“The agamemnon”

Retrospective

Module 1

Instructor: Chris Avigni

ESCAPE STUDIOS
Pearson College London

Index

Summary

This written part of the assignment was made with the intention of being part of the research on obtaining a proper game art workflow, that could be used in future projects and also serve as a basis for any further research/ practice on the subject in focus.

In this document, we will be walking through the overall process for the creation of the module 1 assignment titled “Agamemnon” with a slight focus on the parts of the process that were the most crucial to achieve that goal. It is the purpose of this document, not only to serve as a formal written assignment, but also as a workflow document that could serve as a reference for any future personal projects.

Chapter 1- Art direction (week 1)

Art Bible and Art Direction:

To start the visual research stage, a broad synopsis of the environment, setting of the place , date and context of the level, was conceived.

From there it was a matter of imagining a specific art style in order to begin creating a moodboard. The art style chosen was something reminiscent of “Halo” but with a slightly more vibrant, yet simplified, color palette.

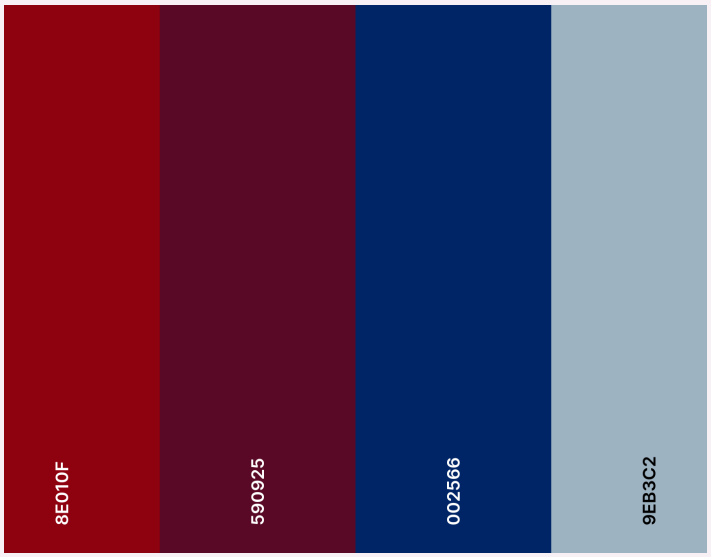


Figure 1 - (The chosen color palette) - The color palette was one of the first things researched, along with the lighting in order to establish the mood.

After that it was time for the creation of the moodboard and to start looking for specific references for the environment and its props. research tools included “Pinterest”, “Artstation” and multiple artbooks.

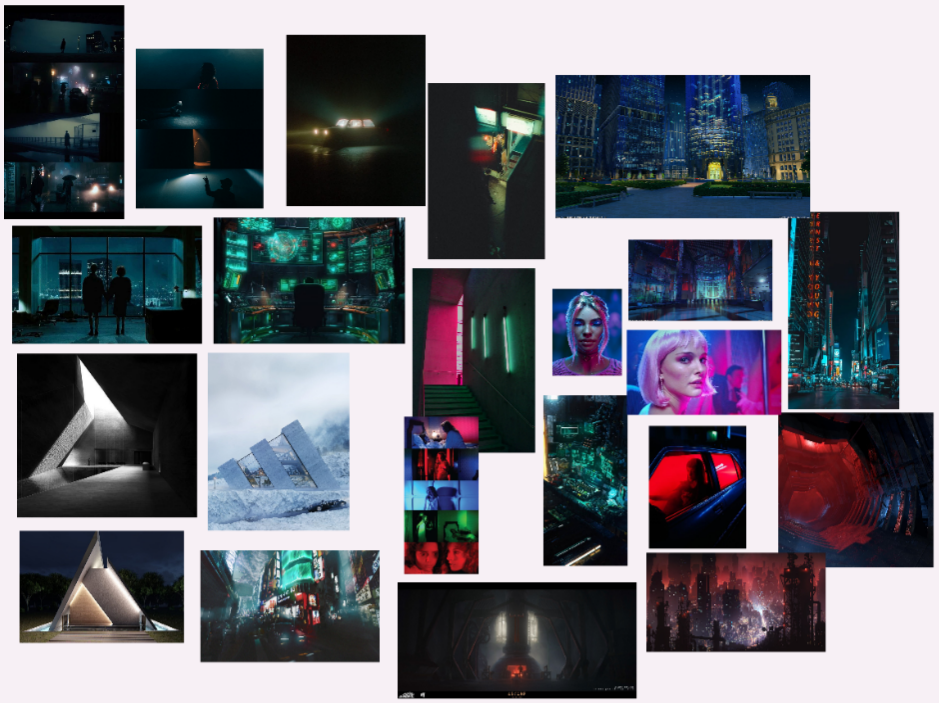


Figure 2 - A few examples of the initial moodboard

It was important to keep in mind that none of the references were to be directly copied so as to avoid ending up with a plethora of different styles and thus breaking the artistic cohesion.

To aid this process a simple “blueprint” of the whole level was drawn in photoshop while the moodboard was being filled with a variety of different ideas.

The final part of this process involved establishing the level by blocking it out in Maya and importing into Unreal in order to check the overall proportions and scale. This process would be repeated throughout the modeling of the props.

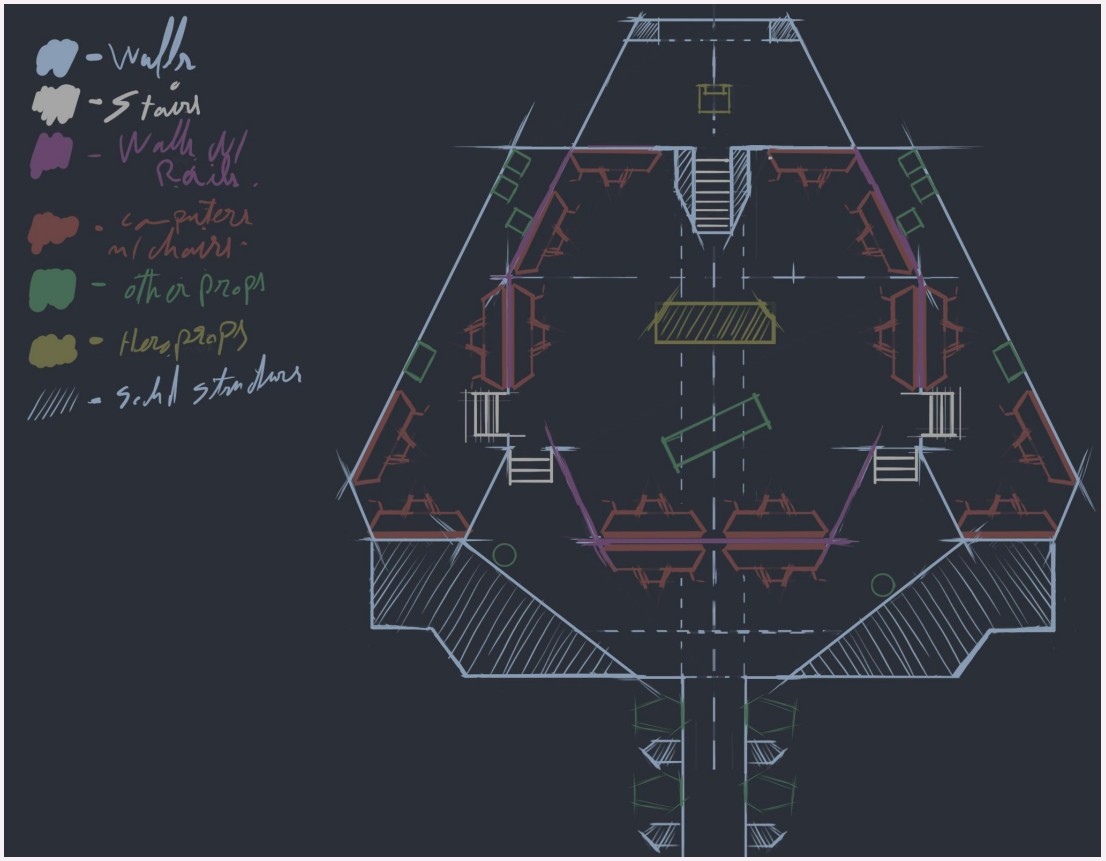


Figure 3 - the Photoshop drawing of the level.

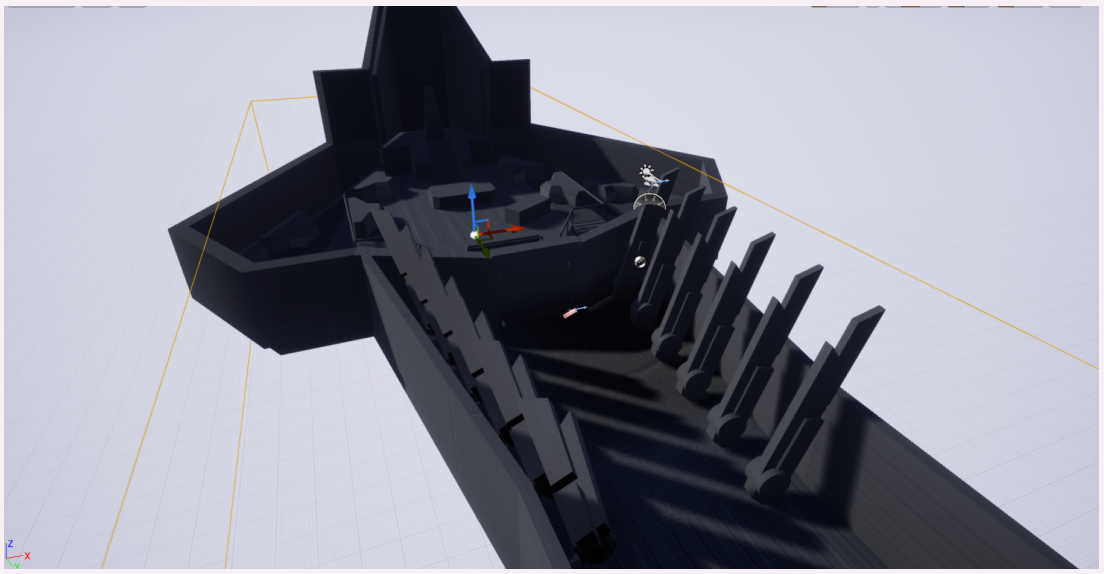


Figure 4 - The initial blockout in Unreal.

Chapter 2 - Modeling and Modularity (weeks 2/3)

Workflow for pipeline optimization

The goal for this project was always to obtain a personal workflow that would allow for the creation process to be as less destructive as possible. So every stage, ranging from the concept modeling to the final modeling and texturing, could be changed on the fly and updated inside the engine as fast as possible. This would allow a smooth iteration between software and to make artistic changes quickly which was absolutely imperative as we're dealing with a sci-fi environment that requires a lot of back and forth when it comes to implementing new ideas. So it was imperative to have a way to conceptualize the level and props and get the concept into the engine quickly to check if any further changes needed to be made. **(Pavlovich, 2016)**

Production modeling (Process)

It was decided to use the references from the Art Bible to quickly conceptualize the props in order to get a certain degree of unity between all of the assets when it comes to their art direction.

To do this, a couple of features were used in Zbrush. A “block out” of the models was imported in to the software and, with a combination of a few IMM Brushes from the XMD toolbox (which I have a perpetual personal license) and some tweaking of the meshes in the IMM Brush using Zmodeler, the concept was quickly kitbashed, changing the overall shape and design on the go. **(Pavlovich, 2021)**

After the basic shapes were blocked out, the tech pattern texture was concepted. This would eventually be used down the pipeline in the texturing stage as a unifying element of all the props. For this It was used “Jsplacement” to tweak and generate a grayscale pattern of panels and ran it through a surface noise layer with some quick UVs generated from the polygroups. **(Pixologic, 2021)**

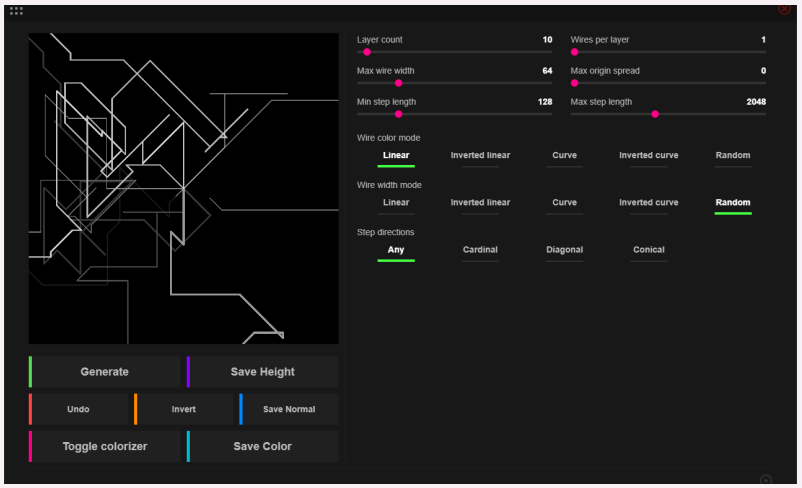


Figure 5 - A pattern generated in Jsplacement

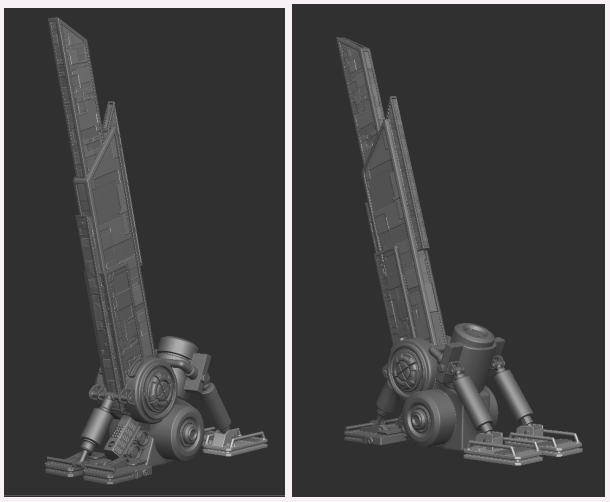


Figure 6 and 7 - An example of one of the prop concepts in Zbrush with surface noise applied

This non-destructive process allowed to check if the idea actually worked before committing to it, since changing that idea later on, in the texturing phase, for example, would require a lot more effort, so this saved a lot of precious time troubleshooting down the line. And, since It was already being sketched in 3D it was fairly easy to extract a low poly mesh from the sketch to then proceed to do the final modeling in Maya. **(Pavlovich, 2015)**

Minor patterns, such as the surface noise pattern and small bolts were not preserved for the low poly mesh since they would eventually only be present as a normal map detail, created using substance painter) .

Using this method allowed, not only to get rough ideas right into 3D but also to create a low poly mesh just to see how it all worked out in Unreal, in order to check how much detail was needed for each piece before actually finalizing the modeling stage in Maya. **(Pavlovich, 2016)**

For this it was a matter of merging all the subtools, running it through “Decimation master”, importing the mesh into Unreal with the correct naming conventions so it could be easily replaced with newer versions, and everytime a model was updated in Maya It would update all the actors on the level itself. This allows for a much faster and streamlined iteration between Maya and Unreal.

The next step was to optimize the meshes for the game taking into account the poly budget for each piece .

Modular kit and UV mapping breakdown:

The process of creating them modular kit used the same principle as when creating the props in Zbrush but this time everything was done entirely in Maya since It was imperative that the geometry placed would keep a proportional consistency to the grid so the UVs could be planned out in order to texture using Trim sheets. **(Finch, 2013)**

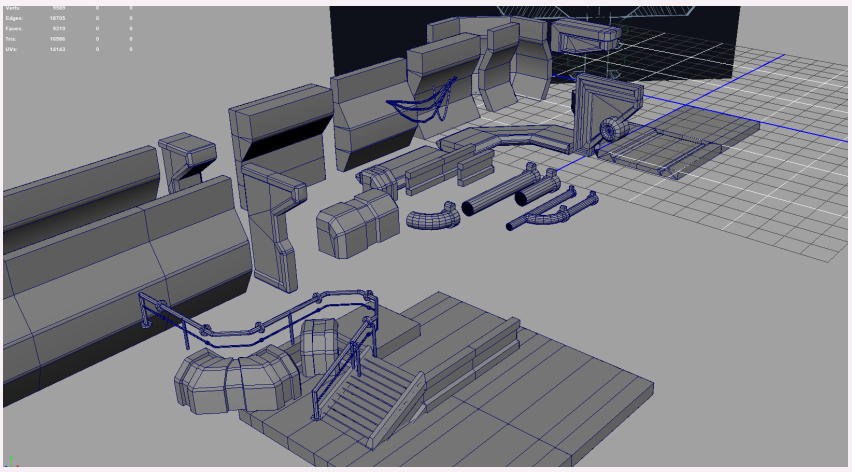
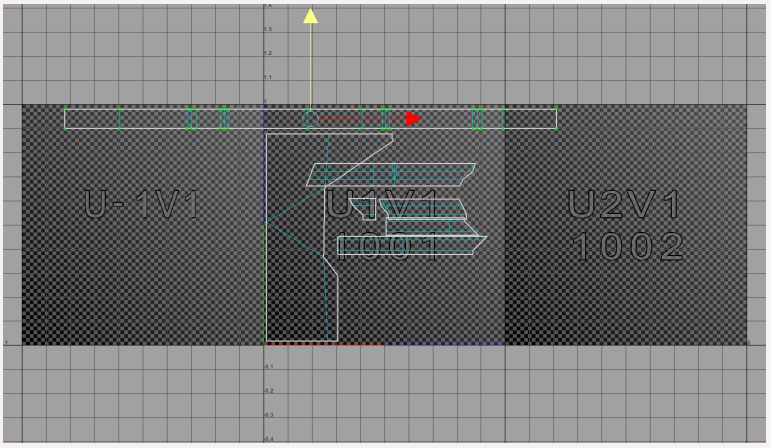


Figure 8 - first draft of the modular kit

First thing to do was to start thinking about where the trim sheets would be reused all over the environment (Including parts of the props on the scene). Any other object or part of an object would be textured using material IDs in Painter. This also provided the correct material IDs inside Unreal as soon as these models were reimported into the engine. **(Finch, 2013)**

Three trim sheets were planned initially . Two of them for the paneling/ vents and another for any tubular shapes on the scene (such as pipes, cables, etc…)



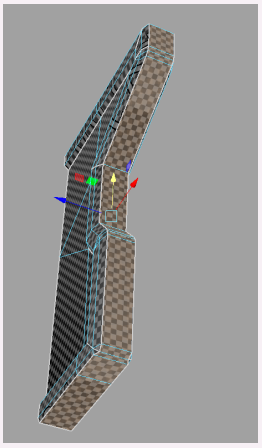
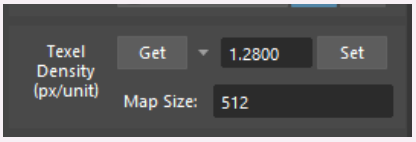


Figure 9 and 10 - Setting material ID and UVs for parts of a modular kit model for the trim sheets.



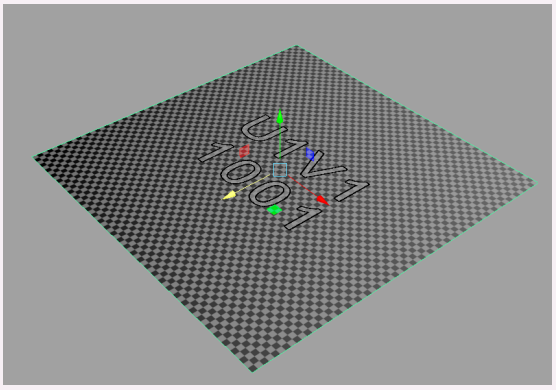


Figure 11 and 12 - setting the texel density in Maya for 1k texture per 400 units

Keeping that in mind, any major changes made to the modular kit (particularly the wall and floors) would have to be consistent to the grid. As an example, If 2 wall shapes were required, at some point, the necessary variation would be done entirely by switching the texture or creating a copy of the same mesh with a different UV placement.

Chapter 3 - Materials and lighting (weeks 4/5)

It was planned out from the beginning to economize the amount of textures as much as possible. for this it was decided to actually use parts of the modular kit trim sheets to texture parts of the props so as to maximize the amount of resolution used for the actual “hand texturing of the props” **(Ahern, 2016)**

Texturing trim sheets and tileable textures

The concepting of the patterns to be used for the trim sheets and tileable textures was made using zbrush.

For the trim sheets I used a simple plane subdivided 8 times to get the correct texture space division that would be needed. the plane would be used to boolean out the mesh in order to only include the patterns within a square shape.

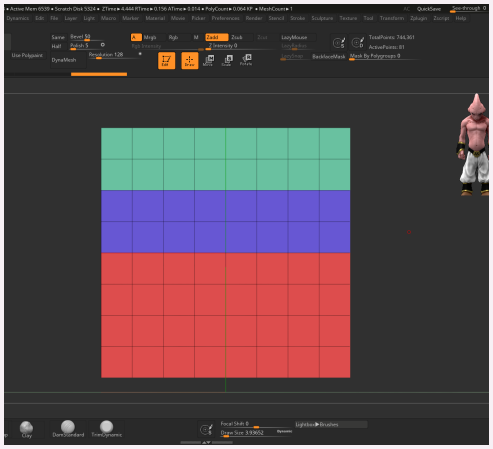


Figure 13 - The ratio plane with polygroups

Polygroups with Live booleans were also used to separate the parts that would include each section of the trimsheet keeping it to the texture ratio.

A plethora of other features were used to actually create the patterns, including live booleans, custom made IMM brushes, Zmodeler, arraymesh and Nanomesh.

After that it was just a matter of turning on “live booleans” and add the plane subtool as an intersection boolean, so anything that wasn’t contained within those extruded faces would be cut out, keeping that particular boolean group to the correct proportion to the texture and still tiling to the left and right.

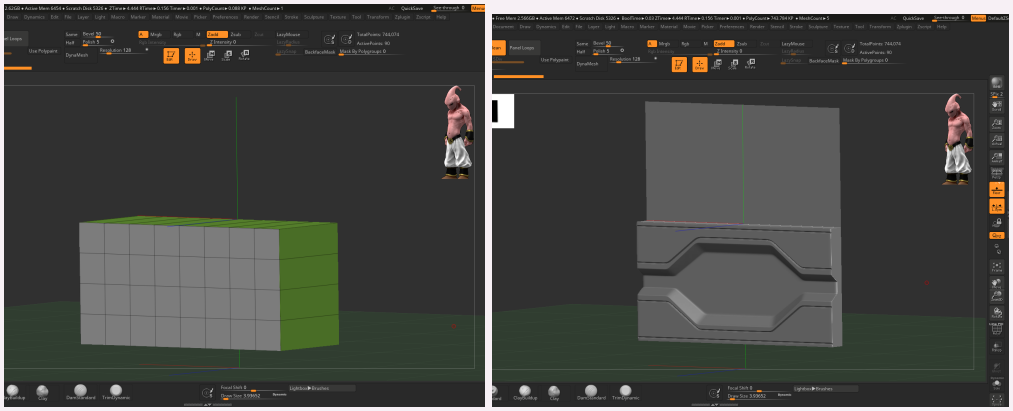


Figure 14 & 15 - An example of the live boolean workflow being used to keep the trim sheet within the ratio desired.

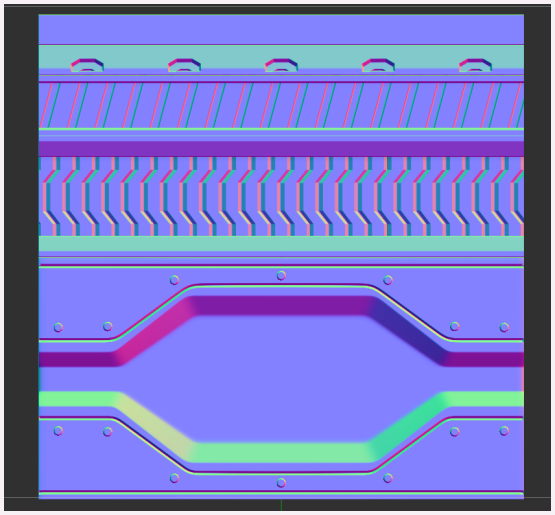


Figure 16 - The final trim sheet with a normal RGB material applied.

The design of the repeating patterns was all made using a simple plane in with subdivisions using morph targets to keep the flexibility of erasing parts of the design as needed, using wrap mode to keep the detailing tileable, turning on “thick skin” to cap off the chisel brush so as to create the illusion of actual paneling and using Array mesh to create instances of the plane in order to preview how the pattern would tile on the level

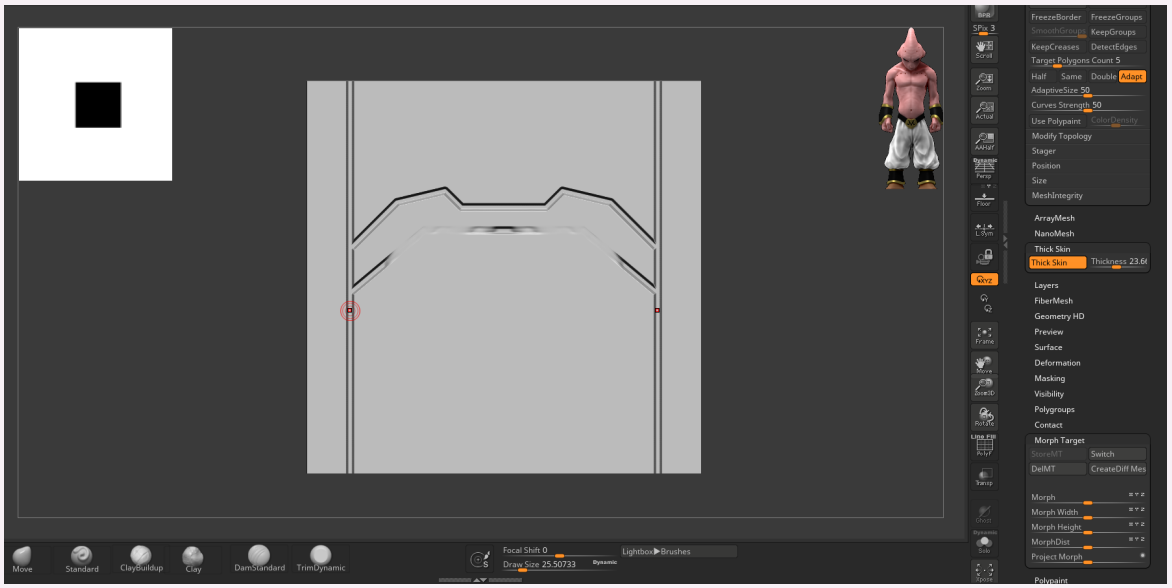


Figure 17 - application of morph targets and erasing with morph brush

Alongside this It was important to use layers to create different patterns for the “high frequency”, placing different patterns generated in JSplacement as surface noise and applying them to the layers.

It is important to note that the alphas that were generated in JSPlacement for this effect were the same ones that were used in the concept stage so it could be reused for the detailing of the normal maps in order to visually unify the different props throughout the level.

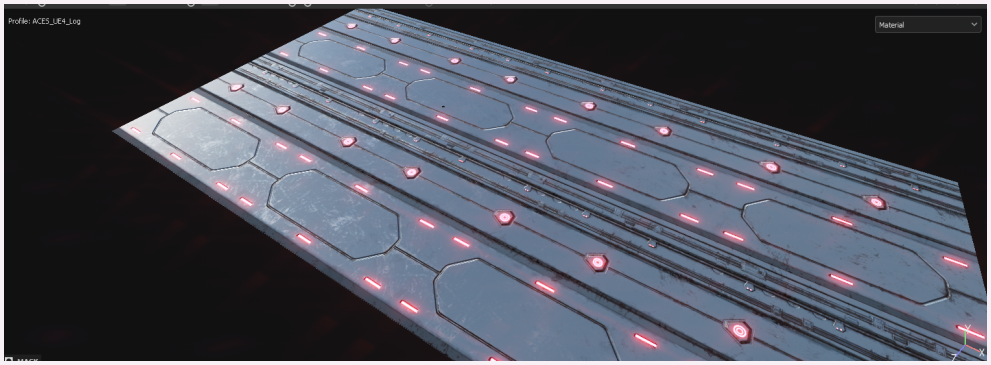


Figure 18 - texturing the trim sheet in substance painter

This process can be repeated as many times as needed in a non destructive way as Zbrush stores the layers, morph targets, And thickskin in the Zproject. So If a new design for the tiling textures is needed later on, it is just a matter of going back to that plane and redoing the sculpting part of the process using the morph targets.

Texturing props

To texture the props It was first considered which parts of the props would be reusing the trimsheets. This was done to allow for higher resolution on the parts that were manually textured in substance. So naming conventions for both the props and textures, creating material IDs in Maya and then exporting the props to Unreal in order to start creating the material networks in the engine kept the pipeline connected as then any change to the texturing of modeling would only be a matter of reimporting that particular asset and it would be replaced throughout the level. **(Ahern, 2016)**

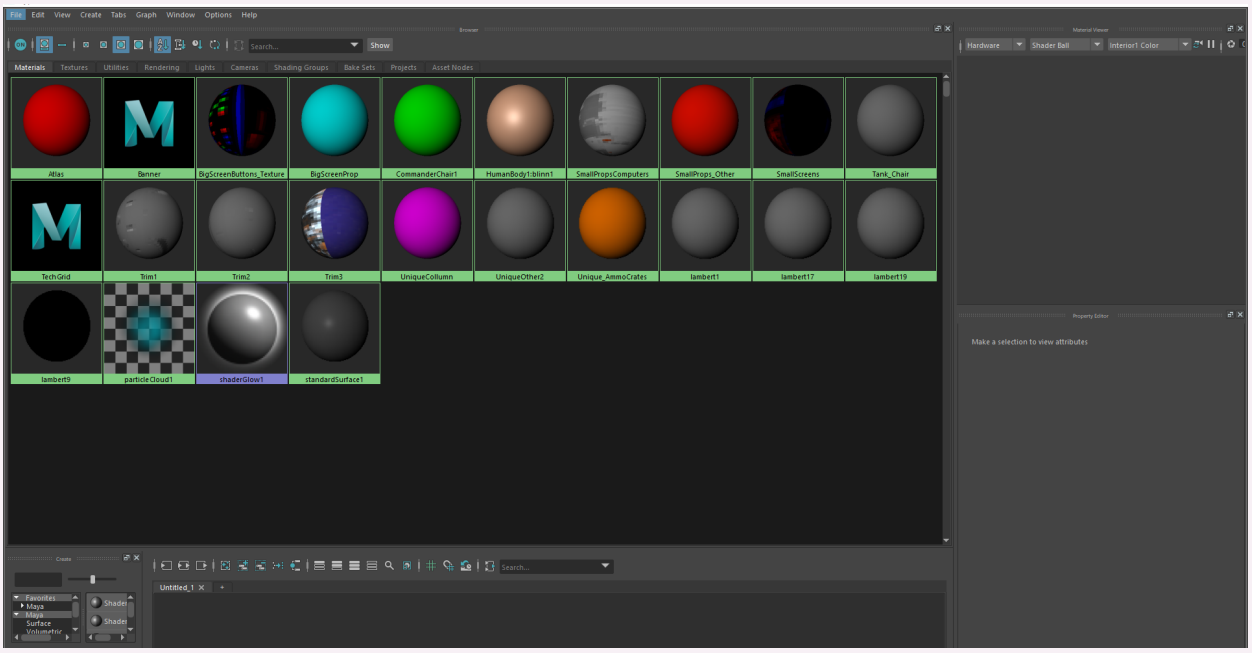


Figure 19 - the material IDs in Maya

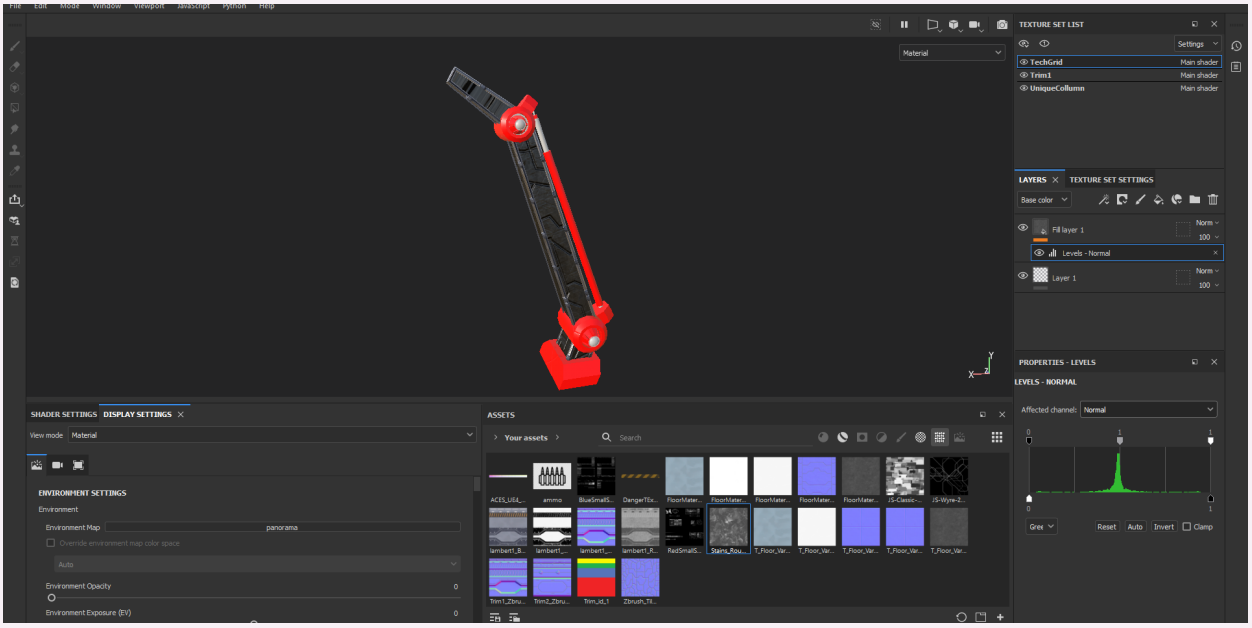
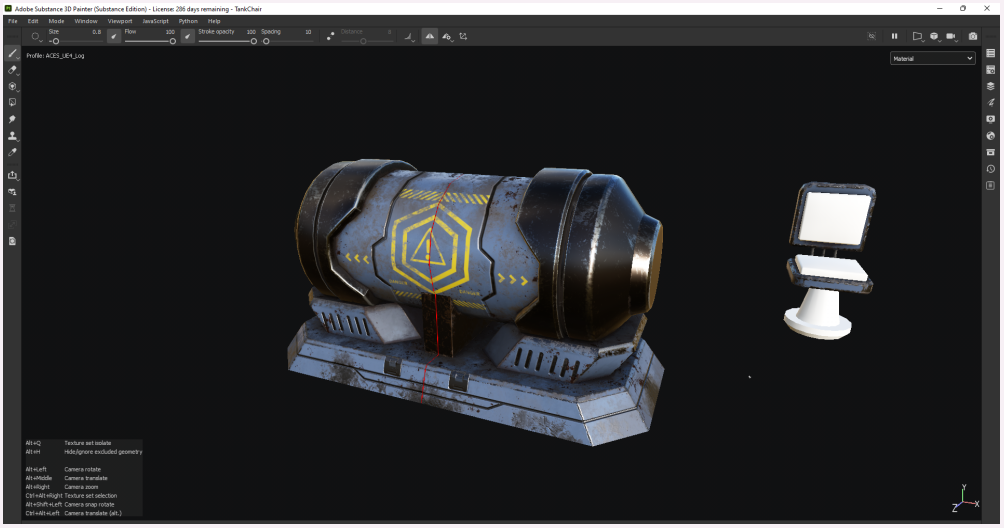


Figure 20 - the material IDs separate into different texture sets in Painter which allows to texture only the parts that have a unique texture applied in the level

When a certain prop did not fill up the entire UV it would share the same ID as another prop so to take advantage of the full resolution of that particular texture.

Any extra detail that was not modeled due to poligonal constraints would be present by painting height maps directly in substance painter.

Figure 21 - examples of textured props that share the same texture space



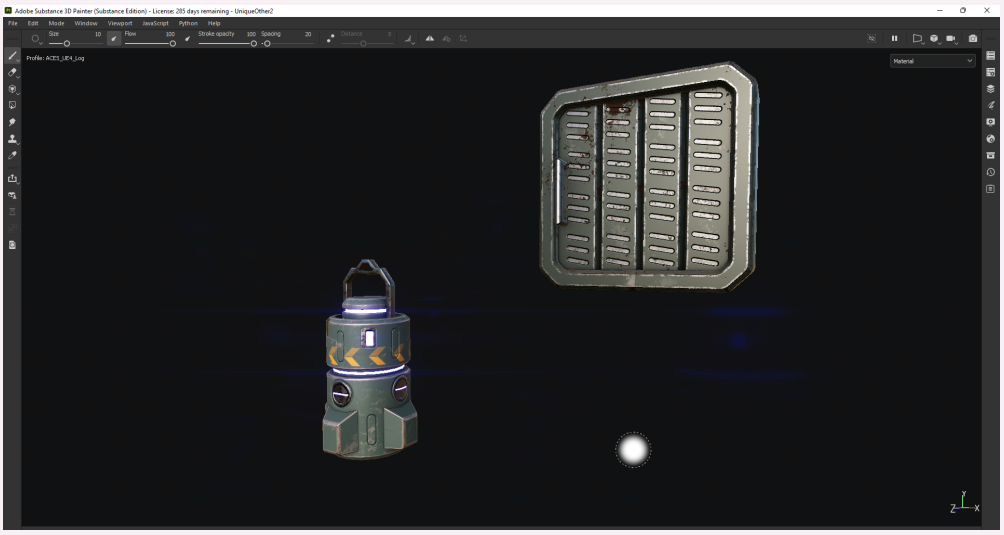


Figure 22 and 23 - examples of textured props that share the same texture space

Besides this, the idea here was to use as few master materials as possible. In the final project there is one material per trim sheet, one material for all the uniquely textured parts of props and materials for the screens that require some extra control over the emissive lights (like changing colors for buttons and controlling the intensity).

Every asset inside Unreal already had the specific material instance assigned to its material Id, which meant that any changes made to either the texturing or the material itself would be updated in the level as long as the naming conventions were kept.

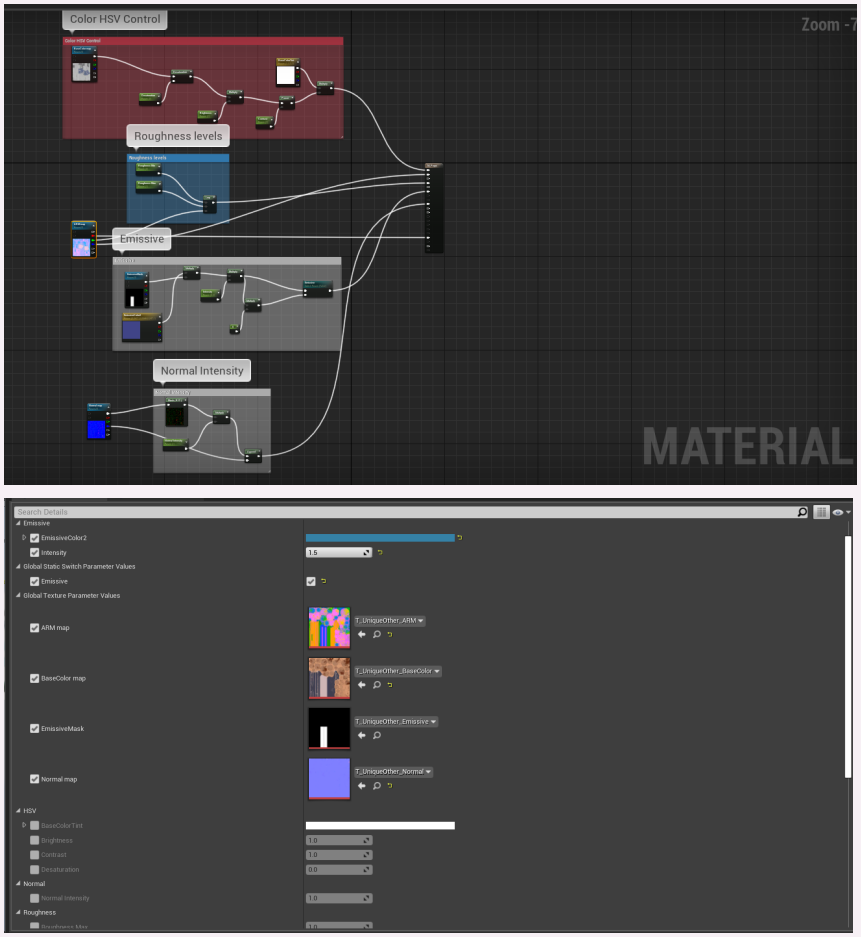


Figure 24 and 25 - example of the prop master material(top) and one of its instances(bottom)

Lighting

The Lighting was pretty straight forward when it came to its overall design. The idea was to have the initial area of the level lit in a more cold color and have that contrasted with the main portion of the level where the warmer tones predominated.

Figure 26 - Initial test of the lighting

Figure 27 - The main light in this example is cold and has a higher intensity, while the "complimentary" light is warm and has a lower intensity

The main lights would cast shadow in order to create high contrast areas to steer the eyes of the player there. All the other light would be composed of a "main" light with the predominant color of that area of the level and then filling in some general tones in the "opposite color".

For example, if the main light in a certain portion is cool with an intensity of 7 then it would have warm light with a lower intensity to fill the parts where the main light wouldn't reach.

Figure 28 and 29 - Final lighting assembly examples

Miscellaneous (Week 6)

Blueprints and extra adjustments

At this stage all that was left to do was create the blueprints for the mechanical door, populate the environment even more (set dressing) and make final adjustments to the lighting.

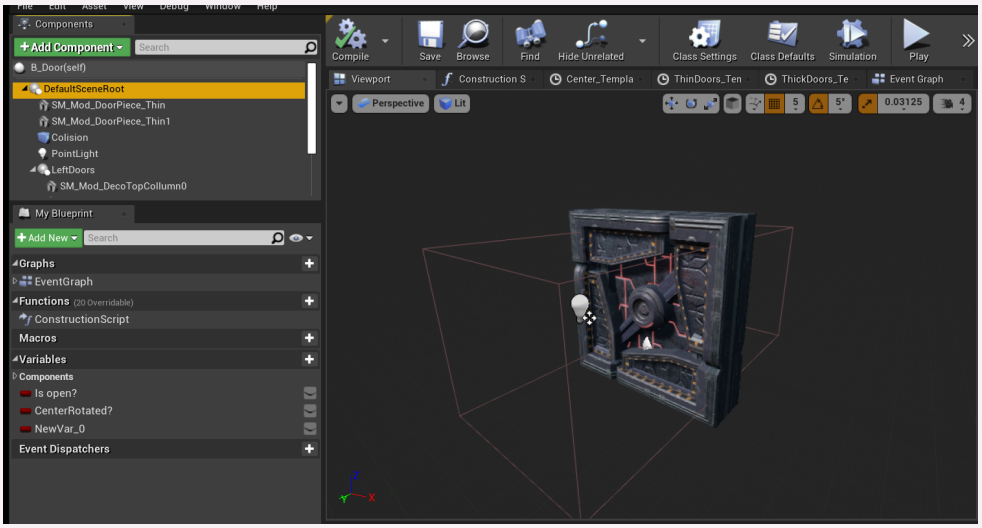


Figure 30 - Assembly of the door in the viewport

For the blueprints it was a matter of bringing in the meshes that composed the door (3 meshes from the modular kit) and assembling them inside the blueprint viewport.

After some testing of how much they should move in order to be fully opened, it was only a question of creating the timeline for each piece and then animating them to look a little more credible using the timelines curves.

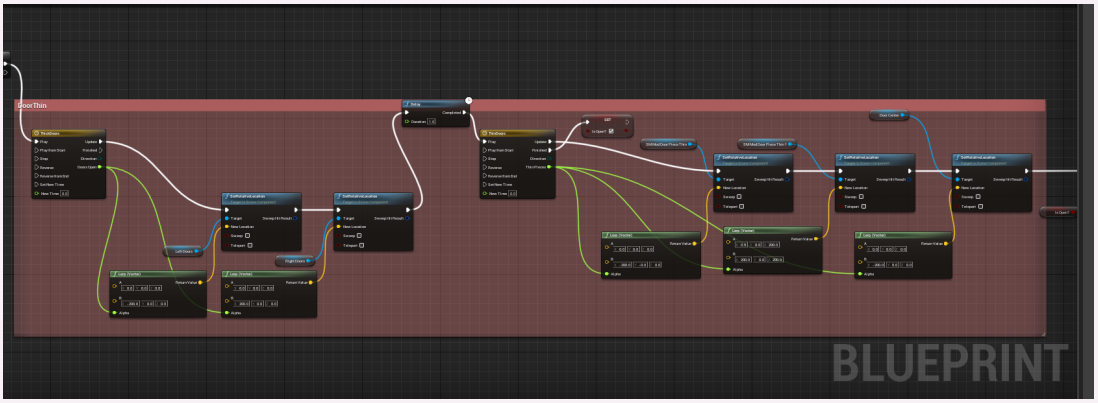


Figure 31 - Blueprints “code” for the door opening animation

The rest of the final days in the process were spent making decals for the documents on top of a previously modeled table, making tweaks to the lighting, textures, materials or making new models to populate any areas that felt too empty.

There were also some issues with the lightmaps that were taking its toll on the overall quality of the lighting which required creating custom UVs for the lightmaps. **(Brady, 2022)**

Figure 32 - The decals for the documents on top of another asset

Conclusion (retrospective)

Overall, and as a learning experience I believe this was a successful project. A total of around 45 models were made, including the modular kit, which meant that the whole purpose of learning a streamlined and non destructive workflow was imperative. In that regard I think that the project was a success. But when it comes to the sheer amount of assets to model and the actual planning of the level it felt too ambitious for a 6 week project. There are a lot of mistakes that will probably not be repeated in the next project, which includes the scale and scope of the level being too big and less than ideal dedication to the lighting.

Bibliography

Youtube:

Michael Pavlovich channel, (2016) *Speed modeling and texturing playlist*. Available at: https://www.youtube.com/playlist?list=PLkzopwqcFevYpcinb91BKc22Kv3-g8pHL (Accessed February 2022)

Pixologic Zbrush channel, (2021) *id Software - A Look Into DOOM Eternal - 2021 ZBrush Summit.* Available at: https://www.youtube.com/watch?v=ujJhjxDXNEk&t=2s

GDC Channel, (2015) *Halo 2 Anniversary Blurring Concept & Production Art,* Available at: https://www.youtube.com/watch?v=072HkIJ_DMs (Accessed: February 2022)

Books and magazines:

Brady, J., (2022) *Optimize game assets for unreal engine, 3D world magazine,* issue 283, p.56 - 61

Finch, A., (2013) *The unreal game engine: a comprehensive guide to creating playable levels.,* United Kingdom, 3DTotal Publishing.

Ahern, L., (2016) *3D game textures: Create Professional Game Art Using Photoshop*, CRC Press

Articles and presentations:

Pavlovich, M., (2015) *GDC 2015: Next Gen Pipelines Presentation slides and documentation* Available at: https://www.slideshare.net/MichaelPavlovich/gdc15-michael-pavlovichupload (Accessed: February 2022)

Pavlovich, M., (2021) *ZBrush Hard Surface - Linear Walkthrough PDF*