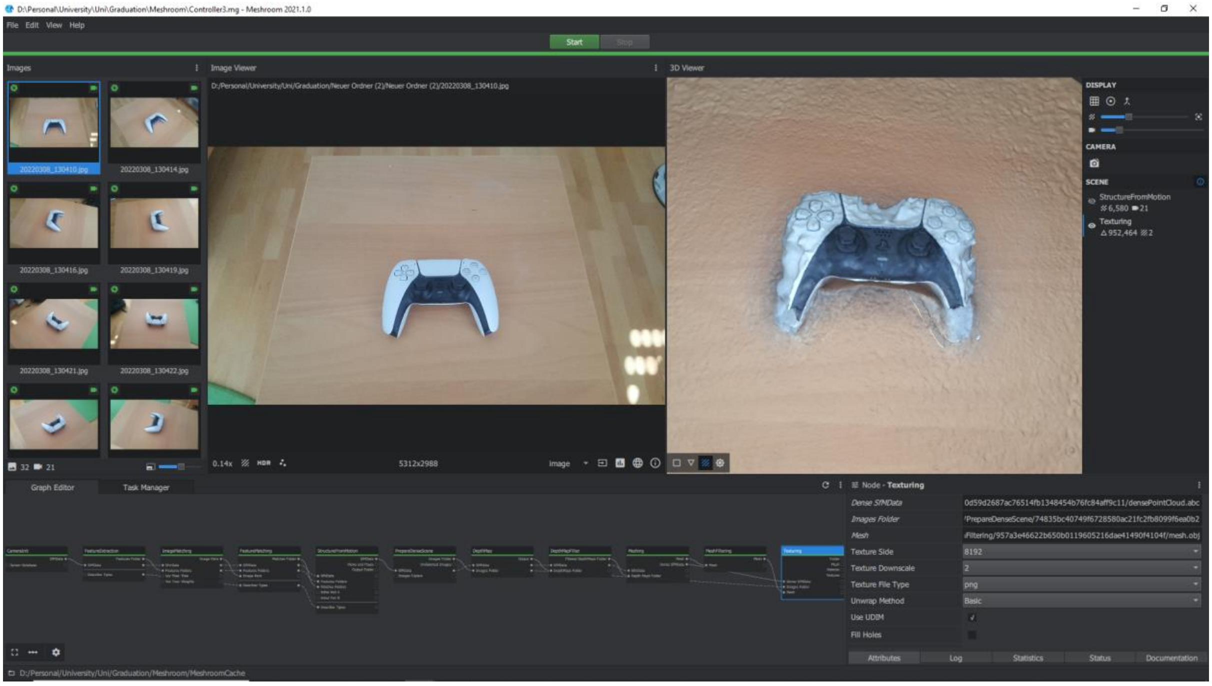


Fig. 14 – Mesh Comparison: Meshroom (left) – 3DF Zephyr (right)

(Screenshot Meshroom 2022), (Screenshot 3DF Zephyr 2022)

Meshroom In-Depth

After all the photos have been shot, Meshroom could go to work. As previously mentioned, the first photogrammetry test was done with a Playstation Controller. The software used to generate the mesh was Meshroom.


D:\Personal\University\Uni\Graduation\Meshroom\Controller3.mg - Meshroom 2021.1.0
File Edit View Help
Start
Stop
Images
Image Viewer
3D Viewer
D:/Personal/University/Uni/Graduation/Neuer Ordner (2)/Neuer Ordner (2)/20220308_130410.jpg
DISPLAY
CAMERA
SCENE
StructureFromMotion
Texturing
5312x2988
Graph Editor
Task Manager
D:/Personal/University/Uni/Graduation/Meshroom/MeshroomCache

Node - Texturing

Dense SfMData	0d59d2687ac76514fb1348454b76fc84aff9c11/densePointCloud.abc
Images Folder	'PrepareDenseScene/74835bc40749f6728580ac21fc2fb8099f6ea0b2
Mesh	Filtering/957a3e46622b650b0119605216dae41490f4104f/mesh.obj
Texture Side	8192
Texture Downscale	2
Texture File Type	png
Unwrap Method	Basic
Use UDIM	✓
Fill Holes	□

Fig. 15 – First Meshroom & Camera Test (Screenshot Meshroom 2022)

It was only through the efforts of the better camera, that improvements could be seen. The process of creating a 3D model with textures is relatively straightforward and can be explained step by step.

The first part of getting a new mesh is to load all pictures previously taken into Meshroom. This can be done by drag-and-drop or opening them through the “Import Images” option. The photos can be looked at the image viewer on the left.

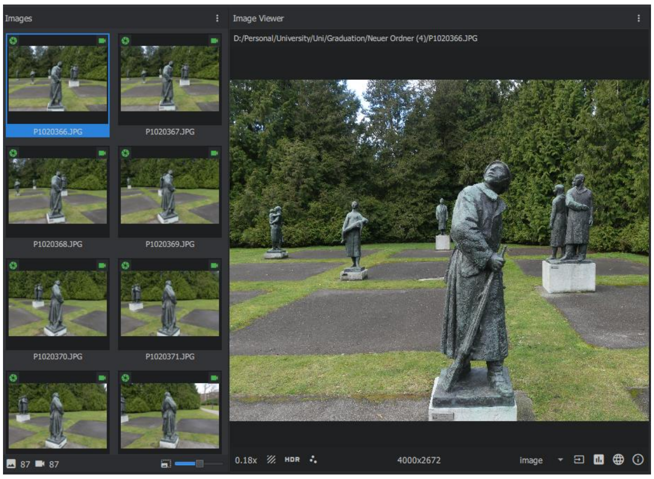


Fig. 16 – Image Viewer (Screenshot Meshroom 2022)

Meshroom operates on a node-based structure, which makes changing values very easy. Without giving any input, the software automatically creates a string of nodes that can directly be used to create a 3D model. To further understand Meshroom, a closer look at the single nodes has to be made. Without any changes, the node string looks like this:

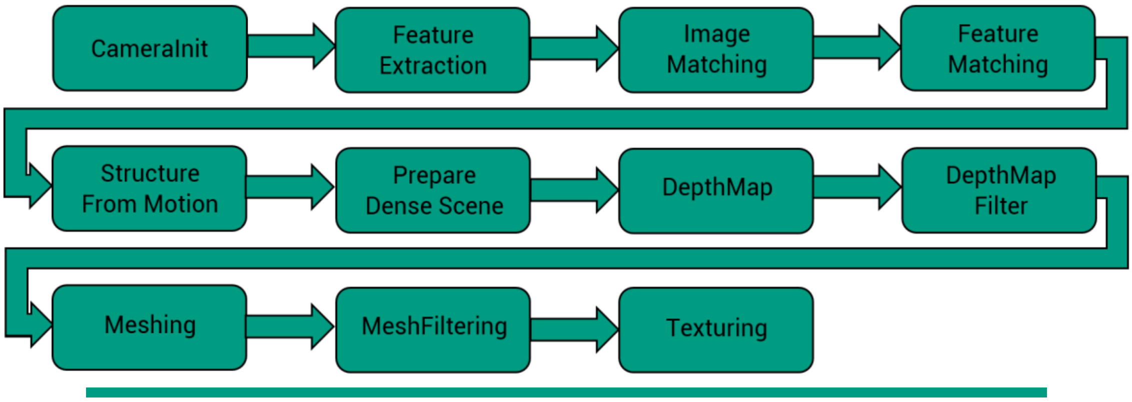


Fig. 17 – Meshroom Nodes

- CameraInit
  CameraInit loads the metadata and sensor data to prepare for the FeatureExtraction

- FeatureExtraction
  This node extracts a certain group of pixels (features). This is done so that the software can deal with viewport changes between the transitions of images.

- ImageMatching
  The node starts looking for images that Meshroom can match together so that it can find specific areas in the scene.

- FeatureMatching
  Meshroom is now starting to match photos using the previously created features and sorts out other pictures that don't match the initially chosen ones.

- StructureFromMotion
  The algorithm starts to create 3D points out of the images.

- PrepareDenseScene
  The node creates .exr images.

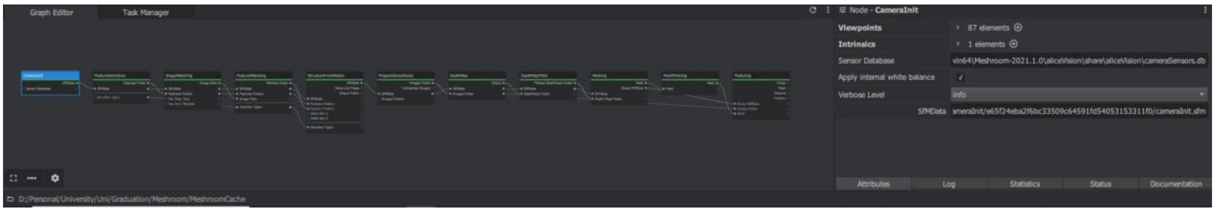
- DepthMap
  DepthMap starts to retrieve depth value for every single pixel, captured by all camera views.

- DepthMapFilter
  To prevent overlapping of DepthMaps, this node isolates areas that are occluded, so depth consistency can happen.
- Meshing
  The 3D model is coming into place through the DepthMap and the point cloud.

- MeshFiltering
  This node filters out unwanted elements.

- Texturing
  The UV mapping and the texture map gets created.

  (Meshroom Contributors, 2020 -a)



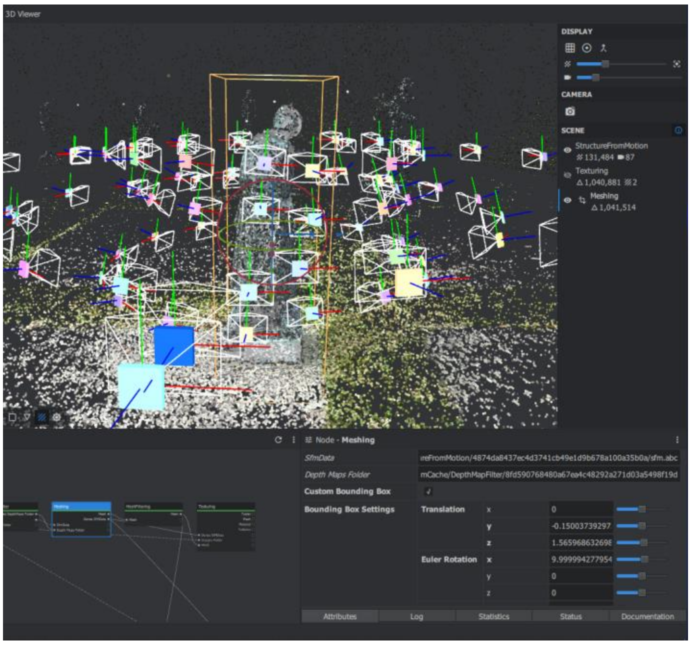
Node - CameraInit

Viewpoints › 87 elements
Intrinsics › 1 elements
Sensor Database: vin64\Meshroom-2021.1.0\aliceVision\share\aliceVision\cameraSensors.db
Apply internal white balance: ✓
Verbose Level: info
SfMData: ameraInit/e65f24eba2f6bc33509c64591fd54053153311f0/cameraInit.sfm

D:/Personal/University/Uni/Graduation/Meshroom/MeshroomCache

Fig. 18 – Image Viewer (Screenshot Meshroom 2022)

While every node has options that can be adjusted, recreating the monuments, only “Meshing”, “Texturing” and “MeshFiltering” need changes. The “Meshing” node has the option of creating a bounding box. The box helps to condense the mesh and only creates a model of the content inside the cube. This is required due to computational constraints.



StructureFromMotion ※131,484 87
Texturing △1,040,881 2
Meshing △1,041,514

Node - Meshing

SfmData: ureFromMotion/4874da8437ec4d3741cb49e1d9b678a100a35b0a/sfm.abc
Depth Maps Folder: mCache/DepthMapFilter/8fd590768480a67ea4c48292a271d03a5498f19d
Custom Bounding Box: ✓
Bounding Box Settings
Translation x 0
y -0.15003739297
z 1.56596863269€
Euler Rotation x 9.999994277954
y 0
z 0

Fig. 19 – Bounding Box (Screenshot Meshroom 2022)

The “Texturing” node lets the user set the desired texture resolution and create the color map for the model. For the UV unwrapping method, Meshroom offers three different methods: Basic, LSCM, and ABF (see Appendix 3). While LSCM and ABF both support UDIMs, the basic method is desired due to the single UV space.

The UV maps that get created, are rough and need to be replaced later down the line if adjustments to the mesh have to be made.

There is one issue with the mesh that gets generated. The problem with the generated mesh is that the polycount for the model reaches over 1 million. That number can be brought down to increase the performance. That's why there is one more node that was put into the string. The node "MeshDecimate" does exactly what the name suggests. It creates a low-poly version of the already existing mesh. The polycount can be set to the desired number. The process works well and almost no detail is lost.

Meshrooms export function is a little bit unique in the sense that every time a node gets completed, it exports the file directly to the hard drive. This allows Mehshroom to immediately start at the same position after a potential crash. That also means that the complete process from start to finish doesn't have to be done in one go.

Fixing the Details

There are often smaller issues with the models after they have been created in any kind of photogrammetry software. These range from geometry that sticks out of the mesh unnaturally, to floating faces that can be deleted as they aren't used. This often happens if the software is missing visual information (e.g. due to the object/building being too high). Those issues usually get solved with drones that can reach high-up places. But since not everyone has access to a drone, these kinds of corrections have to be done most of the time. Even with drones, imperfections are common and have to be cleaned up.

For this project, the software that was chosen for the cleanup was Autodesk Maya. A tool commonly used in 3D modeling that supports the creation of custom UV maps. With Maya, small adjustments to single vertices, faces, or edges can be made. Since Maya supports both modeling and sculpting tools, alterations aren't time-consuming and easy to make.

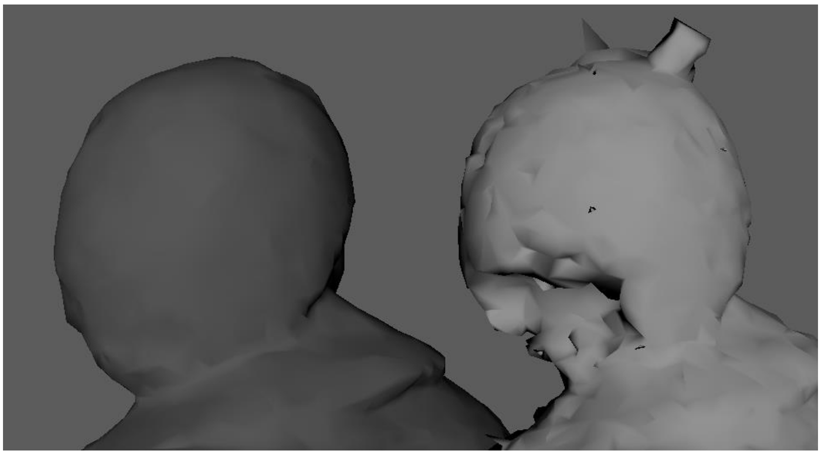
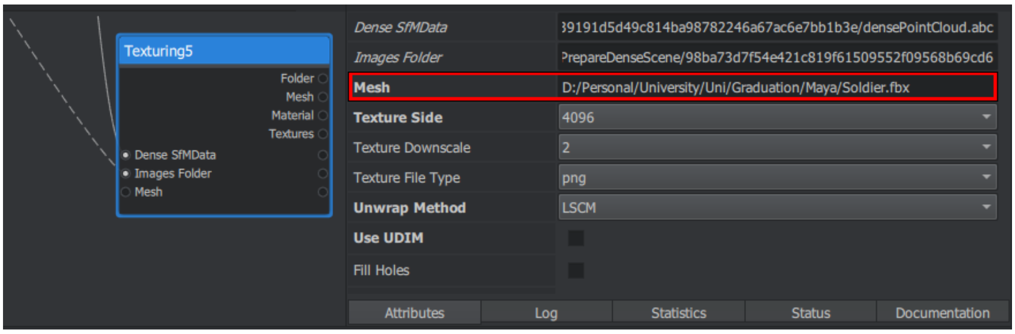


Fig. 20 – Fixed – Original Mesh Comparison (Screenshot Maya 2022)

After all changes on the mesh surface have been completed, new UV maps have to be created. This is necessary for the person later working on the texture painting. A texture artist has to paint over the imperfections that were not properly captured by the camera and will have an easier job with proper UV maps.

With the texture maps finished the mesh has to be put back into Meshroom so that the software can project the color information onto the new UV map. The process for this is done again by the “Texturing” node. Normally, the “Texturing” node gets input from either the “Meshing” or “MeshDecimate” node. Since the model has been worked on in Maya in the meantime, “Texturing” needs the new file exported from Maya. For this to work, instead of using the input from the other nodes, Meshroom can use the folder path and locate the mesh data. An important element for this to work is that the new mesh in Maya cannot under any circumstances be moved or scaled differently.



Dense SfMData	39191d5d49c814ba98782246a67ac6e7bb1b3e/densePointCloud.abc
Images Folder	PrepareDenseScene/98ba73d7f54e421c819f61509552f09568b69cd6
Mesh	D:/Personal/University/Uni/Graduation/Maya/Soldier.fbx
Texture Side	4096
Texture Downscale	2
Texture File Type	png
Unwrap Method	LSCM
Use UDIM	□
Fill Holes	□

Fig. 21 - Texturing Node (Screenshot Meshroom 2022)

After the process is finished, the mesh can be properly painted on. Substance Painter allows the user to use the stamp tool, which makes it easy to paint over imperfections and fix the texture map. This is the final step for the model after which the mesh can be imported into Unreal Engine 5.

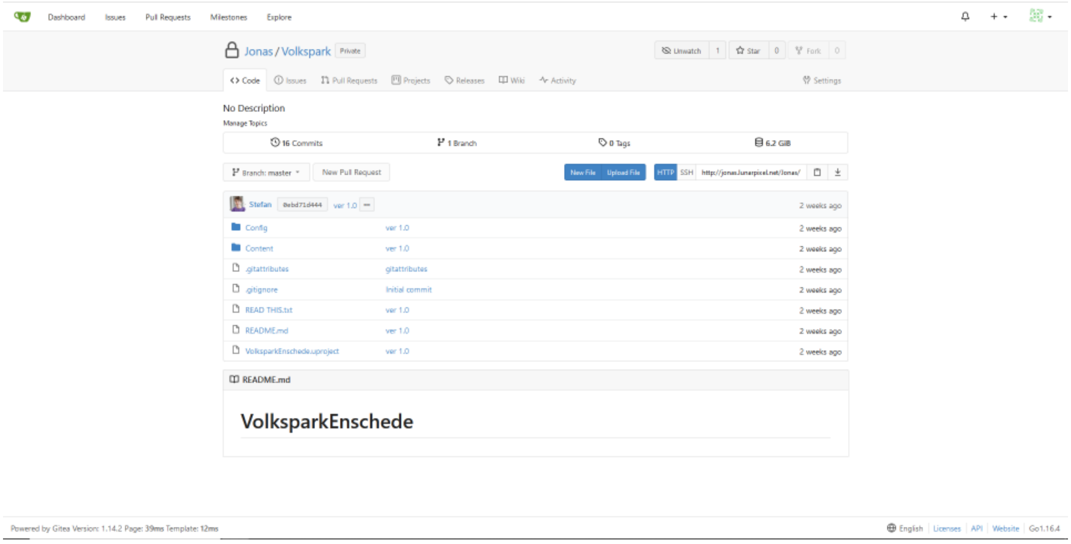
7.3 Using the Unreal Engine

Beginnings with Unreal Engine 4

As previously mentioned, the project was started in the Unreal Engine 4 to ensure the creation of the virtual environment with a realistic art style. The work done in version 4 functioned perfectly fine and was initially supposed to be the software for the whole project. Halfway through the project, Unreal Engine 5.0 was released. This prompted the team to switch over to the newer version. While an early access version was already available in 2021, certain missing functions and bugs like unusable material ambient occlusion (Epic Games, 2022 -b) were considered too difficult to deal with.

Setting up Version Control

It is extremely important to save the changes made for the project. That's why the team decided to use version control. The choice fell on GitHub since it allows more control over each version. The problem with GitHub is that free users are only allowed to transfer a maximum of 500MB per month (Microsoft, n.d.). The solution was to create a private server, which could be connected to GitHub. Companies like Microsoft or DigitalOcean support personal servers with different capabilities. The virtual machine could later be accessed by each team member through a website.



Jonas / Volkspark

No Description

http://jonas.lunarpixel.net/Jonas/

Stefan 8ebd71d444 ver 1.0		2 weeks ago
Config	ver 1.0	2 weeks ago
Content	ver 1.0	2 weeks ago
.gitattributes	gitattributes	2 weeks ago
.gitignore	Initial commit	2 weeks ago
READ THIS.txt	ver 1.0	2 weeks ago
README.md	ver 1.0	2 weeks ago
VolksparkEnschede.uproject	ver 1.0	2 weeks ago

README.md

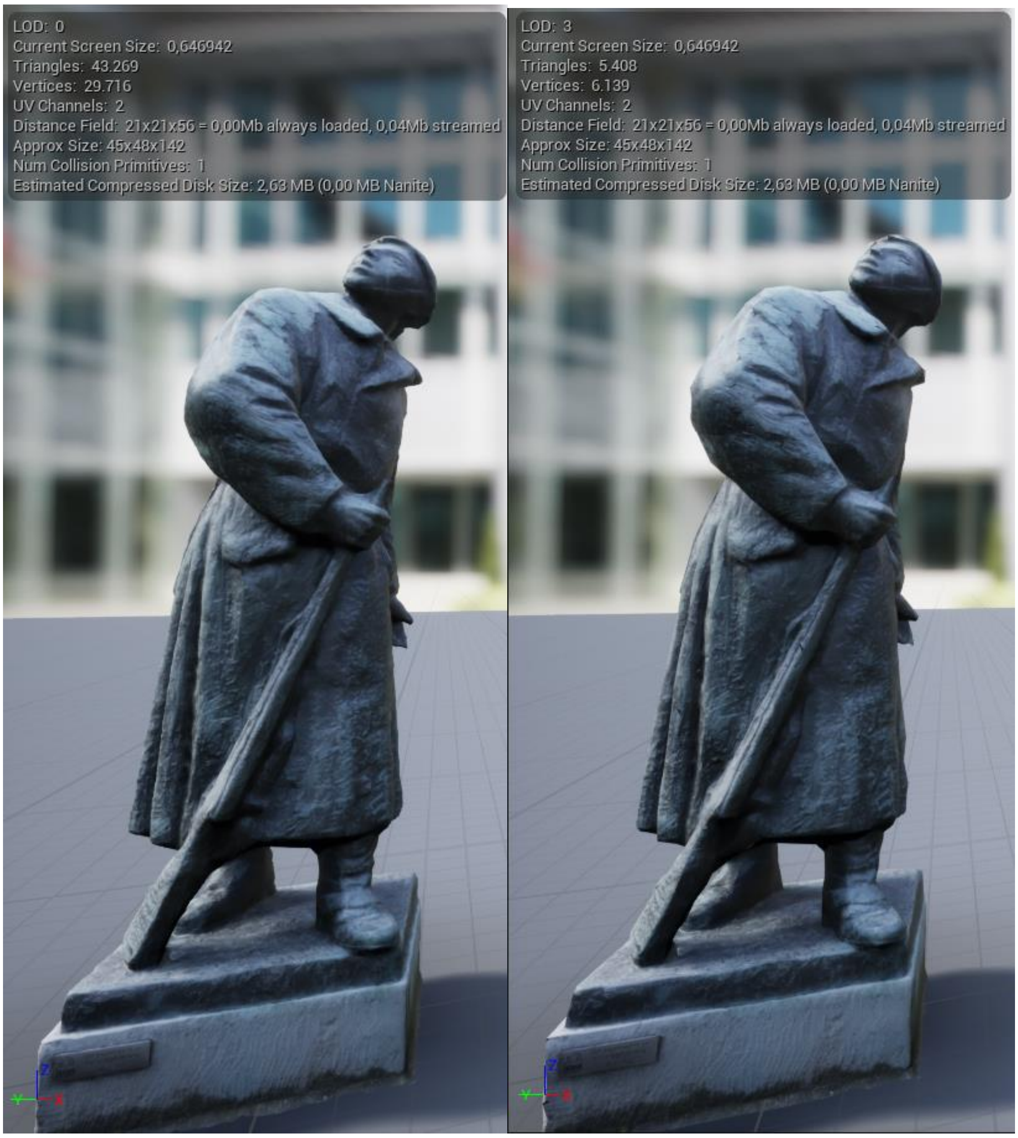
VolksparkEnschede

Powered by Gitea Version: 1.14.2 Page: 39ms Template: 12ms

Fig. 22 – Git Server (Screenshot Lunarpixel 2022)

Measuring Performance

The Unreal Engine 5 can use two built-in methods for 3D models to improve the framerate: LODs and Nanite. While using LODs has been a staple in video game development for over 30 years (David Luebke, 2003), Nanite is a new creation by Epic Games that was first introduced together with Unreal Engine 5 (Epic Games, 2020). While Nanite is exciting, new technology, this project focuses on the use of LODs, due to the prior knowledge of the corresponding workflow. The engine allows the use of LODs for every object. Those can be created during the import of the file, or afterward during the set dressing. Choosing a LOD group automatically splits the mesh into multiple versions, which can be set to the desired view distance. Additionally, the user can choose how many LODs should exist.



```
LOD: 0
Current Screen Size: 0,646942
Triangles: 43.269
Vertices: 29.716
UV Channels: 2
Distance Field: 21x21x56 = 0,00Mb always loaded, 0,04Mb streamed
Approx Size: 45x48x142
Num Collision Primitives: 1
Estimated Compressed Disk Size: 2,63 MB (0,00 MB Nanite)
```

```
LOD: 3
Current Screen Size: 0,646942
Triangles: 5.408
Vertices: 6.139
UV Channels: 2
Distance Field: 21x21x56 = 0,00Mb always loaded, 0,04Mb streamed
Approx Size: 45x48x142
Num Collision Primitives: 1
Estimated Compressed Disk Size: 2,63 MB (0,00 MB Nanite)
```

Fig. 23 – LOD Comparison (Screenshot Unreal Engine 5 2022)

The loss of detail due to LODs is minimal and has a positive outcome on the framerate. To further improve the framerate, other methods like culling have been applied.

Testing & Feedback

It is impossible to create the whole experience without getting feedback from the client. The process of testing the environment is significant and must be incorporated into the environment. To get the most valuable feedback, tests were made with the client in person and additional criticism was gathered through surveys. Thanks to the personal session with members of the Volkspark committee the team figured out where the biggest difficulties lay. This was reiterated through a survey, which was filled out by the same members. The biggest issues were the amount and variety of vegetation. It takes many different trees to create the illusion of a sizable park. With this in mind, the focus was restructured, with more emphasis on foliage. A second survey with similar questions to the first one was conducted, but the team did not get a response from the Volkspark Committee.

8. Justification

During the development of the virtual park experience, certain topics were important. Due to the time constraints given by the university, techniques had to be used, that were efficient in creating 3D models. Photogrammetry was the natural choice when it came to recreating the monuments. It provided a way to accurately display the statues in their current appearance. Manual recreation would have been too time-consuming.

A similar argument can be made for the foliage. While manual work is possible, using SpeedTree is a valid option for realistic foliage creation. SpeedTree incorporates all the tools that are needed for vegetation.

Lastly, the choice of using LODs over Nanite in the Unreal Engine 5 was accepted unanimously. Due to the lack of knowledge, and shortened time frame (due to the release date of version 5), Nanite was not an option at that point in development.

9. Recommendation

If the municipality of Enschede decides to continue the project, certain technical improvements can be made. First and foremost, investments into performance have to be required. Nanite is in an early development phase and is sure to advance to a higher level. Together with Lumen, the Unreal Engine 5 is stable in game development and technology that can be used in all sorts of ways.

Further developments in photogrammetry are also on the way. While Epic Games is currently developing a smartphone app called RealityScan (Epic Games, 2022 -c), Nvidia is working on artificial intelligence that can replicate 3D objects with very few photos (Thomas Müller, 2022). With these new technologies, using photogrammetry for foliage could very well be a reality in the near future.

10. Reflection

The graduation period provided many opportunities to learn. This gaining of knowledge was the result of hard work and the help of fellow students. The project offered the chance to learn the photogrammetry workflow from start to finish. This wasn't always easy, since the process of creating 3D models through photos came with new software and certain hardware limitations. The access to a professional photo camera this time was based on luck. That's why a personal camera is advantageous. The help of the teammates was invaluable and greatly helped the improvement of skills.

Another mistake was the push for version control with GitHub. While version control isn't inherently a bad idea, the teams' skills were severely lacking in this department. This made it difficult to share the project and simultaneously made work very difficult. Learning how to distribute projects properly is an important step in the future.

While the negatives are important elements to improve upon, the positives outshined them. The teammates were great coworkers, which supported the project with the needed skill. It was possible to clearly divide the teamwork and improve the project in every way.

11. Conclusion

In the end, the team considers the project a great success. The learning experience included the basis of photogrammetry, the workflow of the Unreal Engine 5, and foliage creation. The process of creating realistic 3D models was possible through teamwork and research done over 20 weeks. This resulted in six different models that represent the Second World War monuments of the Volkspark.

The foliage creation led to a variety of plants that could be used inside the virtual environment. The base for the plants was investigated through field research by photographs taken in the park.

Thanks to the feedback sessions with the client, the team was able to pinpoint specific flaws that existed within the environment. According to the Volkspark committee, the statues were in good shape, while the amount of foliage could still be improved.

Right now, the client needs to figure out in what direction they want the project to go. They can streamline the upcoming work in a way, which enables a new team to continue the project properly. Since the whole revitalization of the park is set for 10-20 years, it would give the committee enough time to create a fully-functional scene that can be used for various purposes.

References

3DFLOW. (2020, July 07). Retrieved from 3DFLOW: https://www.3dflow.net/3df-zephyr-5-0-is-out-now/

A. Sheffer, E. d. (2001). *Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening.* Springer Verlag London-Limited. Retrieved from https://link.springer.com/article/10.1007/PL00013391

Adams, H. D., Luce, C. H., Breshears, D. D., Allen, C. D., Weiler, M., Hale, V. C., . . . Huxman, T. E. (2011). *Ecohydrological consequences of drought- and infestation-triggered tree die-off: insights and hypotheses.* Retrieved from https://www.fs.fed.us/rm/pubs_other/rmrs_2012_adams_h001.pdf

Alexandrov, G. (2021, December 24). What Camera is Best for Photoscanning? | Photogrammetry Course. Retrieved from https://www.youtube.com/watch?v=ZN8-tzqBLTs

Bruno Lévy, S. P. (2002). *Least Squares Conformal Maps for Automatic Texture Atlas Generation.* Inria Nancy-Grand Est Research Centre, CNRS, France. Retrieved from https://members.loria.fr/Bruno.Levy/papers/LSCM_SIGGRAPH_2002.pdf

Burtch, R. C. (2008). *History of Photogrammetry.* Ferris State University. Retrieved from https://ibis.geog.ubc.ca/courses/geob373/lectures/Handouts/History_of_Photogrammetry.pdf

Carsten Griwodz, S. G. (2021). *AliceVision Meshroom: An open-source 3D reconstruction pipeline.* Retrieved from https://dl.acm.org/doi/pdf/10.1145/3458305.3478443

Centraal Bureau voor de Statistiek. (2022, 01 01). *Opendata.cbs.nl.* Retrieved from https://opendata.cbs.nl/#/CBS/nl/dataset/37230ned/table?ts=1646748260418

Commissie van Beheer over het Volkspark. (n.d.). Retrieved from Volkspark.nl: https://volkspark.nl/geschiedenis/volkspark/

Commissie van Beheer over het Volkspark. (n.d.). Retrieved from Volkspark.nl: https://volkspark.nl/het-park-2/het-park/

David Luebke, M. R. (2003). *Level of Detail For 3D Graphics.* Elsevier.

Epic Games. (n.d.). Retrieved from Unreal Engine: www.unrealengine.com

Epic Games. (n.d.). Retrieved from Unreal Engine: https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/

Epic Games. (2020, May 13). Unreal Engine 5 Revealed! | Next-Gen Real-Time Demo Running on PlayStation 5. Retrieved from https://www.youtube.com/watch?v=qC5KtatMcUw

Epic Games. (2021, May 26). Retrieved from Unreal Engine: https://www.unrealengine.com/en-US/blog/unreal-engine-5-is-now-available-in-early-access

Epic Games. (2022, April 05). Retrieved from Unreal Engine: https://www.unrealengine.com/en-US/blog/unreal-engine-5-is-now-available

Epic Games. (2022, April 05). Retrieved from Unreal Engine: https://docs.unrealengine.com/5.0/en-US/unreal-engine-5-0-release-notes/

Epic Games. (2022, April 04). Retrieved from Epic Games: https://www.epicgames.com/site/en-US/news/epic-games-introduces-realityscan-app-now-in-limited-beta

Grim, A. (2007). *The Origin of the Term Photogrammetry.* University of Stuttgart, Stuttgart. Retrieved from phowo.ifp.uni-stuttgart.de: https://phowo.ifp.uni-stuttgart.de/publications/phowo07/080Grimm.pdf

Haas, J. (2014). *A History of the Unity Game Engine.* Worcester. Retrieved from http://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf

Ho, J. (2015). *Samsung Announces the Galaxy S 6 and S6 Edge.* Retrieved from https://www.anandtech.com/show/8999/samsung-announces-the-galaxy-s6

Hogeschool Saxion. (2019, 11 01). Retrieved from Saxion.edu: https://www.saxion.edu/business-and-research/labs/extended-reality-lab

Interactive Data Visualization, Inc. (n.d.). Retrieved from store.speedtree.com: https://store.speedtree.com/games/

Meshroom Contributors. (2020). Retrieved from Meshroom-Manual: https://meshroom-manual.readthedocs.io/en/bibtex1/node-reference/node-reference.html

Meshroom Contributors. (2020). Retrieved from Meshroom-Manual: https://meshroom-manual.readthedocs.io/en/bibtex1/node-reference/nodes/Texturing.html

Microsoft. (n.d.). *GitHub.* Retrieved from https://github.com/pricing

Panasonic. (n.d.). Retrieved from Photo of LUMIX® Digital Camera DMC-FZ300: https://www.panasonic.com/mea/en/consumer/cameras-camcorders/lumix-digital-cameras/dmc-fz300.specs.html

Paul E. Dickson, J. E. (2017). *An Experience-based Comparison of Unity and Unreal for a Stand-alone 3D Development Course.* Retrieved from https://dl.acm.org/doi/abs/10.1145/3059009.3059013

Rein, M. (2012, October 01). Better with age: A history of Epic Games. (C. Plante, Interviewer) Polygon. Retrieved from Better with age: A history of Epic Games: https://www.polygon.com/2012/10/1/3438196/better-with-age-a-history-of-epic-games

Saxion Hogeschool (Director). (2019). *Saxion opent Extended Reality Lab in Enschede* [Motion Picture]. Retrieved from https://www.youtube.com/watch?v=qJiHJ5HZZQo

Snuif, C. J. (1915). *Familieboek der Van Heeks.* Enschede.

Stichting Cultureel Erfgoed Enschede. (2018, 01 01). Retrieved from cultureelerfgoedenschede.nl: https://cultureelerfgoedenschede.nl/wordpressnew/volkspark/

Sweeney, C. (n.d.). Structure from Motion (SfM). Retrieved from http://theia-sfm.org/sfm.html

Sweeney, T. (2009, May 25). From The Past To The Future: Tim Sweeney Talks. (B. Edwards, Interviewer) Game Developer. Retrieved from https://www.gamedeveloper.com/design/from-the-past-to-the-future-tim-sweeney-talks

Thomas Müller, A. E. (2022). *Instant Neural Graphics Primitives with a Multiresolution Hash Encoding.* Retrieved from https://nvlabs.github.io/instant-ngp/assets/mueller2022instant.pdf

Unity Technologies. (n.d.). Retrieved from Creating And Using Scripts: https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html

Wolf, P. R., Dewitt, B. A., & Wilkinson, B. E. (2014). *Elements of Photogrammetry with Applications in GIS, 4th Edition.* McGraw-Hill Education.

Xrite. (n.d.). ColorChecker Classic. Retrieved from https://www.xrite.com/de/categories/calibration-profiling/colorchecker-classic

Appendix

Appendix 1

Additional development of foliage with SpeedTree:

SpeedTree

SpeedTree is a software for vegetation creation, developed by Interactive Data Visualization, Inc. It lets users create different pieces of foliage in a variety of ways. It is used first and foremost in video games and movie production. Foliage variation stretches from trees and bushes to different types of grass. Users can make adjustments to branches, leaves, trunks, and barks. The software allows to make trees up to photorealism.

SpeedTree can be tested for free without file exportation. For further use, SpeedTree establishes a set price per month, depending on the revenue of the user's product. (Interactive Data Visualization, Inc., n.d.)

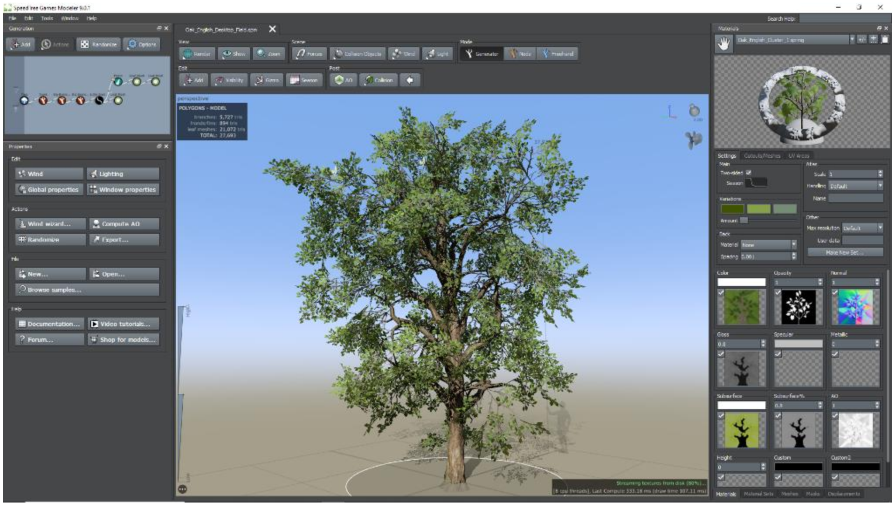


Fig. 24 – SpeedTree user interface (Screenshot SpeedTree 2022)

Why SpeedTree?

The Volkspark in Enschede is home to many different types of trees. While it is too time-consuming to recreate every tree from scratch, certain species that reside in the park are important for the overall feeling that should be evoked through the virtual environment. Foliage generation can be achieved through different methods. Manual creation through software like Maya or Blender would be possible, but very time-consuming in comparison. Alternative programs for 3D foliage creation exist, but pale in contrast with SpeedTree, due to the available options. This is the deciding factor for choosing SpeedTree for this project.

Creating the Base

With SpeedTree, the possibilities for foliage are endless. The tools that the software provides can be adjusted in all kinds of manners and create every type of vegetation that exists. Similar to Meshroom and the Unreal Engine, SpeedTree also uses a node-based system where every part of the tree is separated and can be adjusted accordingly.

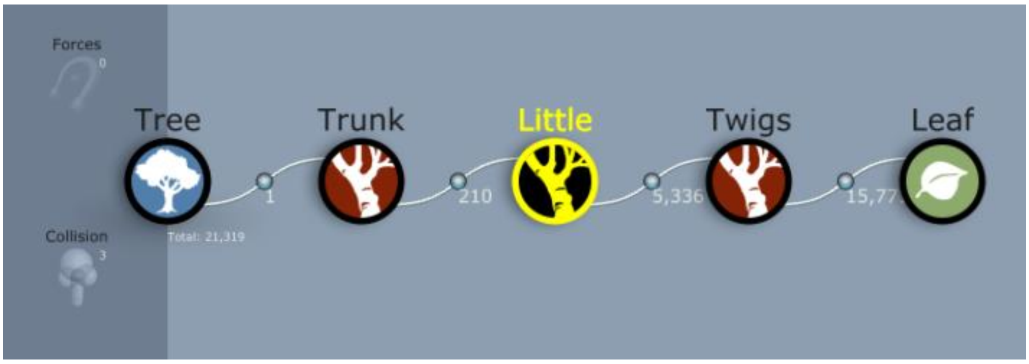
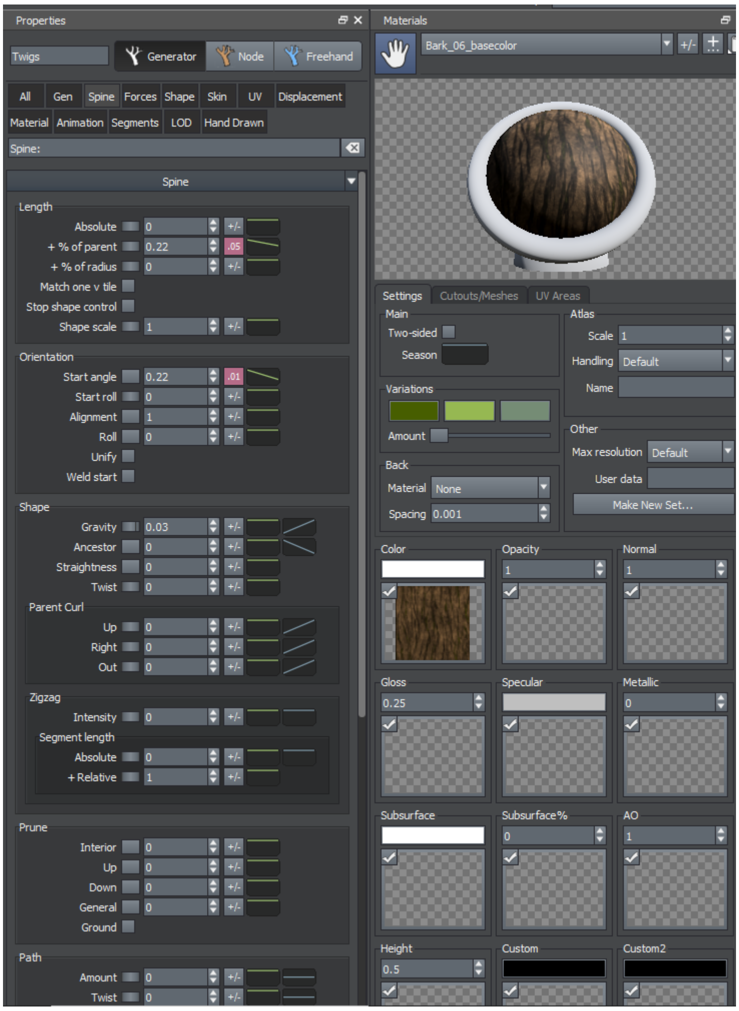


Fig. 25 – SpeedTree Nodes (Screenshot SpeedTree 2022)

The process starts with the trunk. Using the 'Gen' tab in the 'Properties' window, changes for the number of trunks can be made. This is going to be important for the branches, but not the initial trunk. For further changes to the height, width, and other shape-related functions, the 'Spine' tab has to be used. With the trunk set up, the next step is to create the branches. The geometry for the branches can be added directly to the node. It is important to create an evenly spread out collection of twigs so that the tree will look realistic. Additional forces like gravity can be applied in the same tab. To give the tree an accurate look, materials have to be applied to the trunk. This can be done in the materials section. SpeedTree creates a new material that can use other texture maps from outside the software. Detailed texture generation from scratch is not supported by SpeedTree. The material can directly be dropped onto the trunk after the desired texture maps have been loaded.



Twigs

Spine:

Spine

Length
Absolute 0 +/-
+ % of parent 0.22 .05
+ % of radius 0 +/-
Match one v tile □
Stop shape control □
Shape scale 1 +/-

Orientation
Start angle 0.22 .01
Start roll 0 +/-
Alignment 1 +/-
Roll 0 +/-
Unify □
Weld start □

Shape
Gravity 0.03 +/-
Ancestor 0 +/-
Straightness 0 +/-
Twist 0 +/-

Parent Curl
Up 0 +/-
Right 0 +/-
Out 0 +/-

Zigzag
Intensity 0 +/-
Segment length
Absolute 0 +/-
+ Relative 1 +/-

Prune
Interior 0 +/-
Up 0 +/-
Down 0 +/-
General 0 +/-
Ground □

Path
Amount 0 +/-
Twist 0 +/-

Bark_06_basecolor

Main
Two-sided □
Season

Variations
Amount

Back
Material None
Spacing 0.001

Atlas
Scale 1
Handling Default
Name

Other
Max resolution Default
User data

Color ☑
Opacity 1 ☑
Normal 1 ☑
Gloss 0.25 ☑
Specular ☑
Metallic 0 ☑
Subsurface ☑
Subsurface% 0 ☑
AO 1 ☑
Height 0.5 ☑
Custom ☑
Custom2 ☑

Fig. 26 – Properties (left) – Materials (right) (Screenshot SpeedTree 2022)

Twigs

Since the branches have to be filled with leaves, needles, or other vegetation, more additions have to be created. For this example, the choice was made to create a pine tree, so smaller twigs with needles had to be generated. To create small twigs with better manageable details, a second SpeedTree file was created. In the new file, the twigs were created with the same approach as the initial trunk. That way it was possible to let the twig branch out and adjust the needles with the desired values. With the values and materials set, the twig can be exported as a texture map. This is a method to save performance, and can still be adjusted after the maps have been imported back into the other SpeedTree file.

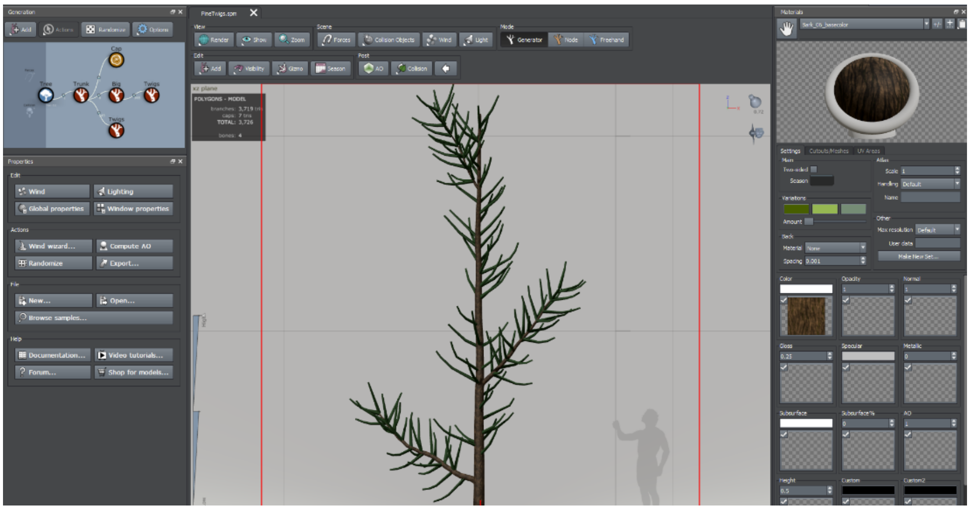
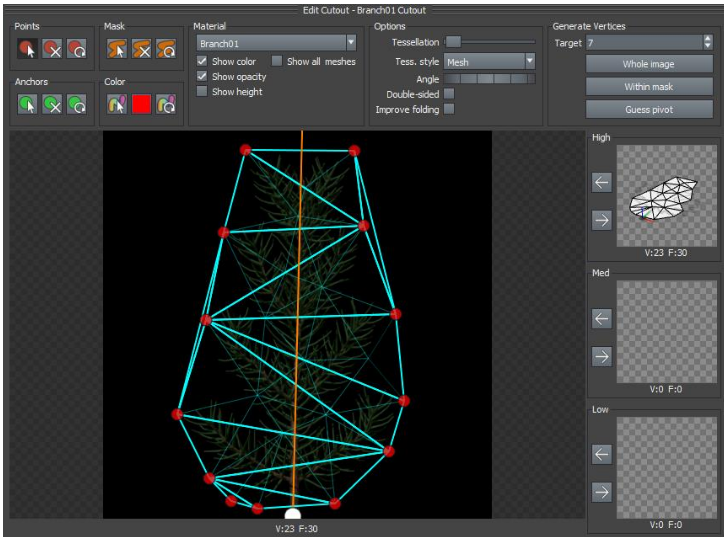


Fig. 27 – Twig (Screenshot SpeedTree 2022)

The reimport of the twig texture maps happens through the material editor. A new material for the maps has to be created. The editor supports a function that lets the user edit the new mesh. That means that the size of the imported twigs can be adjusted and LODs created.


Edit Cutout - Branch01 Cutout

Material: Branch01

☑ Show color □ Show all meshes

☑ Show opacity

□ Show height

Tess. style: Mesh

□ Double-sided

□ Improve folding

Target: 7

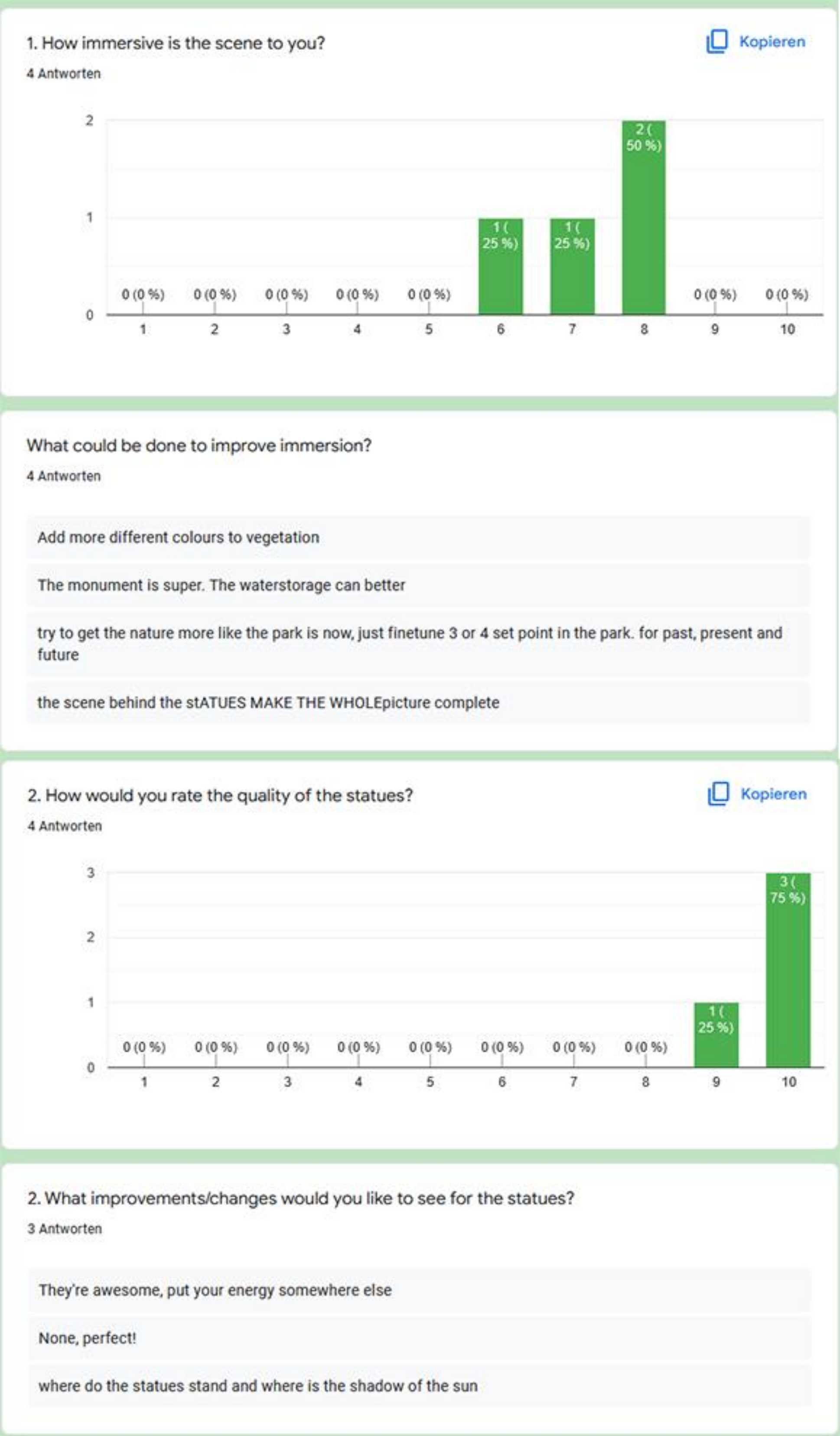
Fig. 28 – Mesh editor (Screenshot SpeedTree 2022)

The final step after the material has been applied to the mesh, is to adjust the values in the 'Properties' setting, so that the twigs look realistic. Additionally, SpeedTree can use external meshes to create a shape of the treetop, if a specific look is desired. Another performance-saving measure is to regulate the number of twigs. In combination with the other settings, it is possible to create the illusion of a dense-looking tree.

This completes the process of tree creation in SpeedTree.

Appendix 2

First survey results with the Volkspark Committee:



1. How immersive is the scene to you?

4 Antworten

What could be done to improve immersion?

4 Antworten

Add more different colours to vegetation

The monument is super. The waterstorage can better

try to get the nature more like the park is now, just finetune 3 or 4 set point in the park. for past, present and future

the scene behind the stATUES MAKE THE WHOLEpicture complete

2. How would you rate the quality of the statues?

4 Antworten

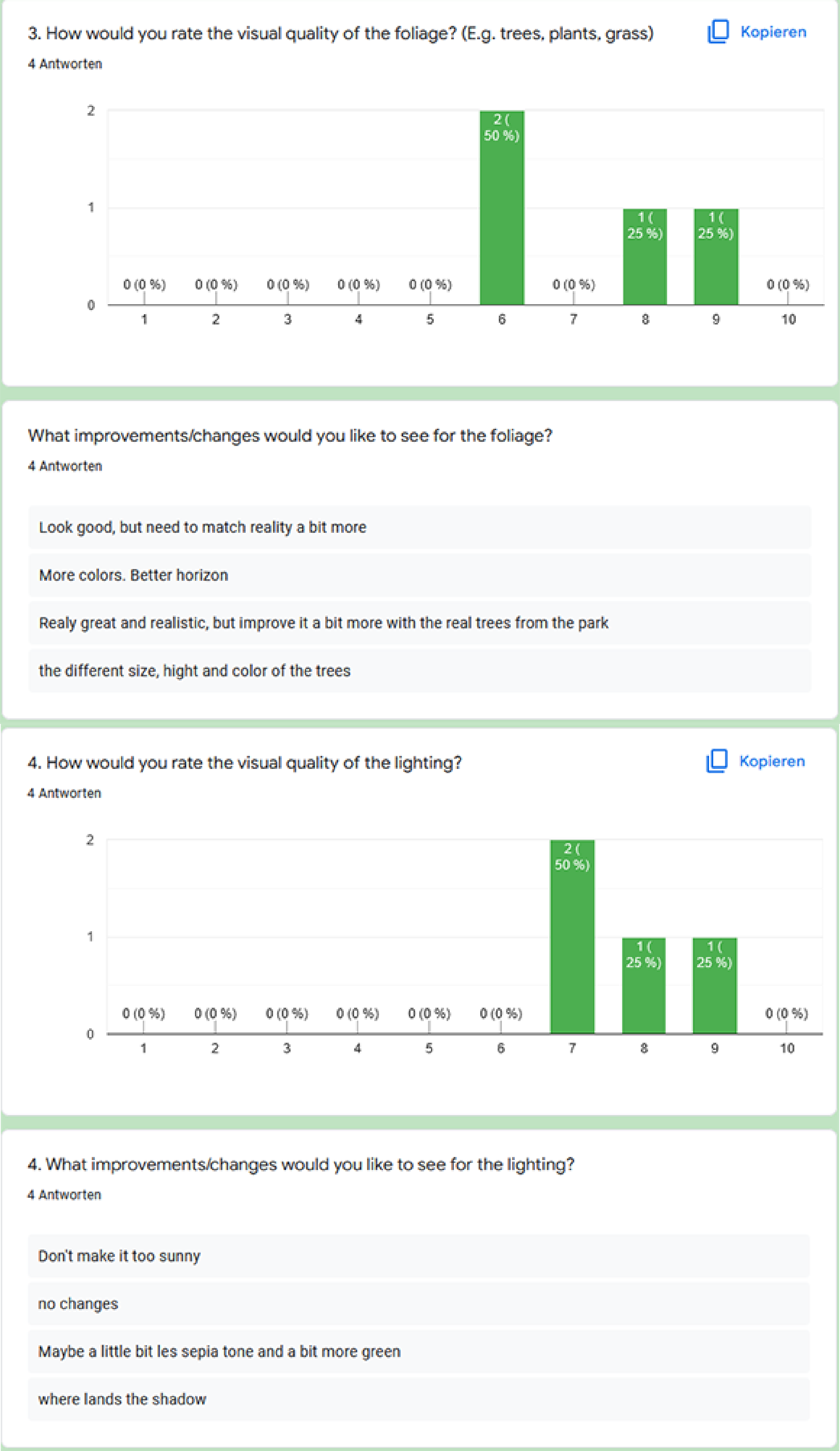
2. What improvements/changes would you like to see for the statues?

3 Antworten

They're awesome, put your energy somewhere else

None, perfect!

where do the statues stand and where is the shadow of the sun



3. How would you rate the visual quality of the foliage? (E.g. trees, plants, grass)

4 Antworten

What improvements/changes would you like to see for the foliage?

4 Antworten

Look good, but need to match reality a bit more

More colors. Better horizon

Realy great and realistic, but improve it a bit more with the real trees from the park

the different size, hight and color of the trees

4. How would you rate the visual quality of the lighting?

4 Antworten

4. What improvements/changes would you like to see for the lighting?

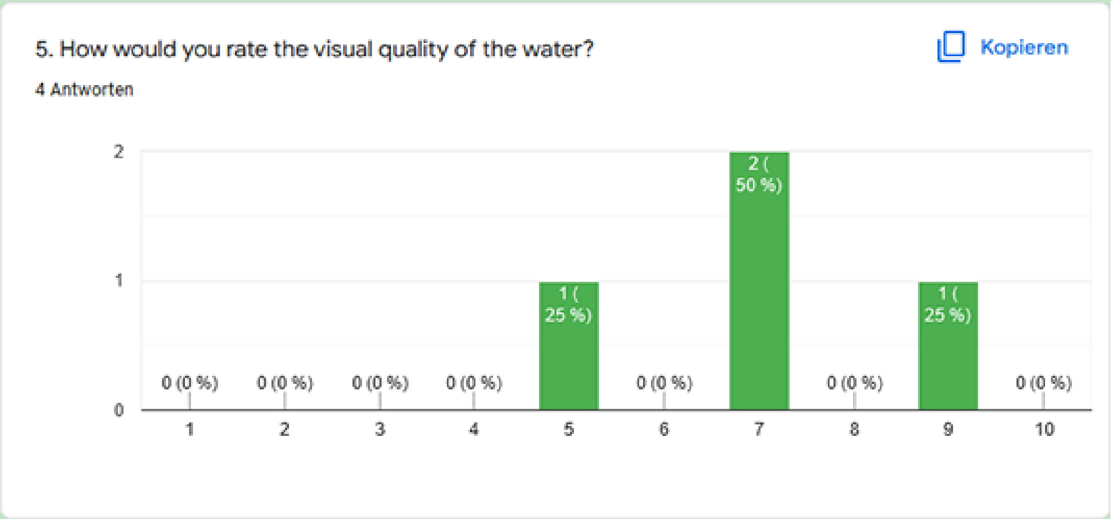
4 Antworten

Don't make it too sunny

no changes

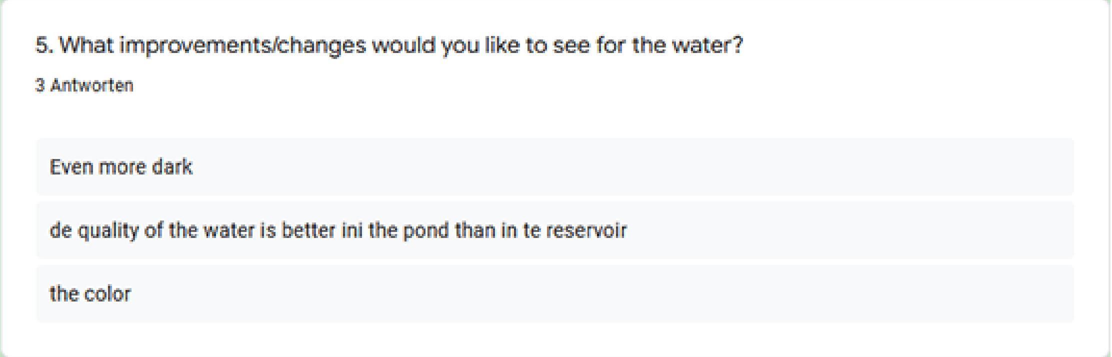
Maybe a little bit les sepia tone and a bit more green

where lands the shadow



5. How would you rate the visual quality of the water?

4 Antworten



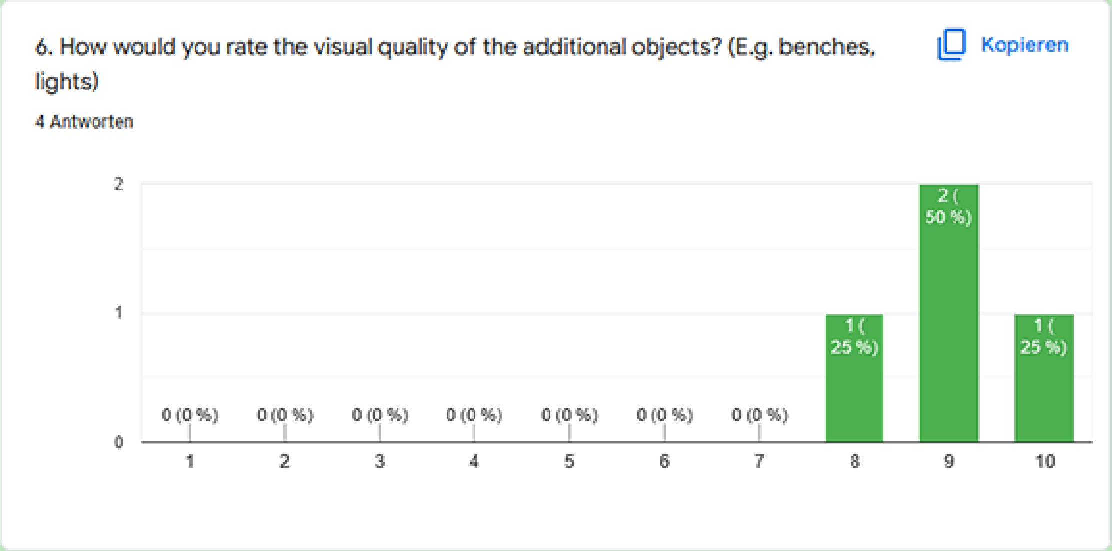
5. What improvements/changes would you like to see for the water?

3 Antworten

Even more dark

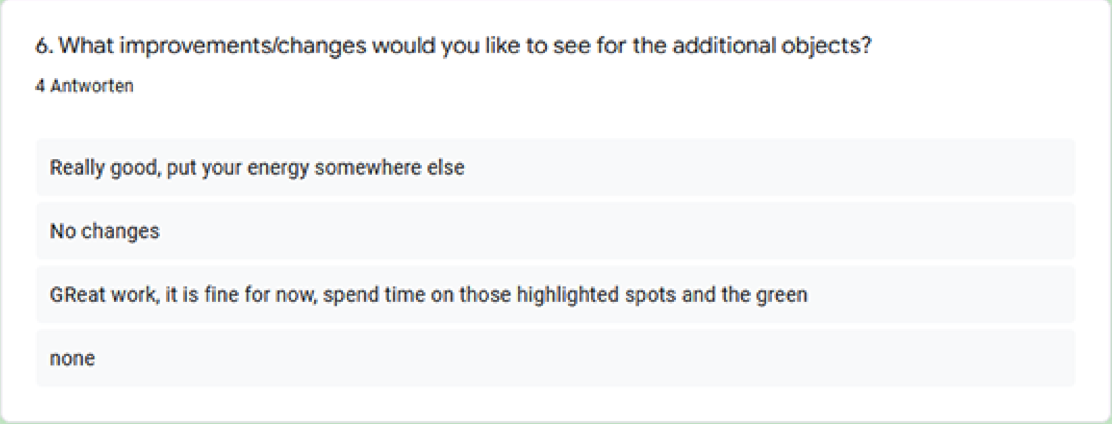
de quality of the water is better ini the pond than in te reservoir

the color



6. How would you rate the visual quality of the additional objects? (E.g. benches, lights)

4 Antworten



6. What improvements/changes would you like to see for the additional objects?

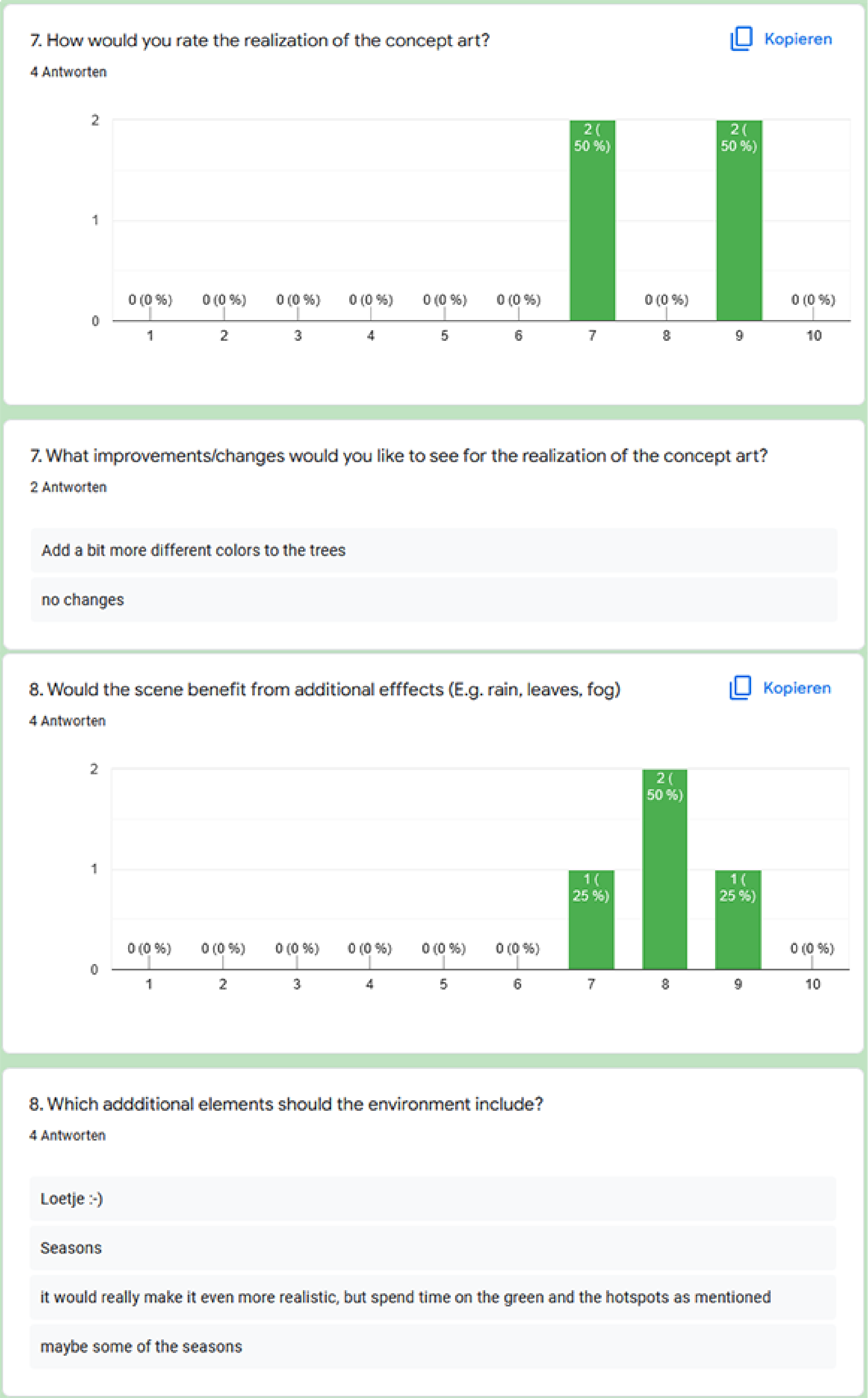
4 Antworten

Really good, put your energy somewhere else

No changes

GReat work, it is fine for now, spend time on those highlighted spots and the green

none



7. How would you rate the realization of the concept art?

4 Antworten

7. What improvements/changes would you like to see for the realization of the concept art?

2 Antworten

Add a bit more different colors to the trees

no changes

8. Would the scene benefit from additional efffects (E.g. rain, leaves, fog)

4 Antworten

8. Which addditional elements should the environment include?

4 Antworten

Loetje :-)

Seasons

it would really make it even more realistic, but spend time on the green and the hotspots as mentioned

maybe some of the seasons

Fig. 29 – Survey (Screenshot Google Forms 2022)

Appendix 3

Different unwrapping methods in Meshroom

- Basic: Meshrooms own unwrapping method, usually used for meshes that are larger than 600k faces. It can create multiple maps for a single model. (Meshroom Contributors, 2020 -b)
- LSCM: The LSCM (Least Squares Conformal Maps) method was established in 2002 for an improved UV workflow (Bruno Lévy, 2002) and can be used for meshes under 600k faces in Meshroom. (Meshroom Contributors, 2020 -b)
- ABF: With the ABF (Angle Based Flattening) method, meshes with less than 300k polygons can be unwrapped (Meshroom Contributors, 2020 -b). The process closely resembles the LSCM method and is interchangeable, depending on the desired outcome. The approach of the method is using a particular parametrization of 3D surfaces that helps with mapping the content to a 2D surface. (A. Sheffer, 2001)

Appendix 4

The Unreal Engine project is available under the following URL (a Saxion account is required):

https://saxion-my.sharepoint.com/:u:/g/personal/471999_student_saxion_nl/EQMPReVfPl9Pjm1uxb1WwwYBUri2xHt4oxMbQ3Dec9l-hw?e=WVxPwJ