CASE STUDY
AN OVERVIEW OF MY WORKFLOW THAT I USED FOR
THE A33 OFFICE PROJECT
USING THE UNREAL ENGINE 4
CHAPTER ONE: INTRO

CREDITS AND BACKGROUND

First, here’s the video that I created for this project: https://youtu.be/iwfvt5LV5rA.

The A33 Office project was concluded during the first half of 2016 under the name of NODE Visual, the archviz studio that I co-founded a couple years ago. My colleagues, Attila Bakó and Attila Fodor did a fantastic job on helping me with the modeling process.

The interior designer for this project was the very talented Kristóf Göbölyös from ART FRONT Hungary.

The goal was pretty simple: to make it a static walkthrough scene so the offices can be explored in Unreal. The priority for this project was to achieve a lifelike (or I'd say V-Ray-like) visual quality that has a wow-effect on people.

THE APPROACH

The first step was to define my approach to this project. Choosing the right workflow is always important. If you do it wrong, you might lose a couple days. In case of strict deadlines, it might be a problem.

The goal was to achieve a smooth user experience on computers equipped with even mid range video-cards, so optimizing the scene to get a good FPS rate was key. When you create content for animations, you tend to not really care about the FPS value - and that’s a big difference regarding to the workflow.

The scene contained lots of furniture and accessories to make it alive. So dealing with high object count was definitely a challenge.

Considering all the above, I used the following approach:

(1) I paid attention to the poly count. Displaying lots of polygons at the same time might require extensive calculations of the GPU.

(2) The challenge was to achieve a reasonable object count. Unreal generates one lightmap for every object (for LODs as well). Lightmaps play a major role in smooth user experience.

(3) I optimized textures. Although Unreal Engine is pretty good at working with any texture size, I had to make textures as optimized for game engines as possible to achieve a smooth user experience.

(4) I used two different workflows for modeling: one for the structure, one for furniture and accessories. I wanted gadgets to be instanced objects in the Editor.

(5) It was very important to me to keep things organized within the project file. Epic made it quite easy to manage your resources in the Content Browser, but it might become messy anyway, thanks to strict deadlines and never-ending modifications. I like to keep my project files tidy to make it easier for others to take over the work if needed.
CHAPTER TWO: MODELING

I used 3ds Max for modeling. My suggestion is that you should go for any 3d application that supports unwrapping (3ds Max, Maya, Blender, Cinema 4D).

If you are using SketchUp or any CAD softwares such as Revit or ArchiCAD, you should further work on your 3d model using one of these 3d applications before importing it into the Editor.

Fabrice Bourrelly talks about why it is a good idea to do so in his great webinar: click here to watch it!

LAYERS

I’ve developed my own layer structure over the years. It just simplifies my workflow.

I used 4 main layers: for structural elements, furniture, accessories and misc objects (such as the covering of the model or reflector planes for the lights). I named them STR, FUR, ACC and MISC. I created sub-layers as well for better transparency.

What’s the point of using this technique?

Well, it’s a great way to keep things organized in 3ds Max and later in the Editor. Before exporting the 3d content, I renamed every object. New names were derived from the layer names.

This technique has been “field-tested” for a couple times and it works like a charm.

UNITS

You should always set proper units in 3ds Max before starting modeling.

I went to Units Setup and adjusted the Display Unit Scale to Centimeters then checked the System Unit Setup as well to make sure 1 Unit was 1 Centimeters.

If these units are not matched you will have scaling issues after exporting your scene into the Editor. The most common mistake is to have your unit set up in Centimeters while you leave System Unit Setup in Inches. As a result, your scene will be 2.54 times smaller than it should be.

STRUCTURE, FURNITURE AND ACCESSORIES

I tend to use 2 different approaches when it comes to modeling in 3ds Max.

Method #1: I create the entire model in Max.

This method speeds up content creation a bit. Arraying objects in Max is faster than in the Editor. But it also has its cons. First, you won’t have any instances in Unreal. Second, it will increase the file size of your package.
I created the whole structure using this method. The overall polygon count of the structural elements ended up at 3.5 million polygons.

**Method #2: modeling the main objects in Max and then arraying them in the Editor, after exporting.**

I call it ‘uasset type of modeling’. If you want to work with instances in the Editor, that’s the way to do it. If you need to modify an object, you just change the model in Max. After reimporting it, every instance will automatically update. It results a smaller filesize too. The downside of it is that it takes more time to array them in the Editor.

For walkthrough purposes you should always use this approach. I created all the furniture and accessories using this method.

About the origin: in the Editor, pivot points are derived from the origin of the coordinate system in 3ds Max. So I placed every object in a way so their geometrical center was in the 0, 0, 0.

One important aspect of modeling was to keep balance between the object count and element size. If I have created just a few objects, I would have needed fewer lightmaps. But those maps should have been pretty high resolution in order to achieve decent lighting quality. Vice versa: having lots of objects would have required much more lightmaps, although with lower resolution. But the amount of maps would have killed a mid range videocard anyway.

I used LODs for the majority of the furniture and accessories. In most cases it is possible to reduce the poly count by even 90% for the LOD objects. Using low-poly elements is key when it comes to optimizing your scene.

*Update: you might want to check out the TS Tool plugin by Tom Shannon as it is possible to export instances from 3ds Max. Fabrice mentioned this tool in his webinar (I will defo test this workflow out in the near future.)*
TEXTURING

In order to achieve a smooth user experience, I had to optimize all the textures for game engines. I did the following:

1. Each dimension had the power of 2 and the majority of the textures were square. This was essential as any other dimensions would have required more extensive GPU calculation.

2. I barely used bigger textures than 512px (for Base Color, Normal, etc). I used 1024px maps but only a few. Sometimes you just don’t want to make any compromise between quality and a slightly better performance.

3. I tested the final package a couple times to check the videocard’s RAM consumption. Mid range videocards tend to have 1-2GB of RAM but I wanted to make an effort to keep it below 1GB anyway (I used the TechPowerUp’s GPU-Z for that).

4. Every texture was saved into JPG with compression applied. Some loss in quality is okay as long as it’s not visible.
CHAPTER THREE: EXPORTING

UNWRAPPING

Before unwrapping, I renamed all the objects. It was necessary to keep things organized in the Editor. Object names were derived from layer names with suffix. For example, Ceiling_001 or Floor_015.

For unwrapping I used the Unwrap Only function in the Render To Texture panel. This way I could add Unwrap UVW modifier onto every object with just one click. But that wasn’t all. You should know that 3ds Max doesn’t make a perfect job in unwrapping. So I needed to find a solution to enhance it.

A plugin called PolyUnwrapper pretty much did the job. I could optimize unwrapping even by 2-300%. The downside of this plugin is that you have to do it manually which can be a pretty tedious process, especially if you have hundreds of objects in your scene.

EXPORTING TO FBX

The content of each layer was exported into a different fbx. File names were derived from layer names. For example, all structural elements were saved into STR.fbx.

This way every object name got a prefix in the Editor that was derived from the filename. For instance, Ceiling_001 became STR_Ceiling_001 after importing it into the Editor.

The reason for this method was to keep things organized and easy to track. I know it differs from what Epic is suggesting (using SM as a suffix) but this approach worked for me much better.

It was also been tested for a couple times and it’s been pretty handy so far.
CHAPTER FOUR: EDITOR

MANAGING RESOURCES

Keeping things organized is key when you are working with complex projects in the Editor. Although Epic made it quite easy to use the Content Browser properly, sometimes it can be a challenge to keep your resources tidy, especially in case of strict deadlines or lots of modifications.

That’s why I have developed my own method for managing stuff in the Content Browser.

I used three main folders:

1. **IMPORT**: I created 4 sub-folders as well. Every fbx was imported to the corresponding sub-folder.

2. **MATERIALS**: I put only materials (and newly added textures) here that were created in the Editor. This way I was only a click away from accessing all the materials and maps that I used for this project. You'll import lots of materials and textures with the fbx that you don't need - just leave where you exported them.

3. **SCENES**: I saved the umaps into this folder.

This transparent hierarchy helped me a lot to speed up the production process. Not to mention that using a simple and clean structure within your source file makes it easier for others to take the project over if needed.

When I was arraying objects (furniture and accessories) in the Editor, I made sure to keep things organized in the World Outliner as well. After a couple hundred objects it might be hard to select certain assets. That’s why I used lots of folders here as well.

LIGHTS

I applied a method developed by koola to set up static lights for this scene. There were four types of illumination that I put into the scene:

1. **Directional Light**: I used the default light source to emulate the sunlight.

2. **Global Illumination #1**: I used the default Sky Light object for that. I adjusted the tone to become a bit cooler to emulate sky light.

3. **Global Illumination #2**: I used planes next to the exterior windows that were lit by Spot Lights.

4. **Artificial lights**: I used Spot Lights and I gave them a yellowish and warmer tone.

(+1) **Lightmass Portals**: Epic did a pretty good job with this stuff. I put these elements at every exterior opening. Using Lightmass Portals resulted in having much cleaner and better quality lightmaps.

The final scene contains 33 light sources overall.
MATERIALS

Optimizing materials was key to achieve a smooth user experience for walkthrough purposes. I did the following:

(1) I used Material Instances a lot. I created a couple of basic Master Materials based upon their nature: wood, metal, wallpaint, plastic, fabric, etc. After that, I converted almost every node into Parameters so I could further tweak them in another (instanced) material.

(2) I used as few textures as possible. If I needed a different tone for a texture, I used the Constant-3Vector and the Multiply nodes to make the original texture blend with another color. I applied this technique to control reflections and glossiness as well.

I created the majority of the materials with this approach. Keeping these aspects in mind resulted in a LOT smoother user experience.
LIGHTMAPS

Lightmaps might make the difference between a well optimized and a slow scene. So it was pretty much crucial to set up the lightmaps properly. I used the following approach:

(1) I had to deal with a great number of objects. The more complex your scene is, the more it matters if you use large lightmaps or not. My goal here was to get away with as few high resolution lightmaps as possible.

(2) Small accessories were tricky. Despite their little size, certain models contained a lot of details that required relatively huge lightmaps. I just couldn’t get away with lower resolutions because it would have resulted in a poor light quality. That’s one thing you want to keep in mind when dealing with small but high-poly objects.

(3) I constantly made test renders to see if the quality of the lightmaps was good enough. I also tested if the package didn’t kill the videocard and wasn’t too large in terms of filesize. It was a trial and error process. (Tip: you should have test renders during the evenings when no one is working on the PCs.)

For instanced objects and LODs I set the lightmap sizes in the Static Mesh Editor. For the structural elements I used the Overriden Lightmap Resolution option in the Details tab. Lightmap sizes varied between 256 and 1024.

BUILDING LIGHTS

I used Swarm Coordinator and Swarm Agent to do the rendering process. It took 13 hours on two PCs equipped with Intel Core i7-4790K processors.

It’s always a good idea to restart the Editor and close all the programs that are heavy on memory usage (3ds Max, Photoshop, etc.) before kicking off the final render. The more time you leave the Editor open, the more RAM it consumes.

The rendering process might eat up 15GB of RAM so you don’t want to risk crashing it, especially in case of strict deadlines.

POST PROCESSING

Adding finishing touches was my favorite part of the whole workflow. I tweaked the visual just like I do for still images (broadly speaking).

Basically, I did the following:

(1) I adjusted the overall brightness and contrast. The final result became much brighter and dynamic compared to the unprocessed one.

(2) I applied a Look Up Table to get an appealing color grading. Using LUTs gives the final result a more cinematic feel. I tend to use them a lot.

(3) I played with the saturation and the color balance. Adjusting colors and tones are completely up to someone’s artistic taste. If you have your own preferences that you tend to use for still images, it might be a good idea to try them out during post processing in Unreal.
(4) I left Auto Exposure intact. I pretty much know that some 3d artists like to reduce the dynamics of the exposure but the default settings just did the job for me.

(5) I cranked up reflections’ quality. I set the Intensity and Quality values to 100 under the Screen Space Reflections option.

That was it. After post processing I finished the project by creating the packages.
THE END

Thanks for taking the time to read this case study over. I hope it was helpful and you had some insights that you can apply for your future projects.

I plan to write more articles on Unreal, so in case you haven’t checked it out, just visit my blog:

ANDRASRONAI.COM

In the meantime, if you have any questions, just hit me up at andras@andrasronai.com and I will get back you.

Thanks, and keep it Unreal!

ARE YOU AN ARCHVIZ STUDIO OWNER OR DIRECTOR WHO NEEDS A HELPING HAND WITH PROJECTS IN UNREAL?

Although Unreal is getting more and more popular for archviz purposes, it’s still hard to find reliable 3d artists who are able to create high-end visual quality content and understand archviz.

As a former studio owner, my experience is that having such people around might give you a peace of mind. When it’s time for their help, you just email them or give them a call.

So If you need a 3d artist to help you with creating your Unreal projects, you might want to check it out how I can do that for you:

CLICK HERE FOR THE DETAILS!

Anyway, let me know if you have any questions. Just drop me a message at andras@andrasronai.com and I will get back to you.

Thanks!

Regards:

--
Andras Ronai
www.andrasronai.com