Photorealism in real-time

From photorealistic still renders to interactive environment

Erik Okfors
2016

Bachelor of Fine Arts
Computer Graphic Arts

Luleå University of Technology
Department of Arts, Communication and Education
Foreword

This project was done at Fa Works Sundsvall, Sweden. The time I spent there stretched from the 30th of March to the 25th of May. I am deeply grateful for letting me be with you during that time and thanks for taking your time to help me and giving me the resources to succeed.

Thank you.

Erik Okfors
Abstract

Photorealism is becoming more and more usual in modern media, in movies and commercials, and the game industry is close by. Directly taking a photorealistic scene and making it in real-time can be problematic if not done right. What is it that makes the audience think that what they are looking at is photorealistic, and how can it be done in real-time?

The recreation of the photorealistic real-time scene, from a precreated 3D scene in Autodesk 3ds Max to Unrel Engine 4, was devided in to different parts.

- Analysis, what makes the reference image look photorealistic?
- Optimization, make the assets and models in Autodesk 3ds Max more Unreal Enigne 4 efficient.
- Export and Setting up in Unreal Engine 4, getting the assets from one program to the other and arrange them to match the original.
- Lighting, illuminating the scene with light, using the original scene as reference for the lights position and properties, like color and intensity.
- Materials, applying color and surface to the objects and models. How they look and how they are perceived.
- Camera effects/Post effects, adding depth of field and other enhancing effects to the scene.

The result came out as a walkable livingroom, running at 70-80 fps, though not photorealistic. Most likely due to lighting and material properties being off. Photorealism occur when all elements in the scene, background, characters, objects, etc, does not visually stand out from each other as well as not visually stand out in terms of how the audience relate to them in real life.
Sammanfattning

Fotorealism blir mer och mer vanligt i modern media, så som film och reklam, och spel industri är inte långt efter. Att direkt ta en fotorealistisk scen och göra den i realtid kan bli problematiskt om det inte görs rätt. Vad är det som får åskådarna att tro att de som de tittar på är fotorealistiskt, och hur kan det göras i realtid?

Återskapandet av den fotorealistiska realtids scenen, från en existerande 3D scen i Autodesk 3ds Max till Unreal Engine 4, var uppdelad i olika steg.

- Analys, vad gör att referensbilden ser fotorealistisk ut?
- Optimering, gör modellerna i Autodesk 3ds Max mer effektiva för Unreal Engine 4.
- Export och sätta upp i Unreal Engine 4, få de valda objekt och modeller från Autodesk 3ds Max till Unreal Engine 4 och arrangera dem så de stämmer överens med orginalet.
- Ljussättning, lyser upp scenen med ljus, använder orginal scenen som referens för ljusens position och egenskaper, som färg och styrka.
- Material, applicerar färg och yta till objekt och modeller. Hur de ska se ut och hur de kommer uppfattas.
- Kamera effekter/Post effekter, applicerar en fokus djup och andra förstärkande effekter till scenen.

Resultatet blev en interaktiv lägenhet som renderar med 70-80 fps, dock inte fotorealistisk. Detta troligen på grund av att ljussättningen och material inställningarna var felande. Fotorealism uppstår då alla element i scenen, så som bakgrund, karaktärer, objekt, mm, inte visuellt avviker från scenens globala fysikaliska egenskaper, samt att element inte visuellt avviker i relation till hur åskådaren uppfattar dem i verkliga livet.
# Table of content

1. **Glossary** .................................................................................................................. 1

2. **Introduction** ......................................................................................................... 3  
   2.1 Question formulation ......................................................................................... 3  
   2.2 Purpose .............................................................................................................. 4

3. **Theory** .................................................................................................................. 5  
   3.1 Real-time graphics and game engine ................................................................. 5  
   3.2 Photorealism .................................................................................................... 5  
   3.3 3D programs, Autodesk 3ds Max ................................................................. 8  
   3.4 Optimization .................................................................................................... 8  
   3.5 Lighting ............................................................................................................ 8  
   3.6 Shading ............................................................................................................ 9  
   3.7 Post effects ...................................................................................................... 10

4. **Method** ................................................................................................................. 11  
   4.1 Analysis ........................................................................................................... 11  
   4.2 Optimizing and export ..................................................................................... 12  
      4.2.1 General optimization ............................................................................... 12  
      4.2.2 Exporting and setting up in Unreal Engine 4 ........................................... 12  
   4.3 Lighting ........................................................................................................... 13  
   4.4 Materials ......................................................................................................... 14  
   4.5 Camera effects/Post effects ............................................................................ 14

5. **Result** .................................................................................................................... 16

6. **Discussion** ........................................................................................................... 18

7. **Conclusion** .......................................................................................................... 20

8. **References** ......................................................................................................... 21
1 Glossary

Autodesk 3ds Max
A 3D program, for creating single assets or fully rendered scenes.

Asset
An asset can be anything, mostly referred to a 3D object like a chair or a lamp.

Bake/Baked
A term used in real-time engines when calculating light and dynamics. The engine needs to bake the information to work properly. A scene can be prebaked, meaning that the lights and dynamics has already been calculated.

CGI
Computer generated imagery. A visualization, either image or moving picture, created using a computer.

CrazyBump
A program designed to make maps, such as normal, displacement, occlusion, specularity and diffuse, from photos and textures.

Float
A value that has decimals, like 0.2 or 1.5. If it were an even number, like 3 or 20, it would have been an Integer.

Fps
Stands for "frames per second" and is a show the amount of rendered frames the application does per second.

GoZ
A tool in zBrush that lets you send files and objects between programs, like Autodesk 3ds Max.

Map
A term for a 2D image containing information that can improve the result in a 3D scene, for example a texture map or normal map.

Normal map
A normal map is a way for 3D software to project fake light bumps and curvatures on a low polygon instead of having a high polygon mesh. This information can be calculated down to a 2D image called a normal map.

Polygon
A polygon is a 2-dimensional plane that is n sided, meaning that there’s almost no end to how many sides it can have, though it can have 3 as a minimum. A polygon consists of vertices (multiple vertex) in the corners and connected by edges. The term polygon count means how many polygons a model has.

Render/rendered
A term for the process the computer does when creating a 2D image from a 3D scene.

**Scene**
A term for the 3D workspace in the 3D program. All the models, textures, lighting are contained in a scene.

**Still frame**
A term for a single 2D image that has been rendered from a 3D program. Still frame renders are used in movie production and other frame by frame productions, where the render time does not have to be restricted, as opposed to real-time where the frame needs to update many times per second.

**Texture**
A 2D image, known as a texture map, that is used on models to add custom color or images to the asset.

**Vector 3**
A 3 vector is a parameter with three float values that can define a position, a direction or a color.

**Vertex:**
A vertex is a point in space that defines corners and intersections of geometric shapes. Plural: Vertecies

**VFX**
A short term for Visual Effects.

**UE4**
A shortening for Unreal Engine 4

**UV**
UV coordinates is a way to project a 2D texture on a 3D mesh. The U and the V stands for the coordinates across the 2D texture, since X, Y and Z are used for coordinates in 3D space.

**zBrush**
A 3D program like Autodesk 3ds Max, with a focus on sculpting.
2 Introduction
Every day, the media industry keeps pushing forward to achieve realistic graphics for movies, games and applications, and every day the expectations of the audience rises. 3D, as we see it today is fairly new, only a couple of year ago we thought that Super Mario to the super Nintendo had amazing graphics. Today we see games and movies that looks, once again, astounding.

The bachelor thesis was made at FA Works, working with visualizing and VFX, who sees commercial opportunities for interactive visualizations and wants to research the potentials to implement it in to their production process.

2.1 Question formulation
What is important to perceive a cgi environment as photorealistic?

This project is based on a 3D rendered picture from a 3D scene in Autodesk 3ds Max. Are there any disadvantages with working from a precreated 3D scene used for still frame renders? Are there any tips or good practices that can help when working with a pre made scene.
2.2 Purpose
The purpose of this report was through converting a 3D scene in Autodesk 3ds Max to the real-time engine Unreal Engine 4, make a conclusion to method, quality and consequence when creating photorealistic sceneries for games and other real-time applications.

Figure 1 shows a 2D rendered image from the 3D scene in Autodesk 3ds Max that this project was based on. The conversion was made from that scene and the goal was to achieve the same level of photorealism.

![Vray rendered from Autodesk 3ds Max](http://faworks.se/)

This report is going to cover analysis of photorealism, optimization, lighting, and shading in a game engine. Also, this report is also going to be made in Unreal Engine 4, results and setting may vary with different programs.

---

1 [http://faworks.se/](http://faworks.se/)
3 Theory

3.1 Real-time graphics and game engine

Real-time
Real-time applications run on a constant updating software, which renders a new screen depending on the set frame rate, games frame rate is around 60 fps. Unlike when creating a movie, which only need to render out the whole movie or a scene once. Meaning that the rendertime per frame can be much higher when creating a movie compared to a real-time application that needs to update a specific number of times per second.²

Game engine
A game engine is a software that lets designers and programmers construct games and other real-time applications. Unreal Engine 4 is one of many game engines, other are Unity³ and CryEngine⁴.

3.2 Photorealism

Photorealism
The term photorealism, also known as hyper- or superrealism, points to early artists who used photos to aid there work when painting, and has been around for a long time but flourished during the 1970. In later years this has been more and more common in the 3D world, where 3D artists try to create photorealistic pictures using computers. Photorealism also expresses the freedom of not only creating realistic pictures of what we know and can see, for example portraits, but also the impossible.

Photorealism is not to be confused with realism. Realism is made from real life and everyday experiences, where as photorealism is based on a photograph. Well made photorealistic images comes so close to realism that it is nearly indistinguishable from a photograph

When it comes to real-time, it's more problematic, at least with today's technology. Real-time is as mentioned, something that updates all the time while it's running. Compared to a photorealistic picture that is rendered from a 3D program, with a render time of, for example, 2 hour. If that exact scene were to be put in a game engine, where the engine would try to render the scene 60 times per second, the result would not be playable.

Creating photorealistic imagery
The creation of any cgi, computer generated imagery, consists mainly of four parts.⁷

³ https://unity3d.com/
⁴ http://www.crytek.com/cryengine
⁵ http://www.theartstory.org/movement-photorealism.htm
⁶ http://www.blenderguru.com/articles/achieving-photorealism-in-blender/
⁷ http://www.blenderguru.com/articles/achieving-photorealism-in-blender/
• Modeling
• Lighting
• Materials
• Camera Effects. It is generally called the rendering step when used with still frame rendering, but since this project is rendered using a real-time engine it is called Camera Effects.

References
Use references so it is clear what kind of work is needed. How detailed the models should be and what can be faked using maps. What kind of lighting is there? Where does it come from, what time of the day is it, and how does it bounce in the scene? What are the models made of and what material should be used on it? What camera was used when photographing, are there any distortions or other camera effects?

Modeling
Looking at an asset that needs to be built, how should it be done and how much detail should be used on it. Is it better to fully model a flower's petals, with all its curvature and bumps, or is it better to model the silhouette and use a map to get the desired detail?

It is very important to know the difference between modeling assets for real-time rendering and still frame rendering. Due to performance, models generally should not have as high resolution as still frame models when working real-time applications.

Lighting
Lighting makes us able to view the world in 3D, otherwise without shadows that comes with lighting, objects and surfaces would look flat. We can generally also relate to the time of the day with proper lighting.

Materials
Materials does not only give the object its desired color or texture, it can also provide the necessary detail to achieve photorealism, giving the desired reflection, refraction or roughness. Looking closely at different objects, bumps, wrinkles and edges become visible that heightens the illusion that the object is real. These can be added with maps, such as normal or height map. Other types of maps used with materials are specular or glossy map.
Camera effects
Since photorealism is the recreation of photographs, knowing what a camera does to the picture is important for the recreated result. The camera is made out of lenses that bends and distorts the image. The person using the camera can add desired effects to the image, such as focal blur or depth of field to give it a more natural look. But other effects appear by themselves.

- Sun flares is when the light hits the lenses and makes an overexposed part in the image. From there, the reflection and refraction in the lenses makes a so-called lens flare.
- Chromatic aberrations is an effect where high contrast areas and low contrast areas clash, making the border split into red, green, and blue.
- Distortion, in many cases barrel distortion which slightly bends the image outwards. This is due to the fact that the camera lenses are concave and convex.
The key to photorealism is in the detail. All the thing that together add up to a believable photorealistic picture. These details include:
- The right intensity and color on bounce light.
- Surfaces on materials
- The right perspective and right proportions.

Common mistakes:
- Completely black or super sharp shadows. In an environment where bounce light occurs, shadows rarely are black or sharp.
- Flat surfaces. A surface without any surface detail, like normal or displacement map, tend to look fake and lifeless.
- No edge bevel, perfect corners. Real life object rarely have perfect edges.

3.3 3D programs, Autodesk 3ds Max
3D programs like Autodesk 3ds Max\(^\text{10}\) or Autodesk Maya\(^\text{11}\) are softwares that allows the user to create, animate and render 3D objects. They are used when creating movies, games and other computer generated projects. When working in a program, you work in a so called scene.

3.4 Optimization
\(^\text{12}\)Optimization is a process that adjusts the properties of the object, texture or scene to enhance its performance. This is important for game or other real-time applications where the creator wants high detail and fast rendering. A limit is set based on the hardware of the platform the application is used on and the software engine that is used to create the real-time application.
There is a large number of steps that is taken in to consideration to reach this limit but some of the main ones are polygon count, textures and lighting.
- Polycount is measured in number of vertices per object, the higher the count, the more processing power it takes to render the scene.
- Texture performance is measured in its size and the number of textures. Often many small textures are combined in to a larger, atlas map, since it is often more effective for the game engine to use one big texture than many small ones.
- Lighting setup can range from a simple setup to a complex. If the lighting setup is simple, computational cost for the light will be small, while having a complex lighting setup can have high computational cost.

3.5 Lighting
**Static, stationary and dynamic lighting**
Lights are essential for making any graphical visualization. As a result, it is good to know when to use static, stationary or dynamic lights in the scene.

\(^\text{10}\) http://www.autodesk.se/products/3ds-max/overview
\(^\text{11}\) http://www.autodesk.com/products/maya/overview
Static lights are used for completely prebaked lighting and cannot be changed in anyway during the game. Also it doesn't change or recalculate while the game is running.

Stationary lights are lights that can change values like color or intensity during the game but cannot change position.

Dynamic, or Movable, lights can change its values, like intensity, color and position, during runtime, meaning that the engine needs to calculate the lights intensity, where it is, and how the light affects the scene every frame.\(^{13}\)

### 3.6 Shading
Shading is the step that gives object their color and surface in a game engine, and it is done with materials in Unreal Engine 4. An example can be a wooden table, where the object is a table and the material is wood. It is important to define what the material is so that it can be replicated correct, in 3D. The following subheadings are a part of shading.

**Physically based shading**
\(^{14}\)Physically based shading means that the engine approximates how the light would react in real life based on the material properties resulting in a more accurate representation of how light behaves in real life. It is important to use references, for example photos or props, and to know how the material you are making looks and reacts in real life.

**Base color**
Base color determines what the overall look of the material is going to be. An object can be green by assigning a material with the color green to the object. Textures can be plugged in to the base color to be used as the base. Same a scolor, but with an applied 2D image to be used as base for the object.

**Roughness**
Roughness is a value that determines the smoothness of the material surface, this gives the material its glossiness, and gives a natural specularity and light bounce to the material.

Roughness value of 0, a completely smooth surface where the light bounce of perfectly and make the material very glossy. Roughness value of 1, the surface is filled with bumps and pits which reflects the light in different directions when it hits, spreading the light over the surface.

In real life, there's hardly any material that is completely flat, every material has some bumps that spreads light. This is an important factor in making a realistic material.

\(^{13}\)https://docs.unrealengine.com/latest/INT/Engine/Rendering/LightingAndShadows/index.html

\(^{14}\)https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/PhysicallyBased/index.html
Metallic
Metallic is a value that determines the metallic property of the material. It is determined by a value between 0 and 1, where 0 is nonmetallic like rubber or plastic, and 1 is pure metallic.

Specularity
Since the parameter for how shiny the surface is the roughness, specular is there to change cavities and shadows, but it’s on a small scale. It is set to 0.5 as default since all materials have specularity, even though it can be hard to see.

Material Instances
Instance materials are based on a parent material and can only change parameters that it inherits from the parent. Ex. a parent can choose to only bestow color or intensity to the child. These instances take less performance to render and are good for multimaterial objects, like a detailed lightbulb, which can have 4-5 different components.\(^\text{15}\)

3.7 Post effects
Post effects allows the artist/designer to tweak the overall look and feel of the scene by adding different effects. These effects include Bloom\(^\text{16}\), scene fringe\(^\text{17}\) and Depth of field\(^\text{18}\)

\(^\text{15}\) https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/MaterialInstances/index.html
\(^\text{16}\) https://docs.unrealengine.com/latest/INT/Engine/Rendering/PostProcessEffects/Bloom/index.html
\(^\text{17}\) https://docs.unrealengine.com/latest/INT/Engine/Rendering/PostProcessEffects/SceneFringe/index.html
4 Method

The method explains the process of creating a photorealistic real-time scene through conversion from a premade 3D scene in Autodesk 3ds max.

The method of creating the scene in unreal engine 4 was divided into different tasks to better explain the process. The tasks are in chronological order:

- Analysis
- Optimization
- Export and Setting up in Unreal Engine 4
- Lighting
- Materials
- Camera effects/Post effects

4.1 Analysis

In the analysis, information is taken out of the picture, about what is it that makes this picture photorealistic.

The following information about the reference image, figure 3, are the foundation of the photorealism in the picture.

Models

The models are visually well made, with good silhouettes and no sharp edges or ill-formed geometry. The textured assets have no visible uv seams or noticeable uv tears.

An example, the towel hanging in the kitchen, shows a good representation of how real life objects behave, with natural gravity for the towel. The same effect can be seen on the chair, the spacing between the planks where they are joined shows that it was built from pieces, not carved from a solid block.

Lighting

The reference image shows strong light coming from outside the apartment, white sunlight with relatively sharp shadows suggests that it is probably midday.
This is the main light source. The lamps in the scene cast a soft yellow light, which breaks away from the sunlight. The light bouncing from the floor and walls spreads evenly in the open area and less in the kitchen area where there are fewer windows. The small lights under the cupboards are dimly lit but can still be seen on the kitchen tiles.

**Materials**
The reference shows apparent materials that are recognizable and therefore are easier to relate to real materials. The wooden floor shows no seams between the boards and has a shiny reflectivity meaning that it probably has been treated. The pot in the kitchen has a highly reflective, shiny metal material that reflects most of its environment. Compared that material to the kitchen sink, which has a matte metallic surface, still reflecting light but no clear environment reflection. The different wood material on the furnitures in the livingroom. The livingroom table which has a matte finish surface, compared to the chair seat and the livingroom bench which has a shiny finish. The leaves on the plant has a translucent material that brings out the organic feel to the object.

**Camera/rendering**
The only noticeable camera effect in the rendered image was a small depth of field effect that lightly blurred the farthest corner in the livingroom, and a bloom effect that shines through the windows, adding to the intense sunlight effect.

### 4.2 Optimizing and export
The original scene in Autodesk 3ds Max was made for still frame rendering and therefore not optimized for real-time solutions. For this project to work, optimizations was needed to fit the real-time application.

#### 4.2.1 General optimization

**Textures**
Taking the original texture from the scene in Autodesk 3ds Max and optimizing them, mostly by shrinking them to a smaller version. Other maps were added together to save memory.

**Polygon count**
Keeping the silhouette of the model intact from all important angels and cutting away detail on less noticeable areas. The detail was then added later with maps.

#### 4.2.2 Exporting and setting up in Unreal Engine 4
Not everything in the original scene was exportable, only the models were exported from Autodesk 3ds Max. Other elements such as lights or materials could not be directly used in Unreal Engine 4. Materials in the Autodesk 3ds Max scene was used with a different rendering method. The lights in the original scene are specific to Autodesk 3ds Max and are not compatible with Unreal Engine 4. Textures were taken from a library outside the program that came with the 3D scene and did not need any exporting.

---

The importance in the exporting step:
- World Position. This information made sure that the assets kept their world position. By doing that, the relationship between assets stayed the same in both programs.
- Material index. Models with multiple materials or textures were assigned different materials in Autodesk 3ds Max for each part. Ex. Material 1 for left arm and material 2 for right arm.

4.3 Lighting
Going back to the scene in Autodesk 3ds Max to find out what kind of lights it was using and what their position were in that scene. Using that as a reference, the scene was then constructed with static light, since nothing in the scene were going to be dynamic. The first lights put in the scene were the smallest light, the ones under the kitchen cupboards and fan. Then proceeded to place the lights in the middle of the room, over the kitchen corner, and then the light in the open area. Lastly the main, outside light source, since it was the largest. By doing the process smallest to largest, there was a noticeable view how the lights impacted the scene, as for doing it the other way around where the small lights may disappear in the large light.

The lights properties, for example color and intensity, were then matched with the lights from the original scene for maximum photorealistic effect.

Figure nr 4
Blue = Smallest lights
Yellow = Secondary lights
Red = Main outside light
4.4 Materials
In the export and optimization step, the number of materials needed for a specific asset was set in Autodesk 3ds Max. By using the original 3D scene as a reference, the materials were set to the different assets. The focus lied in the material properties, for example roughness, color and texture and transparency. It was also during this step that detail maps, for example normal maps, were added to give the object the desired texture and detail. All the materials were recreated as close to the materials from the original scene in Autodesk 3ds Max to heighten the photorealism.

4.5 Camera effects/Post effects
Final step included adding camera effects and post process effects such as matching color and brightness, depth of field, and bloom. Unreal engine 4 has a post process effect volume\(^20\) that, when placed in a scene, can apply different effects within the volume:

- Anti-aliasing
- Ambient cubemap
- Bloom
- Color Grading
- Depth of field
- Eye adaptation
- Lens flare
- Post process material
- Scene fringe
- Screen space reflection
- Vignette

An ambient cubemap effect was added as well as a color grading effect to match the reference image. The ambient cubemap was used to brighten up the whole scene and the color grading for changing the final color setting. These effects were used to recreate the general look to the original scene in Autodesk 3ds Max.

Next were the effect that represent the camera and real-time experience. These were used to raise the photorealistic illusion in the scene.

Eye adaptation
Eye adaptation, also called auto exposure, is a post process effect that mimics the eyes ability to change its light intake when moving between different exposures, from a dark room to a bright and vice versa. Since the scene has bright lights in almost every area, a small value to the eye adaptation scale gave the scene more life. This was to enhance the real-time photorealistic experience, since this effect does not occur on a still frame image.

Depth of field
Applies a blur to the scene in front behind the camera focal point. This effect was added to enhance the spectator’s perception of depth in the scene and heighten the photorealism.

---

**Bloom**
Bloom is an effect that brightens intense light, making them flare up creating an illusion of intense light. This effect’s intensity was set quite high to give the scene a bright light.
5 Result

Figure nr 5.1, Result of in-game photage.

Figure nr 5.2, Reference photo

Conversion
The image shown in figure 5.1 was the result, an apartment that was interactive and rendered in real-time running at over 60 fps.

Models were optimized down to a more performable level of detail while keeping large photorealistic details and silhouette.

The lighting was set up using the original Autodeks 3ds Max scene as a reference, matching the position and direction of the lights between the real-time and the original scene.

The color and texture in the material match well between the scenes, but reflective and roughness properties were off, making most surfaces look unnaturally plastic and non lifelike.

A small depth of field effect was added to enhance the feeling of depth in the scene.
Photorealism
The models had good visual silhouettes and no visible tears in uv or noticeable seams.

The lighting was underdeveloped and had too much ambient light, light that illuminates the whole scene without direction. Which also resulted in the absence of distinct shadows.

The materials had good color and texture but lacked the proper representation of the material properties such as reflectivity and roughness.

The bloom effect was scarce while the depth of field and eye adaptation complements the scene well.
6 Discussion

The optimization and integration of the models were fine and not a problem, the silhouettes and surfaces were well done and had no visible defects that would ruin the image. Even the smaller objects in the scene, like the vegetables in the kitchen counter, had convincing photorealistic models. Though looking back to the polygon limit set during the optimization step, it could probably have been higher. The end result had an fps rate over 60 fps, without problems.

As for the camera effects, the only real effect that was relatable to photorealism was the depth of field and it worked well. The bloom effect did not add as much as envisioned during the method. This could have been because of the eye adaptation effect, even though the eye adaptation added to the real-time photorealism.

Looking at the reference image, there was a good representation of sun light coming in from the windows and lighting up the room. Also, the light with the materials in the scene makes you believe that the floor was wood and the carpet was cloth because it was correctly recreated and could therefore be related to real life. The most noticeable differences between the result and the reference photo can be seen on the shadow in the kitchen, the light bouncing in the living room and the reflectivity on large surfaces. The material attributes were off and it resulted in plastic looking surfaces as well as the wrong roughness setting, which made the reflections look concentrated and made the surface look metallic. Which was good for the plastic and metallic surfaces like the pots and kitchen tools but not for every material in the scene.

Most of the problems occurring in the result was that they did not look and convince the spectator that what they were looking at could be a photo.

A common practice when working with photorealistic images is that they can be worked on after they are rendered in image editing program like Adobe Photoshop. Often used to highlight or decrease rendered parts of the image to maximize the photorealistic result, like adding contrast or changing the tint of color. This option of after render production, is taken away when working with real-time solutions, therefore elements, like models and materials, needs to be perfectly made and balance each other to achieve photorealism as well as balance the need for detail against performance in the real-time engine. Adding to the pressure of creating all the steps, modeling, lighting, materials and camera effects, perfect.

A small disadvantage that happened from time to time when working with the premade scene was optimizing unnecessary large and complicated models. Models that were modeled in high detail in places no one could see it. Other than that, no disadvantages were discovered and mostly good came with working this way.

During the optimization step, a polygon count limit, compared to the size of the model, was set. If the models polygon count was under that limit, it was skipped in the optimizing step. This saved time and unnecessary work.

http://www.adobe.com/se/products/photoshop.html?sdid=JRSIX&s_kwcid=AL!3085!3!79955686523!e!gt!photoshop&ef_id=VaVZSwAAAPTQGqOR:20151213174147:s
Another good practice was that looking at textures that could be merged together, without changing the uvs of the model. This did not only save the amount of textures needed, it also saved memory for Unreal Engine 4.

It was revealed at the beginning of the project that Unreal Engine 4 did not, at the time, have a good solution for rendering mirror surfaces. While trying for a period of time to fake the effect with multiple cameras and reflective surfaces and post effects, it was later decided to remove it from the scene. This was a setback since having the mirror could probably heighten the illusion of photorealism in real-time.
7 Conclusion

The result did not reach photorealism. The main problem was most likely due to the lighting and materials, and therefore the balance between them was not equal. The materials were not recreated well enough to convince the spectator that what they were looking at was the desired recreation. Many materials had the same properties, like reflection the roughness. Though they were different types of materials, like wood and polished metal, they behaved the same.

The lighting was off, making the shadows next to nonexistent, which in turn made object look that they were disconnected from other object, like a floating chair.

All the step in the production did not reach a level of perfection and resulted in an inbalance between the elements that made the result loose its photorealism. This showed that the audience will most likely perceive a scene as photorealistic if the scene is properly made, the elements in the scene, background, characters, objects, etc, does not visually stand out from each other, and the elements does not visually stand out in terms of how the audience relate to them in real life.

The precreted scene had a few overcomplicated models that needed optimization. Models that would probably be faster, not necessarily better, to remodel from scratch than optimize. Other than that, no real disadvantages when working from a premade 3D scene.

Good practices can be to really dwell down in references. This is one of the reasons why this project failed, the lack of reference material. Also, to know what the end goal is. During the optimization step, less detailed could have been taken away since the scene was so small.
8 References

Links
Photorealism:
http://www.theartstory.org/movement-photorealism.htm

http://www.blenderguru.com/articles/achieving-photorealism-in-blender/

https://documentation.apple.com/en/aperture/usermanual/index.html#chapter=r=18%26section=11%26tasks=true

http://www.dxomark.com/About/In-depth-measurements/Measurements/Distortion

Unity
https://unity3d.com/

CryEngine
http://www.crytek.com/cryengine

Adobe Photoshop
http://www.adobe.com/se/products/photoshop.html?sdid=JRSIX&s_kwcid=AL!308513!79955686523!e!!g!photoshop&ef_id=VaVZSwAAAPTQGqOR:20151213174147:s

Game Optimization:

Unreal Engine 4, physical material
https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/PhysicallyBased/index.html

Unreal Engine 4, material instances
https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/MaterialInstances/index.html

Unreal Engine 4, lighting basics
https://docs.unrealengine.com/latest/INT/Engine/Rendering/LightingAndShadows/index.html

Unreal Engine 4, post process effects

Autodesk 3DS Max
http://www.autodesk.se/products/3ds-max/overview

Autodesk 3ds Max, fbx exporting
AutoDesk Maya
http://www.autodesk.com/products/maya/overview

Livingroom reference picture, Fa Works:
http://faworks.se/
&
http://www.mitthem.se/CM/Templates/Article/general.aspx?cmguid=25d2d105-8b2a-4ff8-89fe-efef9479bd3c

Crazybump website:
http://www.crazybump.com/

File types
.tga
http://www.fileformat.info/format/tga/egff.htm

.fbx

Books