Working from home project report

Introduction

This report covers the progression and techniques we used to achieve satisfactory renders. We will discuss our problems and work method before we conclude with the resulting renders. Our project idea was to render a typical work environment in different lighting conditions. We wanted to experiment with how light behaves at different times of the day and how natural light interacts with different objects and other light sources.

Method

Scene

We decided on building the room using standard geometrical shapes seeing as the room only would consist of four walls, the floor, and a ceiling. For furniture and gadgets, we used free *Blender* models which we converted from a *.blend* to *.obj* format. We formatted the object to *.pbrt* using the *obj2pbrt* command. The room structure is built up by six thin squares all connected at the different corners resulting in a sealed room.

All the squares are made up by two triangle meshes. Three walls are the same color, and one wall is colored differently to illustrate how reflection from different colors interact with each other. The left wall is constructed with multiple squares as there was need for an opening in the wall to make room for a window. The room is placed in a 3D environment surrounded by an *EXR* image, making the view outside the window change when the camera/eye is moved around in the room. This allowed us to change the light inside the room as the lighting conditions outside changed. The *EXR* image is implemented as an infinite light source.

Objects

The objects used in our renders were all taken from the webpage *"TurboSquid"* (Shutterstock, 2022) . The only requirement for imported objects was that they had to be in the *.blend* format, and ideally had their textures in a separate folder as *.png* images. This was needed to map the textures to their corresponding objects, lastly, we used *pbrt* to assign appropriate material types to the different objects from the *pbrt* library.

As the objects were *.blend* files we could manipulate them in *Blender* to separate the different parts of the objects we wanted to coat/cover with the corresponding textures. These separated objects got exported as *.obj* files, and later converted to *.pbrt*. When imported, we could use the textures downloaded with the *.blend* file to create a texture and coat the object. An example of this is that the laptop needed to be separated into the screen and the keyboard, each part coated with its respective textures.

To re-assemble the pieces into the objects in the *pbrt*, we had to assign every piece of an object the same translation and rotation values, resulting in the object being constructed back together as it initially was shown in *Blender*. This did however result in a lot of work since we now needed to move each part of an object separately when we wanted to make changes.

To fix this we tried grouping the separate parts of an object into one file/object. This way, changing multiple *Translate* and *Rotate* variables when moving objects around would be easier and the file structure would improve drastically. However, we did not manage to do this, and we believe this kind of object construction is only possible with *.ply* files (like the provided car-geometry object from assignment 2).

Lighting

For lighting, we used three main light sources:

- The environment light source from the outside scene
- The ceiling light
- The lamp on the desk

When configuring the positions and intensities we used *assignment 2* as well as the *pbrt-v3 – input file format* page (Matt Pharr, 2022) as reference material. We set the *Integrator* value to 8 to ensure solid light reflection. This gives the objects a more realistic light setting (since the light rays can reflect eight times). This was especially important for objects such as the glass in the window since bad reflection hurts the "realism" of the render. The outside environment uses the *EXR* file as the main *LightSource*, the ceiling light uses an *AreaLightSource* defined as a square to fit the ceiling light and lastly, the lamp uses a diffuse area light instead of a spotlight, this will light up the light bulb and reflect the light back into the room resulting in a more realistic spotlight than the actual spot attribute.

Problems

Texture

As stated above, when converting from *.blend* to *.obj* and from *.obj* to *.pbrt* the object did not preserve its texture. Multiple solutions were tested; only creating monochromatic objects, single-textured objects, or creating a texture map and wrapping it around the objects. We did find the first two options non-realistic. The last option was constrained by our lack of knowledge in *pbrt* and its *texture* system.

We also tried to use *pbrt-v4* for its support for *assimp*. *Assimp* would per our understanding let us convert *.fbx* files to *.pbrt* files directly. *fbx-files* preserve its texture map when exported from *Blender* to *pbrt* which would have solved our problem. We decided to test if textures would be preserved when converting to *.pbrt*. We never got this far however, due to problems installing and configuring *assimp*. The final option, and the one we ended up with was to split objects into multiple parts (smaller objects) even though it resulted in a bit more work, but we found it to be the best solution to preserve looks and limit workload.

Glass

To make our renders possible a window had to be added to the scene. Our window is built up of a metal frame and glass with the *"glass"* material, its default translucency is equal to one. We followed the same procedure when adding a window, like the other objects we added to the scene. When updating the scene, we noticed the glass was completely black.

Our initial thought was that the window had its transparency value set to zero. However, this was not the case. After various attempts without any luck, a TA was able to find the issue noticing that the max depth (*Integrator*) was set to 1. This means the bounce of light is set to one, meaning that the light will not bounce off other objects. An example is that the light will not bounce from the walls and other objects.

Scene building

The group wanted to create a scene by downloading a scene from libraries online, with an office room. After downloading the scene and rendering it, we encountered a problem when changing the camera's position. We quickly understood that the scenes available online were 360-degree images. Since 360-images don't have any depth, we were not able to move around the scene. We were only able to choose specific degrees to look at, not for instance the coordinates(position) of the camera. Because of this we instead had to create the scene from scratch, building the walls, ceiling etc. manually.

Window

Since the lighting conditions depended on where the sun was at a given time, we naturally had to insert a window in one of the walls to make our renders realistic. Initially, we wanted to remove triangles from within the triangle mesh file but quickly realized that a much faster and better method was to build up the left wall of four sub-walls, leaving an open square in the wall to act as the window. Before scaling these sub-walls to create the window.

What work each member of the team did

The team wanted everyone to experience every aspect of the project (scene building, light configuration, and rendering). However, this became difficult because we were only working with one scripting file, which made it hard for everyone to work at the same time. Initially, we wanted to implement git branches but since the scale of the project was small, we concluded that this was impractical.

Instead, we decided to work in pairs (pair programming) with one person being the "main" programmer and the other person quality checking and providing guidance. This way everyone got to experience the different parts of the project. Tormod was responsible for creating the scene/room. During this process, he worked together with Aleksander to try and construct our room. Having created the main foundation of the render. Responsibility was transferred to Aleksander:

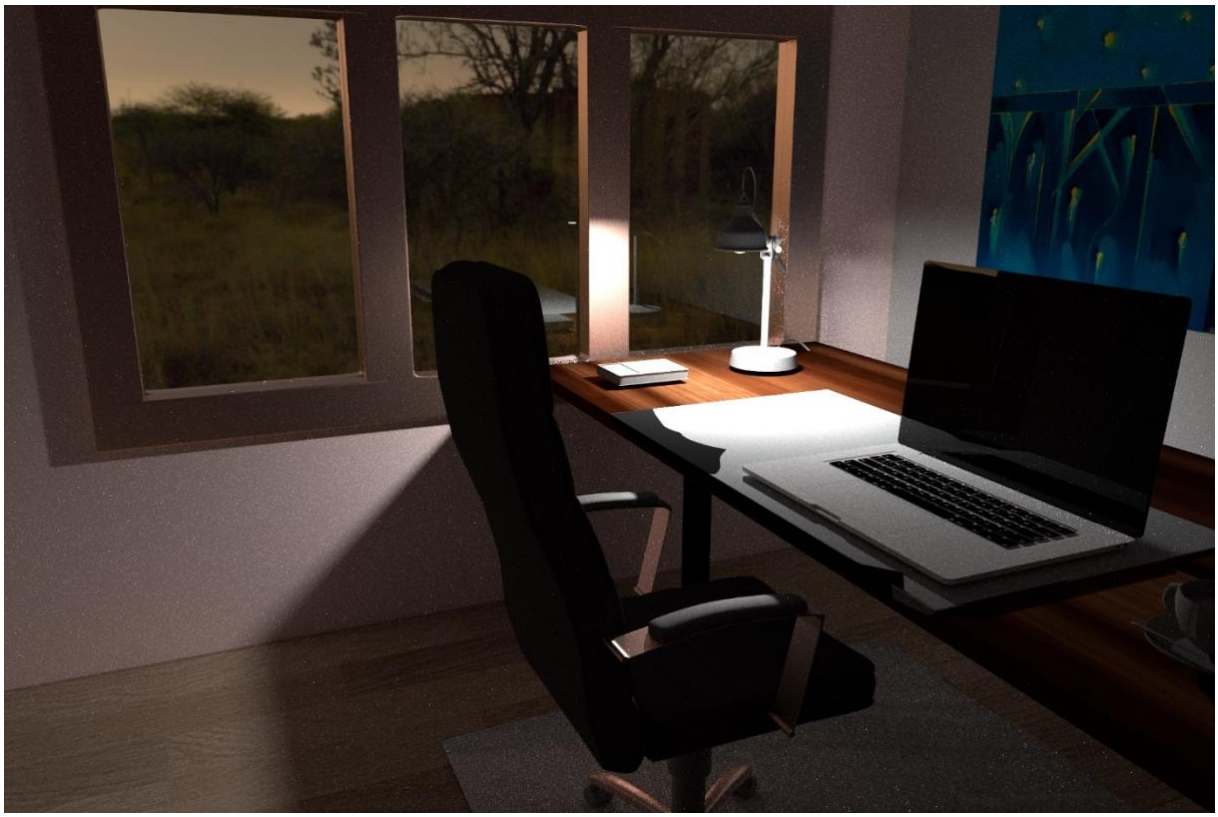
Aleksander was responsible for separating the objects and applying the correct texture. He worked with Martin on this, when the objects had been separated and converted into *.pbrt* files. We began the process of placing the objects in their selected place inside the room. Here Aleksander mainly worked alongside Tormod since Tormod had the main "vision" for how the scene was supposed to look (he also had more knowledge regarding coordinates and placement from scene building). Having placed the objects all that was left was to place and configure light sources and responsibility was given to Martin.

Martin´s main responsibility was to add lighting to the different renders. The lights in the scene are made up of the background light, which will shine light through the window. In addition, the room will be lit up by lamps placed around the room, such as the desk- and ceiling light. For this process Martin worked alongside both Tormod and Aleksander since we wanted every members opinion on the lighting effects as they are essential a realistic render. Lastly, all members worked collectively on the group report and presentation.

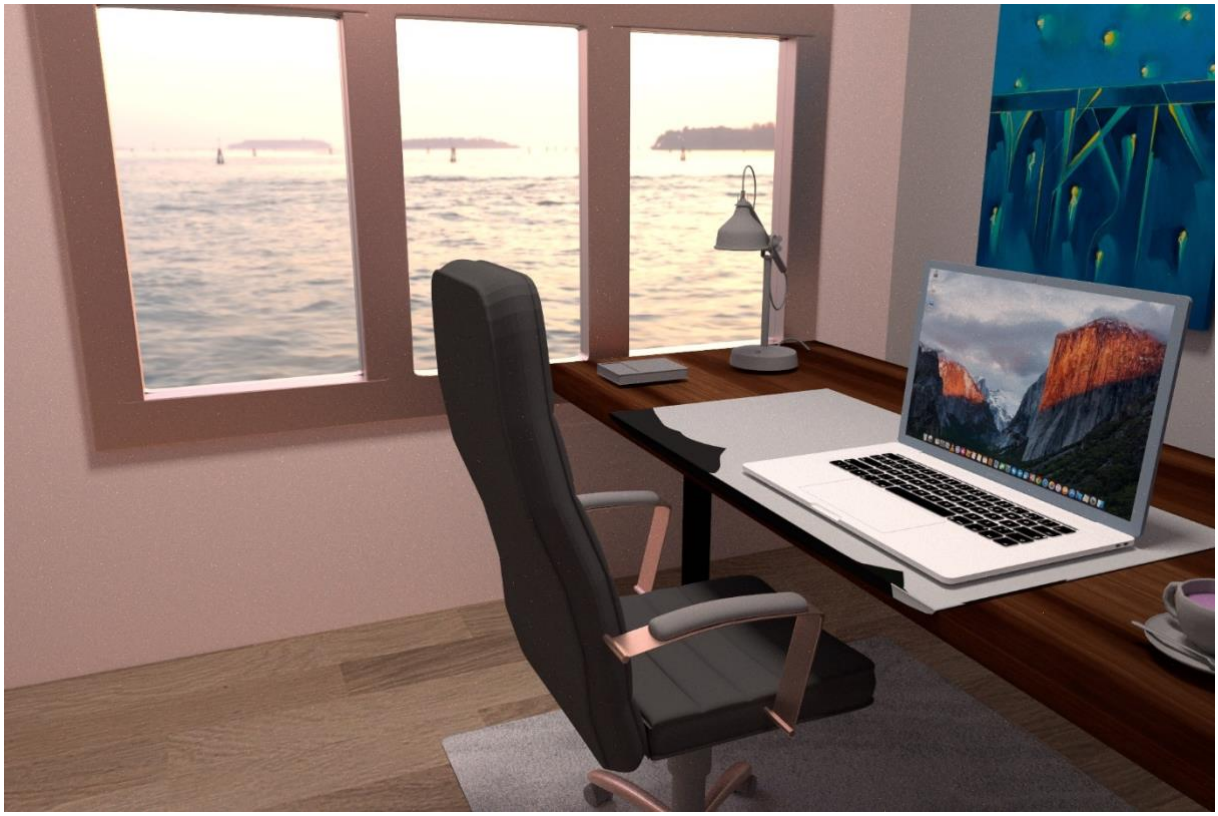
We found pair programming useful both in terms of learning and problem solving. By working in pairs the main programmer always has someone to "spar" or discuss with leading to more effective problem solving.

Results

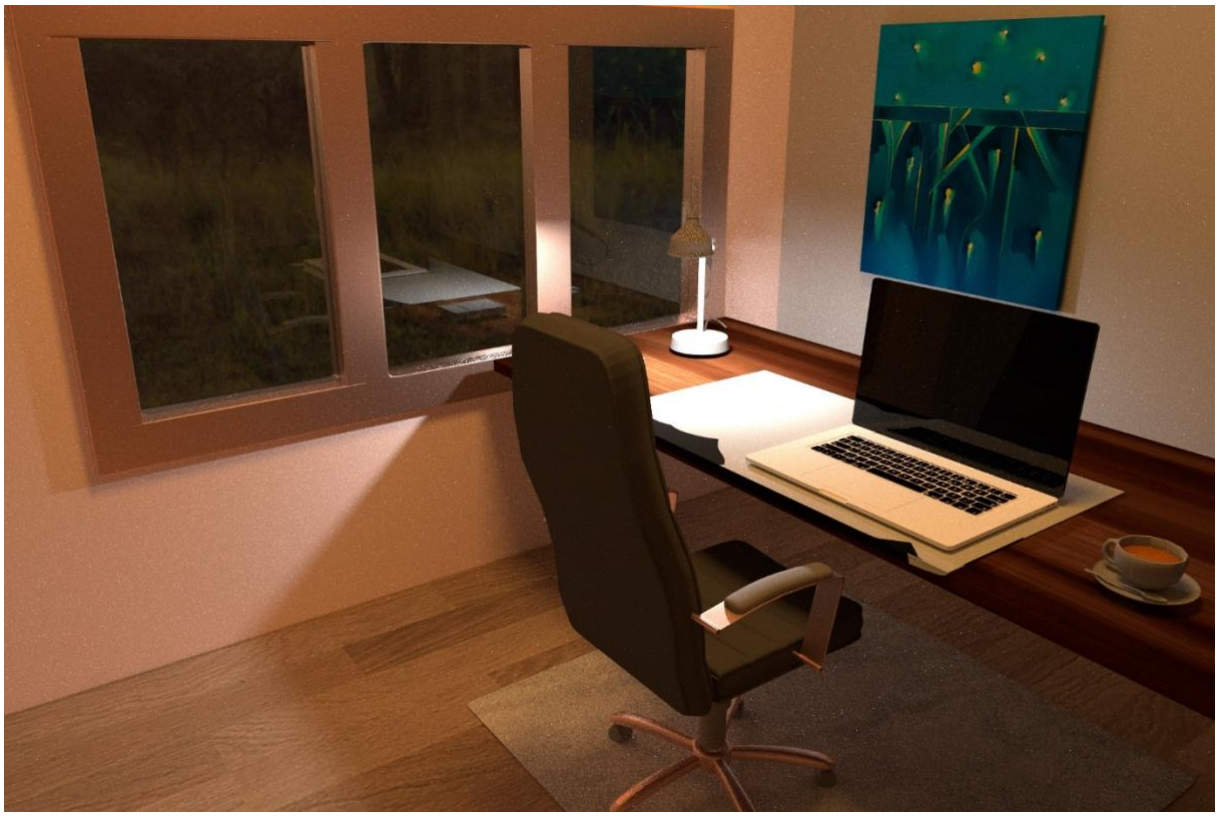
The resulting images show how light behaves at different times of day and how natural light interacts with objects and "synthetic" light in an office environment. We tried making the lighting and objects appear as realistic as possible, this was mostly done by reading on the *pbrt V3 input file format* page (Matt Pharr, 2022). Fine-tuning values was done by trial and error (changing values and building the scene). We also experimented with the depth of *pbrt*`s ray tracing algorithm and ended up with a value of 8 rays. Anything above this made minimal difference to our renders at the cost of much higher computational cost. It is worth mentioning that the images might to some degree look unrealistic. We believe this is due to the provided materials in *pbrt* not matching the textures corresponding to the objects and computational cost. Ex. leather has in our case the material «matte».



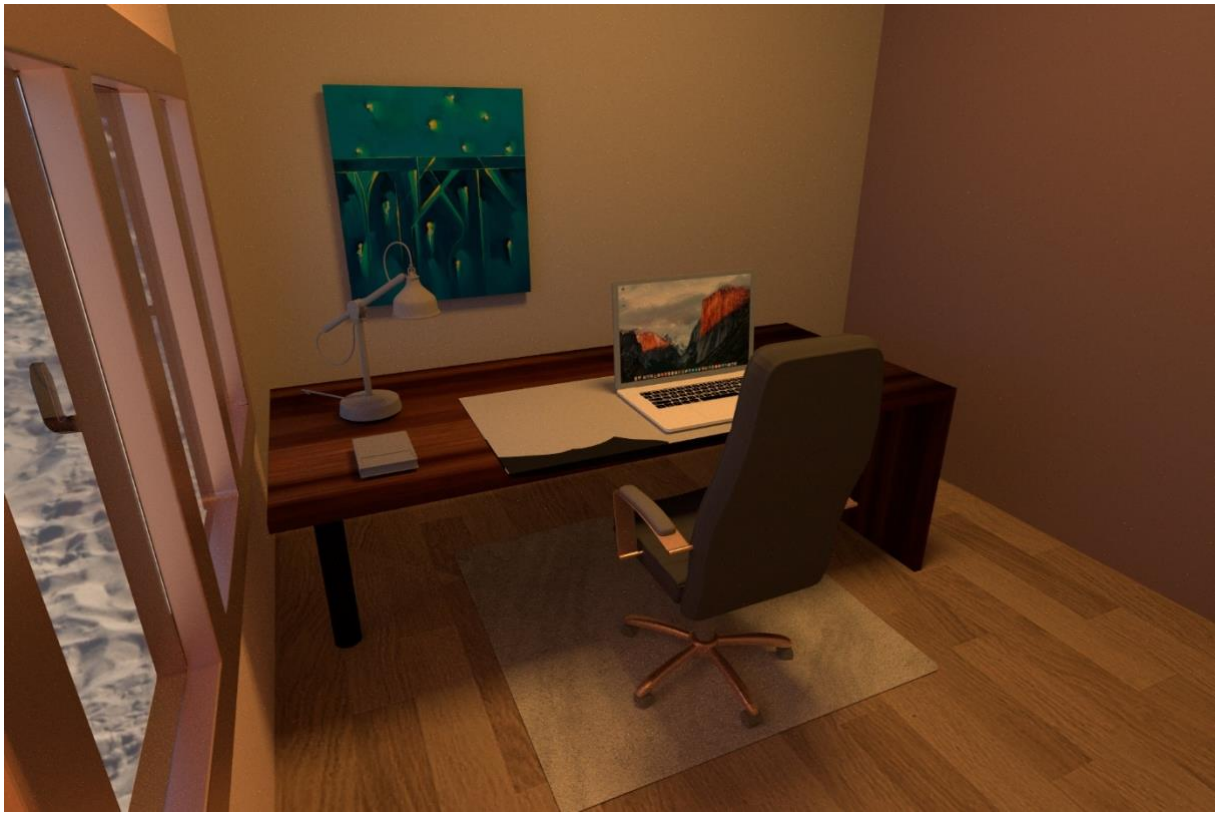
Figur 1 Scene at night with desk light and Macbook turned off



Figur 2 Scene at day with celiling light and Macbook turned on



Figur 3 Scene at night with both desk- and ceiling light. Macbook turned off



Figur 4 Scene at day with ceiling light in warmer lighting. Macbook turned on

Reference material and sitations

Matt Pharr, W. J. (2022, 12 04). *pbrt.org*. Hentet fra pbrt documentation: https://pbrt.org/fileformat-v3

Shutterstock. (2022, 10 19). *Turbosquid*. Hentet fra Turbosquid 3D models for professionals: https://www.turbosquid.com/

Steven Yves Le Moan, T. (2022, 10 28). *Lecture notes* . Hentet fra NTNU: https://ntnu.blackboard.com/ultra/courses/_34652_1/cl/outline