

The Character Production Pipeline Project:

Analysing and improving a production pipeline according to the company's needs and resources.

By Aioanei Iolanda, 452342

June 15, 2021

Company: VSTEP Simulations

Company supervisor: Camille Pauw

Saxion supervisor: Taco van Loon

Abstract

This paper concerns the character creation workflow at *VSTEP*, a company which specializes in making 3D simulators for emergencies and nautical training. The character pipeline, while productive, needs to be documented and modified when it comes to certain steps. These improvements have the goal of expanding the current character library with characters that have specific characteristics, such as ethnicity or uniform, in an efficient manner and making use of the resources available or added to the pipeline.

During this project, research has been done into what the industry pipeline standards are at the moment and what tools can be used to further automate the creation process. Taking into consideration what the company needs are when it comes to characters, the results of the research have been weighed in to decide which tool would suit the company best. Following this, a new pipeline has been established and documented for future use at *VSTEP*.

Table of Contents

Abstract.....	2
1. Introduction.....	4
1.1 Company Outline	4
1.2 Problem setting	4
2. Problem Definition.....	5
2.1 Scope.....	6
2.2 Research questions.....	6
3. Theory	7
4. Methodology	9
4.1 The Current Library	9
4.2 RS needs.....	10
4.3 The current pipeline	11
4.4 Resources and Tools for Improvement	12
5. Investigating the tools	14
6. Results.....	21
6.1 Determining the solution.....	22
6.2 Reorganizing the folder structures and templates	23
6.3 Documenting the new pipeline	26
7. Conclusions and Discussion.....	27
8. Recommendations.....	27
9. References.....	28
10. Appendix.....	29

1. Introduction

3D character production is one of the most complex tasks when it comes to game and software development. The many steps and software involved require a certain experience in the 3D modelling field, as well as the artistic skill to be able to sculpt credible features, whether they are realistic or stylized. These factors also give the process a significant time cost, which is why establishing and developing a pipeline helps consistently create higher numbers of characters.

An organized and documented creation pipeline helps better understand the separate stages of production. By having these stages laid out in a clear fashion, they can be analysed and modified to include more automation, which would then lead to a more efficient pipeline. With the industry constantly evolving, companies have the chance to better streamline their production as long as the proper research is dedicated to what can and cannot be changed. This is the focus of this project.

1.1 Company Outline

VSTEP is a company creating simulators for training and safety purposes. Through their software, users can manipulate 3D environments and interactable assets to create a wide variety of emergency situations or training settings.

Currently there are two ongoing projects: *Nautis* and *RS – Response Simulator*. *Nautis* is a naval training platform, complete with physical hardware which simulates the control panels of a ship. *RS* is a virtual emergency simulator with which personnel can be trained and evaluated for conduct in a situation of emergency. Both projects have national and international clients, with the company offering services such as creating environments and assets tailored to specific countries or corporations.

Characters are a very important part of the *RS* project. As a real-world simulation, it is important for characters to look as close to reality as possible. They also require a long list of features for properly contribution to the scenarios a client would intend to simulate.

1.2 Problem setting

At the moment, the company has a library of fully operational characters integrated in the application. There are, however, additions that are necessary to improve the quality of *RS*, according to the clients of the software. Because additions by means of features or new library items are time costly, it is the company's wish to dig into how the current pipeline handles the character production.

Until now, the character creation tasks have been the responsibility of two members of the content creation team. With their recent departure from the company, the remaining content team has been left with no experience on the character creation topic. They are also disadvantaged by the fact that the pipeline has not been documented in a way that would help them understand the process.

Investigating the pipeline and the influencing factors involved – such as the source file library, the resources available for improvement and the character needs of the application – will help shape a refreshed and upgraded pipeline, better accessible by the content team and other future team additions.

This project aims to analyse the current character creation workflow and the character database of the company. The goal is to research, document, and test more efficient ways of creating characters, with

distinct features (body type, ethnicity, injuries). The needs of the company, which the project's research range will cover, stand as following:

- An extended character database – the client needs more characters for scenario use, with more diverse features, body types and specific uniforms.
- A documented workflow – the company needs a documented pipeline to ensure that even as the content team changes, character creation tasks can be fulfilled in an efficient and satisfying manner.
- Database analysis – the source files database can be a good resource pool for reusable assets and for understanding the workflow of the previous character artists, but in order to assess this, it needs to be thoroughly investigated and organized.
- A more efficient way of generating characters – currently the characters are all created by hand, but the company is interested in finding more efficient ways of generating characters.

With the characters created through the improved workflow, the application would be able to offer a wider array of possibilities for simulating scenarios. Some of the detail is crucial in some cases, such as portraying characters with injuries on the site of an accident, or personnel with uniforms specific to the country they reside in.

The assets generated through the pipeline will ultimately be a feature for the users of the application. While the company has an interest in developing the pipeline, the clients of the company will be interested in the concrete results. They will be able to ask for specific characters which can also be created at a faster pace.

2. Problem Definition

As mentioned before, the client's product software, *RS*, focuses on giving the user the means and assets to create scenarios as close to reality as possible. From a character perspective, this means both personnel and civilians in various states (different equipment, status, injuries). Currently, the application provides around 55 different character models, with most of them being emergency personnel and about 20% being civilians.

The characters are created to resemble reality as close as possible through their features and clothes. Most of them have the same body type and some variation in terms of features. There are also little options for making the characters visibly injured. Because the application focuses on training personnel, the characters representing this category have a higher priority when it comes to creation and detail compared to the civilians. They are also the most time consuming to create of the two – because uniforms differ between countries, each personnel individual needs a uniform modelled to look specific to the style of the country.

The limitations of the project are of different natures, some of them technical, time-bound or related to skill:

- Experience and support

As of the beginning of March, the lead character artist of the company will be leaving his post. This means that, while he will provide all the support and training possible in the timeframe before his departure, in the future, the assistance of his experience will be missing. As my own experience in this field is very little, this means the process might be hindered at times when it comes to troubleshooting issues.

- Timeframe

Because the project has an established timeframe, the needs of the client must be prioritized accordingly. While researching and developing the pipeline is more important, generating new library items may be a secondary and not mandatory goal.

- Development branch

The engine and platform on which the application is made is mostly controlled by the development team of the company, as opposed to the content team. This implies that while the workflow of character creation can be adjusted, the end result must still meet the requirements of implementation in engine, and modifications to parameters and options in-engine are harder to achieve.

2.1 Scope

During this project, the focus will be on documenting and adjusting the pipeline according to the influencing factors discovered during the research phase. The adjustments will follow both the research and the client's needs – creating realistic characters on a shorter time span, while fulfilling the requirements specific to their category, whether they are civilians or personnel.

The following conditions of satisfaction exist for the resulting products of the internship:

- The character creation pipeline is fully documented in the form of tutorial/guide – this documentation will help new additions to the company adapt to the workflow easily, therefore shortening the timespan of getting accustomed to the working environment
- The folder structure of the current library is organized in an easily understandable and accessible manner – this will ensure that future personnel can easily navigate the library for resources
- Different character creation practices are researched and tested, to decide on which fits the company needs the best – the research will cover multiple aspects and the input of the company will decide which will be invested on and introduced to the new pipeline
- Some new library items are created – these items will serve as experimentations and tests of the pipeline and its adjustments

Creating new features that would enable complex real-time adjustments to the characters is a difficult topic, as it is dependant on the developers and the engine of the application. Therefore, the project will not focus on real-time adjustments in-app much further than what is already possible in terms of engine.

2.2 Research questions

Considering the information and the material gathered, the main question of the project would be:

How can the current character creation pipeline be optimized in order to decrease the time cost of character creation tasks and increase the character library with items tailored to the needs of RS, while making use of the resources available?

In order to fully answer the main question, the following sub questions must be pursued:

1. What is the situation of the current library: items, organization?
2. What are the needs of the Response Simulator application?
3. What does and should the character creation pipeline consist of?
4. How can the pipeline be improved in efficiency time wise?
5. What resources are available to improve the pipeline?

3. Theory

Creating a work pipeline has been a software development practice for many years.

Dane Edward Bettis, a pipeline developer with over 15 years of experience in software development, defines a digital production pipeline as a set of different components that can be rearranged and adapted into new structures when needed. These components refer to people, digital hardware and software systems, and the production pipelines is the synergy of these components resulting into the beforementioned adaptable structures. (Bettis, 2005)

An established workflow ensures that the process is time efficient, easy to learn and makes the best use of the available resources.

There are many tutorials and guides available online when it comes to the process of character creation. They usually focus on a single aspect of the process, such as sculpting or retopology, as pipelines vary between companies and individuals, and it is not easy to document the entire process fully. This offers a wide array of options to research as to what can be adjusted or added to the current pipeline.

Because my own experience in creating characters was very little at the beginning of the project, understanding all these steps of the process was a very important part of the research. Multiple sources were investigated on the topic, which can be seen in [Appendix I](#).

The base of the pipeline usually consists of the same steps:

1. Identifying the requirements of the character and gathering references

In the game industry, this is a very important part of the character creation process. Not only for providing the foundation of the character, but because specific requirements can change how the pipeline handles a character. The better the *what*, *why* and *how* of the character are identified, the easier it will be to keep an efficient workflow without interruptions and backtracking for fixes.

Siamak Roshani, a 16-year veteran of the industry with positions such as senior character artist in *Call of Duty* and lead character artist in *Ghost of Tsushima*, talks about how this part of the pipeline influences the process. For example, a hero character will most likely have a much higher polycount and detail compared to a side-character. That would mean more time spent on perfecting the mesh. Every different character type can influence the workflow and the time cost of the task and foreseeing the ways these influences play out can help have a smoother and better equipped pipeline. (*Character Art Pipeline in Video Game Industry Part01*, 2020)

2. Creating the high-poly mesh, sculpting

The high-poly mesh is the character asset at the highest quality possible. The process of creating it usually happens in a digital sculpting software, such as Zbrush. It can take up the most time out of the workflow, as this must be a fully polished, detailed representation of the character.

The sculpting process is complex and often requires moving between different software depending on the part of the character that is being created. As mentioned before, Zbrush is a popular and often opted for software when it comes to organic modelling, which refers to any kind of living organisms. Often times there can also be hard surface modelling required, which focuses on inorganic objects, like equipment or accessories of the character. Then there is software specifically tailored to create particular items - one such example would be Marvellous Designer, a 3d modelling program which realistically simulates cloth and is becoming more widely used for creating clothing base meshes for characters.

Every character artist has their own workflow and practice preferences for this part of the process, which is in a constant change as they gain more experience and as software advances.

3. Creating the low-poly mesh, retopologizing

The low-poly mesh is a version of the high-poly mesh reduced enough in detail to ensure a good functionality and disk space preservation in-app, while also striving to retain as much of the high-poly quality as possible. The process of creating this mesh is highly technical, with some artists striving for automated tools, such as 3dsMax's optimization modifiers, and others preferring to create it manually. The second option is most often chosen when the artist wants to ensure that the topology is of a higher standard.

Topology refers to the flow and distribution of a 3D model's edges, faces and vertices. The way a mesh is structured can affect its rendering, the way a texture wraps around the mesh and how it reflects light. Henning Sanden and Morten Jaeger, 3D modelling artists with over 10 years' experience in the industry and founders of the *FlippedNormals* platform, argue that it is important to analyse what the mesh is intended for before retopologizing it manually, which is a time-consuming process. Ultimately, if the purpose of the mesh is well understood, then the best decision can be made regarding how to manipulate the topology and what standard it needs to rise to. (FlippedNormals, 2018)

4. UV mapping

UV mapping is the process of flattening a 3D model's surface onto a 2D plane, to be able to project textures onto the mesh later on. This is also a highly technical process, but a common one among all pipelines involving any 3D assets. This is also the part of the process where the surfaces can be merged or separated by assigning materials.

When it comes to characters, as well as other 3D assets, UV mapping requires identifying the best options for the seams of the mesh – for a character, the seams should not be left, for example, on the face or visible on the front of the body. The placement and resolution of the UVs on the map can also be a complex process, depending on how many maps the artist wants to stick to.

There are both automated and manual ways to execute the process, depending on how much the artist wants to control the precision of the result.

5. Baking and texturing

At this stage, the character can be textured and brought to life. Most artists' go to software would be Substance Painter, which has many complex features that allow for detail oriented and impressive textures. The software also offers good texture options for skin and clothing, which are very useful for characters. Depending on the style required, the step can be time consuming, especially when it comes to realistic characters.

Baking maps is the process of projecting the details of the high-poly mesh onto the low-poly. This step is essential to retain the quality of the high-poly mesh on the optimized mesh, and is a highly practiced technique in the industry.

6. Rigging, skinning, and weight painting

With the character's mesh completed, making it able to move according to animations is the next part. The process of creating a skeleton that will enact animations is called rigging – the rig being the skeleton of the character. Using rigs can be a tricky process, as for the mesh to follow the animations in a natural way, the rig must perfectly fit with the mesh, and this means that the two of them are co-dependent – for example, they must be the same height and the joints must be positioned perfectly

over each other. So even though humans have the same basic bone formation, there are small details that can influence the rig.

After the rig is completed, the mesh must be bound to the rig in a process that is called skinning. While this is mostly an automated procedure, it is important to check over how each joint influences the mesh – most of the time the automated process needs small adjustments and fixes to make the animations look more natural.

7. Animating and testing

This is the last step of the process, which means taking the character into the engine and testing the results for any issues in the mesh, textures or rigging. From engine to engine, the implementation process can be different and complex in itself.

While these are generally the steps of a character creation pipeline, this process, like any other, is highly adaptable and prone to changes. Each artist's skillset and preferences always influence the workflow. For this reason, an established pipeline should always have room for flexibility and should act as a support for the artist instead of a restriction.

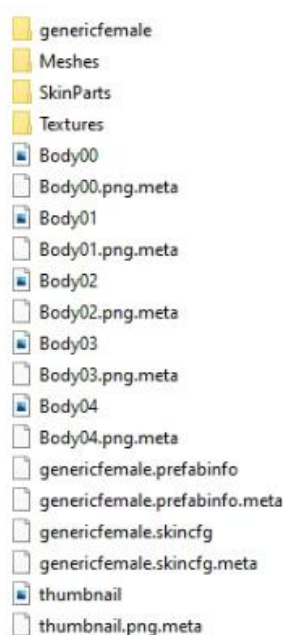
With the in-depth understanding of the general industry-level pipeline in mind, the next step focused on covering the multiple research topics that influenced VSTEP's own process.

4. Methodology

To answer the sub questions that lead to the main issue of the project, various researching methods were used on the topics of the project, to gather the information necessary to assure a correct solution.

4.1 The Current Library

At the beginning of the project, tasks which were related to fixing small issues or optimizing characters already implemented in the engine were assigned by team lead Erwin Wolf. This served to get me accustomed to the library, both outside the application and in-app, as well as give me a better understanding of what files were required to put together a character asset.



A typical character folder setup in the engine would have four folders where the meshes, textures, materials and nodes were stored. Nodes are items specific to the Unigine engine, a capsule where the meshes are stored. Outside these folders and into the root folder are kept two files which are required to implement the character in the RS application, the *prefabinfo* and *skincfg* files. Besides them, there are the thumbnail png files, which are pictures of the characters as they will be portrayed in the library of the application.

Figure 1. Folder setup in the engine and application

By interviewing Erwin, it was also apparent character assets would sometimes be stored in the engine library without fully respecting the recommended folder structure and naming conventions. This could sometimes cause issues in the application and confusion when it came to backtracking to the folders, such as not understanding a file's purpose or files that should not have been stored in the first place.

This situation was also apparent with the outside library. The source files for the characters that had been created by now were not organized in a consistent manner, with many folders named in an unclear manner, misplaced or without the necessary source files stored. The offline library also had character models that were not yet implemented in the application, despite the fact that they had been fully completed.

These two aspects of the library emphasized the lack of an established and documented folder structure model. By solving this issue and implementing a standard for storing both the in-app files as well as the source files, some of the hurdles impacting the pipeline would be solved: firstly, small in-app issues and their necessary fixes would be avoided, secondly, there would be no lost or unused resources that the team would not be aware of, and thirdly, a new employee would understand the files that are required for a character asset in a faster, easier manner, therefore becoming accustomed to the pipeline in less time.

Investigating the in-app library, the characters as they were usable in *RS*, gave a better overview of the quality standard that had been followed until the present and metrics regarding character variety.

The characters had a believable, realistic look and feel to them, although there were some distinguishable differences in detail between older assets and more recently made assets. There were also discrepancies when it came to gender diversity, with 19 adult male characters and 11 adult female characters, no elderly female characters while there are 3 variations of an elderly male character. The same discrepancy could be noticed in ethnicity, with only 6 characters out of the total number of characters with a visibly distinct ethnicity.

Through this qualitative research, part of the current pipeline issues could be documented:

1. Library in need of organization further cataloguing the existing resources
2. Establishing default folder setups for engine and outside library
3. Discrepancies in the character library regarding quality standard, gender, and ethnic diversity

These issues were confirmed by the team lead, Erwin, to be valid and worthy of consideration in the pipeline adjustments project.

4.2 RS needs

As mentioned previously, two kinds of character types were identified by investigating the *RS* character library. These two are, namely, the personnel characters and the generic characters. These categories have slightly different focuses, which can be seen in the following table:

Personnel characters Firefighters, Medical staff, Police ...	Generic characters
<ul style="list-style-type: none">- More detailed in features and clothing- Uniforms specific to company/country- Extra equipment (breathing masks, helmets)- Various additional properties	<ul style="list-style-type: none">- Usable for crowds and standalone- Generic clothing- Ethnicity- Body type

While personnel characters are usually requested separately by RS clients, the content team manages the production of generic characters on its own, having made the decisions internally on what kind of new generic characters to add.

My initial interview with the company, preliminary to the project start, assessed that the character needs of RS were generally more characters created in a shorter time and a way to portray the characters injured, as it could be a significant addition to simulating emergency scenarios.

However after discussing with Alexios Limnios, one of the contacts with the RS clients, it was brought up that what the clients wished to see more of was gender and ethnicity diversity, as can be seen in the following Figure.

Alexios Limnios Wednesday 11:55

Hey lolanda, thanks for reaching out 😊 Yes the two main sticking points are lack of gender diversity and ethnicity. Its not so much the body size however from a medical/triage point of view, it would definitely add value to have characters of different body sizes

I do not have much more specific information other than that. But for example, we mainly have white characters (apart from one Asian character for Changi airport) and this is an issue for some users/prospects

When it comes to more specialised characters such as controllable firefighters, police, paramedics, we only have a few female variations and all are white as well

Figure 2. Conversation extract about character needs in RS

This discussion put into perspective the fact that while generic characters were generally too few, simply increasing the number without focusing on the lacking diversity would not have been a positive outcome for the company.

4.3 The current pipeline

The existing pipeline of the company has the basic steps that character creation workflows usually consist of, which were established by the previous character artists of the company, Kenny Wood. During the initial stage of the project, he offered assistance in understanding the steps by compiling a series of video tutorials on some of the pipeline topics, such as skinning and importing the asset in the RS application.

With his input and the knowledge of the application's particularities, the pipeline presented itself in the following manner:

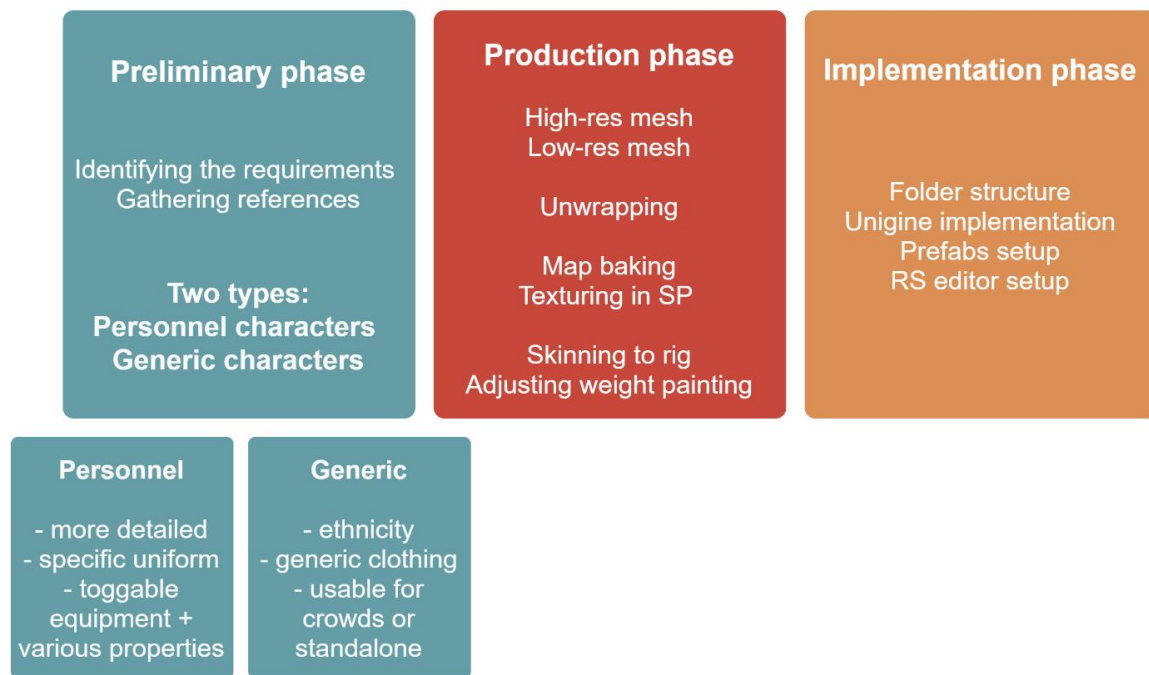


Figure 3. The current character creation pipeline

Improving the pipeline from a timely standpoint revolves around researching what can be automated or what can be used to shorten processes which would otherwise take a more significant amount of time. Currently, the most time-consuming part of the process is sculpting the characters. Therefore, the research conducted to solve this question is related to how this aspect can be changed to take less time, whether that is making use of existing resources (other meshes and models), including new techniques or tools to help in the matter (Houdini or Maya tools which could combine parts of different meshes and generate new items), or pursuing outside resources (character creation kits).

4.4 Resources and Tools for Improvement

There are different types of resources that the company can make use of in the context of the character creation pipeline, to make the process more efficient. First, the current character library, as mentioned in the previous research, contains a lot of source data that can be reused to create new characters.

Some of the improvement options which were explored relied heavily on the existing source files. Creating tools in Houdini or Maya which could procedurally generate a new character from existing mesh pieces were two ways of quickly increasing the character library. Another idea was to standardize the UVs of the existing meshes, so that new characters could be created simply from swapping textures.

However, these options did not become viable after a better understanding of the problem. RS's need for a diverse cast and variety could not be fulfilled by reusing the same existing meshes. Therefore, a way to generate completely new characters became the next focus of the research.

An important circumstance was that the company had confirmed they would be willing to invest in specialized tools. This, however, only as long as their purchase could significantly add to the value of RS and shorten the character creation pipeline.

During my research, I focused on two character creation tools that presented potential.

Tool 1: Character Creator 3

CC3(*Character Creator 3*) is a 3D character generation tool produced by the company Reallusion. It has an intuitive and easy to understand interface, that allows for high level manipulation of the character's features. Users can control the body type of the character, ranging from highly defined muscles to overweight, and many separate elements of the body, such as the broadness of the shoulders or the prominence of the cheekbones.

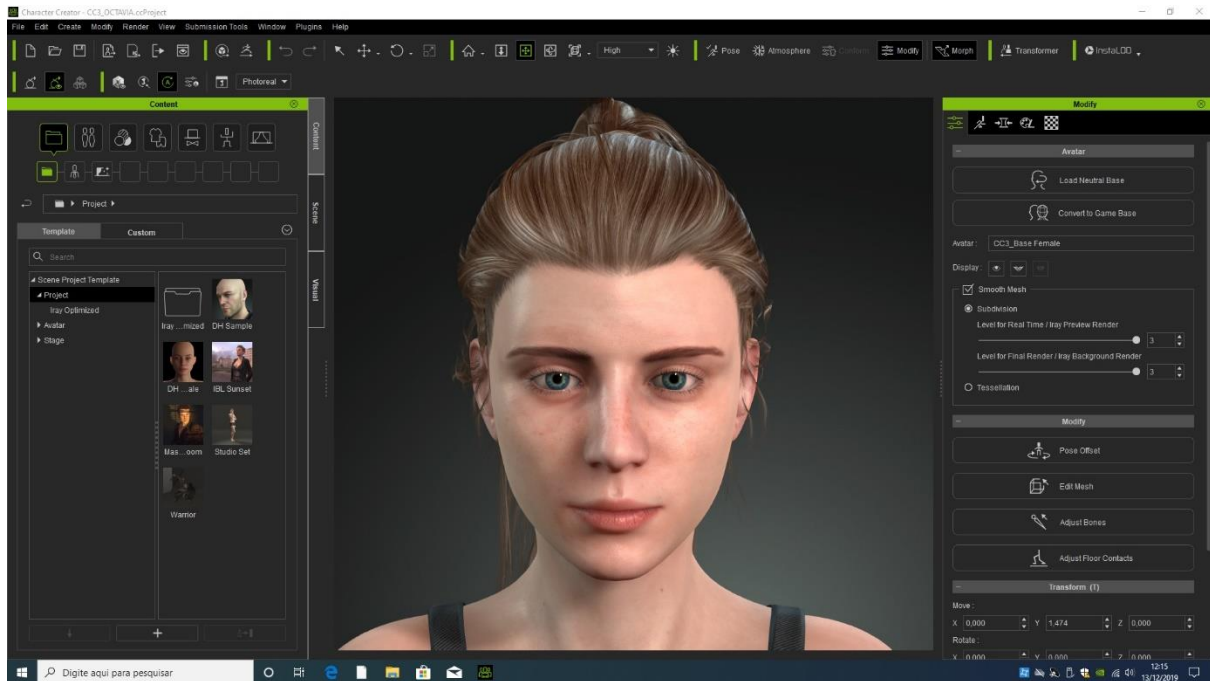


Figure 4. Character Creator 3 Interface

The tool features libraries for skin tones, clothes, hair generation, and posing. By default, these libraries have limited content, but new content can be added by purchasing *Reallusion* store assets or importing and configuring already owned assets.

Besides those libraries, the tool allows for technical adjustments of meshes. Textures can be altered or replaced entirely, meshes can be edited with an interface which allows vertex and face manipulation, they can also be optimized by poly reduction.

The tool allows for many exporting options as well. When exporting a character, the user can choose to merge materials or export different LODs, choosing the number of levels and their respective resolution.

The tool is actively and continuously developed by the owners, who periodically provide updates to the software. Purchasing it also provides a perpetual license, which means that the user is allowed to download and update their tool to the newest version. This makes CC3 a solution with prolonged potential.

CC3 covers a big part of the character creation process. Through it, sculpting, texturing, optimizing, and rigging are steps that become automated and significantly less time costly. For these reasons, CC3 became one of the tools to be tested for use in the VSTEP character production pipeline.

Tool 2: Autodesk character generator

Autodesk has also created a webtool for 3D character generation, which can be easily accessed by making an Autodesk account. After choosing two versions from the item library provided, the user can influence the features of the character with sliders, weighing towards whichever version they wish the character to look more alike.

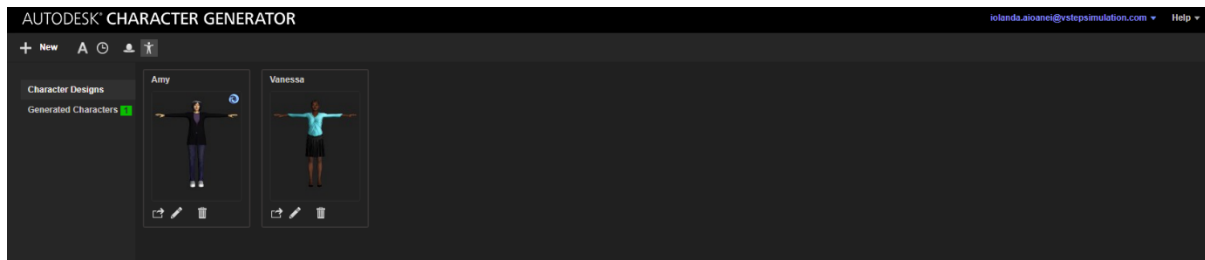


Figure 5. Autodesk Character Generator Interface

This tool has a set library of characters, skin, hair and clothes to choose from, which cannot be altered or added to, as there is no market or import option available. However, it has a good resource library for creating ethnically diverse characters, which is one of the focuses of this project.

The tool also allows for exporting multiple LODs, and automatically merges materials. The meshes are optimized to make minimum use of polygons by merging the clothes to the body mesh and allowing a high polycount only for surfaces that need detail, such as the face.

The generator license rights can be bought on a yearly subscription, but the price is very small by comparison to the CC3 software. While this is beneficial, the tool does not seem to be actively updated by Autodesk, which can be a bad sign for the future.

Like CC3, the Autodesk Character Generator covers sculpting, texturing, optimizing and rigging for the characters, which made it a good candidate for testing in the pipeline.

5. Investigating the tools

In order to assess how the tools work in the context of the pipeline, characters were generated through each tool and taken through the process of fully implementing them in the RS application. These iterations offered a clear overview over the pros and cons of each tool.

Iterations: CC3

Because CC3 offers many exporting and optimization options, the iterations that happened revolved around the different versions that could be exported, their pros and cons.

In the first iteration, I exported the default CC3 mesh on the highest resolution (Fig. 6).

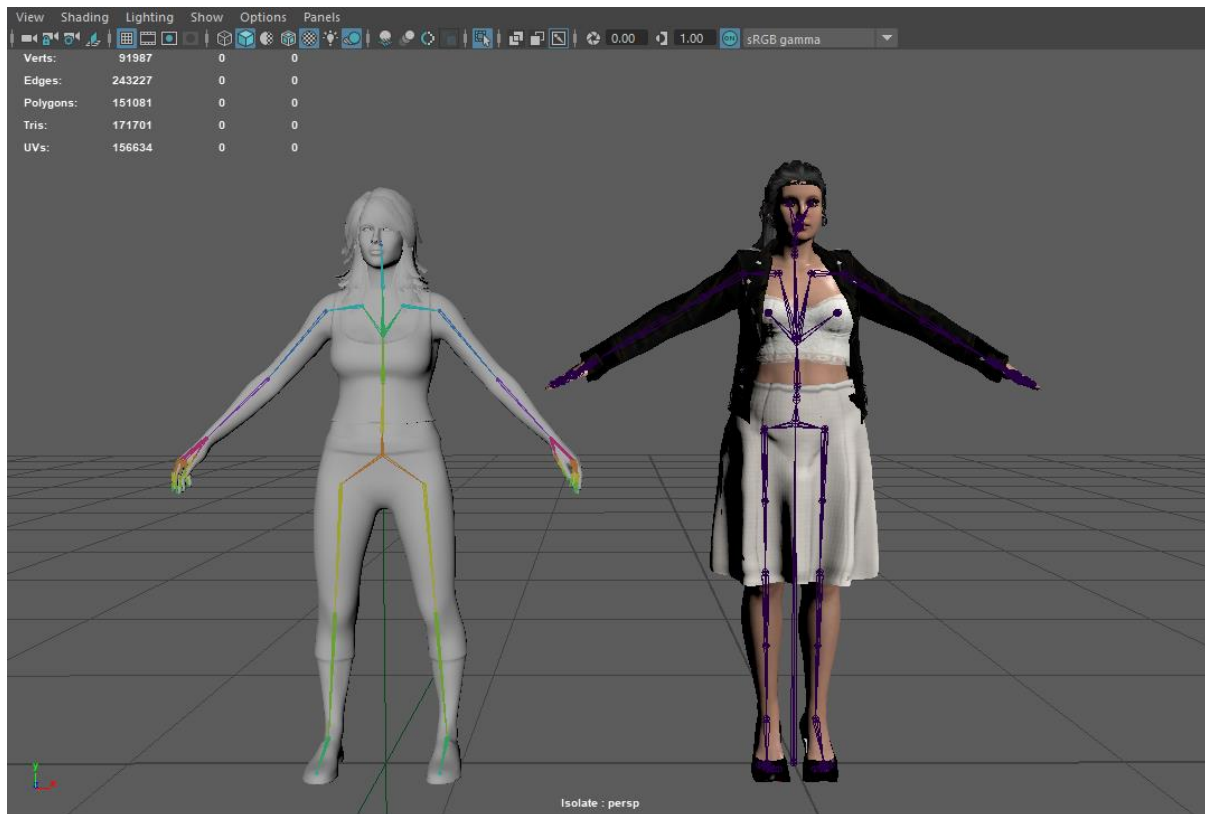


Figure 6. VSTEP rigged character (Left) and CC3 rigged character (Right)

By comparing the rigged mesh of the CC3 character(right) to the rigged mesh of one of the existing RS characters(left), some of the differences between the two surfaced. First of all, the CC3 mesh was rigged in a way particular to the character generated, as it took into consideration that the character wore heels. Even without that, the rig was too different from the default RS rig, which meant the same animations couldn't be applied for the CC3 rigged character. This presented a problem, as re-skinning the mesh would have meant taking away from the automation of the process.

After investigating this issue, it was apparently not possible to create characters on a custom imported rig in CC3. While the software has its own plugin for rigging, the plugin can only be used for creating the rigs from scratch, not for importing rigs and then creating characters on them.

However, CC3 provides options of posing the character from a library of default poses and exporting them with the chosen pose. Through testing, one of the poses from the CC3 library turned out to be similar to the default RS rig. This would make the process of skinning the CC3 mesh to the RS rig easier, as the CC3 mesh would need less modifications.

When importing the CC3 character in the RS engine, Unigine, the texture maps also proved to cause certain issues. Unigine materials use 3 texture maps, one for the general base color input, albedo, the PBR map, shading, and the normal map. Because the CC3 character generated separated transparency, metallic and roughness maps, they could not be used for the Unigine material. Therefore, the transparency map had to be manually added to the albedo map, and the metallic and roughness maps combined into a single map. This process can be done in Photoshop and is a common task for other Unigine asset materials.

The general aspect of the CC3 character was very close to the characters already existing in RS, both seeming to be at the same level of detail and style. The CC3 character, however, was still at its highest resolution, which made it too heavy for a general character.

For the following iterations, I exported another few generic characters, but using the automated optimization features. The characters were exported with the total mesh resolutions to a maximum of 36k polygons, the resolution of other RS characters, along with some other automatically generated LODs.

While the realistic features of the characters remained intact, there were certain mesh issues that were noticeable at a closer look. Because the actual clothes meshes were placed over the body mesh, the CC3 automated optimization algorithm would try to merge the two into a single mesh, which resulted in jagged edges and bad topology (Fig. 7).

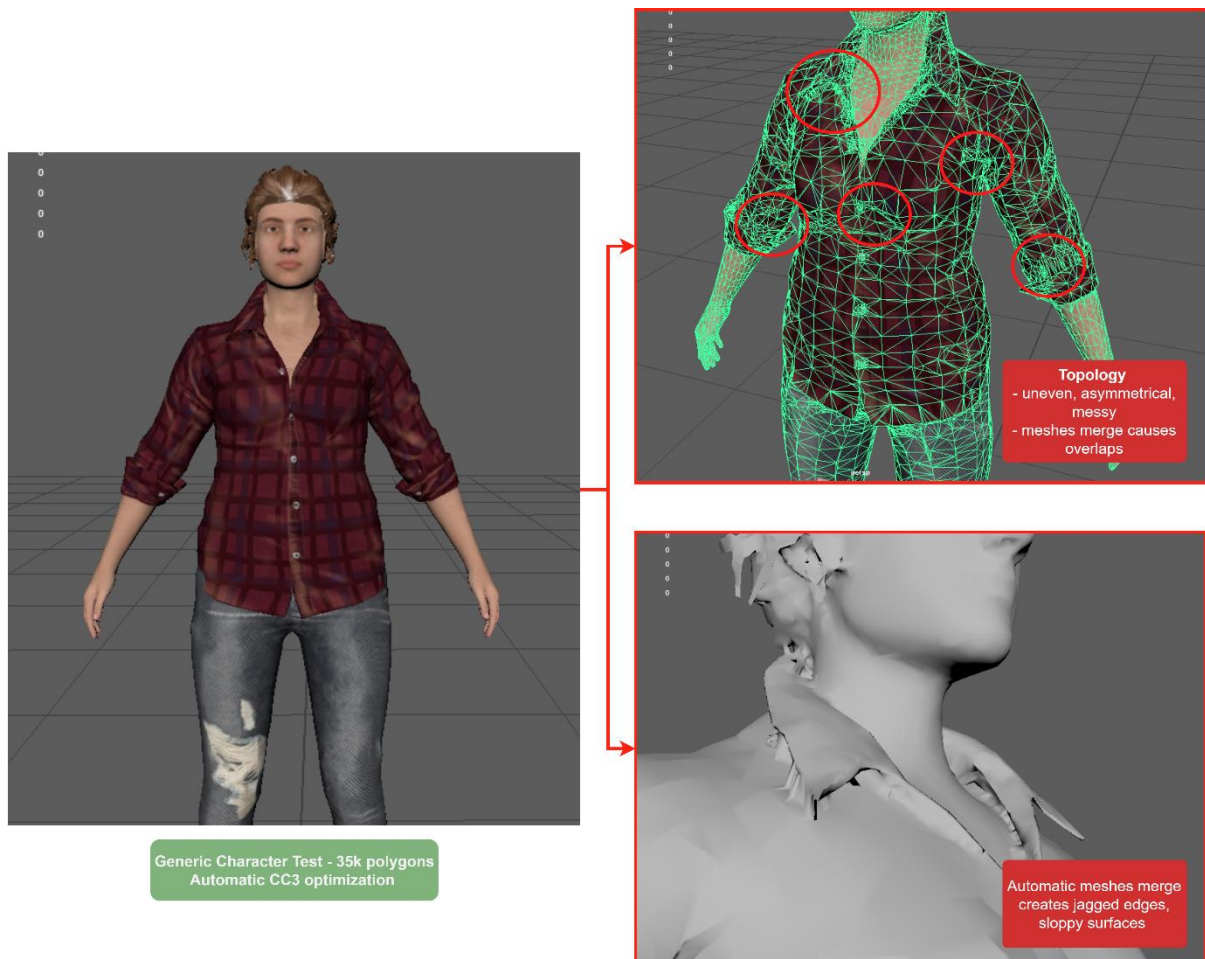


Figure 7. CC3 mesh optimization issues

These issues would cause the workflow to slow down, as they would have to be fixed manually through a tedious process. For this reason, the automatic optimizer from CC3 was a not feasible option.

After experimenting by optimizing separate parts of the mesh without satisfying topology results, colleagues brought up the option of using 3dsMax for fixing the issues. In particular, the 2021.3 version for 3dsMax has implemented new retopology and mesh optimization tools.

To test them, I took two different characters in their original, high resolution state, to be optimized in 3dsMax. The results were satisfying – the topology generated was significantly cleaner and there were little to no mesh issues with the character itself (Fig. 8).

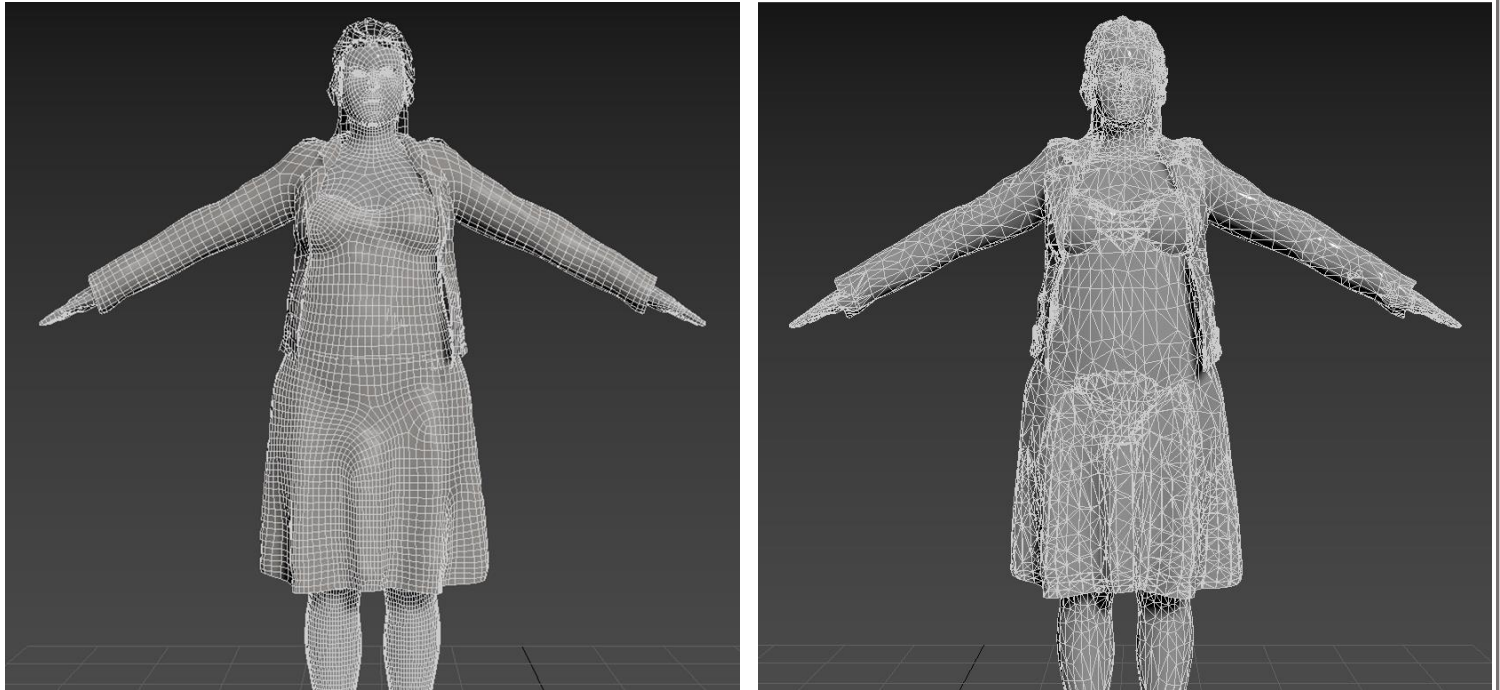


Figure 8. 3dsMax optimization (Left) and CC3 automatic optimization (Right)

However, if overused, the tools could damage the features of the character, and the optimization erased the UVs of the original mesh. Thanks to 3dsMax's projection tool, the UVs could be reconstructed, although they, too, had small issues that needed manual fixing.

Compared to the mesh issues that the CC3 automatic optimization caused, the 3dsMax optimized mesh needed far less and easier to handle manual fixes. This made the use of CC3 a viable option again.

Another very important CC3 option investigated was asset importing. The user could import their own FBX files in the library for direct use in app, and together with the many mesh transformation tools and material editing, it was possible to create a fully custom library of, for example, clothing meshes. This feature meant that any item from the RS source file library could be reused in CC3, and the company could expand its CC3 library solely through their own content.

The CC3 testing concluded that the software would cover a significant part of the character creation pipeline, but with certain conditions:

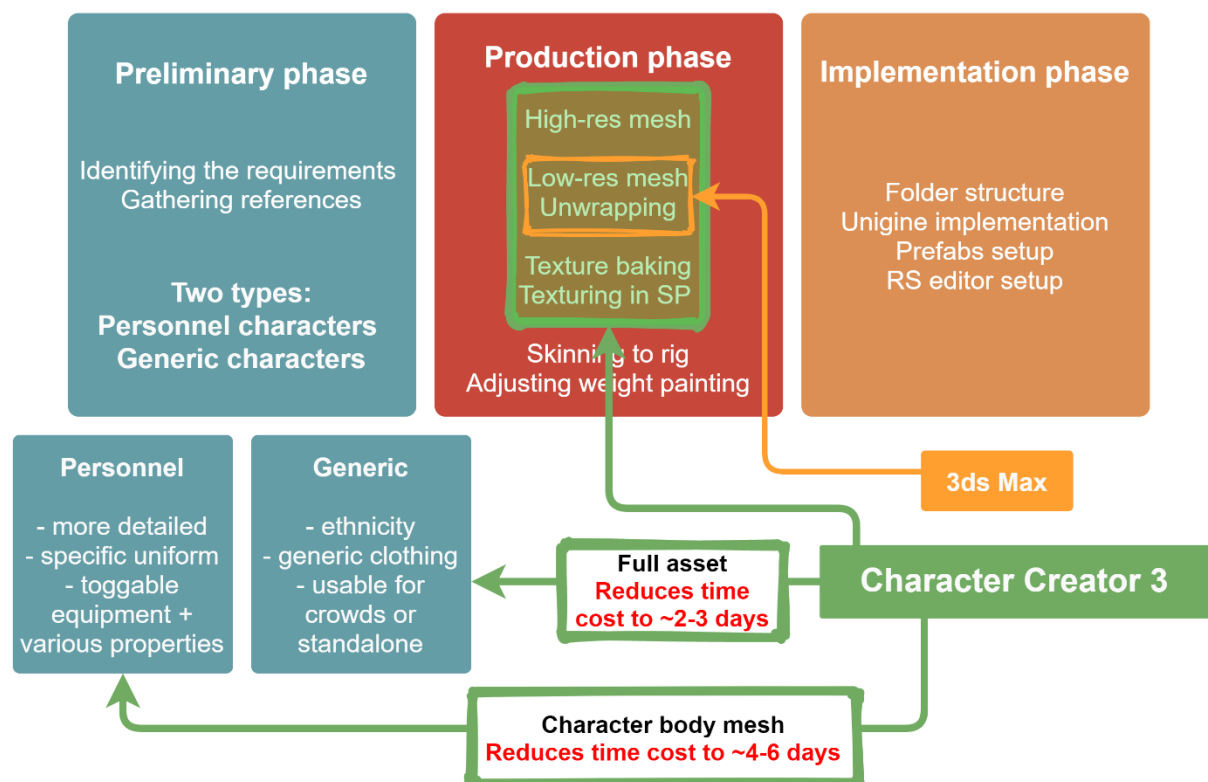


Figure 9. Character creation pipeline concept using CC3

As it can be seen in the graph, CC3 would assist with creating the high-res mesh and texturing of the character, while the low-res mesh would need to be created in 3dsMax.

Through the iterative tests, I estimated that, with the use of CC3, the time cost of producing a generic character would become two to three days. Because of the easy to use interface and convenient options, creating a character can take as little as 30 minutes, which can then be followed by all the technical necessities to bring the character to RS.

However, since personnel characters need specific uniforms that will, more often than not, need to be created manually, the time cost is reduced to four to six days. These estimates are influenced by several factors, such as the experience of the artist with the pipeline and the uniqueness of the character and can be prone to fluctuations.

Iterations: Autodesk Character Generator

On the first iteration of the Autodesk Character Generator, I explored how the system worked and the assets library.

The character's features are formed as a medium between two different character meshes. The user has control over certain details and can choose through sliders for the character to resemble more of one or the other mesh.

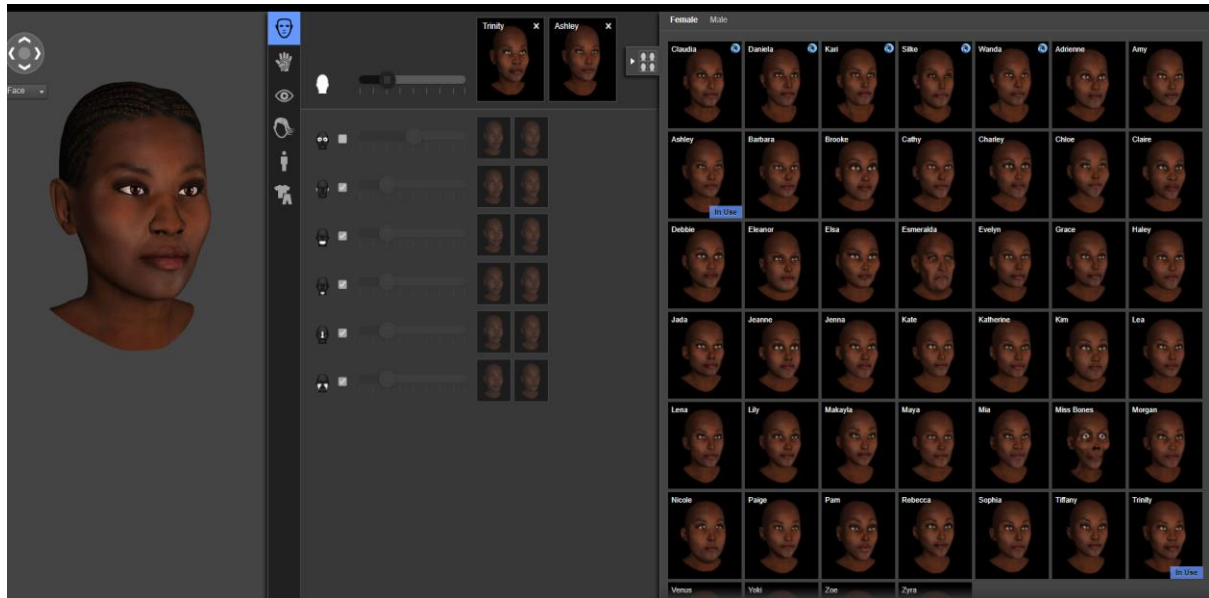


Figure 10. Autodesk Character Generator interface

While this approach could seem limited, the application provides plenty of base meshes to choose from, which can in turn create a high number of characters with unique looks.

On the other hand, the hair and clothes library were limited. In contrast to the CC3 application, the Autodesk platform does not have a way to import or use new items in the library. This meant that the library could not be extended by any means.

The exporting options were simple and efficient – Autodesk would export a fully merged mesh, with the option to merge the materials and export different LODs. However, the high polycount LODs were only unlockable by investing money in the application in advance, so the iterations presented here occurred with the low poly LOD that the application allows.

Opening the exported asset in another 3D software, as mentioned before, revealed a greatly optimized mesh. The topology was very well structured, and polygons were used wisely, with sections that were denser for the sake of fidelity, such as the face. The clothes were merged into the body mesh, which was an aspect better executed than in the CC3 application.

Like CC3, the Autodesk Generator exports characters rigged by default to a custom rig. Because of this, the character's limbs need to be positioned to match the RS rig before being skinned to it. The characters would always be exported in a default T-pose.

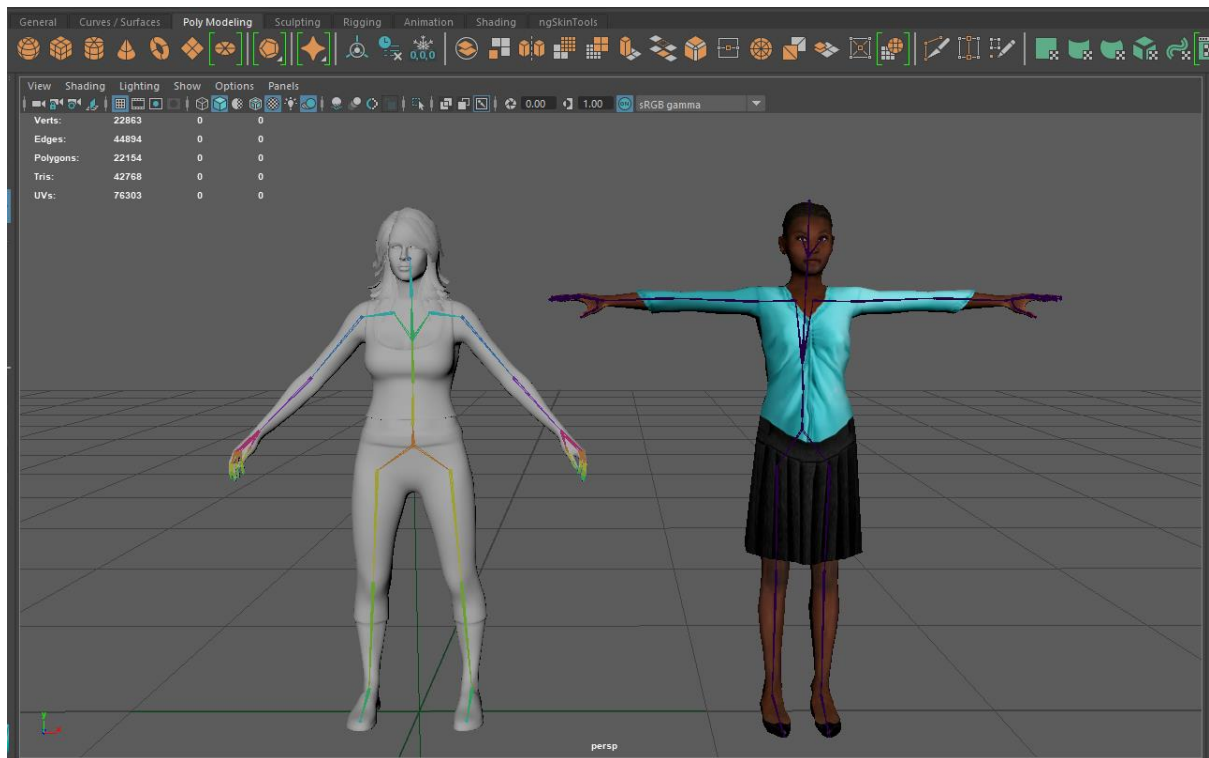


Figure 11. VSTEP rigged character (Left) and Autodesk rigged character (Right)

Importing the character in RS, though, and comparing it with the existing characters brought up some apparent issues with the general look of the model. Compared to the existing characters (Fig. 12), the Autodesk generated character looked proportionally less realistic – the features tilted slightly towards a cartoon style rather than a realistic style.



Figure 12. CC3 character (Left), Autodesk character (Middle), existing VSTEP character (Right)

In order to add more detail and quality level to the character, a possible option explored was taking the mesh into Zbrush and sculpting on top of it. That solution proved to be problematic, at the moment. Organic sculpting in Zbrush is best done with a graphic tablet, which turned out not possible through a remote working environment since the driver installed on the remote computer could not recognize that the tablet was connected to the computer. Because of this, the only solution would be to work on site at the company computer, which would not be possible until further into the future.

The Autodesk Character Generator testing had results that were similar in ways to the CC3 conclusions, however when looking more in-depth, there are considerable differences:

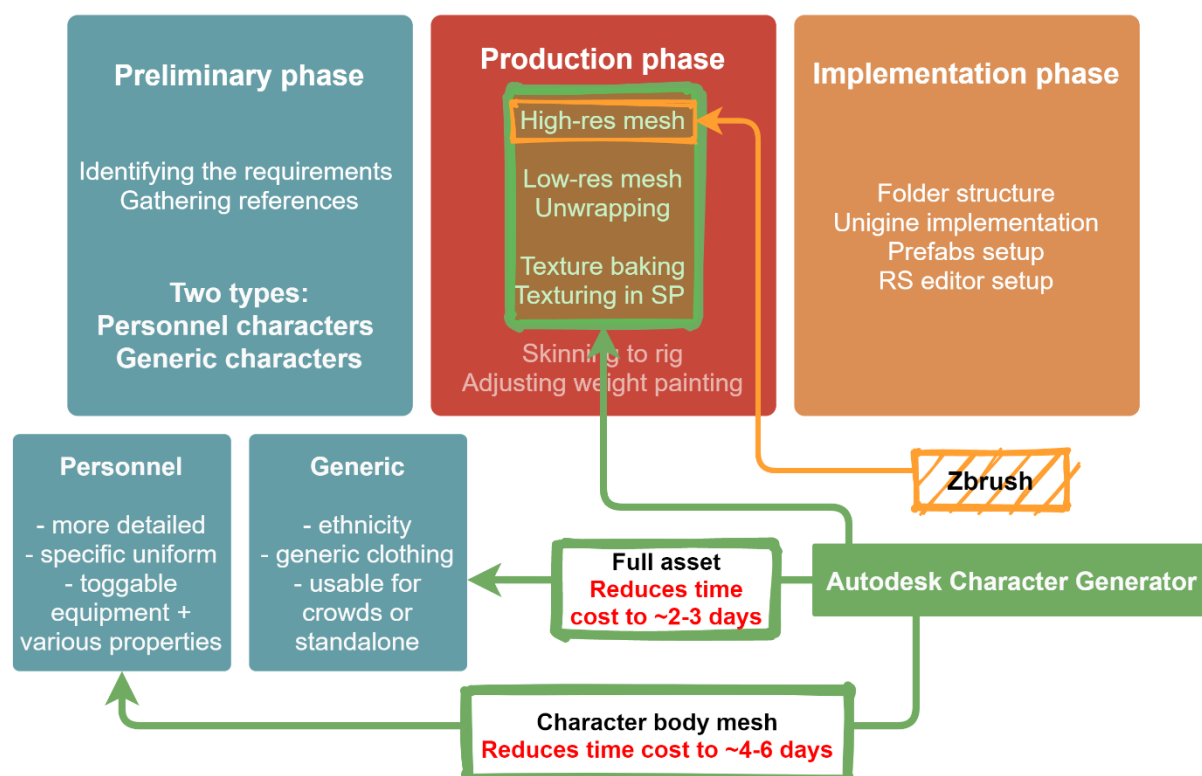


Figure 13. Character creation pipeline concept using Autodesk Character Generator

As with RS, Autodesk Generator covers a big part of the production phase. However, the high-resolution mesh that it generates is not enough as it is provided, since the overall appearance does not match the RS standards. The best solution for the issue, adding more detail to the character through Zbrush, could not a viable option at the moment because of technical restrictions.

The Autodesk Generator could also reduce the time cost of creating generic characters to two or three days: generating the character itself can take less than one hour of work and the mesh itself needs minimal adjustments to bring it to the RS standards. Like CC3, because of the easy to use interface, creating the character itself can happen in under an hour, with the rest of the time allocated to the technical process. Unfortunately, this tool does not shorten significantly the process for the personnel characters either, as the uniform would still have to be tailored specifically to the request.

6. Results

With the two tools tested, the next phase revolved around completing the character pipeline guide based on the decision made by the company about the tool purchase.

6.1 Determining the solution

With the two main options tested, a presentation for the other stakeholders in the company ensured about how the tools tested would affect the pipeline and my recommendations. After explaining how each of the tool operated and what features they have, a list of pros and cons was assembled for each of them, to better portray the differences and similarities.

Option 1 – Character Creator 3

Pros:

- Characters can be created/adjusted easily with the CC3 interface(by anyone)
- Character quality to a high standard
- A lot of options for customization and buying extra assets to create characters with variety
- Custom assets can be imported
- Baking and texture map generation
- Many exporting options(merged mesh, separated mesh parts, merged materials, LODs)
- Materials can be edited

Cons:

- Generated characters can create technical issues
- Mesh optimization has to be done outside
- The default clothing assets are not enough on their own
- Characters cannot be created on a custom imported rig
- Maps generated not fully fitting unigine requirements
- Ethnicity diversity limited without buying the SkinGen expansion pack

Option 2 – Autodesk character generation

Pros:

- Characters can be created/adjusted easily(by anyone)
- Meshes stick to the essential
- Good resources to create diverse characters
- Sculpting and texturing process is almost completely removed
- Meshes highly optimized
- Baking and map generation
- Can export different LODs

Cons:

- Character quality is lower than the existing RS meshes
- Customization options limited – custom assets can't be added
- Characters cannot be created on a custom imported rig
- Maps generated not fully fitting unigine requirements
- Does not seem to be actively updated by Autodesk

Beside the two options, the presentation also explained how the pipeline would function if remaining unchanged: a way to shorten the process would be to make use of the source files as much as possible and invest in smaller assets, such as already created body types, to make up for the lack of experience in character creation in the content team. This would not shorten the time cost of the process however, and would not establish a constant and coherent workflow.

After a week of deliberation, the stakeholders decided to invest in the CC3 tool to help support the pipeline. With their confirmation, the project reached the stage where the pipeline needed to be polished and established properly.

6.2 Reorganizing the folder structures and templates

Together with Camille Pauw, we discussed a new and improved folder hierarchy for storing source files. My recommendation(Fig. 15) was to organize the characters in two main categories, Generic and Personnel, with undergoing structures based on country, age, gender, and ethnicity. This way, navigating the library would be easy and intuitive, with the user following the path to the specific character they are searching for (Fig. 14).

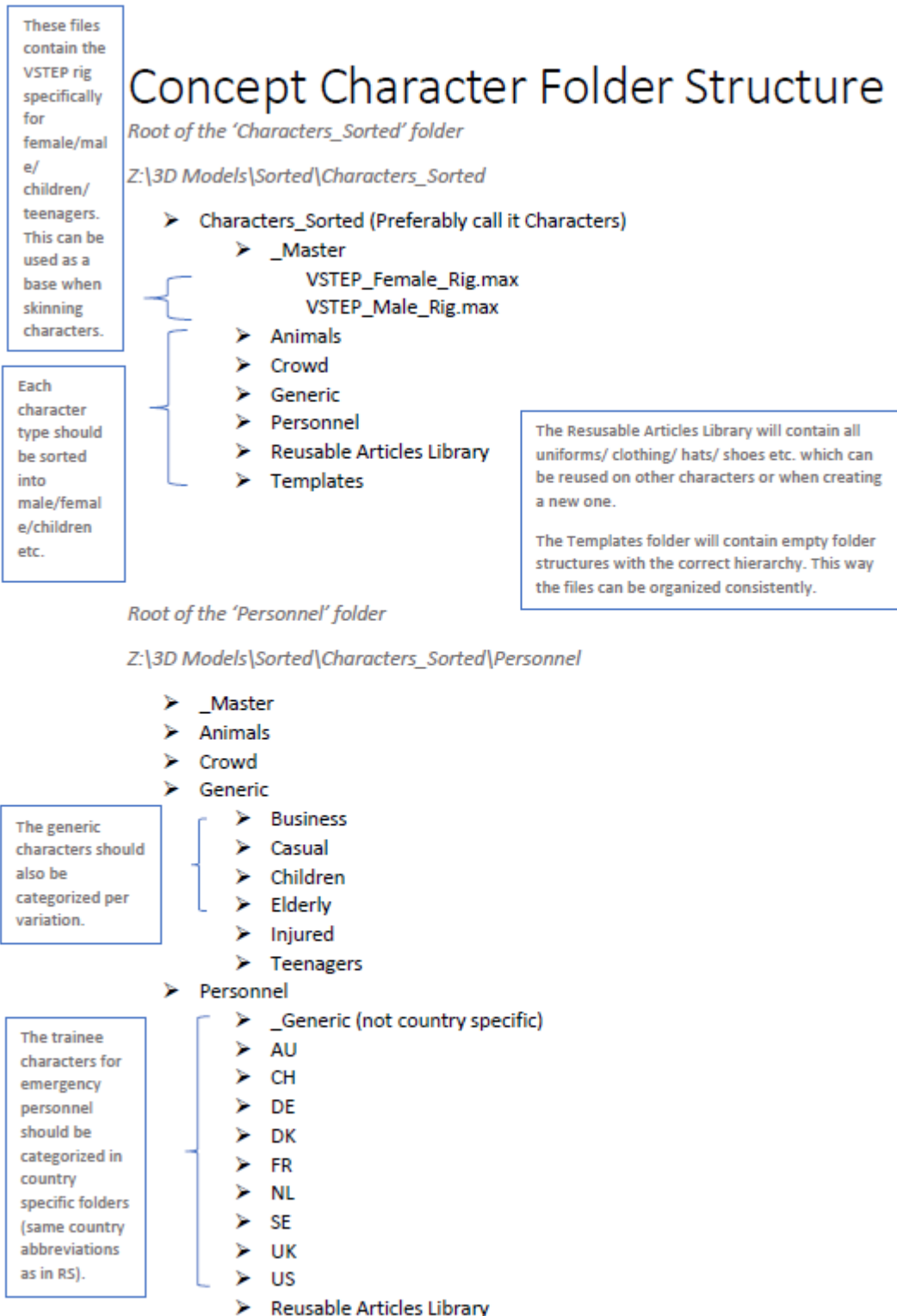


Figure 14. Concept of the reorganized source files library, documentation by Camille Pauw

20/05 12:54

👍 1

as for the big root folder, I like your idea and it's very neatly laid out and easy to understand, but I was also thinking how it would be like if we consider the focus to be ordered the other way around, character type>purpose>location(for personnel)>gender. what do you think? usually when you go looking for source files on characters is the gender category more helpful to find it quickly? it's just about how we can make it easier for us to find stuff fast, so either of the orders is good 😊) i think we also need to include the crowd category somehow, since it's characters that are the most optimised and not usable as trainees. Could also add it in the Generic folder of your original structure 😞

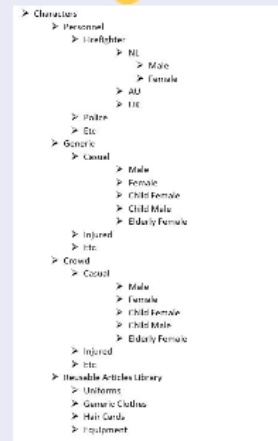
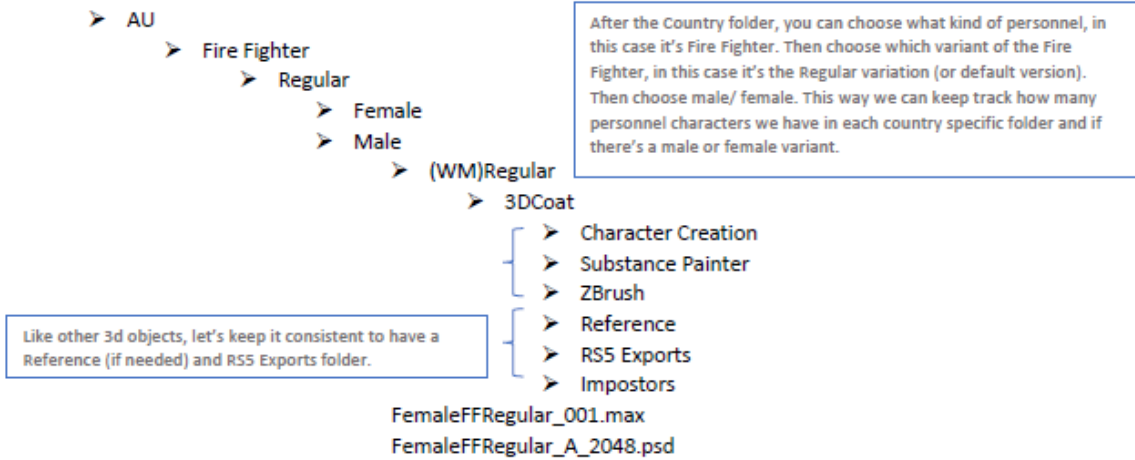


Figure 15. My recommendation for the folder hierarchy

I also created template folders for RS implementation and source file storage, with structures discussed with and approved by Camille(Fig. 16).

Root of the 'Fire Fighter' folder (should also be applied to all personnel folders)

Z:\3D Models\Sorted\Characters_Sorted\Personnel\AU\Fire Fighter



Root of the 'Generic' folder

Z:\3D Models\Sorted\Characters_Sorted\Generic

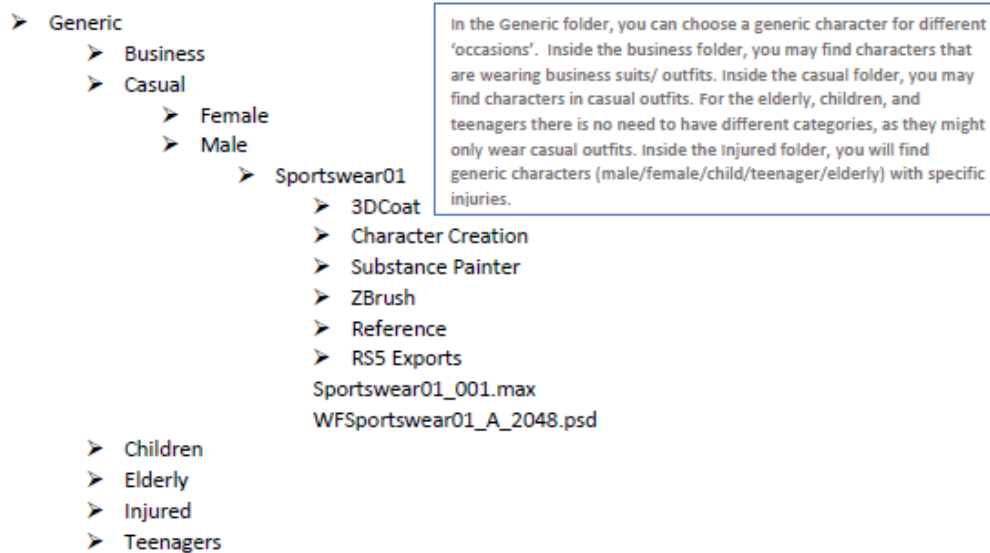


Figure 16. The source files template folder structure

With these restructured folders, the source files could be much better identified in case they could be reused. The process of storing them also become more efficient, as the artist would only need to copy the template folders and store files in their respective places, instead of creating their own folders.

6.3 Documenting the new pipeline

With CC3 approved by the stakeholders, the next step was to properly document the pipeline, which was one of the goals of the project.

Throughout the previous research and iterative phases, an initial draft of the pipeline was made, the stages of the pipeline were shortly defined. To make sure the explanations were understandable enough, visual guides began to be compiled – screenshots of steps from the process, sometimes with annotations or highlights. These guiding screencaps were made while also testing certain parameters

of the 3dsMax optimization process, the CC3 interface and the requirements to fully setup a prefab in RS.

When reaching a first comprehensive draft, Camille Pauw provided feedback for the guide in general and based on a few specific questions, which can be seen in [Appendix II](#). The feedback was mostly positive, with a few mentions of logistical errors which were quickly fixed after revisiting the character node format in Unigine. With Camille's agreement, the decision was made to use video tutorials provided by one of the previous character artists, Kenny Wood, on the subject of skinning a character mesh to a rig and weight painting, as the tutorials were clear and relevant to the subject.

Following this, the *.skincfg* and *.prefabinfo* templates were created by investigating the code of the files and testing certain input parameters. By doing this, it was possible to determine neutral file formats which would configure the character correctly in RS, with added explanations as to what important code alterations were necessary for fulfilling the character's specific requirements.

The second round of feedback from both of the supervisors was also positive. There were some discussions about small details, but overall they were pleased with the guide and found it accurate and useful. The feedback can be consulted in [Appendix III](#).

With the final adjustments following Camille Pauw and Erwin Wolf's feedback, the pipeline guide was finalised and approved. The guide can be read in [Appendix IV](#).

7. Conclusions and Discussion

At the end of the project, the conditions of satisfaction set at the start of the project have been met in a satisfying way:

- The character pipeline has been documented in a manner that makes accessible for any team member with 3D modelling technical knowledge.
- The source file folder structure has been reorganized in an understandable manner, with templates for storing future source files and for implementing the characters in the engine, which will shorten the process of finding source files and keep a consistent database that is easy to navigate.
- From the character creation practices researched, two main tools have been researched and tested, with one of them ultimately chosen to be implemented in the pipeline, which has almost completely eliminated the need to sculpt characters from scratch.

While there have not been new characters added to the library throughout this process, this was not the focus of the project and only a non-mandatory consideration.

Because of the constantly evolving industry and the many ways character creation can follow, there were techniques that required more time and testing to establish whether they could have been a good fit for VSTEP's pipeline. For example, recreating characters from existing source files – while this process may not seem viable, with the correct setup and some more experience, the process could be adopted in an efficient way. Another downside would be that the pipeline tools still need some time investment to reach the best potential possible, aspect which could not be completed for the duration of the project alone.

However, given the current demands of the RS clients, supporting the pipeline with CC3 is the best option to help VSTEP become more independent in its character needs.

8. Recommendations

While the need for sculpting characters from scratch has been eliminated, there are still a few blockers that hinder the pipeline.

One of them is the lack of clothing and hair options for characters. The library for these items can only be expanded by either owned assets or by buying additional assets from the *Reallusion* store.

RS's existing library of characters can provide resources which can be added to the library. Since CC3 has the option of importing custom meshes, clothes and hair meshes from existing RS characters can be detached from the full character mesh and added to CC3 to be reused. This is a task that can be time consuming, as the meshes would have to go through some manual modifications to be correctly implemented in CC3 – making sure the pivot point is correctly set, modifying the UVs. However with CC3's options of adapting clothing meshes to characters and material adjustments that can be done in-app, the library can become decent enough to cover the character needs of RS simply through already owned assets.

The *Reallusion* store provides content created by other artists specifically for CC3 at various prices, depending on the asset. This would be the fastest way to broaden the library, as the assets are already tailored for CC3, and would also bring more variety to the RS character library, since the additions would be completely new. Buying new assets is dependant on how much the company is willing to invest.

Another issue would be creating specific uniforms for personnel characters. Creating a template for such uniforms is, in many cases, not a viable option, so personnel characters will still require a certain amount of manual work to be completed. For any new personnel character, though, it is important to first check if there is any possibility of using an existing asset as a base for the uniform. This is where having an organized source file library counts the most – having an easy to access database with traceable source files will make the process of finding a viable base faster.

As for RS's character needs, with CC3's features and options to create a variety of characters, the next step would be to eliminate the lacking diversity in the character library. This can be done by organizing and assigning character creation tasks. By following through with these tasks the character library can be expanded to a more diverse cast, as per the requirements of the RS clients.

Overall, with the automation added to the pipeline and the pipeline efficiency increased through documentation and library organization, VSTEP can consistently rely on having their character needs met. With enough dedication to continue this project and build a solid library through CC3, the pipeline will continue to develop in even better ways.

9. References

Bettis, D. E. (2005, May). *DIGITAL PRODUCTION PIPELINES: EXAMINING STRUCTURES AND METHODS IN THE COMPUTER EFFECTS INDUSTRY* (Thesis). <https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/2406/etd-tamu-2005A-VIZA-Bettis.pdf?sequence=1&isAllowed=y>

Character art pipeline in video game industry Part01. (2020, February 27). [Video].

YouTube. <https://www.youtube.com/watch?v=qZ8xfVyfvAY>

FlippedNormals. (2018, February 26). Is Good Topology Overrated? [Video]. YouTube.

<https://www.youtube.com/watch?v=s2CYuRxFUSI>

10. Appendix

Appendix I: Character Creation Pipeline Research

Source 1: William Paré-Jobin, [Create a real-time game character](#)

William Paré-Jobin is a character artist at Red Barrels, Montreal. This is a tutorial made by him as to the steps and tricks to create a high resolution sculpt.

- References – getting as many as possible and from any available source
- Sculpting:

Primary form – defining the silhouette and base, work fast and establish the anatomy, reuse parts from other meshes/objects

Secondary form – adding folds, big wrinkles, muscles, defining parts. Add as much “material” feel as you can to get a better idea of the concept.

- Start adding details (default brush or random alphas, project details with maps from texturing XYZ, use layers to handle intensity)
- Hard Surface sculpting in zbrush
- Generating haircards in Maya

Source 2: Adam Fisher, [Create a Game Character: Joustier](#)

Adam Fished is a freelance 3D character artist who has worked for numerous clients such as Creative Assembly, Infinity Ward, Arkane Studios and McFarlane Toys.

An in-dept tutorial that covers all the aspects of creating a game ready character.

1. Blocking in the proportions
2. Sculpting the face
3. Sculpting the armor
4. Creating the armor meshes
5. Finalizing the details
6. Creating the low poly model
7. UV unwrapping and texture baking
8. Texturing the armor
9. Texturing the face
10. Model presentation in Marmoset

- Follows the basic structure of character creation pipeline
- Because the design is highly stylized, many pieces require dedicated sculpting/modelling time
- The process is not automated

Source 3: Tudor Moldovan, [Game Character Production - Applying Industry Standards across Production Pipeline](#)

Tudor Moldovan is a 3D artist and art director at ASSIST Software.

- Goes through the process of creating a realistic character
- Talks about using default tools to save time and other base resources that can be used as alternatives instead of making a resource from scratch
- Importance of base mesh in a production pipeline
- Using Topogun for retopologizing
- Marvelous Designer for creating clothes
- Rigging and skinning in Maya

Source 4: Rich Diamant & Judd Simantov, [The Uncharted 2 Character Pipeline](#)

A presentation by lead character artist Rich Diamant and lead character TD Judd Simantov from *Naughty Dog* Studio about the character creation pipeline for the game *Uncharted 2*.

- This tutorial also involves realistic characters
- Many tools were created specifically for the pipeline which created the assets exactly as needed by the team and automated the process
- Deeper insights into creating a rig

- Explanations as to what needed to be improved to the pipeline from the previous game
- How using base meshes and other assets helped make the process more efficient
- Transferring animations between different rigs

Source 5: Julien Lalleve and Kieran O'Sullivan, [Character Pipeline and Customization System for Far Cry Primal](#)

Source 6: Baj Singh, [Character Creation Pipeline Secrets](#)

Baj Singh is a character artist, project art director at , 'Total War: Warhammer' Franchise at Creative Assembly.

- Gathering references: a lot of anatomy and proportions, clothing, accessories
- Use PureRef to see references
- Feedback from several people helped achieve the look wanted
- Blocking and sculpting in Zbrush, while using other software for different parts: Marvelous Designer for clothes, 3dsMax for hard surface elements(hard edges via smoothing groups are his recommendation)
- Taking each mesh created in a different software back to Zbrush to add some more natural detail – automation can make it seem stiff
- Using Photoshop and Quixel Suite instead of Substance Painter
- Tips on hair cards

Source 7: Antony Ward, [How to create character models for games: 18 top tips](#)

Antony Ward is a digital artists who has worked in the game development industry since 1990's, for some of today's biggest companies and titles. He has written three books on development subjects: Game Character Development in Maya (2004), Game Character Development (2008) and 3D Modeling in Silo (2010).

1. Know your restrictions

Environment settings, how the character will be used in the game, working units, polygon limits

2. Research

Work on references. Make a moodboard, find photographs and concept art to compose your character.

3. Start simple

Don't rush into details. Block out figure and use basic primitives, then stitch them together. Make a broad overview of the character's proportions.

4. Try subdivision surfaces

Useful for creating a smoother model with initial low polygon count.

5. Add detail only where it's needed

Also related to how the character will be used. Ex: if it's seen from distance, no point in adding detail to the fabric of the clothes. Zoom out when working.

6. Be economical

Remove all unseen geometry and remove all unused geometry (anything which doesn't directly benefit the shape of an object or how it deforms)

7. Follow natural muscle lines

You don't need every muscle etching into the surface, however: the normal map, if you can use one, will add nice detail for you. Think of this more as the foundations of how the model will move.

If done correctly, when the arm raises the upper arm and shoulder, geometry will twist and crease as real muscles do.

8. Retain a seamless model

Physically building parts of the mesh to avoid problems like skin poppin through the clothing or separate elements not moving correctly (Maintenance guy).

9. Keep topology clean

Remove anything that isn't directly needed by the model and won't enhance the way it looks or deforms.

Also move around the mesh and look for areas that aren't tidy. Try to keep a clean, grid-type layout across the surface, and everyone will be happy.

10. Check deforming areas

Make sure you have enough geometry in areas that will deform.

11. Image planes

Importing image files as references in the sculpting scene

12. Work smarter not harder

Make things easier for yourself. If you create a good hand model with correct topology, for example, put it to one side so you can use it again on another character.

If the character design you're working on is symmetrical, just model half and mirror it.

13. Keep the scene clean

Correctly name each model, delete unused nodes and shaders, and bake any construction history on your models. Leave the scene as you would like to be given it.

14. Unify normals

Use single-sided normal to highlight potential problems

15. Let the normal map help

Use normal maps to add details to characters, especially objects close to the surface of the skin.

16. Prioritise

Think about what will be seen most in your model and prioritise polygon budget accordingly.

17. Keep regular, sequential back-ups

Set sequential auto-saves to avoid losing work to crashes.

18. Check the silhouette

A quick test to help weed out those jagged, polygonal areas is to disable the lights in your scene.

This will leave you with a silhouette you can move around, making more obvious any areas in need of smoothing out.

Appendix II: Character Creation Pipeline Draft, First Feedback

Appendix III: Character Creation Pipeline Draft, Second Feedback



Camille Pauw 03/06 14:15

Finally got through the last pages. It was easy to follow and nicely written. There are some processes described which might change in the future (for example for creating thumbnails, we might get an automated thumbnail generator), but that's not an issue for the current pipeline. I think with the help of this documentation that anyone from the art team can basically create a character and make it work in RS.

For now I'll say that this character creation pipeline documentation is approved!



Erwin Wolf 04/06 14:58



=== Character Creation Pipeline ===

- The documentation for the character creation pipeline is clearly written with a nice detailed explanation of all the processes and tool involved.

Here some feedback:

- Perhaps already planned but have a title page with name of author.
- Really nice example of the paramedic to show what is important for references. Also the screenshots throughout help a lot explaining the process.

- Step 3 Setting up the source files folder.

Change the template and the image shown here a bit. It should not be called '*RS5 Exports*', but '*RS Exports*'. This as the name RS5 does not exist anymore.

...this helps preserve *memory* space-> this helps to preserve *harddisk* space.

Step 6: optimization for in-app use

The polycount mentioned here (e.g. up to 40k), are they agreed upon with Camille? They seem a bit high for characters but not sure.

Step 7: Skinning and weight painting

The links to Kenny's tutorials (for Maya and 3ds Max are not in yet).

Step 8: Folder structure

Crowd and Generic characters are store in the same -> i would leave out the name crowd as a crowd character is in essence a generic character. So the line could be: 'Personnel character and generic characters are stored in the same...'.
Another name for personnel characters (if this is not clear enough) could be 'avatars' or 'emergency personnel'.

Checklist:

The SkinParts folder contains all the nodes that references *the meshes and used materials*.

The prefabinfo and skincfg files must be named exactly the same as the name of the rootfolder *but all characters of the filename should be in lower case*.

Setting up the lod's:

Shouldn't the Min Visibility of LOD1 just be 0. What does -inf mean?

Creating a correct thumbnail

Save the thumbnail as a .png file type and name it the same as the node which it belongs to. *The resolution of the image should be 256x256px.*

Folder hierarchy and files

... or extra files that are don't need to be stored in the repository -> or extra files that *are not needed* to be stored in the repository.

Overall great job lolanda.



Camille Pauw 07/06 08:03



-inf is unigine's default for 0 I think. It's something that has always been there before 😊 Whenever you add a mesh inside a node, the -inf to inf are the default settings anyway, no need to start changing that now



Camille Pauw 07/06 08:19



Some clarifications for some of Erwin's feedback:

Step 3: Others (like Tom and Maran) have named their export folders for the newest RS version 'RS5 Exports' As we used to have a short time where we would have RS4 exports. Ever since I was at VSTEP I was told to name it like that, so for the sake of consistency (you could check for example the vehicles, ships, buildings folder etc.) I'd keep the 'RS5 Exports'. Maybe in the future there might come a RS6?

Step 6: the polycount for high poly mesh could be around 40k (we don't have a set number). Kenny's high mesh characters are usually between 35k to 80k. For the low poly, it's always the goal to go as low as possible without losing too much quality.

Step 8: Crowd and generic could be the same, however as we are still not sure of how the crowd system will be developed, there might be a possibility that the 'crowd' character needs some requirements (perhaps specific export settings) for the crowd system. It's not really relevant for now, so indeed a separate crowd folder isn't really needed.

For personnel character I'd keep it like this as not all personnel characters are emergency characters (for example, the aircraftpilot, airportworker, roadworker etc.)

Appendix IV: Character Creation Pipeline Guide

VSTEP: Character Creation Pipeline Guide

With Character Creator 3

June 2021

Compiled by: Iolanda Aioanei

Contents

Preliminary phase	40
Step 1. Identifying the requirements for the character	40
Personnel Character requirements	40
Generic Character requirements	40
Step 2. Gathering references	40
Step 3. Setting up the source files folder	41
Production phase	42
Step 4: High resolution mesh	42
Step 5: Exporting the mesh and texture maps	46
Step 6: Optimization for in-app use	48
Restrictions to follow:	48
Step 7: Skinning and weight painting	55
Implementation phase	56
Step 8: Folder structure and prefab files setup	56
Step 9: Unigine implementation	57
Creating the nodes	59
Adding textures	59
Creating the impostor	59
Setting up the LODs	62
Step 10: RS editor setup	62
Prefabinfo and Skincfg files	62
Adjusting the properties	64
Creating a correct thumbnail	67
Step 11: Testing	67

Preliminary phase

Step 1. Identifying the requirements for the character

Information about why and what the character is needed for.

Example: A Lithuanian policeman character is needed. The character needs to have the options of equipping and unequipping a certain vest, cap, and belt. There are two variations of uniforms for the character.

Normally there are two character types:

Personnel Character requirements

The characters with designated jobs. They are the focus of the application.

- Specific uniform according to references or client request
- Male/Female versions
- Will be used as trainee characters
- Extra equipment that can be toggled on and off in RS

Generic Character requirements

Characters with no distinguishable roles – civilians.

- Generic clothing
- Can be used by trainees or for crowds/populating areas
- Focus on ethnicity/body type/gender

After identifying what type the character belongs to, particularities can be drawn out – which equipment is needed, whether a character is expected to be a certain gender or ethnicity.

Step 2. Gathering references

References are important to make sure the character looks and feels as close to reality as possible.

While a generic character might not need a specific reference, it is important for personnel characters to be an accurate portrayal of reality - their uniforms, equipment and provenance must be authentic. This requires a detailed set of references.

Example: Australian Paramedic



Pay attention to small details, they are important to accurately portray country-specific personnel.

Double check your references and sources with leads and requirements.

Step 3. Setting up the source files folder

This part is important to make sure backtracking to source files is an easy process and important files are not lost. Source files should be stored in a folder named after the character, in this root folder: **Z:\3D Models\Sorted\Characters** . **Follow the folder path according to the character you create.**

The folder structure for a character's source files should follow the template that can be found here: **Z:\3D Models\Sorted\Characters\Templates\Source Files Template**

Z:\3D Models\Sorted\Characters_Sorted\Generic

- Generic
 - Business
 - Casual
 - Female
 - Male
 - Sportswear01
 - 3DCoat
 - Character Creation
 - Substance Painter
 - ZBrush
 - Reference
 - RS5 Exports
 - Sportswear01_001.max
 - WFSportswear01_A_2048.psd

Notes:

- The source files for Zbrush and Substance Painter scenes should be kept to 3 versions maximum. These source files need be archived at the end of the task – this helps preserve haddisk space.
- Folders for software that is not used should be deleted from the character folder.
- Name the source files in an understandable, clear manner.

Production phase

The production phase starts with creating the high poly mesh in *Character Creator 3*.

Firstly, you must have the software and its content packs installed: *Character Creator 3*, *SkinGen* plugin and the content packs available in the *Smart Gallery*.

You can open the *Smart Gallery* from the *Plugins* tab on the upper toolbar.

Step 4: High resolution mesh

In CC3, it is important to respect certain parameters, to ensure that fitting the CC3 character with the default RS rigs will not be an issue.

Height Restrictions:

Women (Teenager, Adult, Elderly) – 1.7 m

Men (Teenager, Adult, Elderly) – 1.82 m

Children – 1.13 m

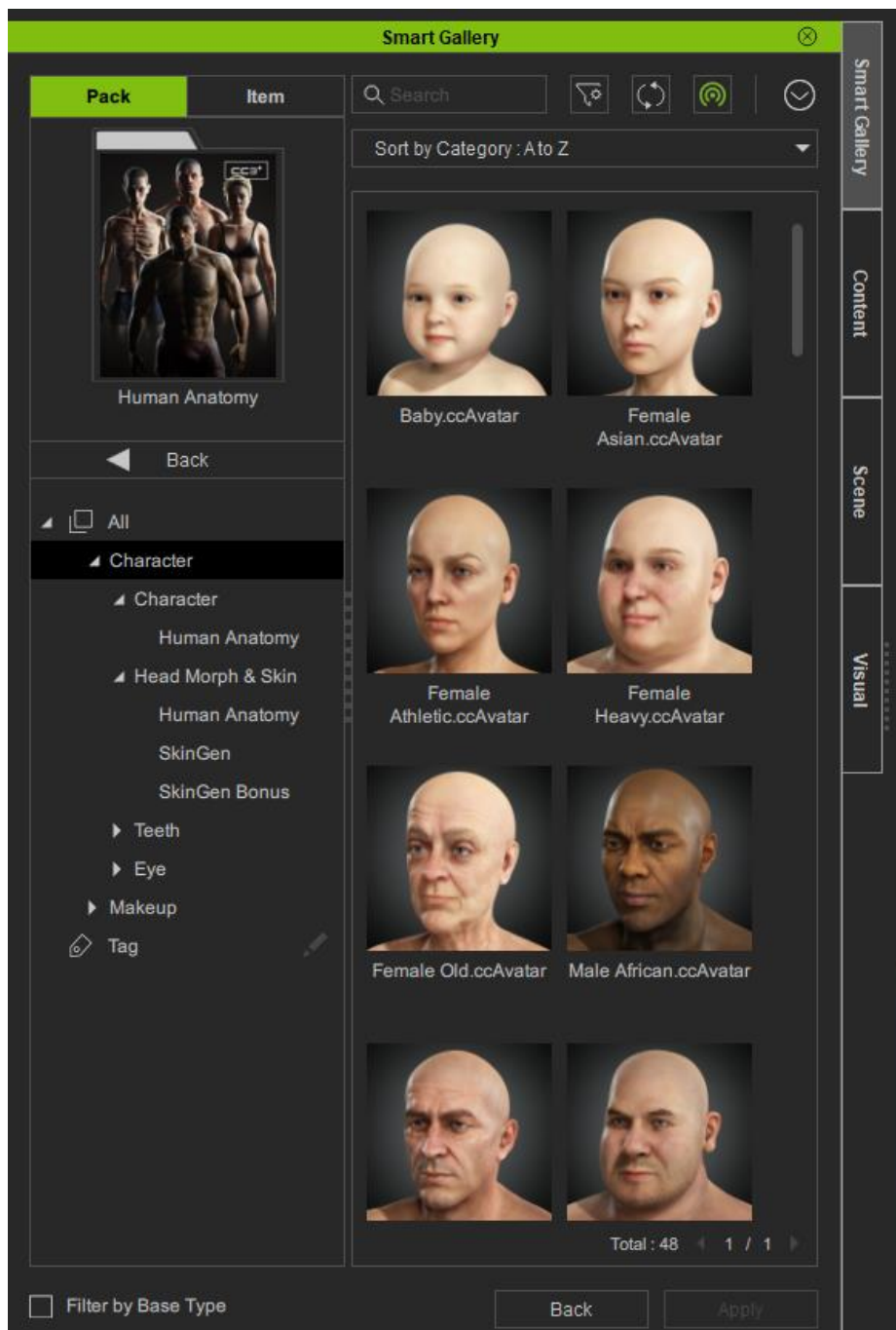
These restrictions can be overridden only by little amounts, as long as the mesh still functions well with the default rigs.

The characters can also have a bigger or smaller body weight than the average, but this must also be tested thoroughly with the rig.

When opening CC3, a female mesh will be set by default. From here, you can start adjusting the features of the character and create a unique person.

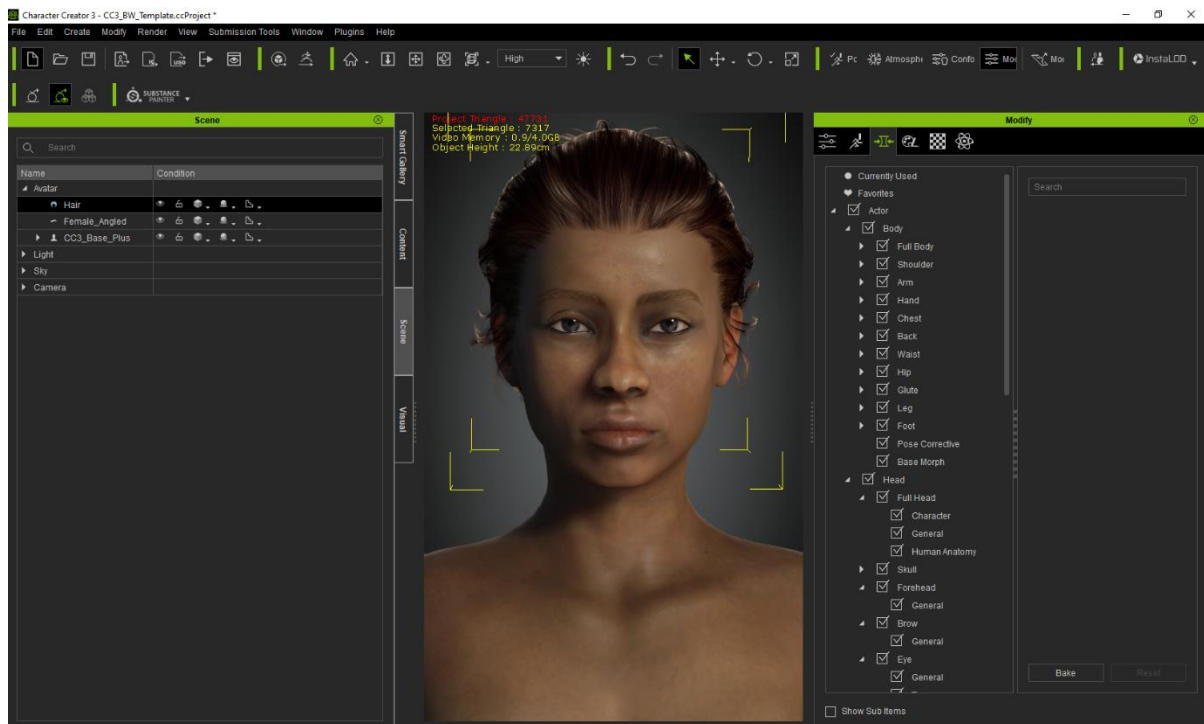
To start from a base of a different ethnicity or gender, go to the *Content* tab on the left window, *Project* and switch from *Template* to *Custom*, open the *Avatar* tab and go into the *Base* folder.

Templates can also be accessed from the *Smart Gallery*, by opening the *Human Anatomy* pack.



To correctly portray unique features and ethnicity, it can be difficult on its own. Use references – on <https://www.3d.sk/> you can find turnaround photo portraits of diverse people.

In the **Morphs** tab you can find the sliders for controlling specific features of the character.



In the **Content** tab, you can select and apply hair, clothes, makeup and other particularities to the character.

The in-depth options for detailing skin are available in the **Appearance** tab.

The meshes can also be adjusted or modified in CC3. In the **Attribute** tab, the **Modify** options can be found.

The **Conform** option is useful for making meshes fit correctly around the body. When applying a piece of clothing that does not fit the body mesh, select the object from the **Scene** tab, go to the **Conform** option and press **Calculate Collision**. CC3 will automatically correct the mesh to fit the body.

The mesh vertices and faces can also be adjusted manually from the **Edit Mesh** option.

Another important step is eliminating the parts of the body mesh which are not visible and therefore not needed. To do this, select a clothing article, then in the **Modify** tab select the **Hide Body Mesh Tool**. This will open a new window which will allow you to hide parts of the mesh which are not necessary.

Materials can also be adjusted or changed from the **Material** tab. Texture maps can be swapped with outside or owned textures. You can change the colour of hair, skintone and any other of the meshes.

Reallusion provides many tutorials for any of the CC3 options that are available on their website and on their YouTube channel:

<https://www.reallusion.com/character-creator/tutorial.html>

<https://www.youtube.com/c/reallusion/videos>

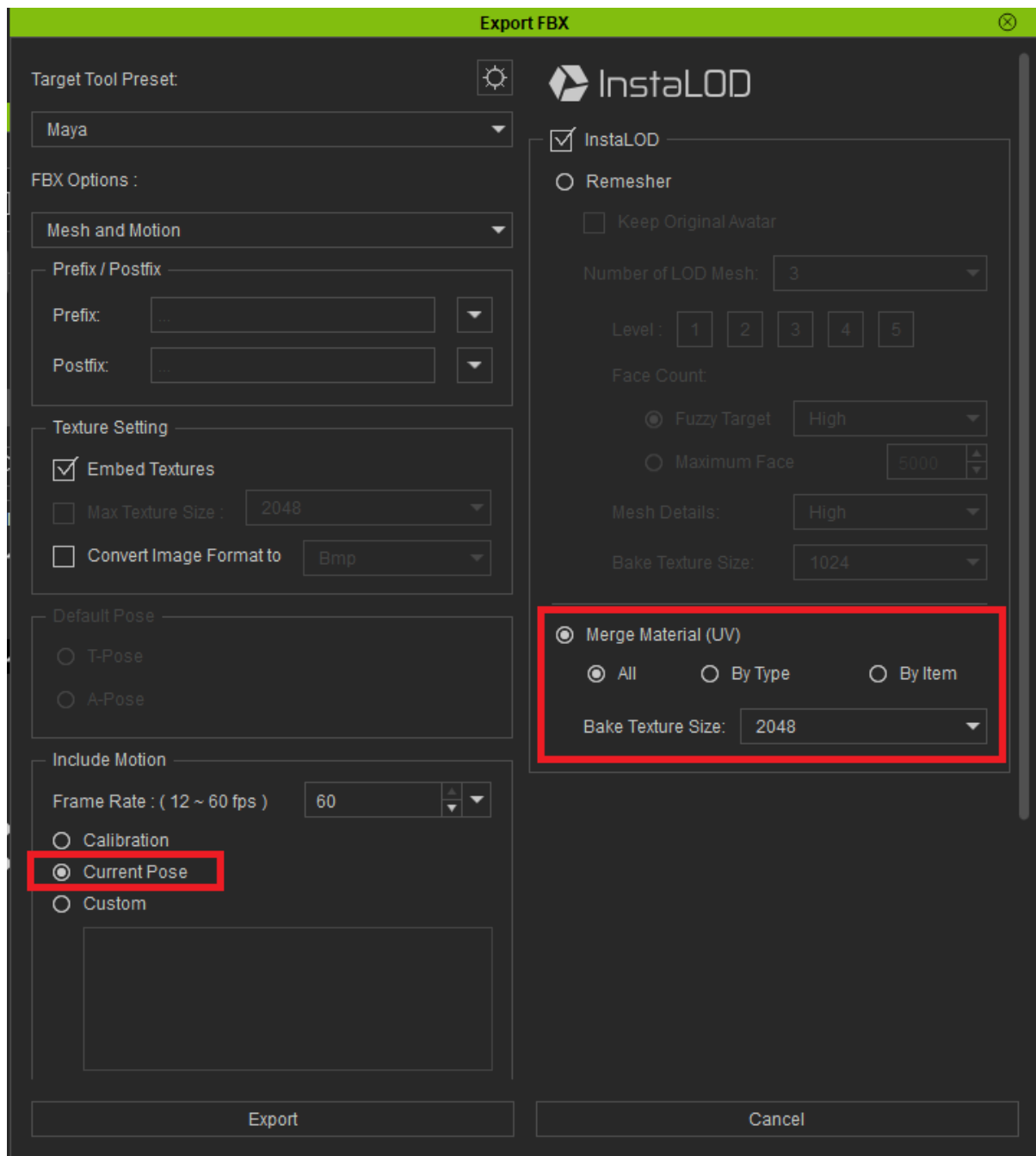
Step 5: Exporting the mesh and texture maps

When exporting the finished character, it is important to choose an exporting pose that is close to the pose of the RS default rig, which will help with skinning the mesh to a rig.

For that, go into the Content tab to the Pose section, choose the Motion folder from the Motion Template and choose the **_OpenA_UE4**. This pose is the closest to the default pose of the RS rig.



After this, you can export the mesh as an FBX. In the exporter window, choose the following options:



The materials can be exported as all merged or separate, and then merged together later, in 3D Coat (Tutorial from Kenny attached).

Unigine materials can only read three texture maps. Because of this, the opacity maps (if necessary) must be combined with the albedo maps, and the metallic maps must be combined with the roughness maps. This can be done in Photoshop.

Albedo + Opacity map:

Open the two maps in Photoshop. On the Albedo map file, add a layer mask to the layer with the Albedo texture. After that, move to the Opacity map file, select the entire canvas and copy. Return to the Albedo map file, hold the Alt key on your keyboard and click on the layer mask – this will ensure that you are modifying the layer mask. After that, Control + V to paste the copied Opacity map into the layer mask. Click on the eye icon next to the layer to see the result.

Export the resulting map as an Albedo map.

Metallic + Roughness map:

After opening the texture map in Photoshop, create a new folder. Set it to screen and add two more layers inside. The top layer inside the metallic folder must have the color RGB Red and the bottom layer 100% black and set them both to multiply. Do the same for the roughness folder but instead of a red layer add a RGB Green layer. In between the Red + Black and Green + Black layers you can add layers inside with grey values. White values result in 100% metallic and reflection or 100% roughness.

Export the resulting map as a PBR texture map.

Step 6: Optimization for in-app use

To bring the mesh down to a resolution that is optimal for in-app use, we will be using 3dsMax's optimization tools.

Restrictions to follow:

Personnel characters: up to 40k polygons

Low-poly personnel: 6-8k

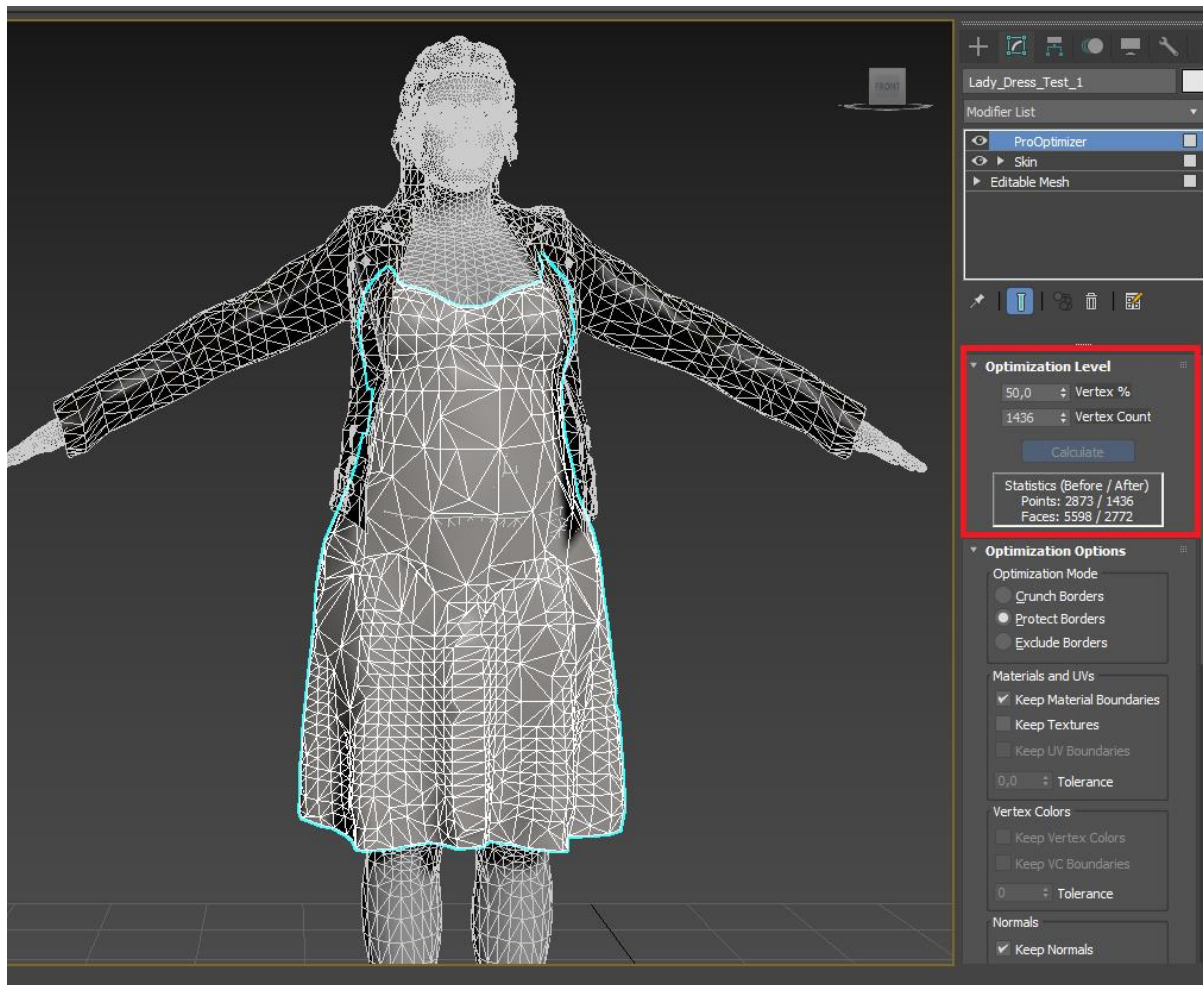
Generic characters: 34-36k polygons

Low-poly generic: 4-7k

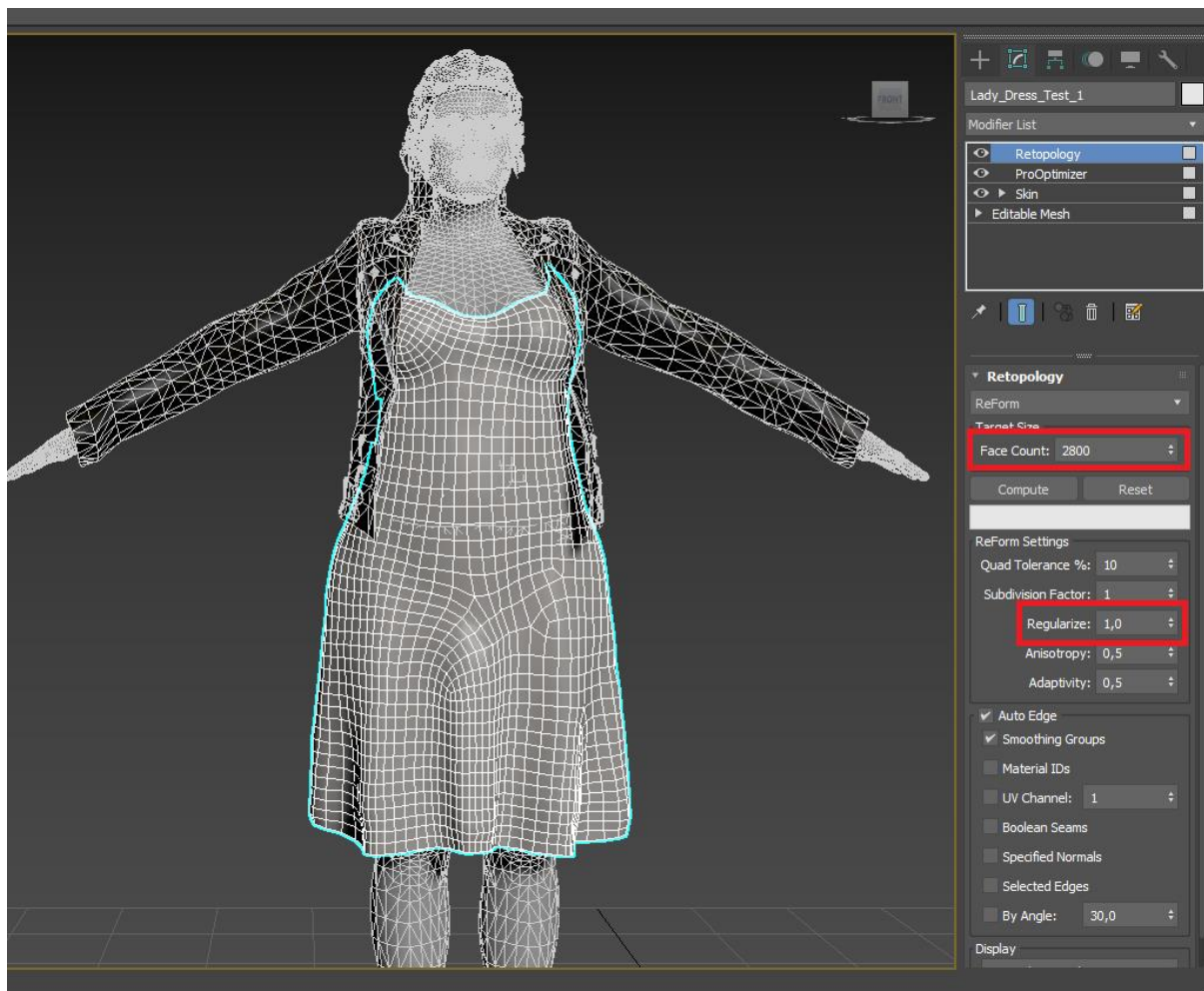
To do that, import the CC3 export into 3dsMax. Duplicate the meshes to keep a high-resolution version in the scene.

For each of the mesh parts, in the modifier tab, choose the **ProOptimizer** tool.

From here you can change the **Vertex %** to test lower resolutions of the mesh.



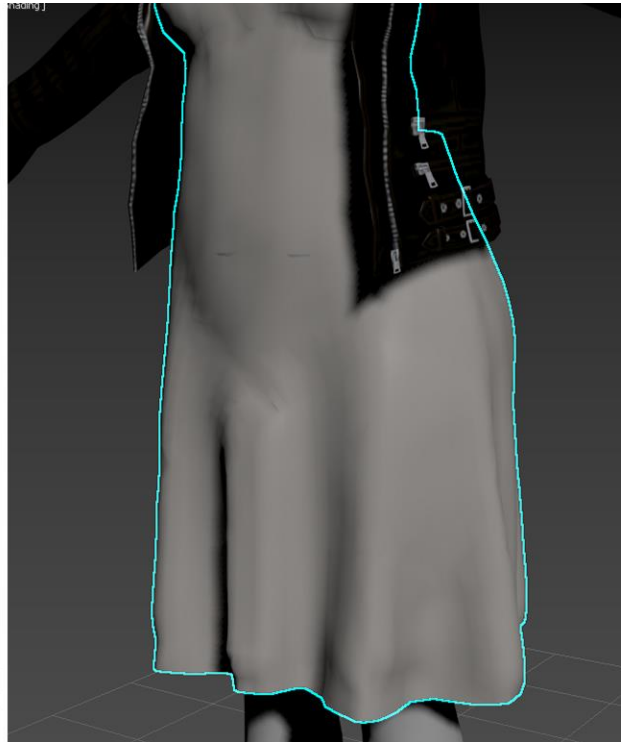
After this, in order to clean up the mesh issues and topology, choose the **Retopology** tool in the modifier tab. Here, make sure to set the **Face Count** to a number close to the one of the previous modification, and set the **Regularize** tab to 1.



Check the wireframe and default shaded mesh for issues that could require a higher face count.



Mesh after *ProOptimizer*

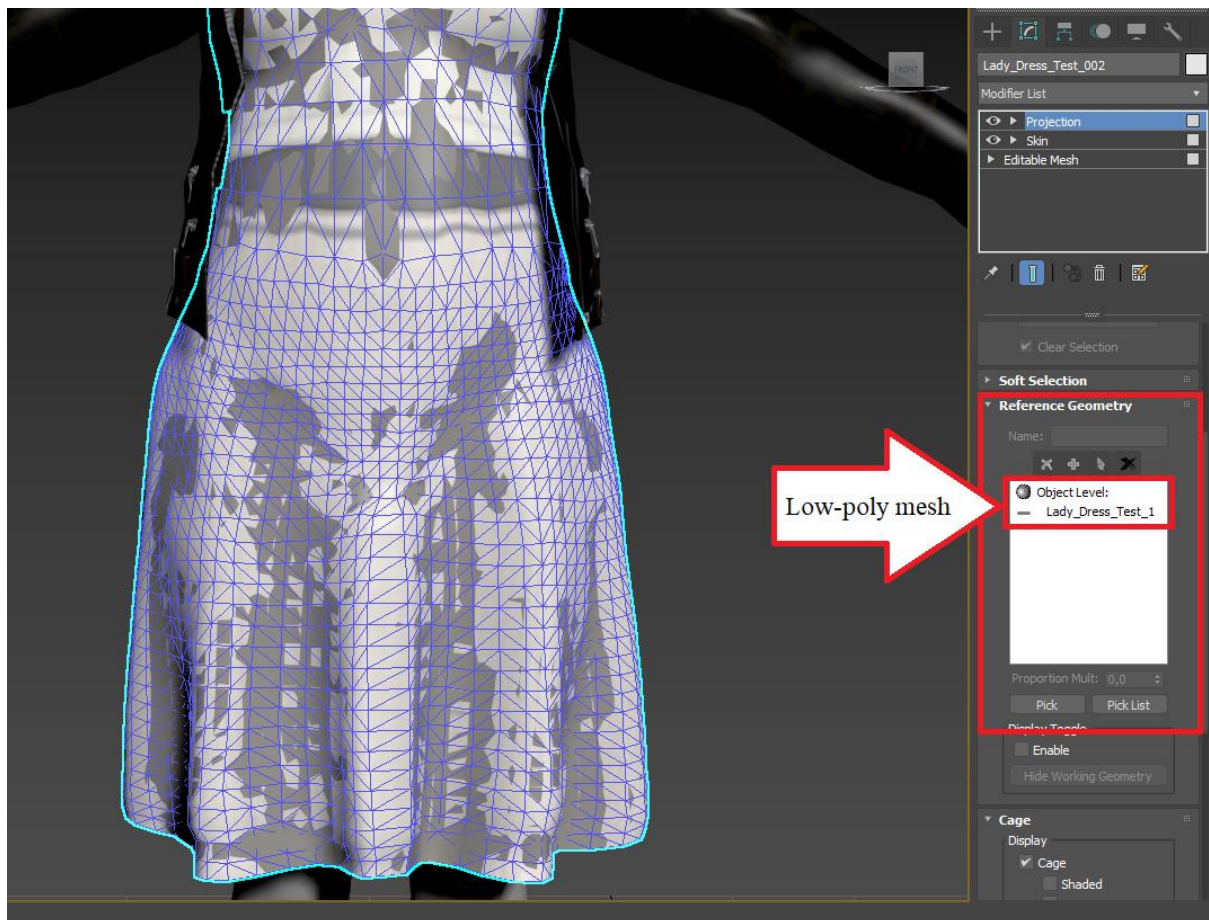


Mesh after *Retopology*

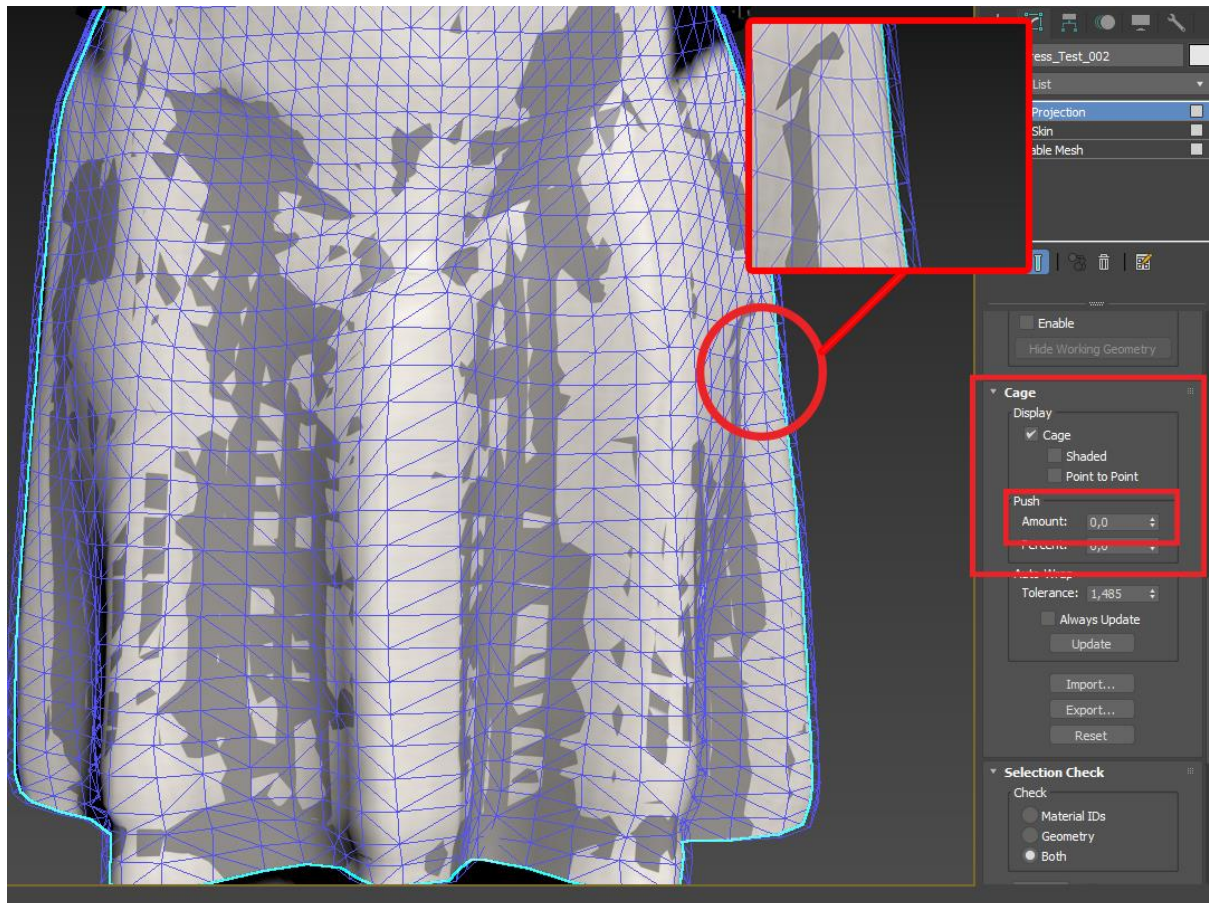
After doing this, the UVs of the high poly mesh will have to be projected onto the optimized mesh. This way, the textures generated by CC3 can be used with the optimised mesh.

To do this, first select the high poly mesh object and choose the **Projection** modifier.

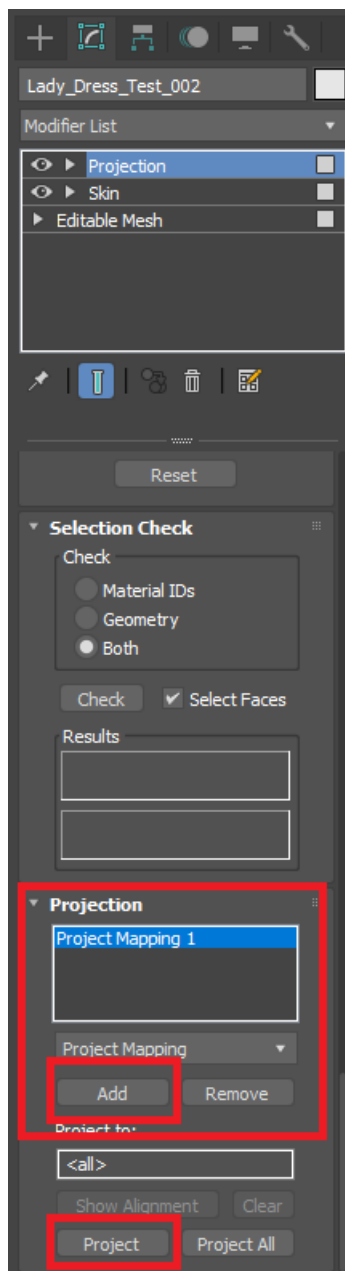
In the **Reference Geometry** tab, press the **Pick** option underneath the white table and select the low poly mesh. It will be added to the table.



Then, in the **Cage** tab, slightly increase the **Amount** – this can be noticed in the viewport, as the purple mesh will increase in size. Make sure the purple mesh overrides the low poly mesh only slightly.



Lastly, in the **Projection** tab, press **Add** to create a new projection. Then, press the **Project** button underneath.

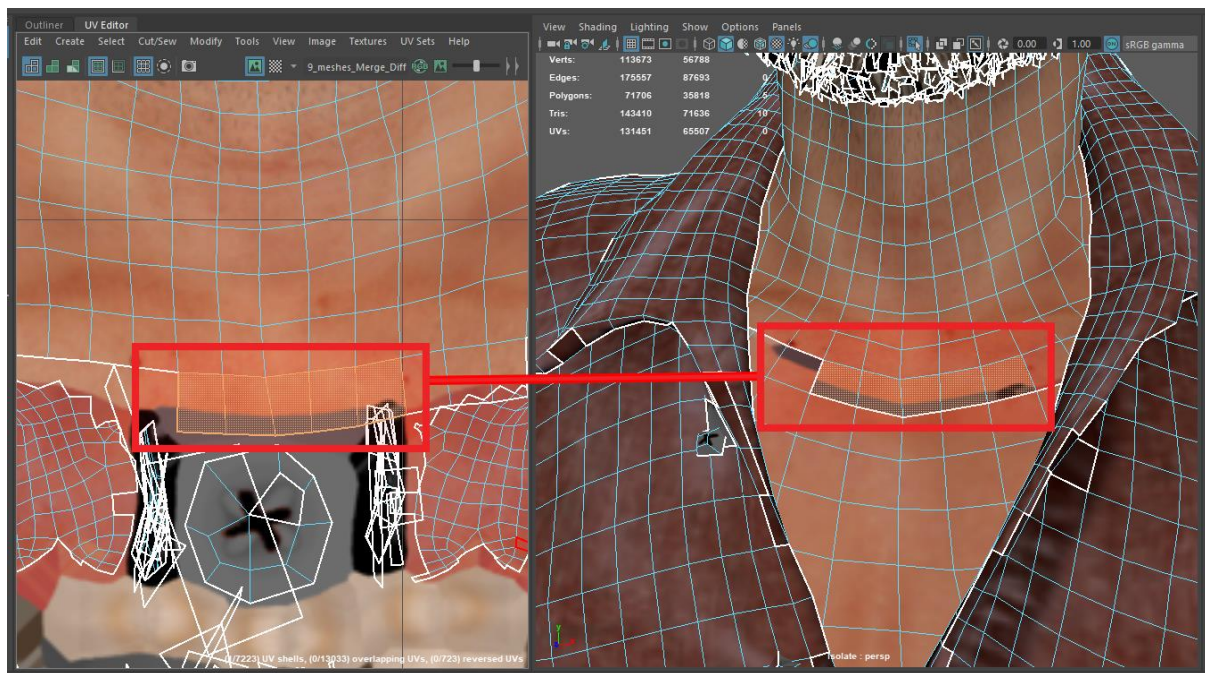


As a result, the low poly mesh should now have the same textures as the high poly mesh.

However, the projection modifier could cause issues at times. In these cases, the issues can be fixed manually, by adding more edges or edge loops where needed, or by moving the faces of the UV unwrap which are positioned incorrectly or too close to the edge of the texture.

This is the stage where these small manual adjustments can be done to optimize the mesh. Details like shirt pockets, buttons, small clothing parts can be eliminated or integrated in the mesh to lower the polycount.

Example:



On this mesh, the edges of the seam are positioned too low on the UV map, making the wrong texture show up on the skin area. To fix this, the rectangles that are too big can be slightly adjusted to not override the texture edge.

Repeat this process to also create the lowest LOD necessary for characters (6-8k polygons).

After these steps, the separate mesh parts can be combined into a single mesh.

This mesh can be exported to be bound to the rig.

Step 7: Skinning and weight painting

The next step focuses on binding the mesh to a rig, in order for the animations of RS to function with the new character.

To do this, import the combined mesh into the master scene corresponding to the gender and age of the character (male/female and adult/child). These master scenes can be found in this folder: **Z:\3D Models\Sorted\Characters_Master**

Save the scene in the correct source files folder.

This process can be done in both 3dsMax or Maya. There are video tutorials for both of the options, which you can access in the Resources folder.

Implementation phase

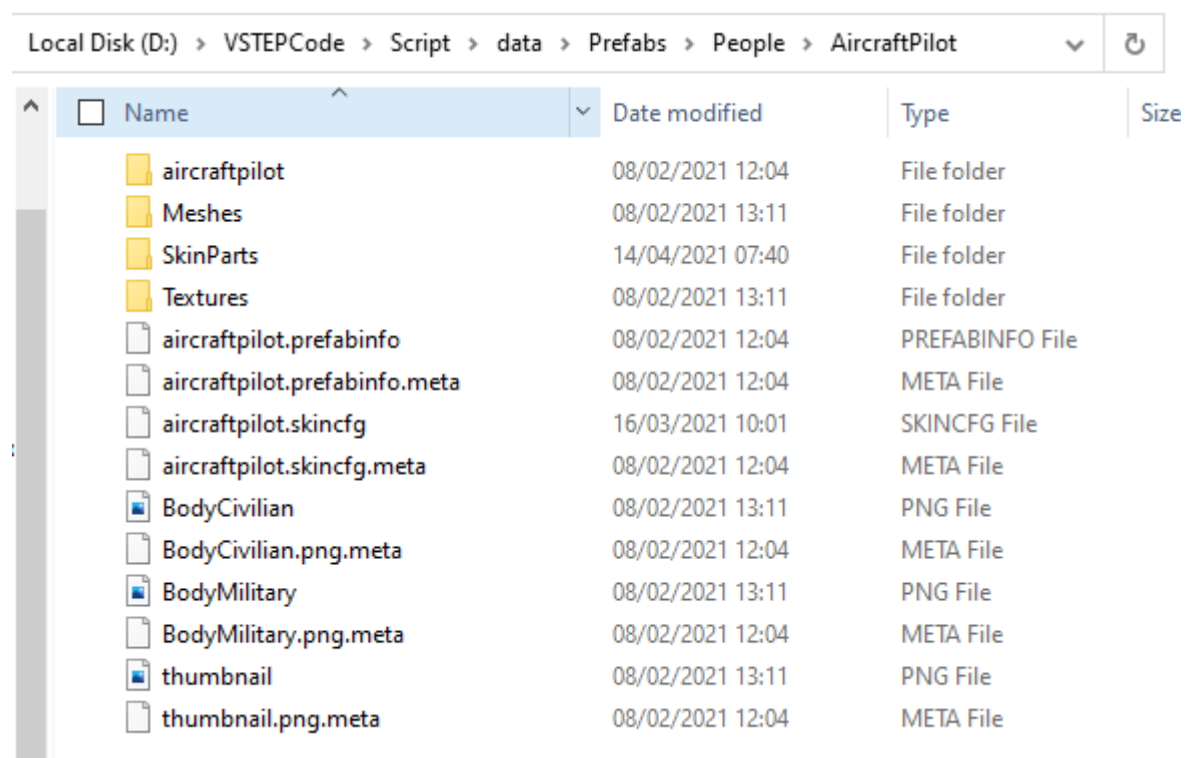
This phase focuses on taking the finished asset into the Unigine engine and doing the final adjustments needed to fully implement it in RS.

Step 8: Folder structure and prefab files setup

Character assets have a specific folder structure and configuration files. These folders create a *Prefab*, which is the entirety of one asset in the RS application. Character prefabs are stored in: **D:\VSTEPCode\Script\data\Prefabs\People**

A new character must have a newly created prefab folder. Generic characters are stored in the same prefab folders, so for a new character you must only add the .mesh, texture maps, new material, thumbnail and configured node to the existing folder.

A template folder for the character setup in the engine can be found here: **Z:\3D Models\Sorted\Characters\Templates**



The screenshot shows a Windows File Explorer window with the address bar set to 'Local Disk (D:) > VSTEPCode > Script > data > Prefabs > People > AircraftPilot'. The main pane displays a list of files and folders. The 'Name' column is selected, and the list includes folders for 'aircraftpilot', 'Meshes', 'SkinParts', and 'Textures'. It also lists several configuration files with extensions like '.prefabinfo', '.prefabinfo.meta', '.skincfg', and '.skincfg.meta', as well as image files like 'BodyCivilian.png', 'BodyMilitary.png', and 'thumbnail.png'.

Name	Date modified	Type	Size
aircraftpilot	08/02/2021 12:04	File folder	
Meshes	08/02/2021 13:11	File folder	
SkinParts	14/04/2021 07:40	File folder	
Textures	08/02/2021 13:11	File folder	
aircraftpilot.prefabinfo	08/02/2021 12:04	PREFABINFO File	
aircraftpilot.prefabinfo.meta	08/02/2021 12:04	META File	
aircraftpilot.skincfg	16/03/2021 10:01	SKINCFG File	
aircraftpilot.skincfg.meta	08/02/2021 12:04	META File	
BodyCivilian	08/02/2021 13:11	PNG File	
BodyCivilian.png.meta	08/02/2021 12:04	META File	
BodyMilitary	08/02/2021 13:11	PNG File	
BodyMilitary.png.meta	08/02/2021 12:04	META File	
thumbnail	08/02/2021 13:11	PNG File	
thumbnail.png.meta	08/02/2021 12:04	META File	

Checklist:

- The ***Meshes*** folder contains all of the ***.mesh*** files.
- The ***SkinParts*** folder contains all the nodes that reference the meshes and used materials.
- The folder named after the root folder contains all the materials.
- The ***Textures*** folder contains all the texture maps.
- The ***prefabinfo*** and ***skincfg*** files must be named exactly the same as the root folder, but all characters of the filename should be in lower case.
- The ***thumbnail.png*** file is needed in the folder, but the actual thumbnail will be an additional ***.png*** file named exactly the same as the node (the character) which it belongs to.
- Every file in the folder needs to have a ***.meta*** file specific to them.

The template folder can be copy-pasted into the repository, but the actual assets should be added through the Unigine engine. This ensures that the ***.meta*** files are generated correctly.

.meta files are automatically generated by Unigine with a specific ID per file:

- By dropping the assets in the correct folder in Unigine
- By opening Unigine after storing the assets in the repository folders

If any changes are made to a file outside of Unigine when it already has a *.meta* file generated, make sure to delete the *.meta* file and reopen Unigine to create a new *.meta* file.

In an asset prefab folder, more than one character can be stored, as long as they are connected in a relevant way.

Example:

The *AircraftPilot* contains two different character types, a pilot dressed in a civilian outfit and a pilot dressed in a military outfit. The two variations can be switched between from the properties window.

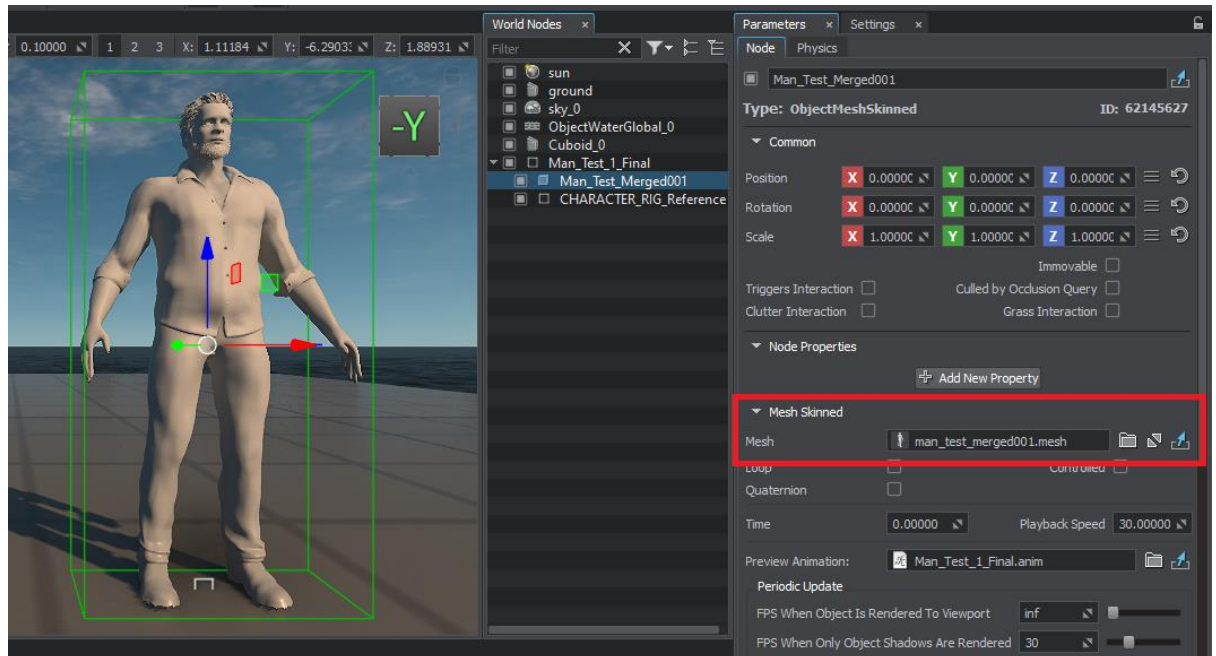
The generic characters are all stored into the same prefab – the client can place the prefab in the scene and then switch to a different character from the properties window.

Step 9: Unigine implementation

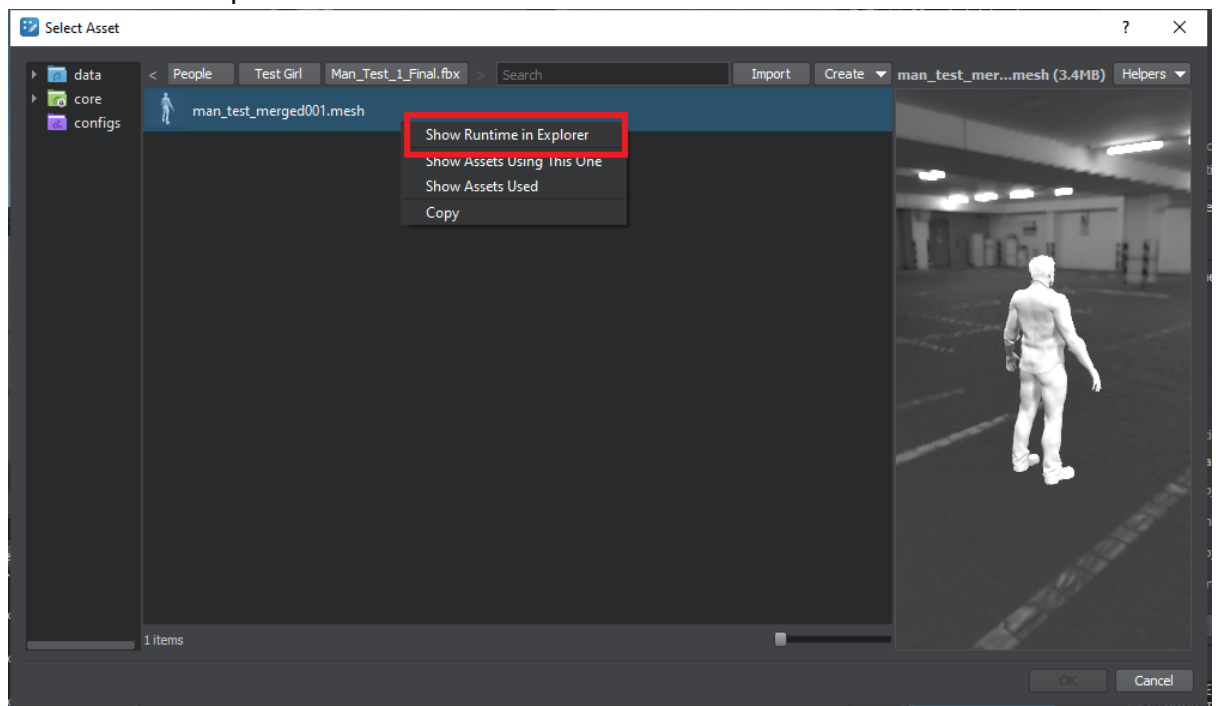
To fully implement the character in Unigine, the files in the prefab folder must be configured in a certain way.

First of all, any FBX files must be converted to ***.mesh*** type files. The ***.mesh*** file type can be generated in two ways:

1. FBX meshes can be exported directly as **.mesh** UnigineMesh export file from the 3dsMax 2017 version.
2. After importing a FBX in Unigine, place the object in the scene. In the Nodes browser, dropdown the FBX node and select the actual mesh. In the **Parameters** window, drop down the **Mesh Skinned** tab. There, in the **Mesh input** panel, the **.mesh** file can be located.



Press **Select Asset**, represented by a folder icon right next to the input panel, and a new tab will be opened with a **.mesh** file.



Right-click on the **.mesh** file and press **Show Runtime in Explorer**.

A runtimes folder will be opened, with a **.mesh** file selected, named with a sequence of numbers and letters combined.

Copy the **.mesh** file into the prefab meshes folder and rename it with the proper naming conventions. After this, delete the FBX in the folder.

Creating the nodes

The RS prefab will not, however, use the **.mesh** file as is. You will need to create a node that houses the mesh file.

To do that, place the mesh in the scene, select it, then go to the **Edit** tab on the upper left part of the screen, and click **Convert to Node Reference**. This will automatically create a node named after the mesh, in the **Meshes** folder. Move the newly created node to the **SkinParts** folder.

Every character component, such as a different body mesh or equipment that can be toggled on and off, needs to have its own separate node.

Adding textures

To add the textures to the mesh, firstly, place the texture maps into the **Textures** folder. Then, in the folder named after the root folder, right click and create a new material – press **Create Child** while having the default option, **mesh_base**, selected. This will be the material of the character. Rename it according to naming conventions.

When pressing on the material, there are numerous tabs for manipulating material properties. In the **Common** tab, let the **Preset** on **Opaque** if the material has no transparency. If there is transparency needed, select the **Alpha Test** option. On the **Textures** tab, you can assign the texture maps to the material.

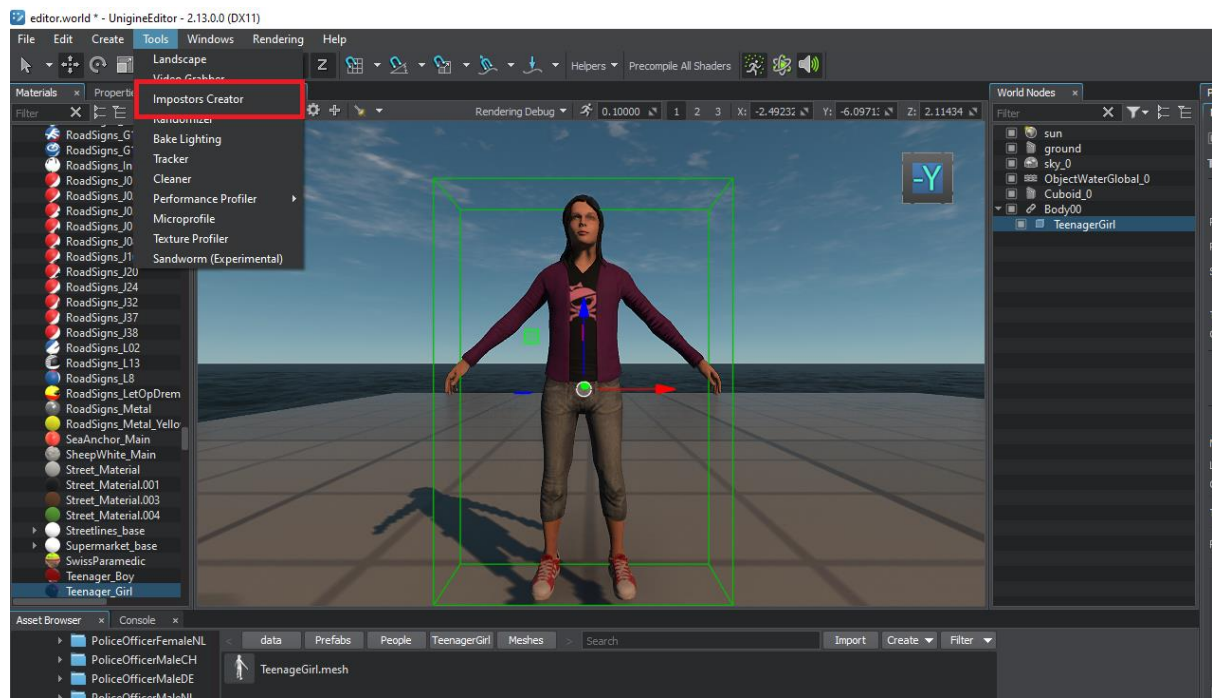
After finishing setting up the material, drag and drop it onto the mesh in the viewport, or assign it manually by selecting the mesh and going to the **Surfaces** tab.

Creating the impostor

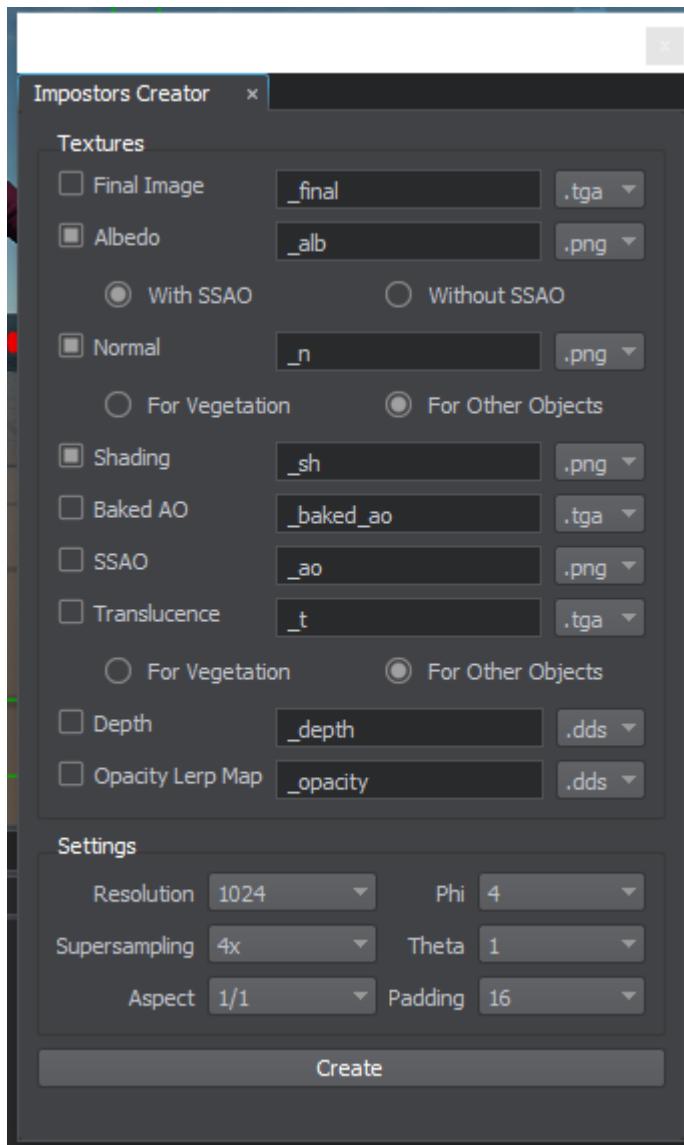
Impostors are the lowest LOD possible for a character prefab. They are 2D projections of the model, seen from distance.

This LOD is created in the Unigine editor through a few simple steps.

First, make sure you have the node with the textured character in the scene. After selecting it and pressing **Edit**, to be able to adjust the node, go to the **Tools** tab situated on the upper left side of the screen. There, you will find the **Impostor Creator**.



The options must be set up in the following way:

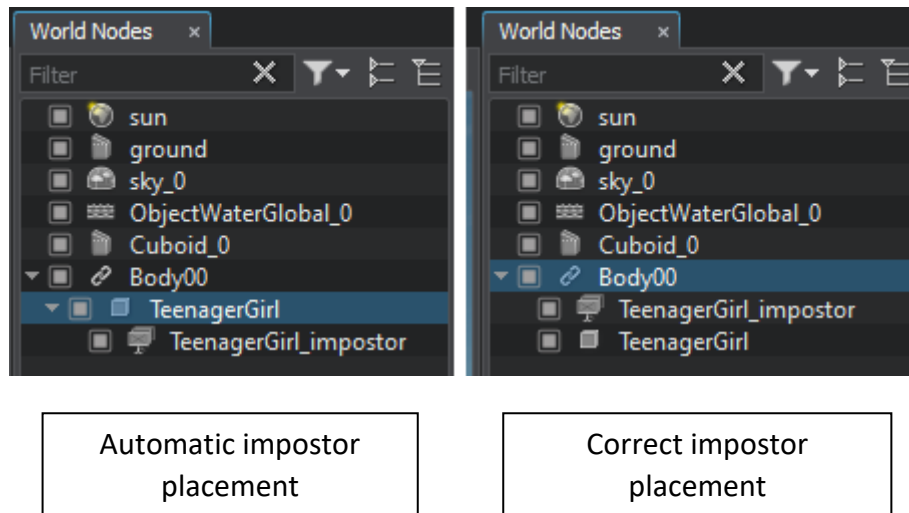


Only the *Albedo*, *Normal* and *Shading* maps are necessary, as PNG file types. The resolution must be changed to 1024, the Phi to 4 and the Theta to 1.

After pressing *Create*, a selection window will open, for which you need to choose the *.mesh* file of the character whose impostor is being created.

The textures and material of the impostor will be generated in the ***Meshes*** folder. They need to be moved to their respective folders and renamed according to naming conventions.

The impostor node will be automatically added underneath the character's mesh node in the node hierarchy. To make sure the impostor is visible, move the impostor into the same root node as the character mesh.



Setting up the LODs

The LOD visibility can be set up from the **Surfaces** tab.

With the node in the scene, go into Edit mode and select the mesh. On the Surfaces tab, you can find the LOD category, which gives the options of Min Visibility and Max Visibility.

The LOD visibility settings must be configured as following:

LOD1 – the high-res mesh

Min Visibility: -inf Max Visibility: 5.0

LOD2 – the low-res mesh

Min Visibility: 5.0 Max Visibility: 45.0

Impostor – the billboard projection

Min Visibility: 40.0 Max Visibility: inf

After setting the LODs up, apply the changes to the root node to save the changes.

Step 10: RS editor setup

To make sure that the character will function correctly and fulfils the requirements, you must make sure that the **.prefabinfo** and **.skincfg** files are configured in a specific way.

Prefabinfo and Skincfg files

The **.prefabinfo** file is important for licensing purposes – this way, certain characters can be allocated to specific packages bought by clients.

By opening the template prefabinfo file with *Notepad++*, the code can be edited. In the file, marked with a green comment, is the line that needs to be edited to add the license name.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <info>
3    <objects>
4      <object name="info" type="Prefabs::People::Info" key="3">
5        <variables class="Prefabs::PhysInfoBase">
6          <_bbMin type="vec3">-0.20000000298023224 -0.300000001192092896 0</_bbMin>
7          <_bbMax type="vec3">0.20000000298023224 0.300000001192092896 1.7999999523162842</_bbMax>
8          <_magneticDeviation type="float">0</_magneticDeviation>
9        </variables>
10       <variables class="Prefabs::InfoBase">
11         <_hidden type="int">1</_hidden>
12         <_tagsList type="Tags::List" dynamictype="Tags::List">
13           <object key="1">
14             <variables class="DigIt::KeyList">
15               <_data type="vector" valuetype="string">
16                 <item></item> <!-- Name of license packages -->
17               </_data>
18             </variables>
19           </object>
20         </_tagsList>
21         <_externBase type="Prefabs::Info" dynamictype="Prefabs::Info">
22           <object key="2">
23             <variables class="Prefabs::Info">
24               <_name type="string"></_name>
25               <_desc type="string"/>
26               <_tags type="vector" valuetype="string"/>
27             </variables>
28           </object>
29         </_externBase>
30       </variables>
31     </object>
32   </objects>
33 </info>
34

```

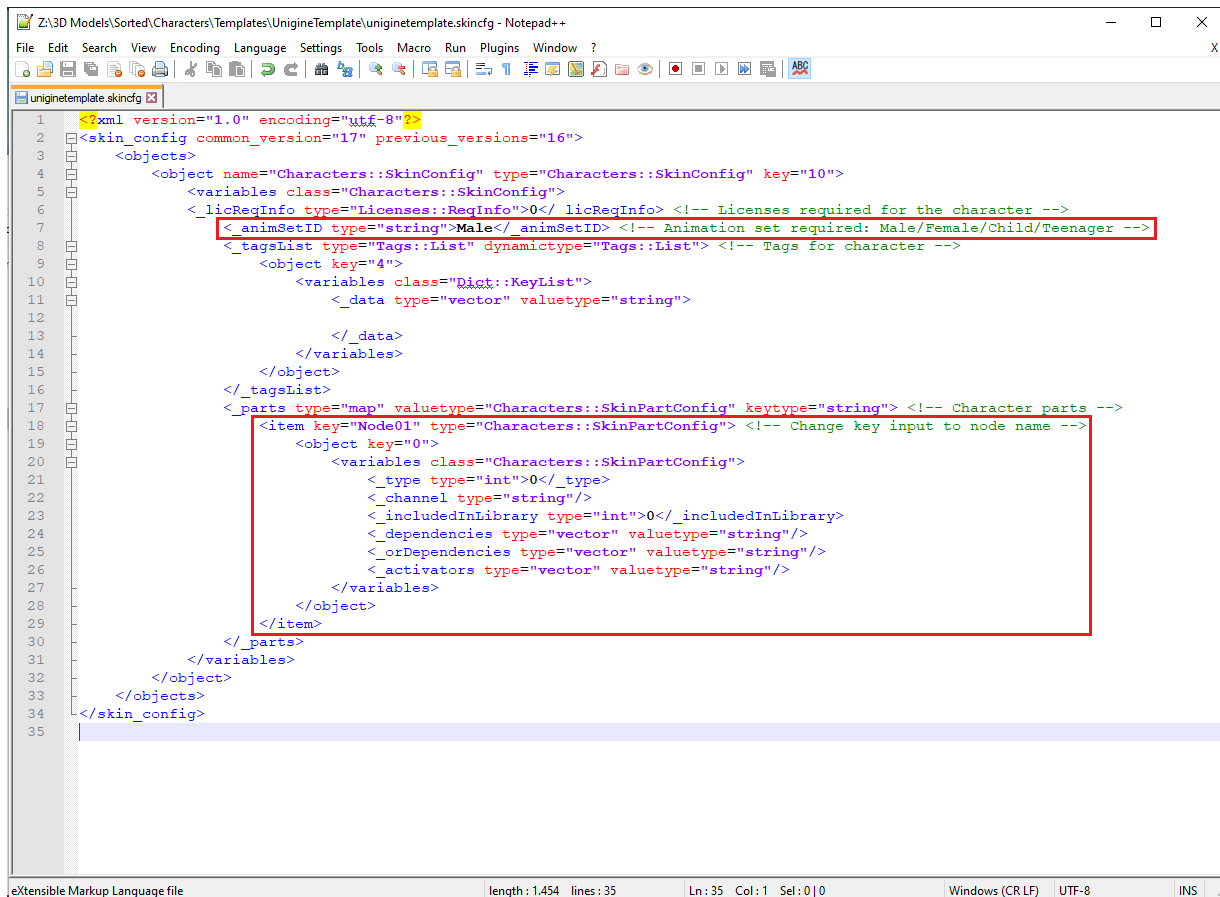
eXtensible Markup Language file length: 1,155 lines: 34 Ln: 34 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

Example: <item>Aircraft</item>

For any additional licences, another line must be added with the same format.

The **.skincfg** file contains the character's configuration in RS. By opening the file in *Notepad++*, you can see annotated the important parts of the code. Two adjustments will need to be made here:

- Change the animation set required to the one that fits the character:
Male/Female/Child/Teenager.
- In the **Character parts** section, a new **item** needs to be added for every node of the prefab.
To do that, copy and paste the code in between the <item></item> markers.
Make sure to rename the key component of each node to the actual name of the node.

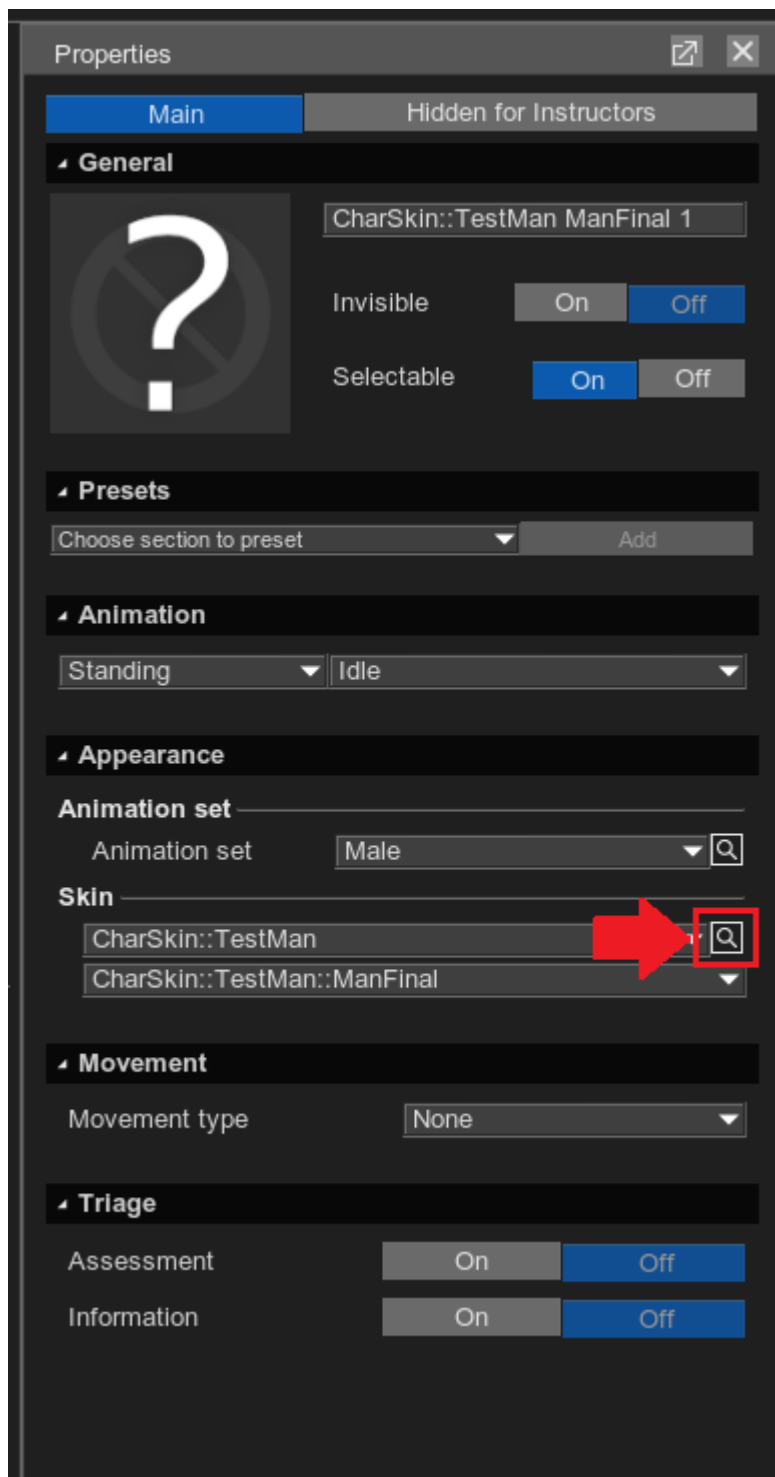


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <skin_config common_version="17" previous_versions="16">
3   <objects>
4     <object name="Characters::SkinConfig" type="Characters::SkinConfig" key="10">
5       <variables class="Characters::SkinConfig">
6         <_licReqInfo type="Licenses::ReqInfo">0</_licReqInfo> <!-- Licenses required for the character -->
7         <_animSetID type="string">Male</_animSetID> <!-- Animation set required: Male/Female/Child/Teenager -->
8         <_tagsList type="Tags::List" dynamicType="Tags::List"> <!-- Tags for character -->
9           <object key="4">
10             <variables class="Dict::KeyList">
11               <_data type="vector" valueType="string">
12                 </_data>
13             </variables>
14           </object>
15         </_tagsList>
16       </_parts type="map" valueType="Characters::SkinPartConfig" keytype="string"> <!-- Character parts -->
17         <item key="Node01" type="Characters::SkinPartConfig"> <!-- Change key input to node name -->
18           <object key="0">
19             <variables class="Characters::SkinPartConfig">
20               <_type type="int">0</_type>
21               <_channel type="string"/>
22               <_includedInLibrary type="int">0</_includedInLibrary>
23               <_dependencies type="vector" valueType="string"/>
24               <_orDependencies type="vector" valueType="string"/>
25               <_activators type="vector" valueType="string"/>
26             </variables>
27           </object>
28         </item>
29       </_parts>
30     </_variables>
31   </object>
32 </objects>
33 </skin_config>
34
35
```

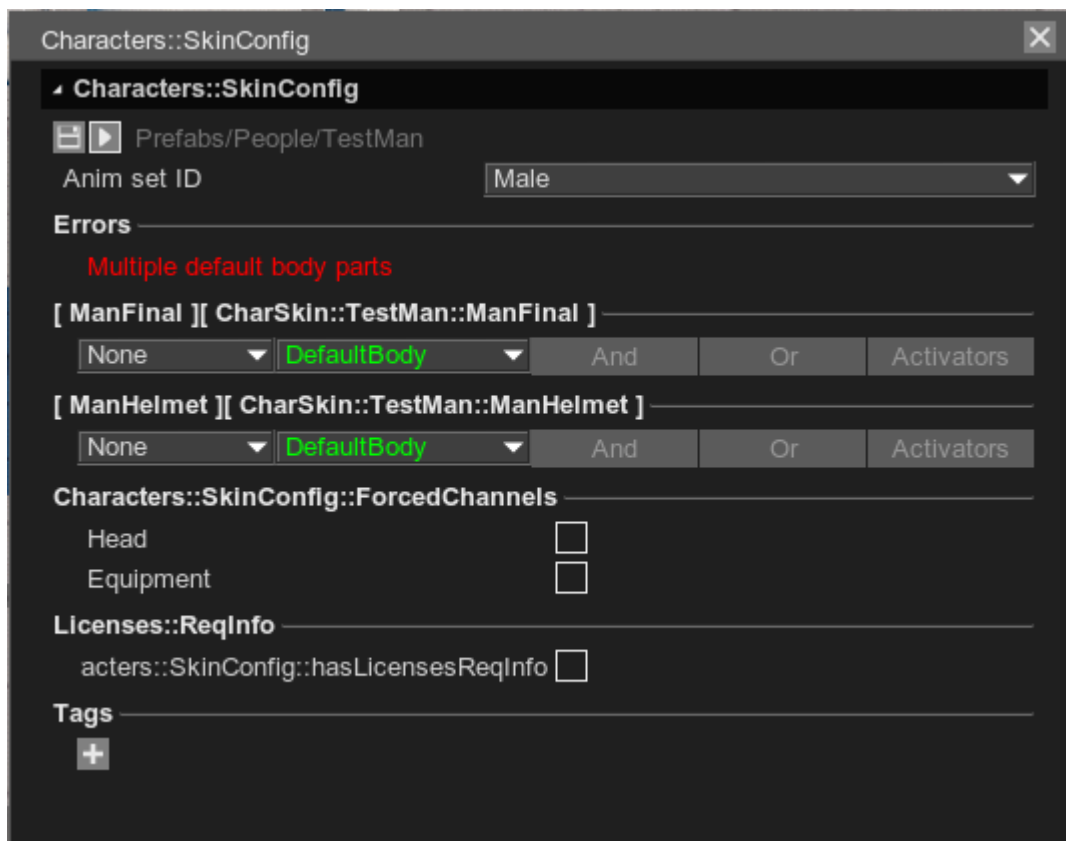
Adjusting the properties

With these two configuration files set up, you now should be able to view the character in the *RS Objects Editor*.

When searching for the new prefab in the library, you will notice that all the components of the prefab are exposed. To change this, place one of the components in the scene and open the properties window. There, next to the **Skin** tab, is a small icon with a magnifying glass.



By pressing that icon, a new window will appear with additional configuration options.



First, you must choose one of the nodes as the **default body**. This will be the mesh that will appear by default when placing the prefab in the scene. The other components of the prefab need to be set to options different than **DefaultBody**:

Body – this is an option for different body meshes of the character, it will make the component be available to switch to from the dropdown underneath the skin tab.

NormalEnabled/NormalDisabled – these options are for equipment nodes of the prefab, they will make the equipment available to switch on and off separately from the body meshes tab; as the name implies, **NormalEnabled** will have the equipment enabled by default when placing the prefab in the scene, or disabled by default in the other case.

Choosing one of these options will also enable the **And/Or/Activators** tabs. These tabs can allow manipulating equipment dependencies.

For example, a firefighter character might need to have a helmet equipped automatically with a breathing mask. To do this, on the breathing mask component, click the **And** option and choose the helmet node as the dependant part. This will make it so that when equipping the breathing mask, the helmet will also be equipped along.

It is also important to assign the equipment to either the **Head** or the **Equipment** category if applicable (which can be chosen from the tab to the left of the component type, set to **None** by default).

In the **Licenses** tab, the required licenses, if applicable, can be assigned. By enabling the checkbox, the options of adding license options to the All or Any categories will become available.

Tags can be added to the characters from the last tab. This is important to make sure characters are categorized correctly when searching through keywords. Personnel characters need to be tagged with the correct country of provenance and profession alignment.

After completing the configurations, first press the **Save** button (represented by a disk icon on the upper left part of the window) and then the **Reinitiation** button (right next to the Save button, represented by an arrow icon). This will save and apply the changed immediately. Sometimes, restarting the *Object Editor* might be necessary for all the configurations to be applied.

Creating a correct thumbnail

With the character textured in the Objects Editor scene, you need to create a correct thumbnail.

To do that, take the existing prefabs thumbnails as reference – position the character in the scene with the sky as a scenery, in an angle similar to the other character prefabs thumbnails. Take a closeup screenshot and edit small background details in *Photoshop* – the backdrop should be evenly blue.

Save the thumbnail as a .png file type and name it the same as the node which it belongs to. The resolution of the image should be 256x256px.

Step 11: Testing

Now that the character is fully implemented in RS, it is important to check for possible errors.

Checklist:

- **Character orientation and overall look**

Check for any mesh or texture map issues.

The character should be oriented facing the bracket of the ground marker.

Toggle on and off the equipment options to make sure they are also positioned correctly over the character's mesh.



- **Character animation**

By pressing the F4 key, the Object Editor will go toggle into play mode.

In this mode, animations will play. This way, the character animations can be tested. Place the character in the scene and change between different animations in the *Animation* tab from the properties. Check for any mesh issues during animations. Test the equipment as well, to make sure the animations work properly and in sync with the body mesh.

- **Properties and translations**

Make sure that the properties are configured according to the requirements of the character: equipment dependencies, toggling, body meshes.

Check whether the tags chosen are correct and whether there are any licences applicable.

Make sure to edit the translations and check them in RS.

- **Console window/Log**

Some errors only appear visible in the application log, so make sure to check the log for any mesh, materials or meta file errors.

- **Folder hierarchy and files**

Check for any files that are not placed in the proper folder, or extra files that do not need to be stored in the repository.