HOCHSCHULE MITTWEIDA
University of Applied Sciences

Faculty of **Angewandte Computer & Biowissenschaften**

BACHELOR THESIS

The Knight's New Armour - Development of a Procedural Tool Prototype for Adaptive Variation of 3D Assets

Ms.
Weidner Fanny

Course of Study:
Medieninformatik & Interaktives Entertainment

Supervisor:
Prof. Dr. rer nat Marc Ritter
Manuel Heinzig

Mittweida, 11. April 2025

Contents

Additional Lists

List of Figures

1 Introduction

In recent decades, the field of video game development has undergone a rapid transformation, driven by technological advancements [Smi15] that have significantly increased the scale and complexity of three-dimensional (3D) assets. Modern games demand increasingly sophisticated characters, environments, and props, placing greater pressure on artists and developers to balance visual fidelity, production efficiency, and technical constraints ([Rod24], p. 7). Open-world experiences, cutscenes and character customisation systems exert significant pressure on scalable asset creation workflows. Procedural techniques, which were initially developed for the purpose of creating textures, subsequently evolved to encompass the creation of 3D objects. Since the 1980s, this field has undergone significant expansion and has become a consistently significant area of research within the domain of computer graphics ([Ebe+02], preface). Procedural techniques have evolved from the domain of programmable graphics processors to encompass the development of software and tools for their creation (e.g. Houdini, Blender Node Editor, etc.) and are being leveraged to create media such as games [Sid24].

1.1 Motivation

A significant challenge in contemporary asset production is the trade-off between manual artistry and automation, now that the necessary tools are available. Traditional hand-crafted workflows allow for maximum artistic control but require significant time and effort, often resulting in production bottlenecks.

Procedural generation has revolutionised many aspects of game development, most notably world building and level design [Smi15]. Industry-leading studios such as Guerrilla Games have integrated procedural tools to generate landscapes, distribute assets and automate complex modelling tasks 3.1. While environment artists benefit from well-established procedural workflows, character artists still rely heavily on manual modelling techniques, as their work is often very project-specific, individual and lacking unified pipelines [Bus21] [Ter17].

As the field of gaming evolves, genres such as role-playing (RPG), massive multiplayer (MMO), and action RPGs are increasingly characterised by the complexity and customisation of characters and non-player characters (NPCs) within the gaming environment (see chapter on 2.2). The artistic community faces a substantial workload, often involving modular customisation, which necessitates the creation of a vast array of armour and attire (see chapter 2.4.1). These elements serve to symbolise the progression and development of players within the game's ecosystem over the course of their playtime and engagement with the game world (2.2). Of particular note is the creation of armour assets, which represents a particularly wide domain within this field. In contrast to static character models, armour comprises modular, layered components that integrate with diverse body types, animation rigs, and artistic styles. The increasing demand for extensive armour customisation, modular equipment, and character variation highlights the need for dedicated tools tailored specifically for 3D character assets. This thesis seeks to bridge the gap between procedural techniques and armour asset workflows.

Throughout the history of gaming, armour has been a constant feature in role-playing and massively multiplayer online games (RPG and MMO games, respectively). Figure 1.1 and figure 1.2 illustrate a selection of armours in video games over time, demonstrating their constant in the presentation of characters and the progression of their journeys [Tra08].

Figure 1.1: Selection of Armours in games through time: 1993 to 2013.

Figure 1.2: Selection of Armours in games through time: 2017 to 2024.

1.2 Research Objectives and Task

The primary objective of this research is to investigate the effective application of procedural methodologies to the creation of armour assets in the context of video game development. The study will leverage procedural frameworks to develop a toolkit that will streamline the processes of *armour segmentation, pattern generation, morph adaptation and modular generation*, while preserving the artistic vision of character designers and enable reusability. The following key objectives will be addressed:

- **Analyse the armour creation workflow and technical requirements:** Contextualise armour in games through examples and investigate the key stages of armour asset development in game production, identifying technical constraints of complexity and quantity, design challenges, and optimization needs (2 and 5.2).

- **Assess armour modularity for procedural generation:** Examine the structure and components of modular armour (2.6.1) and evaluate their suitability for procedural implementation by comparing them to state-of-the-art tools (3.1).
- **Examine the conditions and requirements for tools:** Identify the essential features, constraints, and usability factors necessary for integrating procedural armour generation into existing industry pipelines, ensuring compatibility with contemporary software (3.4).
- **Develop a procedural tool for character artists based on the parameters and conditions:** Adapt procedural content generation techniques to a prototype toolkit based on the analysed parameters, conditions and requirements with the goal of generating base meshes of armour. (4).
- **Evaluate procedural techniques against manual workflows:** Assess efficiency, accuracy, and usability and contextualise the work through expert interviews and technical comparisons (5).

1.3 Thesis Structure

To achieve the previously listed objectives and explore armours in video games and procedural solutions, the thesis follows this structure:

This first chapter lays the **groundwork and fundamentals** which provide a contextual framework for understanding armour in games, drawing upon examples from contemporary games and the realm of customisation. It then delves into the technical intricacies, aiming to foster a comprehensive understanding of the pertinent considerations of output. Subsequently, it meticulously examines the workflow of armour creation, drawing parallels with character artist workflows in games. This approach highlights the state-of-the-art software and process that must be taken into account and further refined for the development of the toolkit. The examination of these intricacies, in conjunction with the workflow, has led to the identification of bottlenecks and the determination of armour components and parameters of armour that are deemed to be of paramount importance and must be addressed by the tool: namely, base mesh modelling.

The next chapter establishes a technical foundation, encompassing the development of tools and the role of technical artists. Additionally, it delves into procedural generation and the existing tools and methodologies. It fulfils the objective of formulating comprehensive requirements and conditions that serve as the foundation for the development of the tool. This analysis informs the **conceptualisation** of the procedural toolkit, outlining its functional scope, necessary conditions, and integration requirements within industry-standard software.

Based on the concept, the detailed **development** of the procedural toolkit using Houdini follows, focusing on the modular components designed to streamline the creation of armour assets. It discusses the technical architecture of the system, including segmentation tools, adaptive fitting mechanisms, and modular generation techniques. Each tool component is examined in depth, explaining how it interacts with the broader character asset pipeline. The toolkit's integration into Houdini, Maya, and Blender workflows is also described.

The effectiveness of the procedural toolkit is assessed through two **evaluation** methods in the following chapter: In the following section, the results of the expert interviews conducted with industry professionals from AAA and AA studios are presented. These interviews provide qualitative insights into the practicality, usability, and impact of the workflow relevance and tool feasibility in real-world game development environments. The technical evaluation is then discussed. A structured comparison between traditional manual modelling techniques and toolkit-assisted workflows is made, analysing key metrics such as efficiency, accuracy, adaptability, and artistic flexibility. This evaluation determines whether procedural armour generation can enhance production scalability while maintaining creative control.

The final chapter synthesises the research findings on procedural workflows for armour assets, evaluating the impact of a developed toolkit on efficiency, modularity and scalability. It also explores avenues for **future research** and discusses the implications for game studios, positioning custom procedural tools as a solution for optimising armour asset production.

The present thesis aims to make a contribution to the domain of technical art and procedural content generation in tools. It will demonstrate how the implementation of structured procedural tools can efficaciously streamline artist-led asset production workflows. The study will undertake a comprehensive study of manual modelling approaches and methodology in order to use the understanding to formulate a prototype of a procedural prototype tool development.

2 Fundamentals of Armour in A Technical Context and 3D Modelling Processes

The design of armour in video games is a multidisciplinary process that merges historical accuracy, artistic direction, and technical constraints within 3D modelling workflows. This chapter establishes the foundational principles of armour creation from a technical perspective, outlining key terminology, industry standards, and case studies of armours in games and how vital customisation is. Furthermore, it examines the evolution of the quantity of armour assets. This contextual analysis underscores the significance of contemporary workflows and technical requirements in achieving both quality and quantity in modern game development. Additionally, it identifies the technical challenges and intricacies involved in creating high-quality armour models, offering insights into the methodologies employed to meet the demands of modern game development.

2.1 Key Terminology of Armour

In this thesis, the developed tools are referred to as a "*toolkit*". The toolkit is a contemporary solution that combines various tools of different routines and functions into a streamlined usage, analogous to the modelling toolkit exemplified by Adobe Maya. [1] During the analysis of perspectives for armour, various game genres and armour types are explored. While RPG refers to role-playing game, MMORPG refers to massive multiplayer online role-playing game [Met19]. Stats, short for statistics, refers to a character's attributes and proficiencies, often depending on their archetype [Met19].

For key terms relating to armour, this research refers to "Iron Documents. Interdisciplinary studies on the technology of late medieval european plate armour production between 1350 and 1500" [Gol14]. Namely parts of armour in order to generalize the correct use of armour parts.

In most video games, armour refers to a piece of clothing that a character can equip that provides some form of damage protection or negation.

Armour in the context of this thesis, given a technical 3D context, is a modular system of wearable components designed to provide protection, aesthetic identity and thematic cohesion to a character model. Technically, armour is composed of visually distinct and functionally categorised elements, such as helmets, breastplates, gauntlets and greaves, which together form a coherent ensemble adapted to the underlying character mesh.

[1] https://help.autodesk.com/view/MAYAUL/2025/ENU/?guid=GUID-D4CD168A-34F2-465B-A39C-DE20B34E5535, visited 02.01.2024

- **Anatomical Coverage:** Components of armour are classified by the body parts they protect (e.g. head, torso, limbs) and further subdivided into specific elements such as helmets, breastplates or gauntlets. *Helmets, gorgets, aventails, greaves, cuisses, poleyns, sabatons, breastplates, backplates, faulds, pauldrons, rerebraces, vambraces and couters*[Gol14].
  In the context of technical design, each piece of armour is meticulously crafted to conform to a character's rigged structure. This ensures seamless integration with skeletal deformations and animations, facilitating a precise fit to anatomical landmarks (e.g. shoulder joints, waistline). Furthermore, the armour is engineered to exhibit compatibility with dynamic movements, such as running, crouching or fighting stances, thereby ensuring the character's mobility and functionality are not impeded by the armour.
- **Structural Characteristics:** The classification of armour is determined by the method of its fabrication. The categories include laminar armour, characterised by overlapping strips; segmented armour, with articulated joints; and plate armour, which consists of single large sections. Supporting elements such as rivets, hinges and edge reinforcements are also considered as integral, highlighting how form influences functionality, mobility and visual cohesion [Gol14].
- **Visual and Decorative Attributes:** Armour terminology includes aesthetic elements such as material types (metal, leather, fabric), surface details (engraving, gilding, embossing), and symbolic designs (heraldry, runes, faction emblems). These features reflect both cultural narratives and player identity within the game world [Gol14].
- **Thematic and Cultural Design** Armour incorporates historical, cultural or fantastical motifs that communicate the character's identity, faction or narrative context. This use case takes all into account, but makes little distinction between cultural requirements and differences, except in the case of an exemplary showcase based on the creative freedom of video games (extreme cases of e.g. French Gothic armour vs. 16th century samurai armour are shown). (e.g. Persian chainmail vs. 4-in-1 French chainmail, French vs. German breastplates).

2.2 Literature on Immersion and Character Customization

Customization refers to the ability for players to modify aspects of their in-game characters to suit their preferences [TA15]. This encompasses both visual customization (appearance, hairstyles, body types, and clothing) and stat customization (abilities, gameplay mechanics, such as combat skills and class abilities). The created connection fosters immersion and agency, key elements in modern game design [Mcm03]. Immersion, defined as being "caught up in the world of the game's story (...) but it also refers to the player's love of the game and the strategy that goes into it" ([Mcm03], p. 2), is amplified when players can project themselves into their characters through customization.

The ability to personalize characters reflects broader trends in gaming, including inclusivity and representation. This feature enables players to reflect their identities, enhancing engagement and satisfaction. As Mether (2019) observed, visual customization has grown in modern MMORPGs as players seek unique identities. The ability to personalize their avatar fosters a stronger connection with the character, leading to a more immersive and enjoyable gameplay experience [Yee07] [TA15].

- **Enhanced Engagement and Replayability:** Customization offers a wider variety of gameplay experiences. Players can create multiple characters with distinct appearances and abilities, encouraging them to replay the game and explore different storylines [Met19].
- **Greater Social Interaction:** As noted in research by Yee (2007) and Mether (2019), Customization in multiplayer games, or even singleplayer, encourages players to share and compare designs, fostering community engagement (forums about how to customize and create strong builds to optimize stats in-game).
- **Increased Player Loyalty:** Studies have shown a positive correlation between customization options and player loyalty in online games [Dom+24]. When players feel invested in their characters, they are more likely to continue playing.

Historical and pseudo-historical influences are crucial in shaping immersive experiences in fantasy and role-playing games. Many games draw on medieval aesthetics and armour design to evoke historical authenticity, blending it with fantasy. Medieval settings in games ground players in familiar iconography, enhancing immersion [Tra08]. These representations are often pseudo-medieval, adapting historical designs to fit modern gameplay and sensibilities rather than being historically accurate. Games such as Kingdom Come: Deliverance and For Honor show the dynamic interplay between historical accuracy and creative expression in medieval gaming. As demonstrated in an article by Kazuma Hashimoto on Polygon, certain games like Dark Souls have been designed to incorporate fantastical armour that is closely related to a real counterpart.[2] This allows players to engage with historically inspired aesthetics while maintaining a sense of power fantasy and identity expression through character builds.

2.3 Game Case Studies

In order to emphasise the implementation across genres, Turkay (2015), Domingues (2014) and Mether (2019) present examples from a variety of exemplified genres. In a similar manner, this chapter provides detailed outlines of reference games with regard to armour and visual customisation. In RPG video games such as Lords of the Fallen, Elden Ring, Dark Souls, Diablo 4 and For Honor, customization and character progression through “loot” and equipment play a central role. These games feature extensive armour systems, with hundreds of pieces to support character individuality and narrative immersion.

In Lords of the Fallen, developed by Hexworks and CI Games, the design of armour serves as a pivotal narrative and gameplay element within the context of its dark fantasy role-playing game (RPG) framework. The game’s aesthetic is heavily influenced by French, Italian and German Gothic medievalism, incorporating a plethora of religious and cultistic symbolism (see 2.1 and appendix A.1 for examples). Its visuals are characterised by battle-worn designs, showcasing intricate details. This is consistent through the narrative themes of divine conflict and corruption, with the visual identity of the armour serving to embed the storytelling. It is possible to recolour each piece of armour using preset colour palettes, which can be collected

[2] `https://www.polygon.com/23391859/video-game-medieval-armor` visited 05.12.2024

throughout the course of the game. The game's visual style is characterised by a number of common elements, including dark metallic plates, glowing runes, chain mail, leather, and cloth.[3]

Figure 2.1: Example Armour Sets from Hexwork's and CI Games' Lords of the Fallen

FromSoftware's Elden Ring, an open-world soulslike action role-playing game set in a dark fantasy environment, provides an exemplar of modular armour design, enabling players to equip helmets, chest plates, gauntlets and greaves on an individual basis. The armour style is inspired by and leans towards a more abstract fantasy rather than a realistic medievalist style, drawing loose inspiration from Celt/Saxon/Roman-Gallic styles with occasional influences from other sources, such as Japanese. The armour in Elden Ring is frequently characterized by elaborate engraving and abnormal proportions. (see 2.2 and appendix A.2 for examples)[4]

The game places significant emphasis on practicality, frequently requiring players to prioritize defensive capabilities and composure over purely aesthetic considerations. This fosters a synergy between functionality and visual appeal. Furthermore, the game adapts the armour to the player's body type, ensuring that those of differing character builds – light, medium, or heavy – can find suitable gear that aligns with their preferred playstyle. Each piece of armour has distinct statistics for physical and elemental resistance, weight, and poise. Players have the option to customize their appearance by combining different armour sets, thereby allowing for flexibility in both aesthetic and functional considerations. Armour pieces are obtained through exploration, boss battles, or non-player character interactions, thereby integrating progression within the game's mechanics. Despite the paucity of diversity in body types, the modular system guarantees considerable autonomy in the customization of gear for disparate builds, including those of heavy tanks and agile mages.

Larian Studios' Baldur's Gate 3 represents a reimagining of the classic role-playing game (RPG) formula, incorporating modern visuals and extensive customisation options. The role of armour in both gameplay mechanics and character personalisation is significant, reflecting its prominence in tabletop role-playing games. Baldur's Gate 3 features a plethora of armour sets, each associated with a specific class and playstyle. The capacity for customisation extends beyond the functional aspects of the game, enabling players to create characters with diverse appearances, including ethnic diversity and gender options. The narrative-driven customisation of characters affects the manner in which players interact with one another

[3] `https://lordsofthefallen.com/`, visited 05.12.2024
[4] `https://de.bandainamcoent.eu/elden-ring/elden-ring`, visited 05.12

Figure 2.2: Example Armour Sets from FromSoftware's Elden Ring

and the eventual outcome of the storyline. The choice of armour affects the efficiency of combat, but its aesthetic features also reflect the character's alignment, class and personality. This depth of customisation allows players to create unique avatars that align with their narrative choices, thereby enriching the role-playing experience.[5]

Baldur's Gate 3 draws upon high fantasy traditions, combining medieval motifs with a hint of Renaissance-inspired elegance as shown in figure 2.3 and further in appendix A.3. The majority of items are designed with a more straightforward aesthetic, particularly in the case of clothing. Depending on the specific region within the game, players have the option of acquiring more elaborate garments. The high fantasy style, while not including the gritty realism of medieval armour in certain instances, incorporates a fantastical element to plate armours and a limited number of chainmail options. However, there is a notable lack of variety in the appearance of cloth armours for wizards. Additionally, armours are designed to fit different races, including dwarves and dragonborn, with four distinct body types for each race.

Figure 2.3: Example Armour Sets from Larian Studios' Baldur's Gate 3

Blizzard's Diablo 4 represents a continuation of the franchise's legacy as an action role-playing game (RPG), with a particular emphasis on loot-driven progression and customisation. The armour worn by the characters in the game serves two distinct purposes. Firstly, it

[5] https://baldursgate3.game/, visited 05.12.2024

enhances their performance in combat. Secondly, it creates a visually striking effect, thereby contributing to the dark and gothic tone of the game as a whole. [6] The armour system in Diablo 4 comprises distinct sets that are specific to each class and align with the unique gameplay mechanics associated with each class. The sets are scalable in accordance with the player's progression through the levels, and they permit the alteration of armour appearances through the use of transmog, which allows players to modify their armour without affecting the associated statistics. Moreover, the game utilises cosmetic items as a monetisation strategy, employing microtransactions to facilitate additional player customisation. The incorporation of armour into the game's loot system provides an incentive for prolonged play through the customisation of appearance. The incorporation of Transmog (explain) affords players the autonomy to maintain a coherent visual style while retaining the functionality and performance of their character.

The fourth iteration of the Diablo series employs a Gothic aesthetic, with a pronounced emphasis on designs that evoke a sense of darkness and foreboding. The armour worn by the characters in the game serves to reflect their individual class skills and abilities. To illustrate (see figure B.1 and appendix A.4), the armour worn by the Necromancer is defined by a primal and bone-laden design, whereas the druidic armour draws inspiration from the natural world. The game incorporates regional and class-specific influences, with armour materials including animal hides, engraved metals, and arcane elements such as glowing symbols. The designs are highly intricate and detailed.

The customisation options allow these styles to be adapted across different body types and genders, thereby maintaining inclusivity. The overall aesthetic evokes the apocalyptic world of Diablo, blending medieval realism with fantastical influences. The armour pieces are intricately detailed, creating a layered appearance that enhances the feeling of immersion and power, which are central to the game.

Figure 2.4: Example Armour Sets from Blizzard's Diablo 4

Ubisoft's For Honor incorporates both hack-and-slash mechanics and class-based gameplay, featuring a diverse array of warriors, including knights, samurai, Vikings, and others. Figure 2.5 and A.5 showcase examples. The armour system is reflective of the game's historical inspirations, whilst simultaneously emphasising the individuality of the player in a multiplayer context. [7]

[6] `https://diablo4.blizzard.com/de-de/`, visited 05.12
[7] `https://www.ubisoft.com/de-de/game/for-honor`, visited 05.12.2024

The customisation of armour in For Honor is largely cosmetic in nature, with the primary objective being the visual expression of the player's identity rather than the alteration of gameplay statistics. Players have the option to personalise their gear with engravings, colours, and emblems, while selecting pieces that symbolise their faction. These options are designed to facilitate engagement in multiplayer scenarios, wherein distinctive appearances serve to signify status or achievements. The system places greater emphasis on aesthetic considerations than on practical functionality, thereby enabling players to project their identity and accomplishments in multiplayer matches. The availability of high-level cosmetics, frequently associated with achievements or microtransactions, serves to foster a sense of pride and exclusivity among players.

The For Honor game features an armour system that is grounded in historical realism. Distinct cultural identities are represented for the three core factions: The game features three distinct factions: knights, samurai, and Vikings. The armour designs are respectful of historical accuracy, yet incorporate artistic flourishes such as intricate etchings and vibrant emblems, thus individualising each warrior. The armour worn by knights is representative of the high-medieval European style, incorporating chainmail and plate armour adorned with heraldic crests. A.5 The armour worn by samurai draws inspiration from Japanese Sengoku-period designs, featuring lacquered plates and bold patterns. In contrast, Viking gear is characterised by its rugged, functional protection, utilising fur and leather. Each class of warrior has a multitude of armours with minimal differentiation between them, with the exception of the legendary armours. A.5

Figure 2.5: Example Armour Sets from Ubisoft's For Honor

2.4 Complexity and Quantity of Armour Assets

The quantity and design complexity of armour in video games provide critical insights into the technical demands, visual fidelity, and genre development over time. It is crucial to recognise that any categorisation of video games is inherently becoming obsolete due to the rapid technological advancement of the medium and the extensive range of approaches, processes and budgets. The objective is to establish an outline of the essential elements that must be taken into account when defining technical complexity and usage of the prototype. While this section does not solely focus on the numerical analysis of the quantity and 3D complexity of armour in games, it also highlights the games that align with the subsequent technical development use-case.

2.4.1 Quantity of Armour in Video Games

The games selected for this statistical analysis represent a cross-section of influential and genre-defining titles across the role-playing game (RPG), action RPG, and massive multiplayer domain [Dom+24]. In the case that a game is known for a series of released titles over time, the following titles are included for comparison. In a multitude of instances, games demonstrate a tendency to exhibit armour of a visually similar appearance. Such armour is incorporated into the count, whilst those that are visually indistinguishable are excluded from the approximations. As the process of collecting precise numerical data and standardising the components of armour is a laborious task and nearly without a unified overview, the figures presented are approximate estimations of the number of armour pieces in the games. An attempt has been made to standardise the components, with around four to five pieces being equal to an armour set.

Genre Impact: It is evident that certain titles have been instrumental in effecting paradigm shifts in the realm of their genre of gaming, or in delineating the fundamental mechanics that would come to define future generations of games. Notable examples of this phenomenon include Diablo [8], Skyrim and Dark Souls, which have exerted a substantial influence on the evolution of RPG mechanics, particularly with regard to character equipment systems. The latter games are widely regarded by gamers and game journalists as two of the most exceptional, as evidenced exemplary by articles on IGN [9], Games Radar [10] and Gaming Bolt [11].

Historical Influence Setting: The utilisation of historical or medieval settings constitutes a pivotal element in the realm of armour design. This notion is exemplified by prominent titles such as *Lords of the Fallen* and *For Honor*, which draw inspiration from historical armour traditions, thereby integrating cultural aesthetics into their fantasy or combat environments. These designs serve to anchor game narratives within historical or lore-rich contexts, wherein armour functions as a conduit between player immersion and the authenticity of the setting (as described in2.2).

Customization Depth: The importance of player agency in the realm of character personalization is exemplified by the customisation systems observed in titles such as *Baldur's Gate 3* and The *Dragon Age: Veilguard*. These games empower players with a focus on shaping the look of their characters to tell their own story in the game world, seamlessly integrating aesthetic preferences with functional customisation. [12,13]

[8] https://www.denofgeek.com/games/rpgs-that-revolutionized-the-genre/, visited 12.12.2024
[9] https://web.archive.org/web/20171210124028/http://ign.com/lists/top-100-games/, visited 12.12.2024
[10] https://www.gamesradar.com/best-games-of-all-time/, visited 12.12.2024
[11] https://gamingbolt.com/30-best-games-of-all-time-2024-edition, visited 12.12.2024
[12] https://baldursgate3.game/, visited 12.12.2024
[13] https://www.ea.com/de-de/games/dragon-age/dragon-age-the-veilguard, visited 12.12.2024

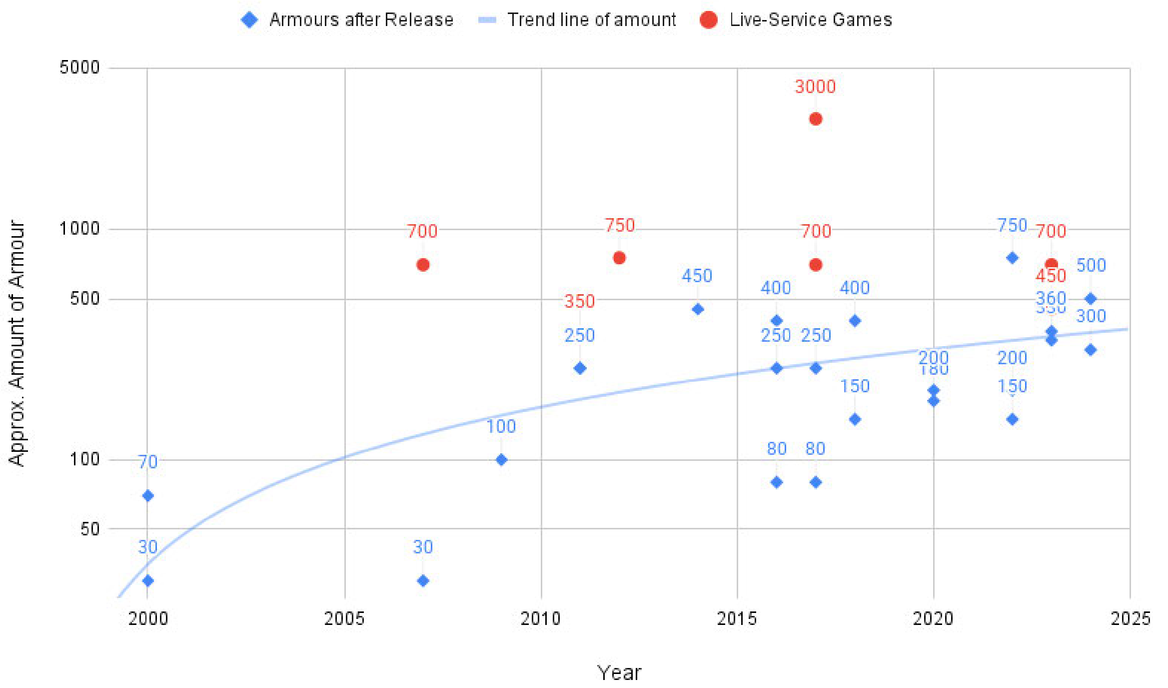

Figure 2.6: Selection of games and their armour over the years since 2000, highlighting varying amounts. Scatter diagram formed from table in appendix B

Advanced Graphics and Technical Complexity: Recent RPGs, including *Elden Ring*, *God of War: Ragnarök* [14] and *Horizon Forbidden West* [15,16], have been pushing the boundaries of graphical fidelity in bringing their worlds to life. These games exemplify advancements in 3D rendering, permanently simulated worlds and realistic graphics.

As illustrated in diagram 2.6 and the table in appendix furtherB, that a growing number of games are adopting large content levels regarding armours. There are games where the exact numbers cannot be fully estimated due to high content levels, updates over time (e.g. *Lord of the Rings Online (2007*), *World of Warcraft (2004)*,...). This tendency is especially apparent in live service games that have been included in the recent years, as their numbers represent the current armour amounts. This is evident in games such as The *Elder Scrolls IV: Oblivion (2006)*, *Path of Exile 2 (2024), Diablo 4 (2023)* and *For Honor (2017)*, which still receive regular updates to expand their content and maintain player engagement. Data indicates a clear trend in the increasing number of armour pieces in different game titles over time, particularly in RPGs and live-service games. For instance, early titles such as *Baldur's Gate 2 (2000)* and *God of War 2 (2007)* featured relatively low armour counts, with numbers between 20 to 90. However, by the 2010s, games such as *Dark Souls 2 (2014)* and *Monster Hunter: World (2018)* have seen a marked increase in the complexity of their armour libraries, with totals of 450 to over 600 armour pieces, respectively. Another noteworthy observation regarding the series of live-service games is that the latest release exhibits a level of armour pieces that is comparable to that of the title in the series released prior to the most recent instalment, with subsequent updates having been made over the course of several years (see Diablo 2-3 and Path of Exile 1-2). More recently, titles such as *Elden Ring (2022)* and *Diablo 4 (2023)* have set

[14] https://mamoniem.com/behind-the-pretty-frames-god-of-war/, visited 12.12.2024
[15] https://www.guerrilla-games.com/read/making-tools-for-big-games, visited 12.12.2024
[16] https://www.guerrilla-games.com/read/Streaming-the-World-of-Horizon-Zero-Dawn, visited 12.12.2024

new benchmarks with over 700 pieces, respectively, reflecting the increasing complexity and player customisation demands of modern titles. The live service models exemplified by *Path of Exile 2 (2024)* and *For Honor (2017)* underscore this trend by continually expanding their armour libraries, emphasising the growing complexity and customisation demands of modern titles and setting benchmarks of over 1000 to even 3000 individual armour pieces. It can be posited that, given the recurrent figure of approximately 700, a baseline for substantial content games and armour quantities in live-service and new releases has been instituted.This development underscores the necessity for extensive content pipelines to accommodate the growth of game content. As articulated by Guerrilla Games' Michiel van der Leeuw in his presented work on tools and procedural content in games, the expansion of games is substantial, and teams are compelled to meet expectations by adapting through novel means of development (see later chapter 3.1). [17]

2.4.2 Complexity of Armour in Video Games

In games such as *Elden Ring*, the substantial quantity of armour assets necessitates the implementation of efficient workflows to manage modelling, texturing, and optimization (see [AD19] for further optimisation). This challenge is further compounded by the necessity to maintain high visual fidelity and consistent performance across different character models and gameplay scenarios. Within the workflow of creating armour, different aspects have to be considered which enhances complexity of the work and the model.

1. Geometrical Complexity The term *geometrical complexity* is used to describe the intricate and detailed structure of a given object or model. The geometric intricacy of a three-dimensional (*3D*) armour model can be defined as the complexity of its polygonal structure, which can be broken down into several factors: polygon count, surface detailing, shape complexity, modularity and articulation and level of detail.

An elevated *polygon count* has the potential to yield surfaces that are both more detailed and more smoothly rendered; however, it also demands greater computational power, both during the process of modelling and in the subsequent rendering stage in engine [Kov15] ([AD19], p. 152). Conversely, a reduced polygon count may constrain the achievable level of detail, yet it is often a prerequisite for real-time applications, including video games. In the construction of a polygonal model, it is imperative to ensure the correct topology is employed. This is essential for ensuring the integrity of the model's animation and engine integration, while also averting any collisions with shaders [Ter17] [Kov15]. It is common for armour designs to feature complex curvatures, overlapping pieces, or intricate shapes such as plates and spikes [Gol14]. As the design becomes more intricate, the number of pieces comprising the armour increases, leading to a corresponding rise in the polygon count. The development of armour designed to accommodate diverse body types or incorporate movable components, such as capes, introduces an additional layer of complexity due to the necessity for precise simulation or modularity (e.g. capes and damage of pieces). In instances where the model is required to accommodate an additional body type, an extra model is stored within the engine.

[17] `https://www.guerrilla-games.com/read/making-tools-for-big-games`, visited 21.12.2024

The *level of detail* (LOD) is used to describe the varying levels of detail in a 3D model, which can be adjusted to suit the desired visual representation. “Each part of the model should always be at the same level in terms of details.” ([Ter17], p. 74) Level of detail and realism in an armour model, from intricate engravings to worn textures, has a significant impact on its complexity. High-poly sculpts are frequently employed to achieve realism, but these must be optimised for game engines through techniques such as normal mapping and displacement [Kov15]. The implementation of LOD systems enables the display of lower-detail versions of the model at greater distances, thereby reducing the computational load ([AD19], p. 215). A model with several LODs can be complex but still optimised for performance (REF). The addition of photorealistic textures and realistic materials (e.g., metallic and reflective surfaces) adds further complexity, necessitating advanced shaders and real-time lighting adjustments. Jesus Merino emphasises the process of creating a high-fidelity, CG-ready samurai armour with a high level of realism, which exemplifies the added effort involved utilising a number of texture maps to enhance the look. [18] The incorporation of features such as dents, scratches, and engraved patterns has been demonstrated to result in a substantial augmentation of the intricacy of a given model. It can be done both in the step of sculpting or texturing; as the augmentation is achieved through the utilisation of additional polygons or the employment of normal maps, which facilitate the simulation of intricate textures without an attendant increase in the polygon count.

2. Material and Textural Complexity The materials used in the construction of 3D armour models are of equal importance to their geometry in defining the level of complexity. The materials and textures are the determining factors in visual fidelity, and can also impact rendering performance. [19]

Texture resolution is employed to denote the level of detail in a texture map. The utilisation of high-resolution textures serves to enhance the realism of the visual representation; however, this approach concomitantly results in a substantial increase in memory usage [Ter17] ([AD19], p. 236). The texture maps for armour models typically comprise diffuse, normal, specular, and roughness maps, with each contributing to the visual complexity [Ter17]. [20] The composition of armour pieces frequently involves a combination of materials, including metal, leather, fabric, and cloth. The utilisation of distinct shaders, mapping, and rendering techniques is imperative to attain the desired visual outcome, with the number of materials employed directly proportional to the amount of memory utilised. The incorporation of *procedural techniques* to generate random patterns[Ter17], weathering, or age marks introduces further intricacy (e.g. forms of wear and tear on attire after and during combat, showcased in Helldivers 2 [21]).

3. Rigging and Skinning In the context of interactive models, such as those utilised in games or animations, the intricacy of the rigging and skinning process assumes considerable significance. This process encompasses the following elements: Bone Structure and Weight Painting The intricacy of the *bone structure* determines the manner in which the armour in-

[18] `https://www.artstation.com/artwork/28BG8x`, , visited 26.12.2024
[19] `https://dev.epicgames.com/documentation/en-us/unreal-engine/texture-streaming-metrics-in-unreal-engine`, visited 26.12.2024
[20] `https://www.artstation.com/artwork/28BG8x`, visited 26.12.2024
[21] `https://www.playstation.com/de-de/games/helldivers-2/`, visited 20.12.2024

teracts with the character. In order to ensure natural movement, a highly detailed skeleton, comprising a multitude of bones and joints, may be required [Ter17] [Kov15]. *Accurate skinning*, done through a technique called *weight painting*, guarantees seamless integration of the armour with the character. Complex rigs may necessitate the utilization of advanced skinning techniques to accommodate factors such as snug armour-fitting, cloth simulations, or realistic deformations. “Weight painting allows the rigger to define how much the movement of each bone or joint affects each of the vertices of the 3D model” ([Ter17], p. 34).

2.5 Armour Creation Processes in Game Development

This chapter is concerned with an examination of the core stages of actions and methods taken in armour creation, from blocking and sculpting to texturing, rigging, and integration into game engines. In doing so, it first addresses the specifics of technical hurdles and requirements in armour creation in games and their origin. The role of industry-standard tools such as Maya, ZBrush, and Substance Painter is highlighted.

2.5.1 Workflow Visualisation

The workflow flow chart visualisation draws from various industry interviews, articles and papers, as listed in the appendix D.2. The process is abstracted in parts from common character modelling and design workflows to reflect armour modelling exclusively. Main references for the flowchart shown are furthermore Ian McOran-Campbell’s presentation to 80.lv [22], archived Unreal Engine 4 documentation wiki [23] and Andy Davis’ documented 3D Character Pipeline [24]

2.5.2 Expanding on the Workflow Steps

In the course of the research, a variety of data has been collected pertaining to the time frame involved in the creation of high-fidelity 3D armour for gaming and computer graphics applications. The duration of this process is contingent upon various factors, including the experience of the artist, the complexity of the armour, the objectives and the intended application. The range of this variability is from 40 hours to 200 hours, with the possibility of a longer duration D.2. A more comprehensive examination of this topic is provided in the expert interview evaluation section (see chapter here).

Concept and Reference Gathering

From the outset, visual references and design concepts guide the process. These references may encompass a variety of media, including historical imagery, in-game style guides, and detailed sketches by concept artists. Digital tools such as PureRef and Photoshop are frequently

[22] `https://80.lv/articles/learn-how-to-craft-intricate-clothing-historic-chainmail-in-3d/`, visited 02.02.2025

[23] `https://nerivec.github.io/old-ue4-wiki/pages/weapon-modeling-tutorial-shock-rifle.html`, visited 02.02.2025

[24] `https://robonobodojo.wordpress.com/2021/11/26/3d-character-pipeline-part-ii/`, visited 02.02.2025

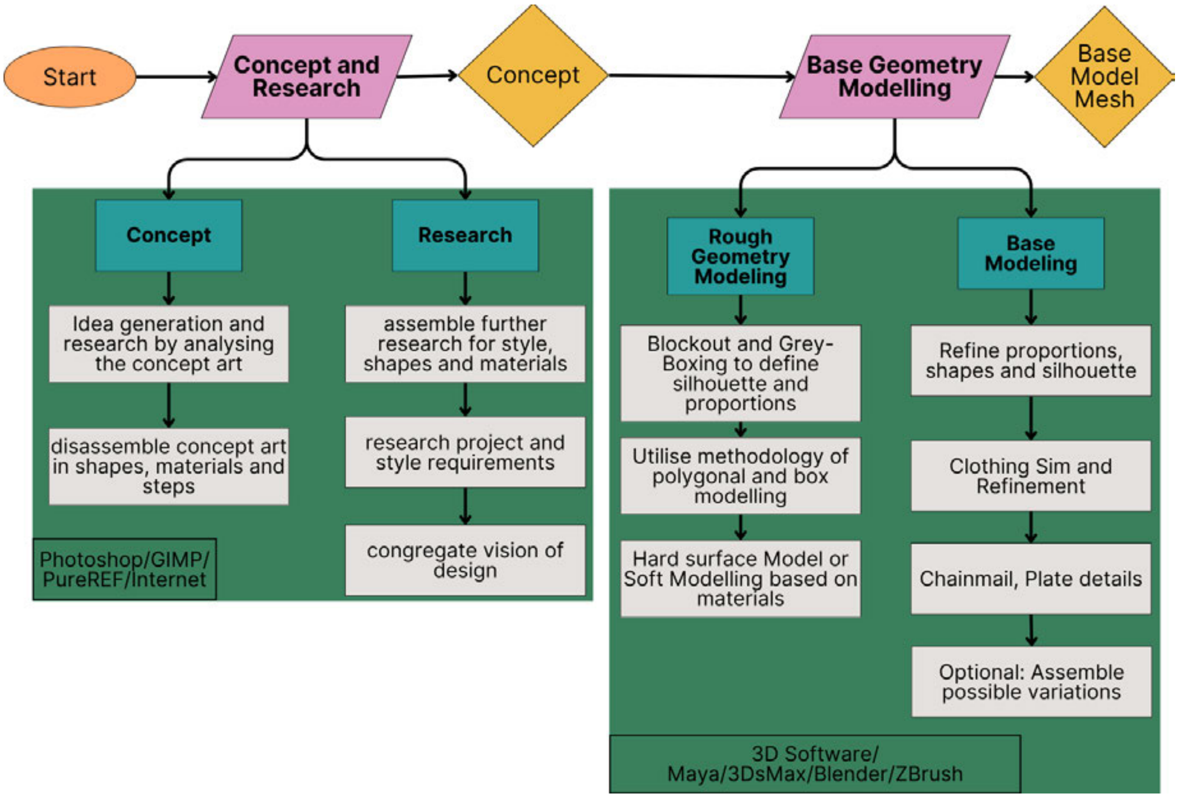

Figure 2.7: Part 1 Workflow Visualisation: Concept, Research and Base Modelling

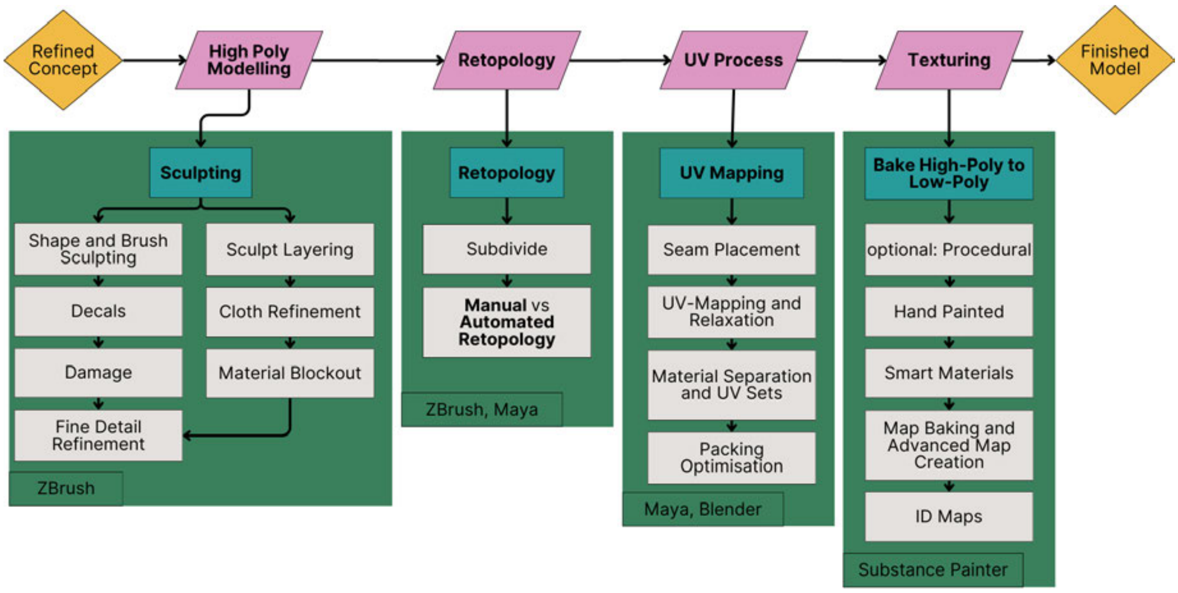

Figure 2.8: Part 2 Workflow Visualisation: High-Poly Modelling, Retopology, UV Process, Texturing

utilised to collate and visualise these assets, ensuring clear alignment with the project's artistic vision. The model is then developed to align with or draw inspiration from these references, thereby ensuring accuracy in terms of proportions, style, and the project's goals [Ter17]. The final output should be a good understanding of the base model to be created, which answers questions of material composition, proportions and purpose. This may be achieved through the use of organised reference materials or sketches.

Base Geometry Modelling

The blockout phase is the foundational step in the 3D modelling workflow. During this phase, a low-poly representation of the armour is constructed. The primary objective of this stage is to define the overall proportions, shapes, and spatial relationships of the armour. This is achieved while adhering to the artistic vision and technical constraints established during the concept and reference gathering stage.

Shape prototyping is a process that involves the use of primitive geometries, such as cubes, cylinders, and spheres, to delineate the primary structural components of the armour. Key features, including chest plates, greaves, and gauntlets, are modelled through prototyping to ensure precise anatomical proportions and alignment with the character rig. This stage emphasises non-destructive workflows, facilitating iterative refinements based on feedback. Preferred tools for this stage include Maya or ZBrush due to their sculpting and polygonal modelling capabilities supported by tools like modifiers. Artists prioritise maintaining the armour's silhouette while leaving fine details to later stages. Greyboxing is a critical method during blockout, where simplified geometry is applied to represent the armour's primary components and verify their proportions. This allows teams to evaluate whether the design aligns with the character rig's proportions and the broader visual style of the game. This approach enables the modification of components at a comparatively lower resource intensity compared to subsequent stages. Modifiers are defined as automatic operations that affect the properties of an object by performing effects that would be difficult to achieve manually. They enable the modeller to work in a non-destructive manner, with greater flexibility and speed. Modifiers alter the manner in which an object is displayed and rendered, without affecting the underlying geometry. This means that the changes made to the object are not permanent until the modifier is purposefully applied, or "collapsed".

The utilisation of *modular design principles* (see 5.2 and 3.1) during the blockout process ensures the interchangeability of individual armour components, such as shoulder pads and helmets, facilitating seamless integration. Symmetrical designs can be created and tested through the employment of modifier techniques, including symmetry, mirroring, and duplication. Shape prototyping for various armours is predominantly enhanced by hard surface modelling, a technique employed for elements such as plate armour and helmets, with a focus on angular edges and rigid geometry. Conversely, chainmail, padded components and more organic elements are modelled using soft surface or organic techniques. A notable method employed in conjunction with ZBrush is the masking method, wherein a mask is manually painted on the existing body mesh and subsequently extruded as a shape base for the armour.

Primitive geometries are utilised to establish the baseline flow and proportions of cloth, with key attributes such as the natural draping of fabric and its interaction with the armour being defined. Placeholder geometry for cloth elements (e.g. capes, tabards) is introduced during the blockout to represent the intended size and movement range. It is imperative to ensure that initial geometry avoids collision with solid armour pieces and aligns with character movement. Subsequent refinement of the drape and fit with accurate cloth physics can be achieved through the use of tools such as Marvelous Designer in this or the High-Poly step.

Placeholder grids or simple overlapping patterns are employed to block out the regions where chainmail is to be incorporated, for example, under plate armour or between shoulder pads and breastplates. Base shapes are prepared with predefined areas where chainmail detailing is to be added, ensuring alignment with modular pieces such as gauntlets or leggings. Chainmail frequently serves as an underlayer beneath other components. The creation of a chainmail mesh typically involves the modelling of a base of interlocking rings, which is then applied to a flattened version of the mesh like a grid to be used. This pattern is then transformed to achieve the desired shape. [25]

It is important to consider the blockout stage that will result in the armour being split into different wearable parts in the game, in order to avoid intersection with other parts. Furthermore, this stage facilitates the establishment of orientation points, which are instrumental in ensuring that animators can avoid angles of movement that would otherwise hinder the character's fluidity. To illustrate this point, consider the necessity to avoid animating a scenario of a character lifting their arms while wearing overly large shoulder pads.

High Poly Sculpting

Sculpting represents a pivotal stage in the 3D modelling process, at which point the armour transitions from rudimentary shapes into detailed, high-resolution models that capture the intricacies of the design. This transition builds on the foundational forms established during the blockout phase, refining surfaces and adding layers of detail that align with the creative vision and technical requirements of the project. This phase integrates a blend of manual and tooled techniques to achieve precision of artistic vision.

The process commences with the secondary shape definition stage, wherein medium-level details are meticulously crafted to enhance the primary silhouette and proportion established earlier. This stage entails the delineation of major components such as overlapping plates, segmented armour, and fabric folds. The utilisation of tools like ZBrush and Maya empowers artists to operate on the model with subdivision techniques, thereby facilitating enhanced control over shape refinement without compromising the integrity of the original forms. As the sculpting process evolves, the emphasis transitions to the refinement of details, introducing elements such as stitches, rivets, grooves, and diminutive decorative embellishments. These elements are instrumental in conveying material authenticity, as evidenced by the aged texture of leather or the hammered surface of metal. Techniques such as brush-based carving and the utilisation of alpha stamps for patterning facilitate consistent and precise detailing across armour sections. Subdivision modelling is instrumental in facilitating this level of intricacy, ensuring that even the minutest details are discernible without compromising the underlying geometry. This step does not pay significant attention to correct topology or polygon count, as these will be addressed in later retopology and baking processes.

Surface layering is an indispensable aspect of sculpting, offering a non-destructive approach to the creation and adjustment of details. Artists create individual layers to address specific features, such as damage effects, fabric textures, or decorative engravings. These layers facilitate modular edits and simplify iterative revisions, as adjustments can be made without

[25] `https://youtu.be/PSw7BhdvEnE?si=Pdb-62efX-3AUhd7`, visited 02.01.2025

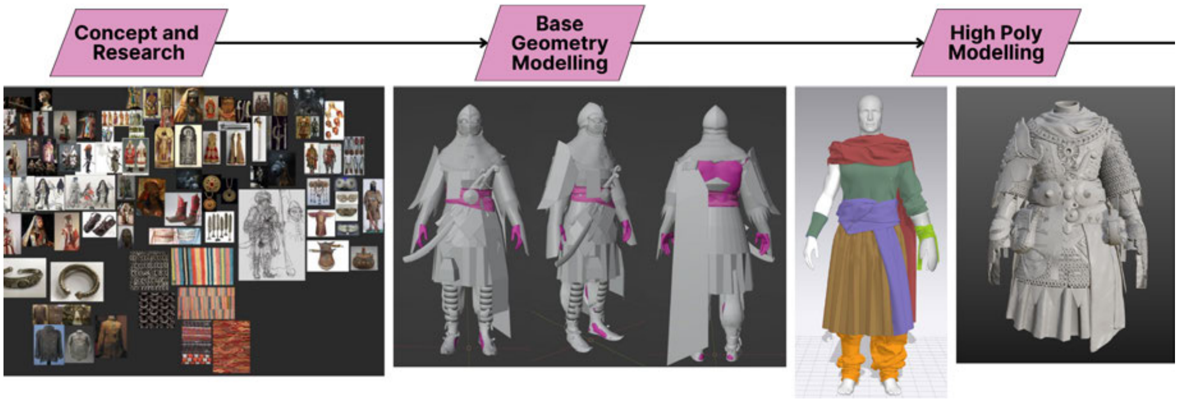

Figure 2.9: Step Images 1 showing Concept Research Board, Base Geometry Mesh, Cloth Simulation and Sculpted Armour by Ian McOran-Campbell's presentation to 80.lv

disrupting the integrity of other design components. This methodology ensures that creative explorations remain fluid throughout the sculpting process. The application of decals and decorative embellishments serves to augment the individuality and narrative of the armour. The employment of projection tools and stencil systems facilitates the precise application of symbols, engravings and embossed designs. A notable feature of this stage is the incorporation of wear and damage details. The addition of scratches, dents and cracks serves to enhance the realism of the final product. The utilisation of tools such as scraping and chipping brushes facilitates the creation of localised damage effects, while procedural damage workflows in software such as Substance Designer or ZBrush automate repetitive patterns and enhance creative freedom.

Dynamic cloth elements, such as capes, straps, or flowing fabric components, require specialised attention to ensure realism. Marvelous Designer can simulate natural folds and draping, which are then refined in ZBrush with custom brushes to harmonise with the rigid armour sections. This careful treatment enhances the authenticity and visual appeal of materials like silk, leather, and heavy cloth.

The sculpting phase is concluded with a process of surface cleanup, a crucial step in preparing the model for subsequent workflows such as retopology or texturing. Errors such as intersecting geometry, uneven topology, or extraneous polygons are identified and resolved. Mesh analysis tools in ZBrush and Maya streamline this process, ensuring a clean, polished surface. This refinement is essential to maintaining efficiency in the later stages of the production pipeline.

The outputs of the sculpting process include high-resolution, intricately detailed geometry that captures the intended aesthetic and functional aspects of the armour. The modular nature of layered edits ensures flexibility for iterations and feedback, while cleaned and prepared surfaces facilitate seamless transitions into the retopology and texturing phases.

Retopology

The goal of this process is to create a topology that allows for seamless animation and texturing, integrating the majority of the vital details. Prior to creating new geometry, artists analyse the high-poly model to determine critical areas of detail, such as curved edges, joints, or highly visible surfaces. Edge loops are planned for deformation zones (e.g. elbows, knees) to ensure smooth animations and efficient UV mapping. The artists employ Maya or Blender to manually create a clean, quad-based low-poly mesh over the high-resolution model. This process demands precision to retain the sculpt's essential form while reducing the polygon count for real-time optimisation. Alternatively, artists may opt for automated retopology tools such as ZBrush's ZRemesher, which can expedite the process but necessitate additional manual refinement due to their inability to meet specific topology requirements.

UV Mapping

UV mapping is an essential process in the creation of 3D armour, whereby a 3D model's surface is translated into a 2D coordinate system for the precise application of texture. The initial step in UV mapping is seam placement, which involves the division of the 3D geometry into sections that can be unwrapped into a flat, 2D layout. The strategic positioning of seams along less visible or natural segmentation points, such as plate boundaries or edges beneath folds, minimises texture distortion and prevents noticeable transitions. RizomUV and Maya's UV Editor are two tools that aid this process, by both detecting and refining the UVs automatically and manually. Unwrapping flattens 3D geometry into an evenly scaled UV map, using algorithms such as Least Squares Conformal Mapping (LSCM) to ensure minimal distortion and balanced texel density .Relaxation techniques refine problematic areas, while symmetry mapping accelerates workflows for mirrored components. Following this process, UV shells are compressed into a texture atlas, thus optimising the use of available resolution while preserving proportional scaling between components. This step is crucial for ensuring that detailed areas of the armour, such as engravings or embellishments, receive higher texel density, thereby enhancing their clarity and quality.

Armour models frequently necessitate multiple materials, including metal, leather, or cloth, each with distinct texture and UV requirements. The models are then divided into distinct UV sets or assigned dedicated regions within the same texture atlas. Material IDs can also be encoded to streamline workflows in texturing software such as Substance Painter.

Texturing

In the realm of 3D armour creation, texturing plays a pivotal role in achieving both realism and artistic consistency. This process entails the application of surface details and visual aesthetics through various workflows. The initial phase involves the baking of maps, which entails the transfer of high-resolution details from a sculpted model to a low-poly mesh. This process includes the creation of Normal, Ambient Occlusion, and Curvature maps, which serve as the foundation for texture detail and shading.

Subsequently, artists employ software such as Substance Painter or Quixel Mixer to create textures procedurally or manually to attain finer artistic control. Procedural methods utilise smart materials that adapt dynamically to the model's geometry, enabling rapid adjustments and consistency across assets. ID maps generated during the UV mapping phase are instrumental in isolating regions of the model for targeted texturing, such as distinct materials like metal, leather, or cloth.

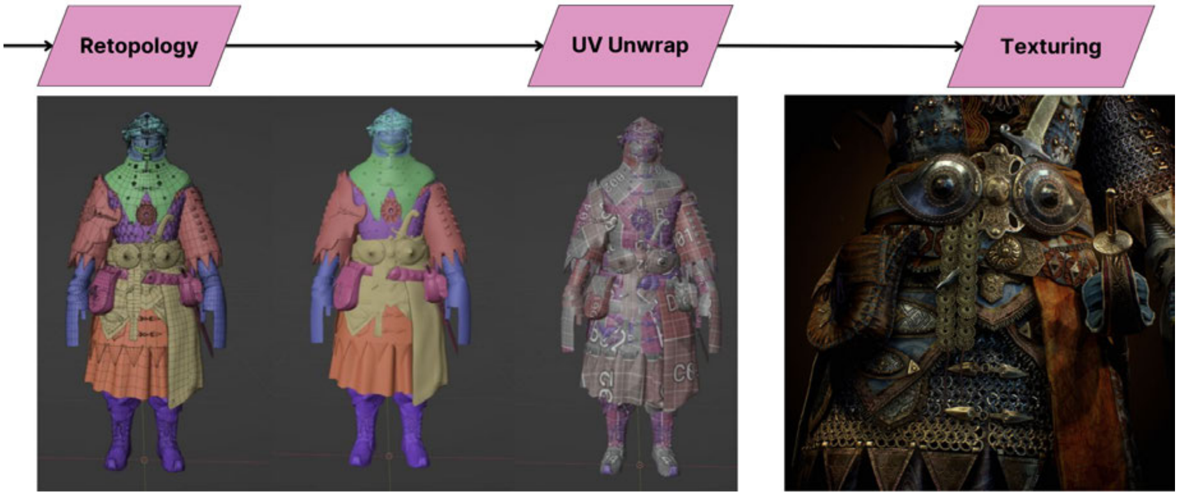

Figure 2.10: Step Images 2 showing Retopology Output, UVs and Textures of the final armour model by Ian McOran-Campbell's presentation to 80.lv

The application of advanced techniques, such as PBR (Physically Based Rendering) texture workflows, involves the incorporation of maps for Roughness, Metallic, and Specular properties. These maps are developed to ensure accurate interaction with light and shadows in real-time rendering environments. In the case of armour, additional effects such as weathering, scratches, and dirt are layered to enhance realism, reflecting wear and tear from use. The outcome of this process is a fully textured model ready for integration, with optimised materials to balance visual fidelity and performance. Finally, the exported assets are rendered in standard formats compatible with game engines such as Unreal Engine or Unity.

Skinning

Skinning is the process of binding a 3D armour model to a character rig, thereby enabling smooth deformations during animation. This step ensures that the armour moves naturally with the underlying skeleton while maintaining its structural integrity and alignment. To this end, artists utilise software such as Maya or Blender to weight individual vertices or regions of the model, assigning them to specific bones in the rig.

For rigid pieces, such as plate armour, weights are concentrated on specific bones to maintain their solid appearance, while flexible components, like cloth, are influenced by multiple bones to allow natural motion. Automatic skinning tools, such as Maya's Smooth Bind or Blender's Automatic Weights, expedite initial setups but often require manual adjustments to refine deformation in problem areas, like joints or overlapping components. Special attention is given to layered or modular armour, where sections may overlap or shift during animation. Constraints and additional rigging elements such as helper joints are applied to preserve behaviour and visual coherence In the case of armour involving cloth or chainmail, soft-body dynamics and additional simulation setups may be incorporated. The final output is an armour model fully rigged and weighted, seamlessly integrated with the character's animation pipeline, supporting realistic movement and dynamic interactions in gameplay.

2.5.3 Standard Software

In the domain of 3D modelling, a range of software packages assume pivotal functions during distinct phases of the workflow. Minimizing the number of software programs used can improve coherence and avoid file transfer errors. Despite the importance of focusing on the usage of fewer different software to streamline the process, each package boasts distinct strengths and features, catering to specific tasks. Kovanen's work provides a comprehensive overview of the available packages and their respective impacts [Kov15]. Furthermore, Mitchell [Mit24] describes key developments in gaming relating to 3D and Virutal Clothing in her book "*Advances in Sewn Product Technology*"

Maya and Blender represent two comprehensive 3D modelling software solutions employed for a variety of tasks, including UV mapping, retopology and general modelling. Maya functions as a central tool for the creation of low- and high-poly models, in addition to topology adjustment for animation purposes. It is also utilised for rigging and animation, a functionality shared by Blender. It is evident that both offer modifiers, which are tools such as subdivide and mirror, that have the capacity to assist an artist's work. While both software solutions offer sculpting functions, ZBrush demonstrates superior performance in this regard. ZBrush is a 3D sculpting software that has been developed for the purpose of creating highly detailed models. It is particularly well suited for digital sculpting. It can be used as the primary software for the creation of armour from scratch, or for the sculpting of a pre-made 3D base model.

Marvelous Designer is a software program that is designed for the creation of clothing and textiles. It is capable of simulating fabrics in a realistic manner, which is why it is frequently employed for the modelling of garments for 3D characters. The garments created using this program can then be further processed in ZBrush or other programs [Mit24].

Adobe's Substance 3D Painter is a software program utilised for the texturing of three-dimensional models, with a particular emphasis on the creation of PBR (physically-based rendering) textures. The program enables the user to paint directly on the model, and offers a variety of ready-made materials and Smart Materials that expedite the texturing process. Substance 3D Painter is employed for the purpose of texturing 3D models, with a particular emphasis on the creation of PBR textures. It facilitates direct painting on 3D models and offers a variety of ready-made materials and Smart Materials that expedite the texturing process. Substance 3D Sampler is a program utilised in conjunction with Substance 3D Painter for the creation of textures, particularly for embroidery and other complex surfaces. Substance Designer is a node-based texture editor used for creating procedural materials, which can be used in Substance Painter or exported to other programs.

Furthermore, RizmovUV is utilised for the purpose of UV unwrapping, which is defined as the process of mapping two-dimensional textures to three-dimensional models.

2.6 Technical Problem Description

The subsequent chapter will examine the fundamental parameters that define the design of 3D armour, categorising them into technical and visual aspects. The establishment of these parameters serves as the foundational framework for the conceptualisation and subsequent development of the toolkit.

2.6.1 Parameters of 3D Armour Design

Given the information from the previous chapters, and considering the work of [Gol14], a basis of parameters has been created into which armour and its workflow can be arranged.

Technical Impact

- **Anatomical Coverage:** Armours are inherently linked to anatomical coverage, with components classified according to the body parts they protect, such as the head, torso, and limbs [Gol14]. Specific elements like helmets, breastplates, greaves, and gauntlets are designed to conform to the underlying wearer.
- **Rig Adaption:** The typical design 2.5.1 of armour includes is such that it is intended to fit a bipedal character model which is already rigged and will be animated.
- **Iterative Design:** The armour undergoes numerous modifications throughout the development process in order to ensure that it is consistent with the artistic vision and the requirements of the gameplay. 2.5
- **Standard Components:** It is common practice within the field of armour design to utilize repeating components, which may include shoulder, chest, arm, leg, belt and glove armour. The construction type also has a limited variety as presented by Goll (2014). Armour designs often utilize laminar structures (overlapping strips), segmented forms (articulated joints), padded armour, or plate designs (single, large sections) ([Gol14], A.5; A.3) or also more fantastical armour builds as highlighted by A.2 and A.1.
- **Modularity:** The integration of individual modularity into the game facilitates the customisation of armours for players. Typically, armour sets comprise between four and five components (as presented by 2.4.1).
- **Cloth Simulation:** A substantial proportion of armour is constituted of cloth elements, and the incorporation of dynamic elements, such as capes, robes, or draped fabric, necessitates the implementation of realistic cloth simulation techniques (2.5).
- **Variant Designs:** In the context of For Honor A.5, it is notable that numerous archetypes maintain their fundamental armour styles, exhibiting only variations in terms of armour designs, materials, patterns, or components.

Visual Impact

- **Cultural Identity:** The designs of armour are inspired by historical and cultural sources, reflecting specific periods and cultures. Previous games have been shown to exhibit recognisable patterns relating to particular eras and cultures (e.g. lacquered plates, neck guards and odoshi (lacing) are characteristic of Japanese samurai armour. They are seen in For Honor and Elden Ring).

- **Material Variety:** The use of recurring materials such as metal, chainmail, fur, cloth and leather are a constant basis for almost all medieval armour [Gol14]. Other materials, such as feathers and bones, are specific to certain types of armour (A.4).
- **Repetitive Patters:** A notable feature of medieval-style armour is the repetition of patterns in layered, overlapping and grid-like arrangements. For example, Plates or pieces are layered on top of each other, padded armour is arranged in grids or lines, and chain mail is arranged in repeating, offset hexagonal grids.
- **Damage States:** A significant proportion of armour sets have been found to contain damaged components, including dents, tears and scratches(A.1 A.4).
- **Anatomical Variety:** In the extreme case of Baldur's Gate 3 (2.3) and Dragon Age: Veilguard [26] armours in the game need to consider largely varying body types of male, female, body fat and muscle amount, sizes and races with possibly even animal-like features (e.g. tails, scales).
- **Embellishments and Engravings:** Armour is frequently characterised by elaborate engravings, embellishments and other decorative elements that serve to enhance its aesthetic appeal and narrative significance (A.2).

2.6.2 Analysis and Hypothesis

The increasing demand for comprehensive and varied three-dimensional assets in the field of AAA game development poses significant challenges for artists and developers alike. The creation of high-quality armour models for characters is a time-consuming process, necessitating the repetition of manual work to adapt designs to different body types, sizes and artistic styles.

These challenges are particularly pronounced during the initial stages of the modelling process, such as concept design and blockout, where iterative adjustments are essential for aligning assets with style guides and gameplay requirements.

The employment of procedural content generation (PCG) tools, such as those constructed with Houdini, signifies a promising solution to the aforementioned challenges. It is already customary in the gaming industry to utilise a variety of procedural workflows in order to enhance the content of their games. However, existing tools often lack the necessary specificity for armour modelling, particularly in terms of addressing the modularity and parametrization required for adaptive workflows that prioritise maintaining control of the artist. This research investigates the potential of Houdini-based tools to facilitate the initial phases of 3D armour creation through the provision of parametric control and modular design. The hypothesis is that this will enable more rapid iteration, a reduction in the workload, and consistent adherence to artistic and technical constraints.

The objective of this project is the development of an effective toolkit comprising discrete modules that facilitate the blockout process and enable artists to create high-quality, adaptive armour assets. This approach is consistent with the evolving role of the technical artist, who serves as a conduit between creative and technical domains with the objective of ensuring efficient workflows and asset quality in game production pipelines.

[26] `https://www.ea.com/en-gb/games/dragon-age/dragon-age-the-veilguard`, visited 12.01.2025

3 Requirements Analysis and Conceptualisation of a Procedural Toolkit for Armour Design

The gaming industry is increasingly reliant on procedural tools to meet the demand for detailed, customisable assets while adhering to strict deadlines and budget constraints. Conventional workflows, while meticulous and comprehensive, are labour-intensive, frequently necessitating extensive manual adjustments and repetition. Tools, notably those implemented with software such as Houdini, have emerged as a solution to various challenges faced by artists and developers by automating repetitive tasks, enhancing scalability, and preserving artistic integrity. This chapter delineates the conceptualisation of a Houdini-based toolkit tailored for the creation of armour, with a focus on the key parameters, functionality requirements, and theoretical foundations.

3.1 Industry References and Standards

This chapter builds upon the analysis of the workflow, software and game assets presented in the preceding chapter, contributing to the development of a comprehensive toolkit. The chapter undertakes a thorough examination of existing procedural tools, categorising them into distinct groups based on their application in gaming and 3D modelling. It then delves into a more detailed analysis of modelling tools, followed by an examination of tools designed for cloth, armour and character-related aspects.

3.1.1 Example Uses of Procedural Tools in Game Development

The implementation of procedural systems in game studios further highlights the relevance of procedural tools. Guerrilla Games utilized Houdini extensively in the production of Horizon Zero Dawn and its sequel Horizon Forbidden West, employing it for vegetation placement, river systems, and biome construction. [27,28] The vegetation systems demonstrate the advantages of procedural systems for game development when full artistic control is permitted through data, logic and assets. Michiel van der Leeuw, Technical Director of Guerrilla Games, presents an analysis of the substantial increase in data size of games and observes that a significant amount of "time is wasted in-between applications". In their presentation on the subject of "Making Tools for Big Games", he proposed solutions to this issue. [29] The issue of managing substantial quantities of data is becoming increasingly pressing due to the continuous augmentation in the size of games and the number of assets employed in these games. It is evident that these techniques provide a foundation for comprehending procedural principles, which are imperative for optimising modular tools for game production, as exemplified

[27] https://www.guerrilla-games.com/read/gpu-based-procedural-placement-in-horizon-zero-dawn, visited 21.12
[28] https://www.sidefx.com/community/horizon-zero-dawn-guerrilla-games/, visited 21.12
[29] https://www.guerrilla-games.com/read/making-tools-for-big-games, visited 21.12.2024

by Horizon Zero Dawn. Procedurally generated modular assets represent a valuable addition to world-building and the use of asset pipelines ease the workload on artists. Furthermore, data-driven procedural systems underscore the potential for deterministic and art-directable workflows. Guerrilla Games highlights how their tools assist in managing the substantial assets in their database, enhancing performance, reducing storage space requirements, and accelerating iteration times, among other benefits. [30]

3.1.2 Clothing and Character Modelling Pipelines in Games

For direct mesh manipulation, Blender and Maya provide a comprehensive add-on library that offers modelling flexibility and assistance. Blender's Auto-Rig Pro facilitates rapid character rigging, while Retopo MT assists in generating clean topology for sculpted models. The 3D Print Toolbox further extends Blender's capabilities by ensuring mesh integrity for 3D printing workflows. [31] Simply Cloth for Blender offers a system for generating and manipulating cloth simulations with preset fabric types, customisable parameters, and real-time physics adjustments.[32] For high-fidelity surface enhancement, Simply Micro Mesh employs a procedural approach to generating micro-level fabric structures, chainmail, and perforated surfaces. This tool replaces conventional displacement maps by instancing actual geometry across a surface, ensuring higher fidelity in close-up renders and real-time applications. [33]

EA's employment of procedural tools to fit helmets across thousands of character models in multiple franchises is a notable example of the application of such systems. This enabled rapid adaptation across varying head shapes and sizes. [34] The importance of procedural systems in minimizing the workload of artists was emphasised by Hoekstra, who stated that such systems "minimise the workload of artists by automating processes and streamlining workflows" (pers. comm.). He posits the following scenario: "if one were to consider 2,000 heads, each accompanied by eight distinct helmets of three different sizes, the total number of helmets would amount to 16,000. This is further compounded by the necessity to continuously enhance the fit of the straps and heads, necessitating constant re-fitting." (Freek Hoekstra, 3D Content Lead, 80.lv, 2018).

Akira Saito has employed Houdini's procedural capabilities in the context of science fiction mecha and voronoi methodology-based designs. While these designs are oriented towards mechanical science fiction systems and aesthetics, they demonstrate the generation of armour and sci-fi plates on any geometry. His approach to grid-based fracturing exemplifies the utilisation of procedural methods for the efficient handling of complex meshes based on existing anatomy. [35]

[30] https://www.guerrilla-games.com/read/creating-a-tools-pipeline-for-horizon-zero-dawn, visited 21.12.2024
[31] https://extensions.blender.org/add-ons/print3d-toolbox/, visited 21.12.2025
[32] https://blendermarket.com/products/simply-cloth?ref=247, visited 21.12.2024
[33] https://blendermarket.com/products/simply-micro-mesh?ref=247, last visited 04.02.2025
[34] https://80.lv/articles/procedural-modeling-for-gamedev/, visited 21.12.2024
[35] https://asaito.artstation.com/projects/L2mwgk, visited 21.12.2024

Anne Seppänen's work presented on artstation focuses on a layered costume system designed to allow for flexible and authentic costume design in *PaxDei*, an MMO with ambitious medieval-inspired character art.[36] The game's visual progression system embraces the historical practice of layering clothing, reflecting the distinctions between medieval social classes, such as peasants and aristocrats. The system is designed to accommodate multiple layers of costumes and armour, thereby enabling a wide variety of designs, shapes and volumes that are fully combinable with one another. A crucial component of the system is the use of automated corrective processes, implemented via Houdini, which integrate with the Maya character pipeline to ensure the costumes are adapted to varying body shapes. The integration of these tools facilitates the development of loosely fitted costumes alongside more intricately designed, finely crafted garments, thereby preserving the historical accuracy and versatility necessary for character creation.

Artur Musalimov's procedural gear system focuses on generating adaptable gear for humanoid bodies, with a modular design demonstrated through plate carrier armour. [37] This method saves time compared to traditional modelling, especially for creating multiple variations. While still in need of refinement, the procedural approach offers substantial production efficiency, making it ideal for games that require extensive customization, such as multiplayer titles.

Vimpassion's armour modelling guide in Houdini employs the use of rig join volumes for the purpose of selecting specific body mesh parts to be utilised in the fabrication of armour. [38]

Javad Rajabzade highlights the efficiency of procedural workflows in Substance Painter through the use of Smart Materials, which are adaptive material presets responding dynamically to an object's curvature, ambient occlusion, and surface details.[39] These materials allow for non-destructive, parametric adjustments, enabling quick iteration and style consistency across different assets.[40]

3.2 Procedural Techniques: Context, Application, and Implementation

"Procedural techniques are code segments or algorithms that specify some characteristic of a computer-generated model or effect." ([Ebe+02], p.1). As Ebert et al. (2002) demonstrate, a significant strength of procedural generation is its ability to abstract complexity. Rather than storing all details of a model explicitly, a procedural system encapsulates them within an algorithm or function, dynamically generating required elements as needed. This reduces storage overhead and shifts computational effort from manual design to automated generation ([Ebe+02], p. 2).

[36] https://www.artstation.com/artwork/dKZK5e, visited 21.12.2024
[37] https://www.artstation.com/artwork/DLmym9, visited 21.12.2024
[38] https://youtu.be/Iwd3_m7O4wM?si=FZjISN99PrXQtz7w, visited 21.12.2024
[39] https://www.artstation.com/artwork/6NvbPO, visited 21.12.2025
[40] https://helpx.adobe.com/substance-3d-painter/using/materials-smart-materials.html%7D, visited 21.12.2025

Procedural Content Generation (PCG) has revolutionised 3D and game development workflows by providing an efficient, adaptable and creative approach to asset creation. Through the use of algorithmic processes, PCG enables the production of complex, high quality content at scale, significantly reducing manual effort while promoting iterative design capabilities. Proceduralism emphasizes the creation of systems that generate content algorithmically, guided by rules and user-defined constraints.

Modern procedural modeling systems often employ grammar-based rule sets to structure content generation. One such method is Lindenmayer systems (L-systems), originally devised for simulating plant growth. L-systems employ recursive string rewriting rules to iteratively generate complex structures from simple initial definitions [USF10]. While initially limited to organic forms, parametric L-systems extend this approach by introducing numerical parameters, allowing for controlled deformation and structured asset generation.

3.2.1 Core Aspects of Procedural Content Generation

The core of the rule definition is to define how to get from a starting point (A) to a desired result (B). The rules describe the steps and transformations that are necessary [SA17].

A flexible approach to the development of the procedural algorithm architecture is predicated on three design properties that are deemed to be desirable: *modularity, specificity and sequentiality.* Modularity is defined as the capacity for isolated changes to components. Specificity is defined as the aim for the most specific area of expression possible. Sequentiality is defined as the simulation of the designer's step-by-step thinking and working process [SA17].

The categorisation of PCG systems can be approached through various lenses, with a notable distinction being the differentiation between *data-intensive and process-intensive systems*. The former operates with substantial man-made building blocks and utilises lightweight algorithms, whereas the latter transfers design responsibility to an algorithm capable of making creative decisions [Smi21]. Furthermore, Smith's work, "*Understanding Procedural Content Generation*", delineates the five predominant categories of content generators: *generation as optimization, generation as constraint satisfaction, generation with grammars, generation as content selection, and generation as a constructive process* [Smi14]. The most vital points in relation to this prototype are as follows: generation as constraint satisfaction: content is generated that must fulfil certain conditions or restrictions; generation with grammars: systems that use a grammar to create content, where the rules determine the structure and appearance; generation as content selection: content is selected from a library and assembled, which can be simple but prone to recognizable patterns; generation as a constructive process: content is built step by step, often with complex algorithms [Smi14] [SA17] [Smi15].

Procedural content can operate in a *fully autonomous manner or as a collaborative tool*, thereby bridging the gap between designer goals and algorithmic outputs [Smi21]. In their book, Ebert et al. propose another classification of procedural modelling. This taxonomy encompasses two broad categories: *explicit and implicit*. Explicit methods are characterised by the delineation of individual points and vertices, while implicit methods are predicated on the answering of queries concerning specific spatial locations ([Ebe+02], p. 12).

3.2.2 Prototyping for PCG

The process of prototyping is instrumental in ensuring the compatibility of the technical infrastructure, facilitating rapid iterations in support, and enhancing the fluidity of the user experience. Furthermore, as Kasurinen et al. (2013) observed in their interviews regarding tools, prototyping and design tools are vital in both the pre-production and implementation phase of game development [KSS13].

Prototypes function as interactive tools to facilitate the resolution of design issues and exploration of the design space. They assist designers in the generation and capture of ideas, enhance communication within the team and with stakeholders, and conserve resources such as time and financial expenditure by gathering early feedback and evaluating design decisions [Hol20]. In the works of Nelson et al. (2009), the study conducted yielded insights into the manner in which designers' interactions with the prototypes of potential tools refined the model of the early-stage game design process and the capabilities of the tools, thereby informing developers of the tools' potential contributions to the achievement of the ultimate goal [NM09].

In the prototyping process, two primary approaches are distinguished: divergent and convergent activities. Divergent activities gather information to expand the design space and generate ideas, while convergent activities refine concepts through selection and narrowing down. Prototypes can be categorised into two distinct types: exploratory and experimental. Exploratory prototypes are employed to explore novel concepts, while experimental prototypes are utilised to validate system requirements. PCG serves as a rapid content creation tool, particularly well-suited for early prototyping. It facilitates rapid exploration of diverse design ideas, a feature that is especially advantageous in the pre-production phase. The focus is not on optimising the algorithms, but rather on leveraging the strengths of PCG, namely its capacity for rapid content generation. Procedural systems can function as tools to assist designers in specific stages of the design process, enhancing their workflow. These tools can facilitate the creation or iteration of content by designers [Hol20]. Segmentation of procedural tasks into modules enables isolated modification without disruption to larger systems. Sequentiality in procedural design simulates the designer's iterative process, improving predictability and customisability of outputs. Modules can be tailored, substituted, or reorganised individually. It is argued that PCG is particularly suitable for quickly and inexpensively producing raw content that can serve as a basis for further design decisions [SA17].

3.3 Technical Art and Tool Development in Game Production

As games become increasingly complex, the support structure within development teams also expands. Game teams generally comprise a variety of roles, including management, programmers, engine developers, 2D and 3D artists, with further specialisation according to project scope. These teams encounter mounting challenges related to content creation, which necessitate specialised support. In this regard, technical artists assume a pivotal role ([Rod24], p. 5-8).

3.3.1 The Role of Technical Art

As games evolve in terms of both scope and artistic intricacy, the role of technical art has become increasingly pivotal, functioning as a nexus where artists and programmers intersect. Technical artists, uniting expertise of artists and programmers, play a crucial role in the development of streamlined systems, which have the capacity to accelerate processes and enable larger teams to function with greater efficiency [Rod24]. [41] The role of technical artists is to serve as a conduit between the artistically oriented staff and the programmers, ensuring that the tools and workflows are aligned with the technical demands of the game engine, while concurrently preserving the aesthetic vision of the project [Rod24].

This paradigm shift is underpinned by the recognition of the pivotal role that tools play in facilitating the scaling of development operations. As the industry continues to evolve, the demand for producing increasingly intricate, voluminous, and aesthetically pleasing content necessitates the implementation of more advanced technical infrastructure. Roda describes that a foundational principle of this evolution asserts that any task which is performed more than twice should be automated or streamlined through the utilisation of a tool or script. This concept underscores the mounting demand for efficiency in an environment where every minute spent on manual tasks can adversely impact the overall pace of production.The implementation of such tools has been shown to reduce errors, whilst also enabling greater oversight of larger datasets. Consequently, teams are enabled to engage in protracted iteration while upholding elevated standards of quality.

A technical artist's primary responsibility is the creation and maintenance of art pipelines, which are instrumental in the streamlining of the asset creation process. [42] This process involves optimising communication between different departments, which is a crucial aspect of tool development. Another pivotal factor in this context is the target audience. Technical artists must consider whether a tool will be used by themselves or by multiple departments. This has implications for the tool's design, usability and accessibility. Furthermore, the project timeline can greatly influence the development of tools. In some cases, the urgency of meeting deadlines can take precedence over the pursuit of perfection. This necessitates a balance between speed and quality in tool design and implementation [Rod24]. [43]

The technical artist, functioning as both a guardian and a translator, ensures that artists' problems are addressed swiftly and effectively. They serve as the liaison between creative and technical departments, deciphering complex requirements and ensuring that each side's objectives are met. By understanding both the programming and artistic needs of the project, the technical artist's role becomes integral in fostering collaboration and bridging the divide between artistic and programming teams[Rod24]. [44]

[41] https://www.gamedeveloper.com/programming/the-code-art-divide-how-technical-artists-bridge-the-gap, visited 20.12.2024

[42] https://www.gamedeveloper.com/programming/the-code-art-divide-how-technical-artists-bridge-the-gap, visited 20.12.2024

[43] https://gdcvault.com/play/1022289/Technical-Artist-Bootcamp-Proceduralism-and, visited 14.01

[44] https://www.gamedeveloper.com/programming/the-code-art-divide-how-technical-artists-bridge-the-gap, visited 20.12.2024

Additionally, the increasing demand for customization, scalability, and rapid prototyping in game development further highlights the importance of tools designed for specific workflows. For instance, a tool may be designed for modular gear generation for characters, allowing rapid adaptation to various body types and minimizing the need for extensive manual adjustments, which significantly speeds up production times. These tools not only enable faster iteration but also provide a scalable solution that adapts as the development progresses, saving substantial time when multiple variations of assets are required[Rod24].

3.3.2 Design Considerations for Tools in Technical Art

In creating tools, technical artists must weigh several factors. First, the scope of the task is a key consideration: whether the task is frequent or complex enough to warrant a dedicated tool or script. For tasks performed infrequently or without considerable impact on the overall workflow, a quick, temporary solution might suffice. However, when tasks are repetitive or crucial to the project, the development of a dedicated tool becomes indispensable. Additionally, technical artists assess the return on investment when considering new tools—if the effort and time required to develop a tool outweigh the potential long-term benefits, then other alternatives might be explored [Rod24]. The objective is for tools to evolve organically over time, thereby empowering artists to build upon extant knowledge rather than replacing existing workflows with new ones. This natural progression should result in a steady advancement of the toolset, as well as the skills and mindset of the artists. By allowing for reusability and evolution, the pipeline can facilitate continuous growth in both the tools and the artists' abilities. "Tools must be streamlined and constrained in order to be both rigid and flexible" ([Bus21], 06:25). Buschauer (2021) also mentions the development of a tool during the production phase, in conjunction with the integration of workflows already established by artists on a project, carries with it a certain degree of risk. This is due to the potential for the introduction of bugs, alterations, iterations, and the potential abandonment of previously developed tools ([Bus21], 26:36)

Roda describes that the target audience constitutes a pivotal factor in the development of tools. Technical artists must consider whether a tool will be utilised solely by themselves or if it will be disseminated across multiple departments. This has implications for the tool's design, usability, and accessibility. The tool creator is responsible for capturing the user's or designer's expertise [Smi15]. Furthermore, the project timeline exerts a substantial influence on the development of tools. On occasion, the urgency of meeting deadlines can supersede the pursuit of perfection, necessitating a balance between expediency and quality in tool design and implementation [Rod24].

3.4 Design Parameters and Functional Objectives

The development of a procedural toolkit for armour design necessitates a structured framework that aligns with industry standards, technical constraints, and artistic flexibility. This chapter outlines the core design parameters that shape the toolkit's functionality, ensuring it meets the practical needs of artists while integrating seamlessly into existing production workflows. The functional objectives are derived from an analysis of traditional modelling pipelines, procedural methodologies, and expert insights. The chapter thus establishes the foundational guidelines that govern the implementation of the toolkit, ensuring that it remains both technically robust and artist-friendly.

3.4.1 Software Compatibility for Toolkit Integration

The aforementioned tools and the chapter on 2.5.1 highlight the usage of software by the artists. The software Houdini by SideFX mainly finds usage in the VFX Industry. Movies from Marvel Cinematic Universe to Indie Movie Productions utilize it. (source Houdini production showcase) But the procedural use cases of it also allow a good utilisation within the games industry. In games, often many sets of data, large-scale ideas and detailed to realistic simulations and graphics need to be constructed, especially in the AAA game space. Depending on the production, the Houdini Technical Artist is a special role within the team. "Houdini interesting for the game developers is the ability to solve problems of scale and deal with things that are simply too great in number to do by hand or that require a lot of precision, especially in combination with tedious repetitive work which will bore people and make them make mistakes". [45]

Houdini represents a state-of-the-art, established software for procedural tools and pipelines, with existing integrations in Maya, Unity and Unreal Engine. The software is accompanied by comprehensive documentation and support [Sid24], and functions on the basis of procedural methods of model graphs and hierarchical descriptions of models [USF10]. The latest version of Houdini, 20.5, incorporates existing functions such as enhanced Rig-work, which can be an essential part of the armour toolkit. [46] It offers a built-in plugin that facilitates integration with various engines, including Unity and Unreal, as well as Maya and 3Ds Max. [47]

In order to create a tool in Houdini and use it in Maya, Houdini Digital Assets (HDA) are the key. [48] [49] Houdini Digital Assets (HDAs) are defined as reusable, encapsulated toolkits or nodes that are created within Houdini, SideFX's flagship 3D procedural software. An HDA consolidates a network of procedural operations (a "node tree") into a single, manageable package, thereby enabling users to perform complex tasks through a simplified interface. These assets are versatile and customisable, offering parameters that users can adjust without needing to alter the internal logic of the asset itself.

These assets function as modular and adaptable solutions for automating repetitive workflows or creating parametric models, textures, and simulations in a highly efficient and user-friendly manner. The scope of HDAs encompasses basic modelling tools as well as sophisticated systems for particle effects, simulations, or procedural world-building. Once a node setup has been consolidated into a digital asset, it is possible to create a custom user interface for the newly created node. This interface allows all parameters from the underlying setup to be linked ("promoted"). The utilisation of these parameters enables users to adjust the node's output in an efficient manner.

[45] https://80.lv/articles/procedural-modeling-for-gamedev/, visited 26.12.2024
[46] https://www.youtube.com/watch?v=9iOqERDE6jI&ab_channel=Houdini, visited 26.12.2024
[47] https://www.sidefx.com/products/houdini-engine/plug-ins/maya-plug-in/, visited 26.12.2024
[48] https://tokeru.com/cgwiki/HoudiniHDA.html, visited 26.12.2024
[49] https://www.sidefx.com/docs/houdini/maya/assets.html, visited 26.12.2024

HDAs can also be utilised in other plugin-compatible software mentioned before. Houdini Engine enables direct manipulation of HDA parameters within Unreal Engine or Maya, allowing real-time updates to assets or environments. [50]

3.4.2 Conditions and Requirements of the Toolkit

A toolkit offers selective assistance to artists during the creation process, focusing on repetitive tasks and parametrized controls without removing the artist's creative freedom [NM09]. Full automation could eliminate artistic input, reduce the individuality and style consistency, and potentially result in a loss of quality and creativity. Smith et al. describes it as "design support systems rather than design replacement systems". Gavin Annand highlights the pitfalls of over-reliance on procedural generation, noting that excessive automation can undermine artistic expression and lead to homogenized content. [51] The effort necessary for a full automated armour generation system is unrealistic in the face of efficiency.

The conditions and requirements for the proposed toolkit derive from a careful analysis of the armour creation workflow, current software capabilities, and practical challenges faced by 3D artists in game development pipelines. They are derived from the proposed references of tools mentioned before and lay a ground work of state-of-the-art procedural tools. Conditions are defined as the prerequisites for the establishment of a macro-level framework and environment. Requirements are delineated as the specific functionalities and features that the toolkit must deliver. These are categorised as micro-level, technical and detailed features, operational details and outputs, and precise, measurable goals for implementation.

The prototype of the toolkit does not aspire to perfection; rather, it is guided by a philosophy of rapid, scalable solutions that bear resemblance to guerrilla problem-solving tools. These tools prioritise addressing repetitive pain points in workflows without necessitating the disruption of existing systems [Rod24]. The tool must be available for rapid changes and adaptions [KSS13]. The blockout phase necessitates the utilisation of robust tools that facilitate flexibility and rapid iteration without compromising precision. This phase establishes the foundational shapes and proportions of armour elements (see chapter 2.5.1). Errors rectified at this stage result in substantial time savings in subsequent processes such as detailing or texturing (see also 5.2). These observations substantiate the emphasis on conditions and requirements that enhance efficiency, compatibility, and usability during the blockout stage.

It is imperative that the toolkit is fully compatible with contemporary versions of Maya (2025), Houdini (20.5) and ZBrush, aligning with their native workflows and constraints. These are industry-standard tools used across various stages of modelling that are compatible with one another, which the Houdini-based Toolkit needs to adhere to and integrate into (derived from 2.5.1 and [Smi15]). Furthermore, it is essential to ensure seamless usability through geometry-level HDAs in Maya via the integrated Houdini plugin. The utilisation of HDAs facilitates the modularity of the toolkit, thereby enabling the independent use of its various components without the necessity of reliance on external or other module elements (see 3.2.2). HDA's allow for interoperability of Houdini inside Maya, without the artist having to utilise another

[50] `https://www.sidefx.com/docs/houdini/maya/#using-houdini-engine-for-maya`, visited 26.12.2024
[51] `https://www.superjumpmagazine.com/by-design-procedural-generation/`, visited 02.02.2025

software outside of their standard workflow (see 3.4.1). The toolkit modules are required to adhere to Maya-compatible base data formats (e.g. .obj, .fbx) and to require minimal manual intervention, as artists frequently iterate on models. Excessive complexity or manual steps would hinder adoption and reduce the time efficiency of the toolkit [KSS13]. Adherence to common data formats allows for avoidance of conflict of data and saves, ensuring seamless integration. The creation of inputs and settings that are limited to a small number and intuitive eases the handling of the tools, as well as allowing easier comprehensive documentation that ensures the scalability and usability over time. It is important to note that not all users have the same technical expertise, and intuitive, artist-friendly UI elements (sliders, drop-downs) allow broader adoption of the toolkit without excessive training [Rod24] ([Bus21], 05:03). This approach fosters a non-disruptive workflow, independent of the toolkit, with the potential to integrate seamlessly into existing methodologies employed in the domain of armour creation.

In order to comply with the rigorous technical specifications that are required of armour models, there are a number of aspects that must be taken into consideration. These include optimised polygon counts, clean topology, UV unwrapping, and compatibility with prominent game engines such as Unreal Engine and Unity [Ter17] [Kov15]. It is imperative to prioritise optimisation and rendering time to ensure performance efficiency. High-resolution models are not viable in memory-constrained environments and can lead to conflicts due to hardware limitations during the artistic workflow. Unoptimised tools can result in lag, extended loading times, and crashes, thereby impeding workflow. The toolkit will necessitate uniform naming conventions for existing character rigs to facilitate long-term development. The body meshes of models should have anatomically fitting UVs (e.g. arms, hands, head, body, legs,...). Utilising naming conventions and universally fitting UVs allows for the avoidance of errors in the procedural tools, which depend on correct inputs to function. When all rigs are named consistently, the same tool can use different rigs and meshes in the same manner without needing adjustments or further manual input (cite on procedural basis). Similarly, all body meshes with UVs adhering to an anatomical basis support correct output of tools. However, the trade-off must be that the time taken to create a module or part of the tool must be outweighed by the benefits gained. A general rule of thumb is that a script can be applied when a task is performed more than two or three times [Rod24].

Guerrilla Games emphasises the significance of data-driven, deterministic, and locally stable tools that are not only accessible but also art-directable, thereby fostering seamless collaboration between artists and technical teams while maintaining workflow efficiency and creative control [Bus21]. Adhering to these conditions fosters modularity and flexibility. It is imperative that tools demonstrate adaptability to a range of potential use cases and variations of the workflow, with stability and flexibility being paramount. Modularity is a key feature that facilitates easy adaptability and scalability, and new styles of armours, body types, inputs and patterns need to be integrable into the toolkit. Rapid iteration and customisation are crucial to meeting the evolving artistic and technical demands of the creative process, and processes and requirements can change and need to be adapted as fast as possible over the course of developing a game.

Buschauer (2021) draws attention to "filtering ideas from artists" and that "every project is different" [Bus21]. To illustrate the practical application of requirements, consider the following examples: Developer A is tasked with the design of an armour set primarily comprising cloth and chainmail for a humanoid male model, while Developer B is designing padded armour combined with metal laminar plates for a non-humanoid female gnome model. When adhering to the requirements of the input models, the goal remains that both developers can utilise different individual parts of the toolkit or a unified function.

3.5 Toolkit Functionalities: Derived Functionalities from Armour Parameters

Through a thorough analysis of existing workflows and their technical requirements, a set of core functionalities has been derived to address the technical and visual challenges associated with armour asset creation. The toolkit utilises both explicit and implicit procedural methods: explicit methods construct structured assets such as base meshes and segmentation masks, while implicit methods focus on adaptive fitting and pre-existing mesh deformation. Like Philipp Buschauer (2021) presents in his work on "*Building An Artist Friendly CFX Pipeline*", this toolkit also provides templates with which artists can work. Once the initial setup is complete (segmentation), they iterate. Thereafter, artists are at liberty to utilise individual working areas for the finalisation of their work (modules) ([Bus21], 09:53).

The prototype toolkit under consideration has been designed to deliver an efficient and adaptable solution for the creation of armour across a range of body types, rigs, UVs and meshes. The toolkit aims to ensure a consistent and predictable clean output with solid topology while facilitating fast generation and enabling rapid iteration. The singular functions should come together as a type of framework to access easily with real-time rendering in the software of choice. Ideally, it would seamlessly integrate with industry-standard software such as Blender and Maya as a non-destructive modifier, facilitating rapid generation and iteration. The system would intelligently adapt to curvature and volume, minimising the need for manual adjustments while ensuring clean topology and realistic deformation. By facilitating the streamlining of base mesh creation, the toolkit would enable artists to prioritise refinement over repetitive modelling, thereby bridging the perceived gap between procedural efficiency and creative control.

3.5.1 Base Mesh and Component Generation for Modular Armour and Anatomical Coverage

This tool covers armour parameters (as presented in2.6.1): anatomical coverage, rig adaption, standard components. In ZBrush, artists usually utilise a brush for masking to paint a paint to be modelled further (see 2.5.1). A system for anatomical masking can automate the segmentation of the character's base mesh into distinct anatomical regions (e.g. arms, chest, legs) using rig data to define appropriate armour placement, thereby accelerating the armour creation process by providing predefined masks that ensure seamless alignment between the base body and modular components. The body splitting and masking tool is a base component of the altered armour creation process. This is a crucial step in the fitting of armour components, such as helmets, gauntlets, and breastplates, to the character. The tool employs rig data and the UVs of the body mesh to automate the creation of masks for specific body parts, ensuring precise alignment with the natural anatomical boundaries. The refinement of mask edges to follow the contours of the character facilitates clean segmentation for armour fitting, thus accelerating the modelling process by reducing time spent on manual segmentation and ensuring consistent topology alignment for future detailing. The tool can be used repeatedly to access specific body parts when modelling a part of an armour, or alternatively provides a fully split mesh. The employment of diverse methodologies for segmentation could assist the accommodation of preferences concerning masking and the intricacy of armour components during the blockout stage, thereby enabling fast adjustments. When adhering to general data of mesh and rig, it should adapt to any body, able to adhere to shapes with consistent output with which to work with quickly. The challenges posed by model sizes, the number of splits necessary, and the methodologies employed, such as Voronoi, clustering, or grids, must be addressed to ensure the efficacy of the model.

3.5.2 Workflow and Efficiency Enhancements

This tool section attempts to address the following armour parameters: reusability, standard components, variant design. As seen in 2.4.1, during large scale productions armour is manufactured in substantial quantities. The toolkit can integrate a modular asset library, allowing for the reuse of previously created armour components. While primarily beneficial for large-scale productions, the system also supports rapid prototyping by enabling users to quickly access and modify existing designs. The concept of reusability is of particular significance in the context of a pipeline designed with the needs of artists in mind. This attribute enables the transfer of components from one project to another, facilitating a seamless integration of elements across diverse artistic endeavours ([Bus21], 05:52). The various layers and shapes of armour that are available for exploration are analogous to a 2D method of "*photobashing*" [52]. "Photobashing" is a digital art technique that combines photographic elements with painting and manipulation to create highly detailed concept art, illustrations, or textures by combining many individual elements together. This method allows for experimentation and utilising existing works to achieve a desired vision or to generate alternative options. Artists can utilise a library of shapes and forms to expeditiously assemble armour elements into place, a method that is particularly advantageous for games such as *Baldurś Gate*, which feature recurring shapes and design motifs. This tool enables elements to be projected and adapted, by procedural constructionist approach, across different armour sections with minimal manual intervention. As a game project progresses over time, numerous armour models are created, which are then incorporated into the library and can be further utilised for future modelling, fitting or serialisation. As illustrated by Freek Hoekstra's work on the helmet fitting system, Houdini can serve as a pivotal instrument for the management of voluminous object data. Houdini's PDG *(Procedural Dependency Graph)* [53] facilitates batch processing and efficient stashing, thereby constructing a pipeline that integrates diverse processing phases in form of HDAs. It can facilitate the serialisation of armour and the searching for specific models based on filters such as armour parts, armour type and armour name. The concept under discussion is that the artist's interaction with the system is minimal, consisting of a limited

[52] https://www.theseus.fi/bitstream/handle/10024/787725/Viro_Vi.pdf, visited 20.01.2025
[53] https://www.sidefx.com/products/houdini/pdg/, visited 02.01.2025

number of button presses and manual search inputs. This would establish a foundation for a workflow initiation, enabling the artist to utilise serialised armour for mesh prototyping and fitting. Alternatively, it could function as the culmination of a workflow, with the purpose of preserving the created armour and its variations.

3.5.3 Fitting

This tool category, consisting of two sub parts of Morph Fitting and a Wrap Tool, addresses the parameteres of: anatomical variety, variant design, and anatomical coverage.

A morph-based adaptation system ensures that already modelled, or generated, armour components conform to different body types or body parts, and species variations in games like *Baldurś Gate 3 (2023)*. The goal is to utilises morph targets to preserve anatomical integrity and streamline asset scalability across multiple character rigs. An additional part is procedural wrapping tools for cleaner output on closed shapes. Armour elements such as belts, straps, and wrapped fabric require flexible adaptation to various shapes.

The Morph Adaptation Tool is an attempt at addressing the challenge of retargeting meshes and therefore accommodating diverse body types, including gender variations, races, and species-specific anatomical features. By leveraging morph targets and deformable rig systems, this tool ensures armour components adapt precisely to the underlying character structure without requiring manual reshaping for each individual character. The tool utilises pre-defined morph targets based on variations such as body size (thin, muscular, obese), gender, and masks. The tool's parameters can be used to control fit tension (e.g., looser fit for robes vs. tighter fit for plate armour). The tool facilitates iterative prototyping for multi-character armour designs and can be utilised as a form of pre- or post-processing in combination with the library pipeline. The tool's efficacy can be impeded by a variation in body type meshes and topologies, which compromise the integrity of the model outputs.

Due to possible generation challenges, an additional tool is required in order to abstract the Morph Adaption Tool for fully encompassing models or with the goal of generating a new fitting mesh like belts and wrapped rope. The standard workflow for the creation of a belt or wrapped cloth typically involves the detailed sculpting and simulation of cloth in software such as Marvelous Designer or Maya. A curve is manually drawn in the software to delineate the waist or hips, and this is then given a profile. Houdini, in collaboration with technical artists, has an integrated node that facilitates the wrap-around process, which can then be employed in this instance. [54] A challenge that emerges with this is the time required for the node calculations to be completed, as well as challenges with topology and satisfactory results with disjointed geometry and proportions.

3.5.4 Pattern and Detail Generation

As described in the 2.6.1 section of the fundamentals, armours often present repeating patterns, damage states, components and distinct groups of materials based on style and culture.

[54] `https://www.sidefx.com/docs/houdini/nodes/sop/labs--simple_rope_wrap-1.0.html`, visited 26.12

To accommodate repetitive patterns which can be modelled, such as chainmail, lamellar, and odoshi lacing or other patterns as seen in 3.1, the toolkit can enhance already existing modifier tools of scattering, by employing grid-based parametric solutions. These facilitate the creation of structured designs that would otherwise require significant manual effort or be managed through the creation of textures in later processes. The system supports different pattern densities and alignment methods, ensuring seamless transitions across armour sections. The creation of repetitive, intricate patterns unique to various armour types is simplified by tailored parametric solutions. This is essential for armours with complex structures, such as chainmail or lamellar armour, which would require more manual labour if created traditionally. The patterns can be based on a variety of parametric grid-based techniques that can be easily adjusted. For the prototype, some of the garanteed patterns that can be generated should be:

- **Laminar, Padded Armour, Odoshi**: Scale and padding are often base parts of various armours and need to fit body curvature. The distinctive laced designs of samurai armour can be emulated by utilising grid settings on meshes, with adjustable density and custom inputs.
- **Plates**: abstracts from the grid system for easier handling to provide articulated plate patterns through custom pattern input for overlapping armour with adjustable spacing and curvature fitted to the body part.
- **Chainmail**: has the capacity to create preliminary designs of chainmail patterns, as defined by a style guide (e.g. 4-in-1, Elven Chainmail,...). Alternatively, it can utilise settings to abstract hexagonal grid offsets for custom input chainmail. Following this, the chainmail can be bent to an existing mesh, in a manner similar to the previously mentioned common workflow.
- **Fur**: Fur and feathers are frequently represented in games by a number of textured planes scattered over a base mesh. The concept involves the utilisation of an alpha and atlas map of the employed fur texture to cover a mesh with parameters abstracted from the grid.

The Pattern Modules can be implemented as individual Houdini HDAs in one package to allow accessibility, with built-in flexibility for integrating new patterns as game-specific and style demands evolve. It is noteworthy that all pattern modules exhibit a high degree of uniformity in their user interface, incorporating sliders, toggles and texture inputs that facilitate streamlined interaction. The outputs are meshes that are particularly well-suited for use as base meshes; however, it should be noted that chainmail meshes can result in a high density of polygons and therefore be more suitable for high-poly models and baking. It is evident that by combining rapid prototyping capabilities and templates with customisation options, the Pattern Modules improve the iterative design process and variant creation like for armours from *For Honor (2017)* (A.5).

A fundamental challenge in procedural pattern application is ensuring **seamless texture repetition**. When a texture needs to cover a large surface, it is tiled across the geometry. However, visible seams can emerge due to misaligned pixel values at the texture edges. Techniques such as **retouching or procedural blending** mitigate these artifacts, ensuring **continuous visual integration** ([Ebe+02], p.23). This principle is crucial for armour detailing, particularly in chainmail or patterned leather elements, where seamless procedural generation enhances realism and artistic consistency. Like Buschauer (2021), a number of presets and recognisable parameter should be utilised and patterns should be made from individual, adjustable presets (e.g. single pad, single chainmail pattern,...).

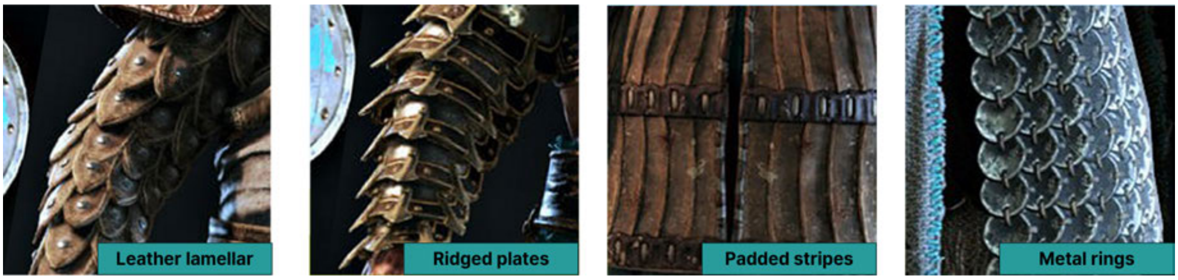

Figure 3.1: Examples of Grid patterns that can be made utilising the HDA. All examples taken from *For Honor (2017)*

3.6 Toolkit Adaptability

The toolkit's adaptability is a key strength, ensuring its continued value across diverse applications and evolving project needs. Primarily designed for the creation of 3D armour with a focus on medieval European armours of the bipedal variety, its modular structure and procedural nature enable expansion. The toolkit can be used for armouring creatures other than bipeds, such as horses or monsters, or for contributing to different armour styles.

The toolkit could support non-humanoid shapes by allowing for adjustments based on character rigs, accommodating designs with additional limbs, unconventional body types, or intricate patterns. This flexibility draws inspiration from works such as Akira Saito's procedural designs, enabling the creation of robots. New patterns of model-splitting, different rigs and extended modules can be used to adopt the same system in different ways, as the inputs (rig or mesh) remain universal.

The toolkit utilises universal procedural principles, which find application across a range of use cases in game development. Its modular development framework facilitates iterative updates and expansion, thereby supporting the creation of new asset libraries and procedural rule sets that are tailored to different projects. In the majority of digital armours, components such as spring-latches, linings, straps, hinges and rivets are not visible, but can be incorporated as a custom module if required. Additionally, the toolkit incorporates a serialisation pipeline for extended data saving and reuse. This feature ensures that assets and their procedural configurations can be efficiently stored, shared, and reused, enhancing workflow consistency and enabling cross-project, engine, and software adaptability.

4 Implementation

The implementation of the procedural armour toolkit is centred on the translation of conceptual methodologies into functional tools within Houdini.The present chapter details the precise development and resources utilised, underlying system architecture, and the integration of procedural techniques for each toolkit aspect.The toolkit is structured around modular Houdini Digital Assets (HDAs), each designed to address specific workflow challenges such as segmentation, fitting, and pattern-based asset generation. The subsequent sections delineate the technical framework, core functionalities, and the considerations undertaken to ensure compatibility with industry-standard software while adhering to the conditions and parameters set in previous chapters. Furthermore, the analysis extends to the altered workflow utilising these implemented tools and the support systems required.

4.1 Toolkit System Architecture

The toolkit is structured as a non-destructive, parameter-driven system that allows for quick extensions and uniform solutions while preserving iterative control over HDAs. Built on Houdini's node-based architecture, each HDA operates independently yet interacts through shared attributes, geometry inputs, and data-driven controls. The system follows a modular data flow model, where geometry progresses through sequential processing stages, each refining the asset through attribute modifications and procedural transformations.

The toolkit operates with three primary inputs to keep the necessary manual data to a minimum:

- a base mesh (including UVs and in case of a human base mesh, a rig),
- an armour piece or chainmail link for scattering,
- planes for the wrap tool.

The HDAs utilise common acceleration techniques employed in procedural methods. The toolkit functions by generating only the necessary data, conceptualised to operate in real time, thus excluding pre-computation acceleration methods. However, it utilises multiple-level procedural models and hierarchical resolution scaling ([Ebe+02], p. 290). It benefits from hard-coded presets for accessibility and maintains overview through subnetworks and in-software documentation. Network structures are subdivided into function-specific subnetworks, with additional optimizations leveraging Houdini Labs nodes for enhanced efficiency. Figure 4.1 shows an example of the internal subnetwork structure from the Anatomical Masking HDA. Extensive visualisations on the structure of the toolkit HDAs networks can be found in the appendix E.3. The integration of a Blender-Houdini export-import bridge further broadens usability, allowing artists to retain procedural control within their preferred software environments (see figure 4.2.).

A master HDA functions as the primary interface from which to work from, its content structure visualised in 4.3, integrating multiple sub-HDAs, each handling a distinct aspect of the toolkit.

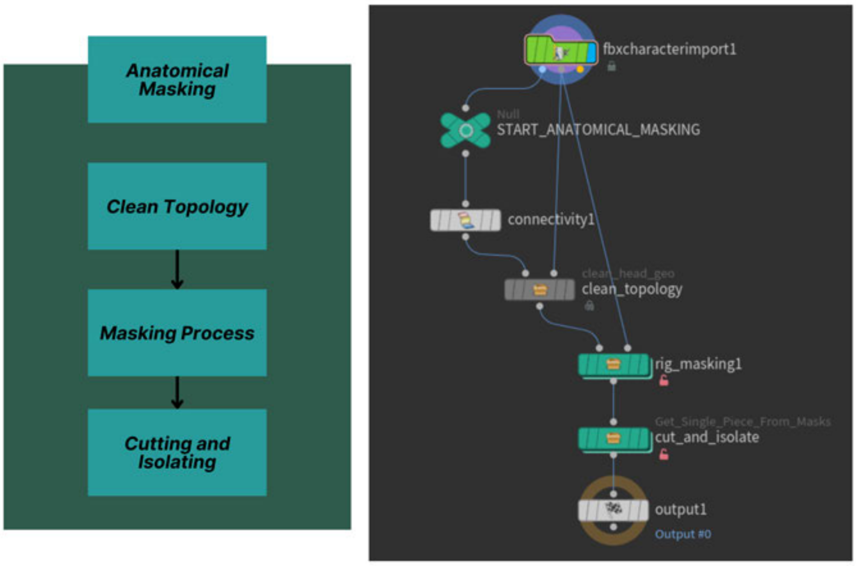

Figure 4.1: Exemplified comparison of Flow Chart of Process and Subnetworks of HDA in the Toolkit

Figure 4.2: Abstraction of Procedural Modelling System as presented by [Ebe+02] to visualise the support bridge from the modelling software to the toolkit in Houdini.

- **Anatomical Segmentation HDA**: Divides the base mesh into anatomical regions, utilizing subnetworks for geometry cleanup, masking, and controlled part extraction.
- **Pattern Generation HDA**: Generates parametric armour components, including chainmail, laminar plates, and segmented armour, employing point-based grid alignment and VEX-driven patterns.
- **Fitting and Morph HDA**: Adapts armour pieces to different body types through volumetric deformation techniques and attribute-driven transformations.
- **Fur HDA**: Automates UV unwrapping and texture integration for layered and tufted fur representations.
- **Library HDA**: Manages caches of saved armour, facilitating efficient iteration and serialisation.

4.2 Segmentation Tool for Anatomical Partitioning

The segmentation tool has been developed for the purpose of dividing a base mesh into anatomically accurate regions. This is achieved by leveraging rig data and UVs as foundational inputs. This segmentation is to support a precise low-poly of armour components, including helmets, gloves, and chest plates, and ensures clean separations while retaining the underlying mesh topology. The process is computationally intensive but requires minimal data, emphasising automation with minimal manual input. The tool's approach is charac-

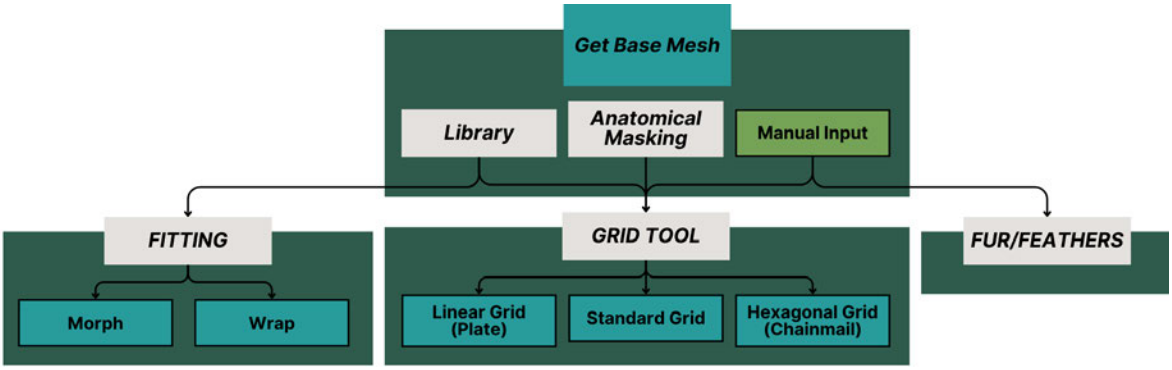

Figure 4.3: Tool HDA and Step System Structure. White: Toolkit HDA. Blue: Subnetworks and methodologies. Light Green: Supporting Plugin

terised by its modularity and sequentiality, underpinned by principles of procedural content generation. Constraint satisfaction, grammar-based systems, and constructive processes are integral components of this methodology. The Segmentation Tool is joined from three subnetworks: Anatomy Clean-up, Rig Masking and Part Isolation.

A **connectivity node** leverages UV data (adhering to the requirements) and point positions to define clean boundaries, splitting meshes into anatomical sections (e.g., arms, legs, torso, head). This node identifies connected islands and segments the model while maintaining the integrity of the mesh structure by outputting the connection data as a *class* attribute. Utilising this attribute, a foreach loop node network leverages it as their iteration steps. For each connected part (e.g. leg right, leg left, body, arm, head) the masking process is handled seperately.

The function of the *anatomy clean-up* subnetwork is to rectify the initial geometry, including interior vertices in the mouth, eye cavities, and other irrelevant areas. This enhances computational efficiency and improves segmentation accuracy later on. The network performs two primary functions. Firstly, it isolates the head geometry of the body and a point from volume isolates. Secondly, it removes unnecessary mesh parts. This is realised through the integration of volume data from designated volume to points nodes which gives out any points inside the head, followed by the calculation of intersections and distances by VEX code. Subsequently, the application of LABS' delete small parts function is employed to circumvent the formation of disjointed single polygons.

The masking process is comprised of two primary approaches, both of which necessitate the utilisation of the rig and base mesh as inputs.

- **Method 1: Voronoi Diagrams** Voronoi diagrams are a method of dividing a mesh into regions based on proximity to designated points. This technique is more advantageous for models that require extensive sculpting, as it enables the clean division of geometry without adherence to its topology (e.g. a knife tool). The points define Voronoi sites, generating cells that form natural partitions. The result is a uniform segmentation of the regions, which are then ready for sculpting or further modelling adjustments.

- **Method 2: Cluster-Based Methods** Cluster-based segmentation demonstrates greater alignment with the mesh's underlying topology, whilst simultaneously providing finer control through grid-based customisation. The cluster node is used in place of the voronoi node and it combines points into groups by spatial or attribute constraints marked by a cluster attribute.

The two approaches, Cluster and Voronoi Node, necessitate point-based inputs from which to generate their respective cells and groups.

- **Option 1: Rig-Aligned Points** Rig-Aligned points are defined as the joints extracted from the input rig of the anatomical base mesh. These points utilised for segmentation facilitate the correct naming and parenting of body parts. The outputs are typically characterised by their precision and clarity with respect to the anatomical regions.
- **Option 2: Grid** The refinement of custom segmentation is accomplished through the overlaying of a manually adjustable grid, thereby enabling precise control of cell shapes and boundaries by refining point spacing through a point wrangle node.A point wrangle node functions as a custom VEX code node that iterates over all points.Moreover, VEX scripting facilitates the alignment of grid directions with rig points, ensuring the execution of vertical slicing for arms and horizontal slicing for legs. The script calculates the directional vector from one rig point to its neighbours and isolates the determining direction (x, z, y) to apply the grid point space to.

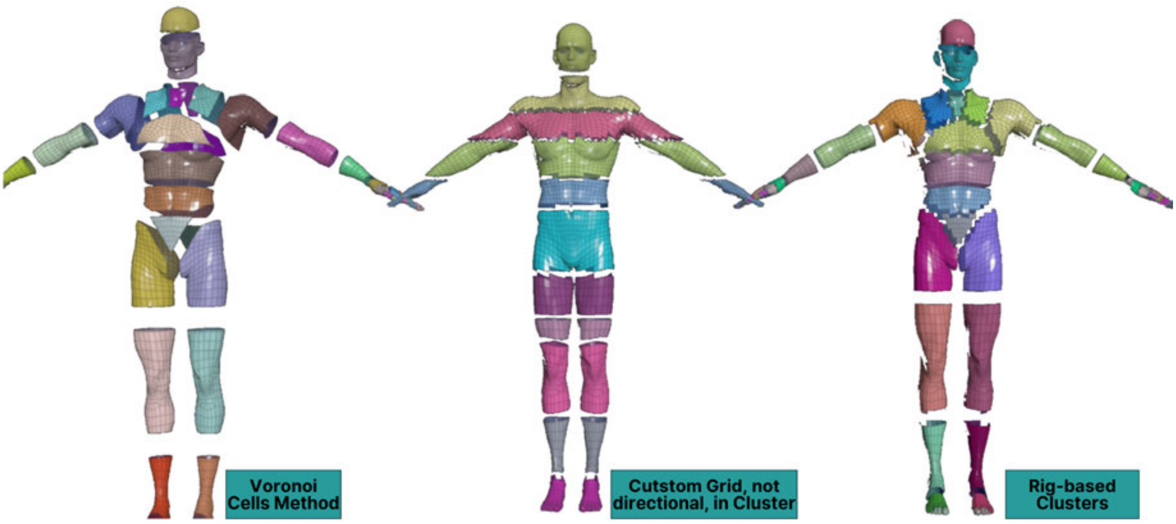

Figure 4.4: Methodologies of segmentation in prototype shown on male body mesh.

Following the segmentation process, the tool facilitates access to specific body parts through the isolation subnetwork. To illustrate this, an artist requiring only the arms or torso for pauldron design can isolate these components. This is achieved through the accessing of the named rig. The different grid parts are connected by attribute to their rig parent and can be called upon by string input. Predefined sections can be rapidly recalled by a temporarily hardcoded menu using naming conventions (e.g., "arm," "head," "hip"), thereby streamlining repeated use. As demonstrated in Figure 4.5, a mesh of a horse and a cage spider from Dark Souls 3 is presented, illustrating non-bipedal rigs and non-humanoid meshes that yield the desired, consistent outcomes when utilising the tool. The outputs adhere to the rig's connectivity, enabling seamless searches for parts. However, incomplete rig naming may necessitate additional manual input (see E). Moreover, highly intricate rigs (e.g. for cinematic animation) have the potential to yield outputs characterised by granularity, attributable to excessively fine segmentation, which can compromise the overall usability of the system. A considerable proportion of rig sections, in conjunction with discrete mesh components, contributes to an augmentation in the processing duration of the tool.

Figure 4.5: Highlighting universal application of the segmentation HDA with a Horse Model and a Dark Souls 3 Caged Spider Model.

4.3 Asset Library and Reusable Components

The library module functions as a centralized repository for the management of all existing armour pieces created. Given the inherently layered nature of armour and the tendency for pieces to exhibit similarities based on style, class, or function (as seen in *For Honor*), this module provides an efficient system for categorizing, accessing, and reusing assets. Each piece is stored in a structured manner, tagged with metadata that includes its name, type, and position relative to the character rig (see figure 4.6), ensuring modularity and ease of access.

Houdini's PDG is the foundation of the library system, offering a scalable solution to manage vast quantities of data. It acts as a task manager that determines sequential dependencies among HDAs, ensuring that only the necessary elements are recooked during updates. The ability of PDG to manage user inputs at scale facilitates the automation of complex workflows, including the management of stashes and the processing of attributes. The PDG pipeline is split into export (saving an armour to stash) and loading an armour from stash. In consideration of disk space and workflow efficiency, all armour is packed, compressed and stored in a single stash of their type and a temporary merged stash during the pipeline cooking, with all models packed into lightweight formats. This minimizes storage demands while allowing for rapid unpacking and reuse. Therefore, the Library HDA is *data-intensive and process-intensive* based on amount of armours serialised and number of steps to process but is categorised as generation as content selection.

Exporting begins with a specialized HDA, "piece export", tailored for handling each armour component, referred to as the *piece export HDA*.

Figure 4.6: Armours Models from Baldur's Gate 3 (2023) Mod Manager utilised in the testing of the Export HDA. Colours indicate different body part IDs

Each connected armour piece (e.g., chest, sleeves, greaves) is assessed individually in positional relation to the rig of the model it was made for. Utilising the object's bounding box, the tool calculates its parent point on the rig. For instance, sleeves are associated with arms and shoulders, while pants correspond to hips and legs. The individual pieces are assigned this attribute as metadata in packed format. During the export process, the serialisation data is applied, including parameters such as the component's name, armour type, and file path. Subsequently, the model(s) are categorised into stashes based on weight class (heavy, medium, light) or function, and then merged into a *merged armour stash* that centralises all components.

Once the armour has been deposited in the stashes, the library facilitates the selective retrieval and filtration of armour pieces through a fetch operation governed by Houdini's Procedural Dependency Graph (PDG). The PDG system retrieves data from existing stashes, and filters based on attributes such as body part, armour type, or name can be applied by the artist to streamline access. The PDG then dynamically processes the filtered models using "filter by value" operations and outputs them as geometry ready for use or further processing.

The library module is founded on the principles of data preservation and future-proofing. As the asset database evolves, the module has the capacity to incorporate a range of features, including expanded metadata for material definitions, animation compatibility, and advanced procedural controls. By establishing robust initial structures, the system is able to adapt seamlessly to growing project complexities while maintaining stability and usability.

4.4 Procedural Modules and Pattern Systems

The development of procedural modules and pattern systems within this toolkit introduces a structured, parametric approach to armour asset generation, facilitating the rapid creation, adaptation, and modification of armour components by leveraging predefined procedural rules and constraints. These modules are based on known approaches to procedural texture generation in common texturing workflows[Ebe+02]. The toolkit integrates pattern-based methodologies, including grid systems, hierarchical segmentation, and adaptive deformation, thus enabling the creation of complex armour structures, such as chainmail, lamellar, and plate designs, with automated fitting mechanisms.

4.4.1 Grid Pattern System and Chainmail

The grid system constitutes a core component of the procedural armour generation process, facilitating the structured, repeatable, and parameter-driven placement of elements such as chainmail, scalemail, and layered armour. The system exhibits a conceptual similarity to the grid-based clustering method employed in body mesh segmentation, wherein evenly spaced points are distributed over a surface to create structured subdivisions.

The initial phase in the system's implementation entails the generation of an adaptive grid that can be projected onto any given surface. This is accomplished by employing a point generation method, where points are systematically distributed across the mesh using the "*pointgenerate*" node in Houdini. These points function as anchors for geometry placement, ensuring an even, structured distribution. The grid generator operates in three distinct modes: hexagonal, linear, and rotational pattern generation. The hexagonal grid layout is particularly effective in achieving even coverage with minimal gaps, making it optimal for applications involving chainmail. The line-based setting allows for structured lamellar or plated designs, while the rotational mode facilitates circular patterning for ornamental or radial armour elements.

The system incorporates offset parameters that facilitate precise adjustments to the spacing between points, row positioning, and rotational variations. Utilising procedural UV mapping, the generated points are wrapped around the object's surface, thereby ensuring that patterns adhere to the curvature without distortion. This UV-based projection prevents elements from deviating from alignment when transferred onto complex 3D forms. Subsequent to establishing the grid, a transformation process morphs the points to the target surface while adjusting their orientation to align with local surface normals. This ensures that any geometry copied to these points will maintain correct alignment, avoiding misplacements or inconsistencies due to surface curvature. The copied elements, such as chainmail rings or plate segments, inherit the adjusted rotations and spacing while preserving their original proportions. The final output allows for both packed and unpacked geometry export, offering flexibility for further refinement in external software or direct integration into high-poly workflows. The structured yet adaptable nature of this system ensures modularity, making it suitable for a broad range of armour designs. The figure 4.7 shows examples of patterns that can be created.

4.4.2 Automated Fur Plane Scattering

The automated fur atlas application process is a methodology that streamlines the integration of fur or feather planes onto three-dimensional surfaces by leveraging texture atlas guidance. This process ensures plane distribution and customizability, thereby reducing manual intervention. It is process-intensive and does not generate a largely new geometry rather than

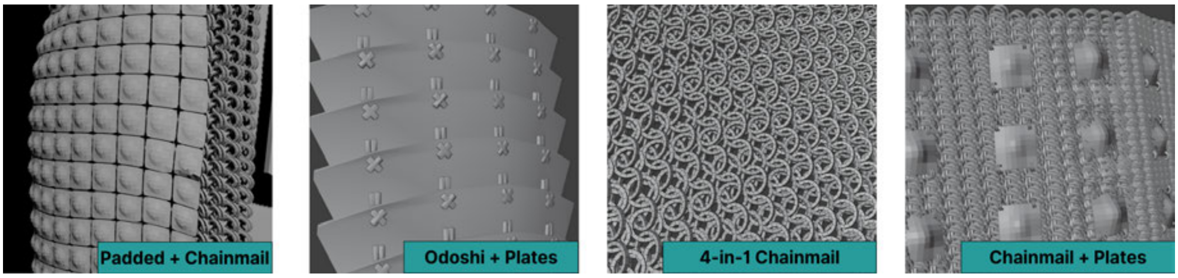

Figure 4.7: Examples of patterns made utilising the Grid Tool.

distributing planes with UVs later used for texturing to create fur. The process commences with the preparation of a texture atlas, which is typically provided in formats (e.g. png, jpg) and represents a blockout of fur or feather textures. This serves as a reference for generating planes that correspond to the mapped textures. A UV projection is applied to align the atlas with the surface geometry, followed by a trace operation to generate individual planes sized according to the texture blocks. These planes are created with a low polygon and point count to optimize computational performance. Alternatively, pre-constructed planes can be input and procedurally distributed.

Each plane undergoes further refinement, including connectivity checks to separate distinct components, size standardization for uniformity, and error correction to remove misaligned or excessively small elements. Visualization techniques are used at this stage to assign unique colours to individual planes, aiding in debugging and setup clarity.

The underlying mesh is prepared for fur or feather application by calculating a directional vector derived from the surface normals. This vector points slightly downward and can be adjusted as needed to meet stylistic or technical requirements. Points are then scattered across the mesh surface, with each assigned a fur or feather plane from the atlas. The selection process allows for randomness or user-defined rules, and parameters such as size, rotation, and density are adjustable to create visual variety.

Following distribution, the planes are then merged with the underlying mesh to form a cohesive 3D model. The final geometry, complete with fur or feather planes, is then optimised for export and integration into modelling, texturing, or rendering workflows. This approach enables efficient creation of complex assets, ensuring adaptability for varied production scenarios while minimising computational overhead.

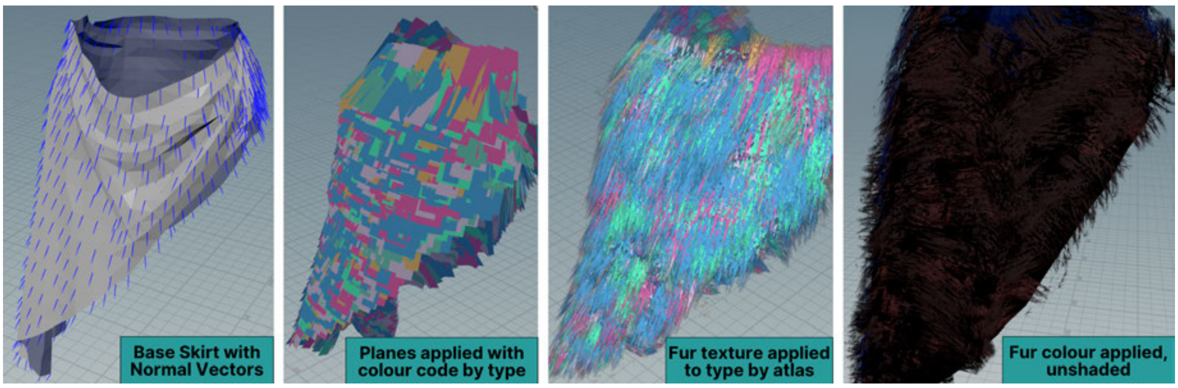

Figure 4.8: Process visualisation of applying fur to a mesh

4.4.3 Belt and Wrap LABS Tool

The Belt Tool is a procedural asset generation tool designed to create belts, ropes and wrapped cloth around models. It leverages Houdini's Labs Quick Rope node [55], simplifying the process of generating belt and rope structures while allowing for both rapid prototyping and detailed refinement. As described in 3.5, the tool enables artists to create elements such as waist guards, decorative sashes, and straps that conform to the contours of a character model with precision, as shown in figure 4.9. "The simple rope wrap is a content creation tool that wraps ropes around objects, using the cutting surface of the surfaces to create a ring around the object and generate ropes. The simple rope wrap tool allows for control over rope thickness, rope profile, and simulation to achieve organic-looking results, and it also produces UVs and reduces the rope for game-ready use," Paul Ambrosiussen describes in his conference talk on "SideFX Labs 2021 New and improved Workflows for Artists". [56]

The curve outputted by the *LABS Simple Rope Wrap Node* constitutes the foundation of the tool's workflow, with customisation possible to suit specific design requirements. Parameters such as segment density, curve radius, and tension are adjustable, providing fine control over the belt's path and behaviour. The Quick Rope node then extrudes planes along the curve to generate belt geometry ("sweep"). Artists can choose from predefined profile shapes, such as rectangular or circular, or create custom profiles to achieve unique designs. Attributes such as thickness and width can also be modified. Following extrusion, the geometry can undergo optional refinement through the application of subdivision and smoothing techniques. These techniques can enhance the visual fidelity and eliminate hard edges, particularly for elements that resemble cloth or flexible materials.

Additionally, UVs are generated automatically during this process, facilitating seamless material assignments. While the tool automates much of the belt creation process, artists maintain control over the results through manual overrides. These artists can edit curves, adjust points before receiving the final result, or modify geometry directly afterwards to achieve precise results.

Figure 4.9: Examples of the LABS Simple Rope tool as used in the toolkit. 1: Extensive use of stress-testing, 2: specified use-cases in model and 3: combined with anatomical masking.

[55] https://www.sidefx.com/houdini-hive/houdini-hive-gamedev-2021/#labs, visited 02.02.2025
[56] https://www.sidefx.com/houdini-hive/houdini-hive-siggraph/#labs, 13:00. visited 19.01.2025,

4.4.4 Anatomical Morph Fitting

The Anatomical Morph Fit process is designed to adapt armour and clothing models to a variety of human body types. The workflow ensures both the preservation of key spatial relationships and the morphological adaptation needed to accommodate different character proportions. This is achieved using a combination of deformation techniques, predefined morph targets, and procedural adjustments that balance manual control and automation. The method is based on deformation techniques, such as *Laplacian Mesh Deformation (LMD)*, *Volumetric Graph Laplacian Deformation (VGL)*, and *Free-Form Deformation (FFD)* [Mit24].

The initial step in this process is to ensure that the base armour or clothing model is correctly positioned relative to the target character model. It is imperative that size, rotation, and bounding box alignment are in accordance with the reference geometry, taking into account the specific scale and pose of the target model. Misalignment of the reference model can result in suboptimal deformation outcomes, manifesting as regions that are either skewed or excessively compressed. Accurate placement is therefore pivotal in ensuring that the deformation process commences with geometrically sound inputs. While automated alignment scripts can assist, manual oversight is often required, particularly for extreme proportions (e.g., child versus ogre-like figures).

A critical step in the process is the definition of masks and restraining points which define shape and size fitting. This enables disparate regions of the model to adapt proportionally to the target morphology. Masks delineate regions on the geometry where deformation is permitted and those where it is not, thus ensuring that key design elements maintain their volume and shape integrity, particularly for rigid materials like armour plates. Masks are defined by point difference and intersections of underlying models and bounding boxes of target areas. In the case of meshes with more complex volumes, such as cloth, additional refinement is often required to achieve smooth deformations while preserving natural shapes. Manual adjustment involves finer modelling or repositioning of the model prior to the utilisation of the tool. Masks can additionally be implemented as morph targets, such as body types or curvatures.

Following the completion of the masking process, the model undergoes subdivision to generate a mesh of higher resolution, thereby facilitating detailed deformation. Adaptive subdivision ensures that deformation-heavy regions receive greater density, while lighter areas remain optimised, thus preventing the occurrence of artefacts or low-poly distortions, which are crucial for materials such as cloth or curved armour plates. Subdivision serves as a necessary precursor to the application of advanced methods, including *LMD* and *VGL*. *LMD* utilises cross-section curves to apply localised constraints, thereby enabling smooth and precise deformations suitable for edges or detailed sections of armour. In contrast, *VGL* focuses on deforming tetrahedral mesh networks to preserve the volumetric relationship between the model and the underlying body.

Advanced workflows frequently incorporate *Free-Form Deformation (FFD)* with volumetric methodologies, such as *VGL*. This amalgamation facilitates indirect manipulation of the armour by transmitting alterations in body shape directly to the attached model. Moreover, hard surface and soft surface techniques are customised to their respective materials, ensuring precise geometry preservation for rigid plates and flexible adjustments for cloth-like elements.

Whilst automation has been proven to expedite processes, there are instances where creative workflows necessitate manual intervention for the purpose of achieving artistic precision. The process shown here shows issues with correct topology and at times proportion fitting, as well as adhering to correct volumetric or shape restraints. An example of the step process of the network can be seen in figure 4.10. Interactive refinement of masks or restraints has been demonstrated to outperform fully automated solutions in certain edge cases. Furthermore, deformation processes such as *VGL* have been shown to be computationally expensive.

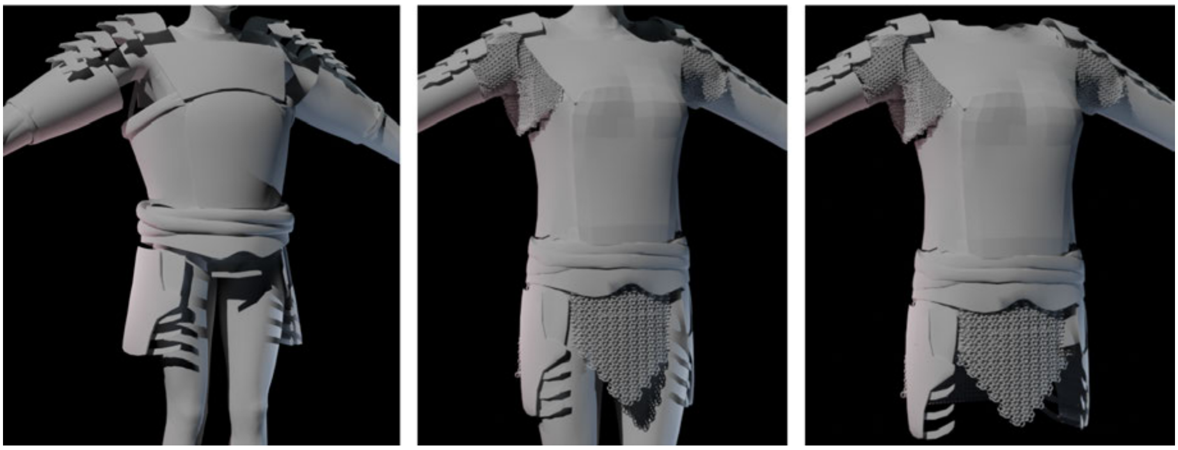

Figure 4.10: Process of Morph Fitting from male armour model to female body. Steps as shown: positional proportion manipulation, morph fit process and adjusting topology.

4.5 Supporting Tools and Systems

In addition to the core toolkit functionalities, several supporting tools and systems have been implemented with the objective of enhancing workflow integration and facilitating the evaluation process of the prototype. These tools have been designed to ensure compatibility across different software environments, to streamline asset handling, and to provide necessary data for performance assessment.

4.5.1 Blender-Houdini Pipeline and Interoperability Plugin

As previously outlined (see chapter 3.6), additional support systems were incorporated to broaden the scope of support beyond Maya, encompassing Blender as well. This approach was undertaken to extend the evaluation target group, thereby encompassing a more extensive demographic. As Blender does not natively support Houdini integration and to accommodate artists primarily working in Blender, a custom Python-based plugin facilitates the transfer of geometry between Blender and Houdini. The tool is based on Simon Verstraete's methodology [57] [58] [59] and the functionality of the system under review here mirrors that of Houdini's native Engine Session Sync feature[60], which allows real-time updates between Houdini and plug-in compatible software e.g. Unity and Unreal Engine. This ensures that modifications made within Blender can be reflected in Houdini and vice versa. The system operates by utilising Python scripts that create temporary files, which are continuously updated upon artist input. These files allow for bidirectional communication between Houdini and Blender, ensuring that meshes remain consistent throughout the workflow. Following the implementation of modifications to the original iteration, the tool's fundamental functionalities can be enumerated as follows:

The process is structured as follows:

1. The artist selects the geometry in Blender for export.
2. The artist clicks on the "Export to Houdini" button.
3. A Python script executes, generating or updating a temporary files in the toolkit directory.
4. Inside Houdini, the artist clicks on "Load from Blender".
5. Houdini reads the temporary file and loads it. The geometry is inserted into the toolkit pipeline.
6. After processing in Houdini, the results are saved and reimported into Blender through pressing the "Export to Blender" button.

It is acknowledged that artists may possess limited experience with Houdini; consequently, the system has been adapted to restrict interaction with unfamiliar entities.

- **Python Module with Callback Scripts:** In the context of the Houdini interface, the conventional import and export functions have undergone a transformation, changed into a modular HDA. This evolution has not only introduced a structured input handling paradigm but has also led to a significant reduction in the complexity of interactions with the Houdini User Interface. The HDA employs Python modules with callback scripts for button-based execution, thereby empowering artists to initiate imports, exports, and file updates within the same User Interface that is utilising the toolkit.
- **Integration as a Houdini Digital Asset (HDA)**: The toolkit has been consolidated into a unified HDA, accompanied by a unified user interface. The tools have been organised into subnetworks, ensuring that artists interact solely with predefined parameters. This approach minimises the necessity for manual setup and technical overhead, while accommodating users with limited Houdini experience. The toolkit is designed to simulate Blender modifiers with a high degree of fidelity.
- **Selection-Based Exporting:** In contrast to the base implementation that exported the entire scene, the modified version enables artists to selectively choose which meshes to export, thus eliminating the need for any additional instances of Houdini or Blender. In this step, the Blender Export Scene Operators are leveraged.

[57] `https://80.lv/articles/tutorial-creating-a-blender-to-houdini-tool-using-python/`, visited 16.01.2025
[58] `https://youtu.be/dc4g1l4rx0U?si=RFLXCFZpfSz_p99V`, visited 16.01.2025
[59] `https://www.youtube.com/watch?v=H5aY9wcbX3c&ab_channel=SimonHoudini`, visited 16.01.2025
[60] `https://www.sidefx.com/docs/houdini/ref/henginesessionsync.html`, visited 16.01.2025

This implementation enables Blender users to utilize the procedural toolkit within a familiar interface while ensuring compatibility with Houdini's procedural workflow.

4.5.2 Houdini Logging System

A logging system has been integrated into Houdini to track tool usage, performance efficiency, and evaluation-related data. This system utilizes Houdini's built-in *Python Logging Package* [61], allowing for the structured recording of interactions and computational times.

The logging system is designed to capture the following metrics:

- **Rendering Times:** The system is designed to record the time required for key operations, such as procedural mesh generation and pattern application. This function is analogous to the performance monitor integrated into Houdini.
- **Tool Usage:** The log documents the utilisation of toolkit functions, the frequency of their execution, and the sequence of operations performed by the artist.
- **Method and Input Tracking:** The system identifies which specific methods within the toolkit (e.g., different segmentation approaches, pattern modules) were utilised and logs the parameter inputs for each operation.
- **Import and Export Actions:** The logging system serialises the importation and exportation of assets, providing a comprehensive record of the workflow stages, thereby ensuring the uninterrupted and verifiable flow of data between Blender and Houdini.

All collected data is stored in JSON format, ensuring the logs are both structured and readable, and can be used for further evaluation. Each HDA has been modified to automatically append its relevant execution data to a unified logging file. This setup enables a detailed post-evaluation analysis of both performance efficiency and user interaction patterns.

4.6 Altered Workflow

The integration of segmentation, pattern-based asset generation, and automated fitting tools fundamentally restructures the 3D armour creation workflow by reducing the need for extensive manual actions. In figure 4.11, the developed prototype was used to create armour base meshes in a brief time frame (each two hours approximately). Traditional workflows rely heavily on manual segmentation, base mesh preparation, and iterative fitting processes, all of which can be time-intensive (see chart of manual workflow for specific actions in appendix E.4). The procedural toolkit streamlines these steps by introducing predefined anatomical zones, automated mesh partitioning, and scalable deformation tools, significantly increasing efficiency. Entire manual steps of modelling can be skipped and replaced by toolkit handling (see grid and wrap tool comparison in figure E.4 and 4.12).

This transformation is particularly beneficial for large-scale productions where character assets require multiple armour sets. Unlike static, single-character models, where armour is often sculpted as a single unit and rigged later, the toolkit ensures modular adaptability

[61] `https://www.sidefx.com/docs/houdini/hom/hou/logging/index.html`, visited 16.01.2025

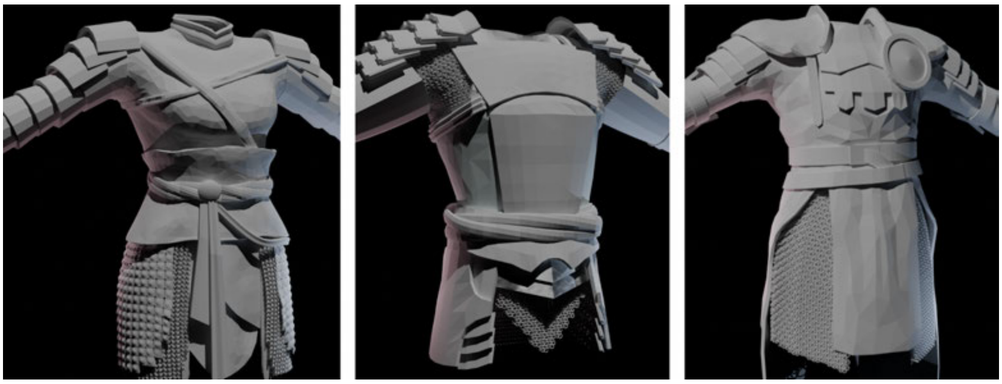

Figure 4.11: Armours created leveraging the toolkit.

across different character meshes, rigs, and morphologies. Automated segmentation allows for quick partitioning without requiring manual masks, while morphological adaptation tools facilitate seamless scaling and fitting across various body types. Beyond efficiency, the toolkit enhances asset reusability and standardisation by providing a centralised armour component library and procedural templates, thus supporting the creation of consistent yet highly adaptable armour sets across different projects and teams. The system maintains creative flexibility by not fully automating artistic decision-making, but rather by optimising repetitive technical tasks, allowing artists to focus on quality and style. However, its applicability varies depending on asset type; while it excels in rigid and structured armour, its use in fabric-heavy designs remains limited.

The toolkit's capacity to rapidly generate, adapt, and iterate on armour components represents a substantial workflow enhancement, thereby bridging the gap between manual artistry and procedural efficiency. Nevertheless, it is imperative for the artist to adapt to a new workflow that incorporates these tools first. In addition, E.3 includes a visual representation of two distinct armour sets from the video game *For Honor* (2017). These armour sets are colour-coded to exemplify the application of the toolkit,

The subsequent part offers flowchart visualisations and flowcharts that demonstrate how the toolkit remodels the workflow in comparison to conventional modelling approaches. The time estimates are derived from the analysis within the designated 2.5.1 section, in addition to observations of the resources and further insights from the chapter on the 5.2.

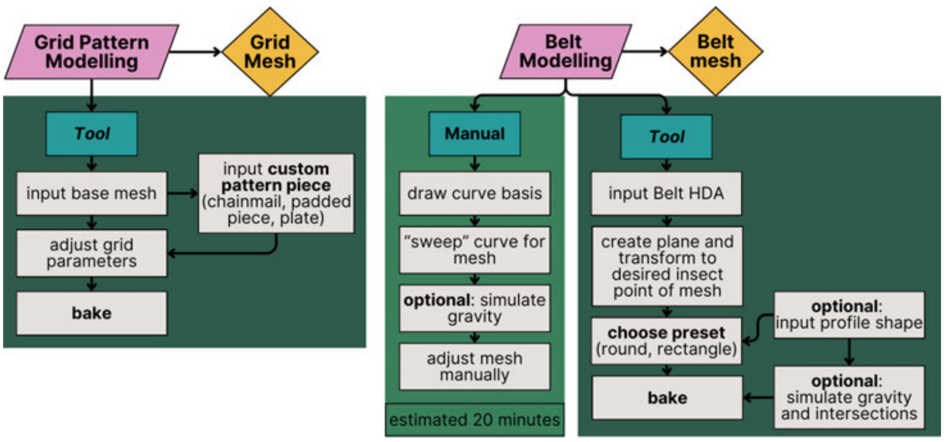

Figure 4.12: Workflow diagram: Grid Pattern Modelling with Tool and Manual Belt Modelling to Tool.

5 Evaluation

The present evaluation approach is analogous to that outlined by Nelson et al. in his work on “A requirements analysis for video game design support tools”, in which he conducted a requirements analysis for a design-support tool for game design [NM09]. The methodology involves validating a concept of the workflow as previously researched, with the aim of contextualising and verifying its reality and accuracy. This evaluation stage is instrumental in informing the preparation of the toolkit for the second evaluation, which is of a technical nature and involves the testing of the tool by character artists with varying degrees of experience. The tool is assessed using a quality questionnaire, and the process of its use is mapped in order to gain deeper insights into how its functionality and compatibility can be enhanced. The overarching objective of the evaluation process is to further contextualise the usability and potential applications of the toolkit in a variety of industry contexts.

5.1 Evaluation Participants and Process

The selection of participants for the evaluation of the toolkit is based on their expertise and familiarity with 3D modelling and game development workflows, particularly in character and armour modelling. The primary group for contextualisation includes experienced 3D artists from professional backgrounds, with participants chosen through targeted outreach to ensure experienced perspectives. The second group of participants will conduct the technical evaluation with extensive assistance on Houdini and Blender pipeline set-up. Participants will have a range of experiences and academic and professional backgrounds, thus creating a diverse area in which to “stress-test” the toolkit in different workflow and skills.

Participant Group A: Group A conducted the interview exclusively and provided feedback on a third-person presentation of the toolkit, as well as offering input during the development process.

- Anton Guzeev (Person 1): Character Art Supervisor at Hexworks
- Ivan de Gregori (Person 2): Lead 3D Character Artist at Jyamma Games

Participant Group B: Group B, comprised of eight participants, all male, age ranging from 20-31 (see F), made use of the toolkit integrated into Blender with a custom export tool to bridge the lack of in-house plug-in and provided feedback on the various steps and categories. As demonstrated in Figure 5.1, it illustrates the varying levels of experience within the group who evaluated their experience between 1 (none) and 5 (advanced) on their own. The majority of respondents claimed to have no experience with Houdini, but a significant proportion reported intermediate experience with working with add-ons and plugins within Blender.

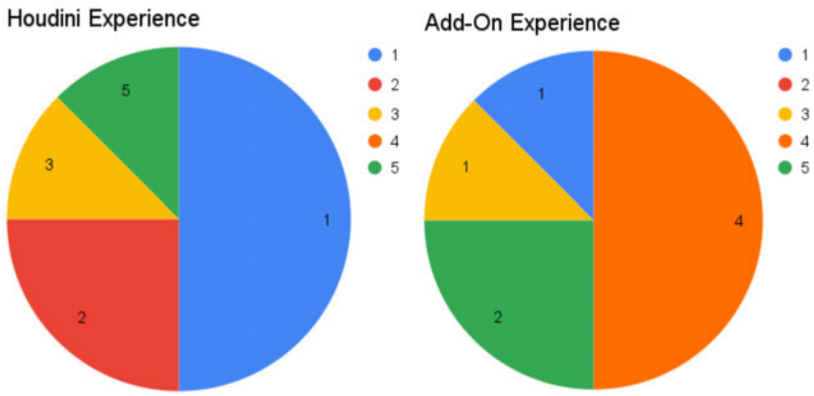

Figure 5.1: Experience of participants with Houdini as a software and with Add-Ons and Plugins within their modelling software.

5.2 Expert Interview: Industry Insights on Armour Modelling

The evaluation group, which was considered for the expert interview, was more precise. It comprised leading 3D artists from the games industry who had contributed work to a game with heavy armour and customisation elements. These artists provided specific industry examples and contextualised the work of a leading 3D artist. This was done in order to draw conclusions for enhancing the toolkit and evaluation hypothesis.

5.2.1 Interview Structure and Objectives

The objective of expert interviews is to gain insights that enhance the comprehension of the research subject. These interviews are utilised to collect expert knowledge from individuals with a direct connection to the research topic where professionals give in-depth responses to various categories. In the present study, non-standardized and semi-structured expert interviews were utilised. Semi-structured interviews employ guiding questions that can be adapted to the respective expert. This approach enables the researcher to gain insights and explore novel concepts and methodologies based on the project context specific to the interviewee.

Topics were divided into the following key areas:

- **Design and Guidelines:** An examination of the conceptualisation of style, the technical guidelines that inform the creation of armour, and the historical inspirations that underpin its design.
- **Production and Workflow:** An evaluation of the life-cycle of armour, collaborative processes, and the role of different teams was conducted in order to form a picture of the context of the toolkit use.
- **Challenges and Refinements:** An attempt at identifying bottlenecks, iterations, and optimisation techniques to consolidate the toolkit approaches.
- **Time and Resource Analysis:** Accumulation of time estimates for creating armour and adaptations to character models in order to compare standardize a comparable number.

5.2.2 Design and Guidelines

As Anton Guzeev, Character Art Supervisor at Hexworks, emphasised, design style guidelines are implemented on a project-wide basis to ensure cohesion across all assets, including armour. The differentiation of character factions is characterised by divergent material and colour schemes; conversely, however, the armour itself adheres to broader artistic principles. For the production of Lords of the Fallen, inspirations span various historical periods and geographic regions, although technical and artistic requirements ultimately determine the final designs.

Enotria, on the other hand, employs a stylised, semi-realistic aesthetic, characterised by an 80:20 ratio of realism to stylisation. Armour designs are rooted in specific historical and cultural inspirations, including Etruscan and Renaissance influences, and regions within Enotria exhibit distinct artistic motifs that directly influence character and armour designs.

A notable aspect of Lords of the Fallen is its comprehensive approach to cultural influences, whereas Enotria is specifically focused on meticulous cultural research and the analysis of influences from discrete historical contexts across various sections of the game.

5.2.3 Production, Challenges and Workflow

Both experts follow similar workflows for armour creation as explored in this thesis (2.5.1). Specific variations exist in that modularity plays a significant role in *Lords of the Fallen*. This is evidenced by the fact that armour must support player customisation, partial swaps, and adherence to skeletal proportions (also see 2.3). During development, various previous assets (from the previous projects and Lords of the Fallen) and skeletons are reused. In contrast to the focus on modularity in *Enotria*, the customisation of anatomy is hindered by exaggerated proportions, necessitating the development of unique armour designs tailored to specific body types. In *Enotria*, the armour is one full design as the player only equips sets in form of "masks". The main points of challenges listed for the game's armours were:

- *Lords of the Fallen:* The management of clipping issues between modular pieces is a persistent challenge. The balancing of polycount limits against an increase in visual detail has the potential to further complicate production.
- *Enotria:* Anatomical deviations have been shown to disrupt traditional workflows, thereby creating difficulties in adapting armour to unrealistic body shapes. Maintaining clarity of design is another persistent issue.

Guzeev's answers identified several key issues impacting the efficiency of armour production. Firstly, the production of armour is constrained by hardware-imposed limitations, including polycount, material restrictions, and bone counts. Secondly, the Xbox One S has been identified as a significant hardware bottleneck for Lords of the Fallen, leading to substantial production constraints. Guidelines are formed in response to such limitations, rather than preceding production, to ensure feasibility. Both Guzeev and de Gregori highlight additional bottlenecks in retopology, as a tedious task, as well as detail to constraints balancing.

Pre-planning helps to avoid unnecessary detailing of invisible sections, thus saving both time and resources. Custom validation tools were recommended to ensure technical consistency without excessive manual checks. As a supervisor, Guzeev serves as a liaison between teams, facilitating communication across art, animation and technical departments. He emphasised the need for technical knowledge and broad skills to manage expectations and avoid miscommunication between outsourcers and internal teams. Guzeev observes that although novel technologies emerge, they seldom integrate seamlessly into established pipelines during ongoing projects and he emphasises that new workflows can significantly alter processes, especially during pre-production phases.

5.2.4 Time and Modularity Analysis

During the time evaluation stage, the unit of measurement employed is man-days, otherwise referred to as “days per man”, which denotes the number of days per person working on the asset (e.g. modeller, texturing artist, shader artist,...). The allocation of time depends upon budget and manpower. Guzeev and de Gregori were given several examples of varying complexity levels of armour. While Guzeev placed the workload at around 20-25 days for basic armour, with variants taking 1-3 days extra. De Gregori has set 6-15 working days for less complex armour, 10-25 for the more complex cases. Depending on the factors, variations may extend 5–10 working days. However, de Gregori made mentions of the limited input on constraints and necessary output, as well as the factors going into the production of the specific armour to be measured. Guzeev also emphasised how careful planning is vital, with reusable elements across different characters or projects, and that context is crucial, with cutscene characters requiring significant time.

5.3 Derived Contextualisation from Expert Interview

The expert interviews provided valuable context and insights that directly informed the development and evaluation of the procedural toolkit. Leading 3D artists from the industry highlighted critical challenges in armour modelling, such as managing clipping issues between modular pieces in titles like Lords of the Fallen and adapting armour to exaggerated anatomical forms as seen in Enotria. These discussions highlighted the necessity for a toolkit that supports robust modularity, ensuring that armour can be efficiently adapted across diverse body types and meshes while minimising manual adjustments.

The insights obtained from the analysis also revealed several specific aspects regarding the toolkit’s current capabilities and limitations. Notably, the toolkit does not account for intersections during the morphing process, though it does address intersections in anatomical masking; this represents an area for further development. It performs well with varying meshes and shapes, making it suitable for stylized designs and adjusted styles, as exemplified by *Enotria*. However, it does not incorporate procedures for ensuring correct topology or poly count and major parts still require manual adjustments or full remeshing. Given that variations of armour typically do not require extended production time, the morph fitting tool should be optimized to outweigh this factor. While the prototype is highly effective in facilitating large-scale modularity and base meshes, it is not yet well-suited for projects that require extensive custom sculpting, as evidenced by examples such as *Enotria* and *Elden Ring*. In addition, the toolkit would benefit from being more closely aligned with the specific restrictions and contexts of each project, as artists may face challenges in adapting these tools within a realistic workflow.

The expert interviews thus serve to reinforce the toolkit's design, vision and objectives. Furthermore, they contribute to the development of the evaluation process and metrics to investigate and identify the areas to focus on.

5.4 Technical Evaluation: Parameter and Metrics

The technical evaluation is intended to provide a systematic assessment of the *effectiveness, usability, and efficiency* of the toolkit in comparison to a traditional manual modelling workflow. In order to ensure consistency and controlled comparisons, the evaluation is divided into four sequential phases: **Manual Modelling, Toolkit Introduction, Toolkit Modelling**. Each phase is timed and recorded for the purpose of performance analysis. While artists work independently, their feedback is documented post-evaluation for further analysis.

The objective is to measure improvements based on topics discussed in the 5.2 when utilising the toolkit across different stages of modelling.

- **User Experience**: The overall satisfaction of the artist with the toolkit, encompassing ease of use, clarity, and user-friendliness. Done through self-reported satisfaction and intuition on a Likert scale and observed roadblocks during evaluation. Includes qualitative feedback and regard to heaviness of pain points.
- **Efficiency**: The time required to complete specified modelling tasks compared to the manual modelling workflow. Total time taken to complete modelling tasks in each phase of the evaluation (Manual Modelling vs. Toolkit Modelling). Breakdowns of time for key subtasks (e. g. , base mesh creation, UV mapping, applying patterns).
- **Realism**: The extent to which the toolkit functions align with realistic, practical modelling scenarios in a production pipeline. This is measured by frequency of use across tasks and feedback on feel of toolkit.
- **Accuracy**: How closely the output of the toolkit matches the quality and fidelity of a manually created base mesh. Side-by-side comparisons of manually and procedurally modelled base meshes. User feedback on the perceived closeness of results to manual standards and their likeliness to use it.
- **Reliability**: The consistency of the toolkit's output, ensuring it meets expectations without errors or missing functionality. Proportion of successful outputs across repeated tasks (e. g. , pattern modules generating consistent chainmail grids). Feedback on instances of failed or suboptimal outputs, such as incorrect masking or misaligned UVs.
- **Performance and Stability**: The responsiveness and load times of the toolkit during task execution and the robustness of the toolkit, evaluated based on software crashes and interruptions during modelling tasks. Average time taken for loading key toolkit modules and processing tasks. Observed lag or delays during user interactions. Per-

formance feedback during time-intensive operations, such as pattern generation or UV mapping. Number of crashes or software interruptions recorded during the evaluation. Observed anomalies, such as freezing or temporary unresponsiveness.
- **Compatibility**: The degree to which the toolkit integrates with existing workflows and external software, meeting artists' expectations. Compatibility with popular 3D modelling tools such as Maya, Houdini, and ZBrush. Technical success in importing/exporting files and maintaining mesh integrity.

5.5 Technical Evaluation: Phases and Methodology

The evaluation process is to be divided into four sections. These sections are to be timed, but not time-limited, and participants are permitted to take breaks at their discretion. It is important to note that the evaluation process is conducted solely between the artist and the evaluator, and does not involve any additional input or feedback from third parties.

5.5.1 Manual Modelling Baseline Assessment

This phase establishes a baseline for comparing toolkit performance. Participants use traditional 3D modelling techniques with no tool support to create a base armour model based on provided concept art. The armour concept provided is limited in its technical complexity.

- **Objective**: Choose one of two armour concepts. Create the base armour model (including e. g. chainmail,. . .) on the given human base mesh *without* assistance from the toolkit.
- **Scope**: Focus on blocking out proportions, basic forms, and structural elements. Patterns like chainmail, padding, ropes, odoshi and plates need to be recognisable despite being required for the High-Poly only. Cloth requires no simulation or sculpting at this stage.
- **Metrics**:
  - Time required to complete the base model phase
  - Observed difficulties (e. g. , repetitive tasks, inefficiencies)
  - Used tools (e. g. , Geometry Nodes in Blender)

5.5.2 Toolkit Introduction and User Onboarding

The purpose of this phase is to facilitate the familiarisation of participants with the toolkit's functionality, its User Interface (UI), and the features that are available to them. It has been designed to permit exploration, troubleshooting, and adaptation to the toolkit's workflows, in addition to enabling stress-testing.

- **Objective**: Experiment with the given toolkit, understand its structure and ask questions before proceeding to the next task. Learn to operate it confidently and ask questions when necessary.

- **Scope**: Cover small, focused tasks like setting up modular armour pieces, applying procedural grids, and utilizing masking and morphing tools. This step includes guided tasks with explanations for key tools (e. g. , pattern modules, anatomical fitting) including a tutorial, as well as a self-driven problem-solving scenarios to build confidence.
- **Metrics**:
  - User adaptation time
  - Quantity of guidance and questions required
  - Problems and bugs encountered
  - Feedback on initial usability and problem areas

5.5.3 Toolkit-Based Procedural Modelling Evaluation

In this phase, participants employ the toolkit to replicate a modified version of the previously provided concept art. This version is characterised by a higher technical complexity and a focus on elements that necessitate the utilisation of the tool (e.g. chainmail, plates, etc.).

- **Objective**: Choose one of two armour concepts. Create the base armour model (including e. g. chainmail,. . .) on the given human base mesh *focusing* use of the toolkit.
- **Scope**: Focus on blocking out proportions, basic forms, and structural elements. Patterns like chainmail, padding, ropes, odoshi and plates need to be recognisable despite being required for the High-Poly only. Cloth requires no simulation or sculpting at this stage.
- **Metrics**:
  - Comparison of modelling time versus manual modelling
  - Output consistency with modified concept art
  - Observation of which toolkit features were frequently used and which were avoided
  - Identification of remaining challenges even when the toolkit is used

5.5.4 Evaluation Hypothesis

The evaluation of the toolkit was designed to assess its efficiency and limitations within procedural workflows for armour asset creation. The following hypotheses were formulated to anticipate the performance, usability and integration challenges associated with the toolkit's key features. These predictions were grounded in existing workflow constraints, prior research on procedural content generation, and expert insights from character artists.

1. **Unrefined Outputs and Overgeneralised Tools:** As this is a prototype, it is possible that certain areas of use have not yet been refined. The Morph Tool is more efficient when used on rigid, simple geometry, but struggles with precise outputs on curved, cloth-based components or larger volumes. Tools such as the grid and morphing tool suggest broader, unspecific functions, unlike their reference tools suggested in the concept chapter. This more general use case can make handling difficult or add to the unspecific outputs which need extensive manual adjustments.

2. **Time Reduction:** Based on the time expectation provided by the expert interview, planned tool concept and altered workflow, it is anticipated that the implementation of these tools will result in a substantial reduction in the time required for asset creation, primarily through the automation of repetitive tasks. Despite the increased complexity, it is projected that the time required will be reduced by approximately one-third (e.g. 7 hours to approximately 5 hours).
3. **Learning Curve:** Artists unfamiliar with procedural workflows may struggle initially but will improve with use [Bus21]. The evaluation phase showed most users rated the toolkit as "somewhat intuitive", requiring some trial and error but becoming manageable over time.
4. **Adoption and Usability:** While the toolkit is expected to enhance efficiency, some tools may not align seamlessly with an artists' workflow, as mentioned before in regards with Buschauer (2021).
5. **Scalability and Asset Reuse:** The library tool and modular design should prove most valuable in large-scale productions, where reusability is essential. However, its necessity in the evaluation's small scale is limited.
6. **Performance and Stability:** In the absence of adequate safeguards, stability issues may arise. However, it is anticipated that performance will remain consistent. Tools might cause unanticipated crashes and therefore workflow disruptions.
7. **Further Development:** The toolkits handling and functions will be informed of their use and further necessary constraints, as the prototype by Nelson et al. in their evaluation [NM09].
8. **Toolkit Neglect:** It is evident that the proliferation of tool functions has the potential to result in the neglect of one or two functions by the artist. This phenomenon can be attributed to a lack of understanding or an ineffective utilisation of the software's functionalities, particularly those that the artist is more acquainted with.

5.6 Evaluation Results and Questionnaire

A Python logging system was utilised to extract data from which portions of the evaluation were analysed. The questionnaire was divided into the metrics mentioned earlier and included Likert-scale evaluations of adjectives to describe tool handling and actions, as well as questions on performance, crashes, best practices and subjective experience. In addition, users were guided and interviewed to gather subjective feedback. The diagrams 5.2, 5.3 and 5.4 visualise the numerical results of the questionnaire filled out by the participants which was sorted into the metric categories as explained in 5.4. The figure 5.5 illustrates some of the armours and tools employed in the creation process during the evaluation phase.

As demonstrated in figures 5.2 and 5.3, there is a clear tendency towards above-average performance in most categories of evaluation. The toolkit demonstrates particular potential in the areas of intuitiveness and navigation. However, the overall overview of the toolkit and its functions was less satisfactory, thereby impacting manageability. The accuracy and consistency of the system is impacted by the expectations and realistic usage of the individual tools; however, the three main tools (anatomical masking, grid and wrap tool) performed well and were commended in the individual feedback, as further demonstrated in figure 5.4.

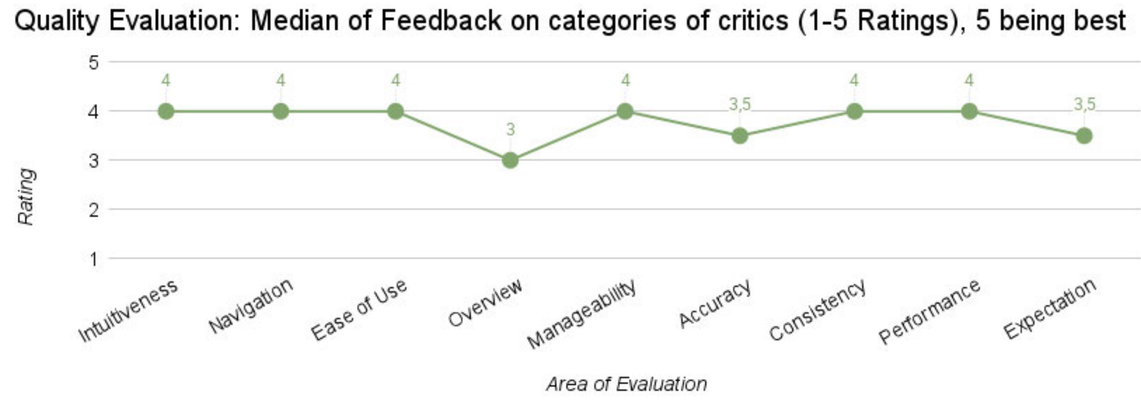

Figure 5.2: Line diagram showcasing the different areas evaluated and the median of all feedback on the category. Ratings 1 (none/bad) - 5 (good)

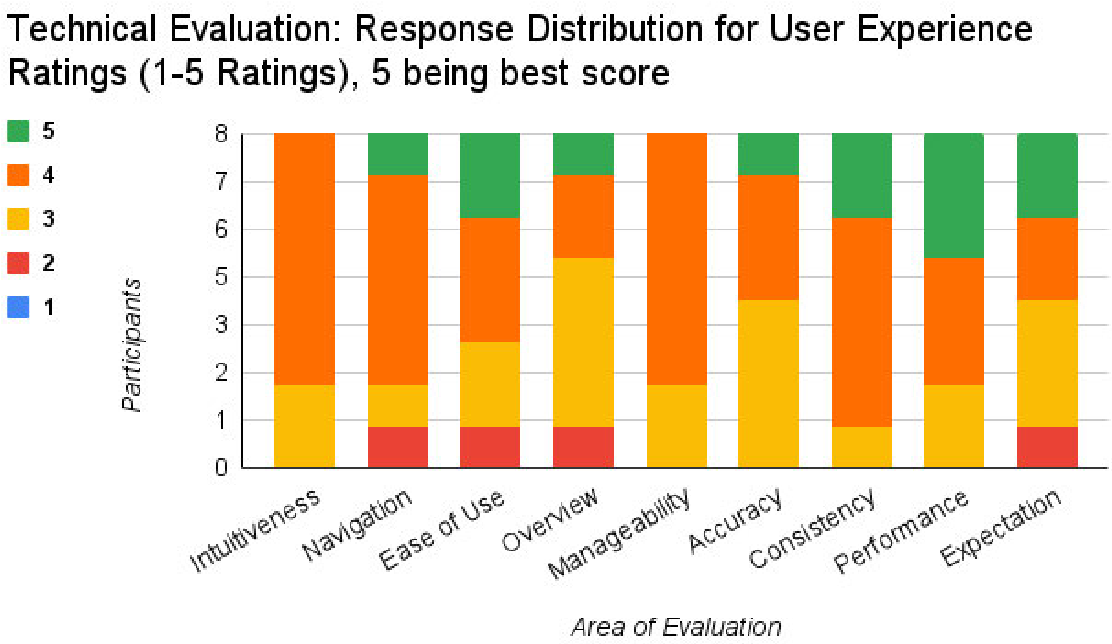

Figure 5.3: Stacked bar diagram showcasing the distribution of answers by participants on the categories. Ratings 1 (none/bad) - 5 (good)

The evaluation of the procedural armour toolkit yielded significant insights into its efficiency, usability, and areas for refinement. All participants, for all evaluation phases, took between a minimum of 8 to a maximum of 15 hours. Through structured testing phases, participants demonstrated a notable reduction in modelling time, increasing adoption of procedural workflows, and adaptability to toolkit functionalities. However, challenges in workflow clarity, tool predictability, and varied user experience levels highlighted aspects requiring further refinement.

A substantial decrease in asset creation time was identified as a critical finding. Initial evaluations showed that participants required between 6 and 10 hours to complete a base mesh armour using standard modelling methods, with an average of 7 hours. By the third evalua-

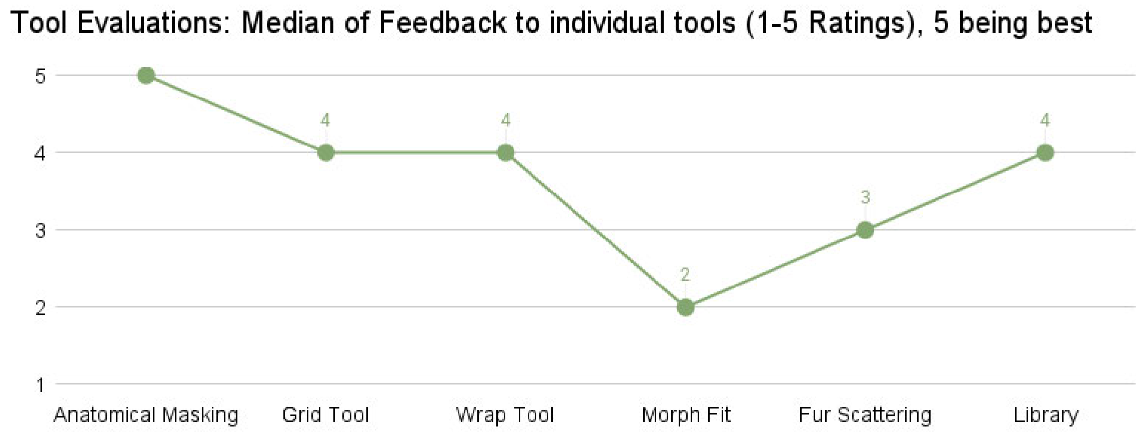

Figure 5.4: Line diagram of median evaluation on individual tools within the toolkit. Ratings 1 (none/bad) - 5 (good)

tion phase, despite working with more complex models, this time was reduced to 3-4 hours with 5 hours at most, demonstrating a measurable improvement in time. This reduction in production time can be attributed to the frequent reuse of settings and parameters, allowing for rapid iteration and adaptability of the tool.

Artists reported that approximately 70% of the final armour models were derived from the toolkit, indicating a high integration rate into the workflow. Once the procedural tools were understood, users found them intuitive for exploration and reusability, allowing for quick adjustments and iterative design. In the event that the tool did not initially meet their expectations, the artists opted to refrain from its utilisation. This phenomenon was primarily observed in conjunction with the Morph Fit and Fur Scattering Tool and their logging files.

Whilst the toolkit demonstrated clear efficiency gains, the ease of adoption varied among participants. The evaluation revealed two primary user groups:

- Artists who fully adopted the toolkit, using it extensively throughout the workflow.
- Artists who used it sparingly or avoided it, preferring traditional methods (e.g. geo nodes or particle systems) for smaller-scale tasks as they struggled to get used to the functions.

A significant proportion of the artists had no prior experience with Houdini, with only one participant having previous experience. However, Blender users with experience in extensive add-ons adapted more quickly to Houdini once introduced to the toolkit's UI and node handling. The present study found no clear correlation between artists with a significant add-on experience level and those without. Both groups required extensive aid and guidance into the work of the toolkit, with some artists demonstrating greater aptitude than others. However, there were also artists with no Houdini experience who could quickly adjust, as well as Houdini experienced artists who could not. The present study found that artists with more experience generally demonstrated a higher level of proficiency. The mean rating of the toolkit's intuitiveness was between 3 and 4, indicating that while it was manageable, it required trial and error and initial guidance. The navigation of workflows and clarity were generally positive, though some features were perceived as cluttered or required additional tutorial support to reduce the initial learning barrier. In the event of artists encountering bugs or components that were not conducive to their workflow, an update could be implemented through a packaged HDA file. This update could be executed immediately and simply be added to the directory of the evaluation. This approach ensured functionality and adaptability, thereby emphasising the success of implementing quick modularity.

The library tool was found to be of limited utility in small-scale tasks due to the lack of content in library and limited need for use-cases. During the evaluation phase, artists found the fur scattering function to be of minimal use, with similar solutions being found through the grid tool or base functions of Blender. The wrap tool exhibited stability issues, requiring safety nets and refinements, and the Morph Fitting tool did not meet expectations. A substantial number of iterations were recorded, yet these yielded no results due to the tool's substandard performance. The diagrams did not incorporate Morph Fitting feedback, as the tool was largely disregarded. In some cases, artists encountered unexpected distortions with scatter projection, indicating that some nodes functioned unpredictably. Despite these challenges, most tools were deemed useful for quick access. Nevertheless, unanticipated outputs and manual intervention remained obligatory, underscoring tool bias in data, particularly with the wrap tool and projection scatter functionalities on curved surfaces.The primary reliable tools were the Anatomical Masking and Grid Tool, but this was determined through a process of trial and error.

While the toolkit under scrutiny has automated a substantial proportion of the workflow, it has concurrently afforded artists the capacity to manually refine outputs. This has been found to ensure that the procedural approach does not supersede artistic control, thus facilitating a nuanced and balanced integration of technology and creativity. Furthermore, some users discovered alternative uses for the toolkit beyond its original design, demonstrating its flexibility in supporting different artistic workflows. However, the learning curve and overview of functions posed initial barriers, requiring explanations and guidance for some artists to proceed. Figure 5.5 shows some of the armours and tools used to create them during the evaluation.

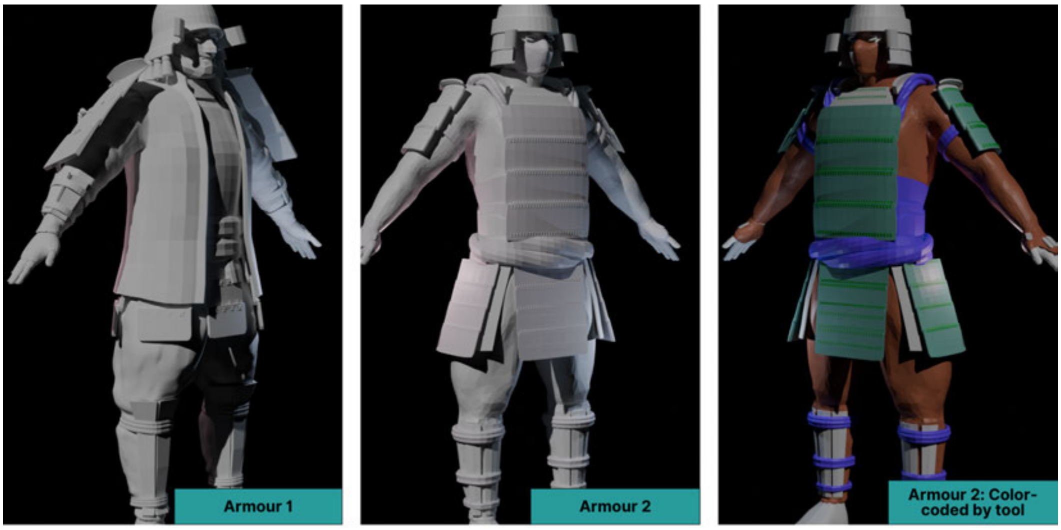

Figure 5.5: Samurai-style armours with reference from *For Honor* (2017) created during the evaluation process. The toolkit's usage is shown visually: turqoise-green represents the grid tool, purple the wrap tool and orange the anatomical masking.

6 Conclusion and Future Work

This thesis explored the development and implementation of a procedural toolkit for armour asset creation in game production. The primary goal was to enhance character artists' workflows by introducing procedural methodologies similar to those already utilized by environment artists. The research systematically examined existing manual workflows, technical constraints, and optimization needs, leading to the design and evaluation of a Houdini-based procedural toolkit.

The work has made significant contributions to the field of procedural armour asset creation. Firstly, it conceptualised and developed a prototype toolkit that functions similarly to Blender modifiers, thereby with the goal of streamlining the creation of armour base meshes. The prototype was designed with a strong focus on modularity and compatibility, drawing on contemporary tools, workflows, and methodological standards to ensure both scalability and future expansion possibilities. The integration of Blender with Houdini was facilitated by the development of a supporting pipeline plugin, enabling the effective utilisation of the toolkit for artists during the evaluation. The work further enhanced modularity and reusability, providing efficient means to generate varied armour designs suited to different body types and morphologies. Moreover, the evaluation demonstrated that procedural workflows can significantly reduce setup time through automated segmentation and predefined anatomical regions, eliminating the need for manual masking and partitioning. Finally, the findings that the prototype toolkit can form a fundamental part and framework to substantially support an artist's work without compromising their autonomy and creative direction.

6.1 Evaluation Outcomes and Validation of Hypotheses

The conducted evaluation process validated the hypothesis that procedural asset generation reduces manual workload while preserving artistic control. The toolkit successfully streamlined iterative workflows, with 70% of the final models integrating procedural elements.

However, certain limitations were identified:

- The toolkit proved most effective for rigid armour but **struggled with cloth-based** designs due to the absence of cloth simulation tools and more intricate mesh geometry.
- The integration with **Blender** required additional adaptation, as Houdini does not include a plugin like for Maya and the workflow differs from Blender's modifier-based modelling approach.
- While procedural workflows increased efficiency, **artistic oversight remained essential** to ensure high-quality and unique designs.
- Consistent tools that enriched the modelling process include: **anatomical masking, grid tools and wrap tool.**
- **Performance** of the tools proved as largely excellent with nearly all tools producing output fast and without crashes.
- The **user adoption varied**. Some artists fully embraced procedural workflows, while others relied on manual techniques for specific tasks (e.g. chainmail with particle systems).

- Certain tools, such as the **wrap tool, morph fit and scatter projection**, required refinement to improve stability and predictability.
- The toolkit's **onboarding** and UI **clarity** needed improvement, as unclear documentation led to underutilization of some features.
- Underutilised tools such as the Library and Morph Fit Tool require further investigation and evaluation. The Morph Fit Tool particularly provides largely unsatisfactory results in some cases and cannot yet be a suitable consistent use.
- During the evaluation process, the toolkit could be readily **updated** via a compact packed file, thereby ensuring the implementation of fixes, modularity, updates and seamless functionality in the event of any encountered bugs.
- The toolkit does not further address **production bottlenecks** or new bottlenecks formed by the altered workflow. New bottlenecks in further evaluation could be suitability of the generated armour parts for engine integration.

6.2 Thesis Limitations and Constraints

Overall, the evaluation demonstrated the strong potential of procedural workflows for armour creation while highlighting areas for further refinement. It is evident that the thesis encountered numerous constraints and limitations with regard to its conductibility, particularly in the domains of technical scope and evaluation. For the technical side, the toolkit relies on Houdini and might pose a limited accessibility for artists.

Technical Constraints:

- The toolkit relies on Houdini, limiting **accessibility** for artists using other software such as Maya or Blender.
- Due to the lack of a direct pipeline, while Maya can benefit from utilizing the tools as modifiers, Blender does not have the same advantage.
- Chainmail and padded armour are intended for **high-poly models** only, whereas the other parts of the toolkit are focused on enabling base-mesh creation.
- The toolkit does not enforce **clean topology or optimize polycount** and lacks integrated remeshing tools, leading to a need for manual adjustments.
- It does not fit well into workflows that rely extensively on Marvelous Designer or ZBrush.
- Certain armour types, such as those without metal plates or chainmail, may not benefit from the toolkit at all.

Scope Limitations:

- Insights into quantity and quality, as well as technical constraints of armour in games is generalised and lacks insights and further examples from specific games and engine solutions.
- The toolkit primarily supports medieval and rigid armour, with limited functionality for cloth-based assets.
- It lacks focus on other styled armours, detailed engravings, or higher-detail models.
- Only a limited number of tools could be developed and refined within the available time frame and successfully tested during evaluation.

- The toolkit function of the library and morph fitting is more suited for larger productions than for small-scale projects, as it is not fully informed by the specific restrictions and contexts of individual projects.
- It is designed primarily for generating modular armour sets, limiting its applicability for projects that require comprehensive, custom sculpting solutions.

Evaluation Constraints:

- The number of test users was limited, and participants had varying levels of procedural modelling experience, which affected the results.
- Due to lack of participants, no testing was conducted on Maya, which may skew insights regarding cross-platform compatibility.
- More **contextualisation** and testing in bigger productions would be beneficial.
- The toolkit's impact on realistic game armour production time may vary significantly depending on the project and artistic team, potentially limiting its **applicability**.

6.3 Technical Art Implications and Future Work

Due to the limitations faced during the development of the prototype, as well as the results of the evaluation, the prototype can be seen as the basis of a framework that can be developed further into two possible categories.

- **Scalable Procedural Armour Generation for Large Game Worlds** The findings of this thesis demonstrate considerable potential for procedural asset generation in large-scale game productions, thus indicating several avenues for future research. For instance, as evidenced by the recent releases of Kingdom Come: Deliverance 2 (2025) and Path of Exile 2 (2024), there is an increasing demand for a diverse array of armour sets tailored to complex narrative and gameplay scenarios.In Kingdom Come: Deliverance 2, extensive cutscenes and in-game siege battles require a vast number of distinct armour variations for numerous non-player characters across various factions. The library is designed to facilitate the creation of variations and customisations for a substantial number of soldiers, with the potential for enhancement to accommodate factions and styles (e.g. chainmail, padded, metal) in addition to weight classifications. The generator could be upgraded to operate based on a user pattern input, such as necessity or pattern, in a manner analogous to the Houdini House Generator. In the LABS House Generator, a pattern constitutes an input following a ruleset which is identified by the system to translate into the manner in which the house should be assembled, given the defined modules (e.g. number of windows next to one another, floors). The artist is thus able to specify the components of a given suit of armour, whereupon the system generates the relevant pieces, including the mail coif, gorget, vambrace, cuisse, plackart, and doublet. This system draws upon classifications and known components of armour to define and recognise the pieces, in a manner analogous to the anatomical definitions.

- **Exploration of new modules and methodologies**

The development of the toolkit's contents should be a primary focus, with the implementation of novel methodologies to adjust, enhance and optimise. This approach would facilitate a greater emphasis on smaller workflows and productions by potentially incorporating a pipeline framework into software such as Marvelous Designer, Substance Painter or ZBrush, or by utilising tools that are already well-established within a production environment. Additionally, contexts should be considered for when the toolkit is adapted mid-production with existing meshes and rigs. A generalised definition system could be adapted to input new rules for rigs and meshes that are pre-established. More methods for modules regarding cloth or detailed ornament armours, as seen in *Diablo 4* and *Path of Exile 2*, can further be considered, along with new methodologies regarding simulations, atlas reading, and approaches in data input via scale-based blocking.

One of the most significant implications of this research is the demonstrated viability of procedural methods for prototyping tools for character and armour design, as well as the compatibility with Blender. While procedural workflows have been widely adopted for environment assets, their application to character-based assets remains underdeveloped. As game productions increasingly demand a high volume of unique, reusable assets, procedural generation must adapt to large-scale pipelines. This necessitates that procedural generation must adapt to large-scale pipelines. The expansion of procedural armour generation for faction-based systems, as evidenced by the recent developments in *Kingdom Come: Deliverance 2*, suggests that rule-based pattern generation could serve as a scalable approach to asset creation.

In summary, the present study provides a robust foundation for procedural toolkits for armour generation. The study emphasises the necessity for enhanced customisation, cross-software compatibility and scalability. By addressing these areas, procedural workflows can continue to evolve into a powerful asset creation method that complements traditional artistic processes while meeting the growing demands of modern game development.

Appendices

A Armour Examples

A.1 Lords of the Fallen

Figure A.1: Examples of armour 3D models from the game Lords of the Fallen (2022).[62]

Figure A.2: Further examples of armour 3D models from the game Lords of the Fallen (2022).[63]

[62]https://www.artstation.com/dv3d_gaming
[63]https://www.artstation.com/a46studio

A.2 Elden Ring

Figure A.3: Examples of armour concept art from the game Elden Ring. [64]

Figure A.4: Further examples of armour concept art from the game Elden Ring. [65]

[64] Elden Ring Artbook I & II
[65] https://www.artstation.com/artwork/8wE4NQ

A.3 Baldur's Gate 3

Figure A.5: Examples of armour 3D models from the game Baldur's Gate 3. 6667

[67]https://www.artstation.com/keos-masons
[67]https://www.artstation.com/metadumplings

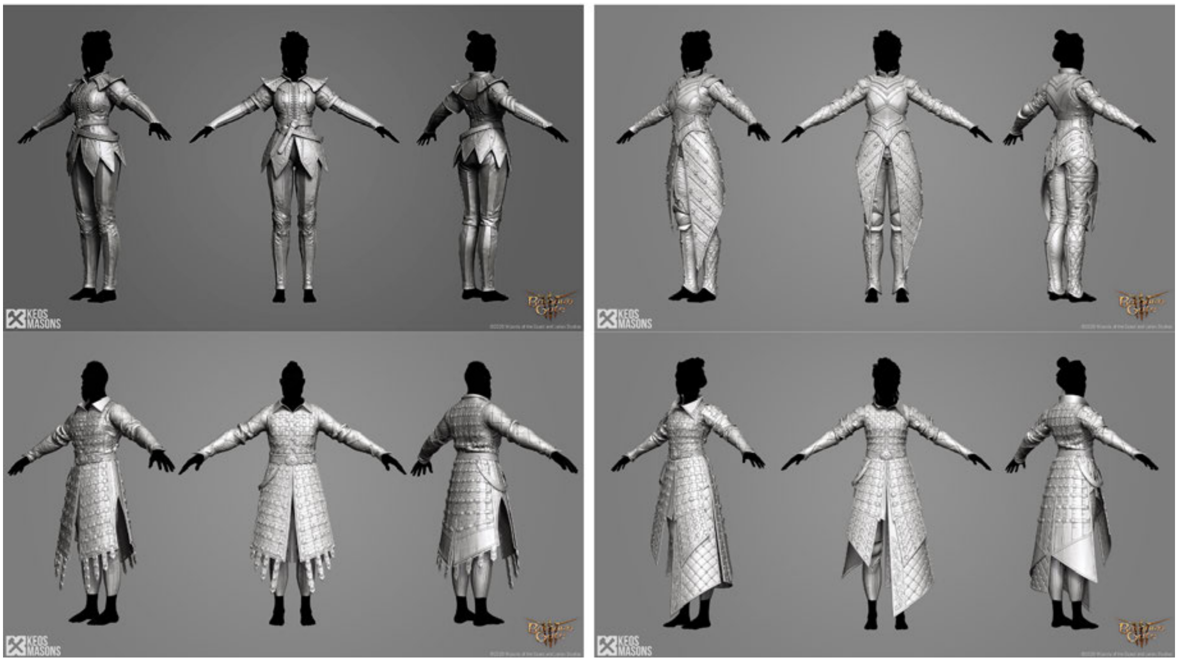

Figure A.6: Further examples of armour 3D models from the game Baldur's Gate 3.

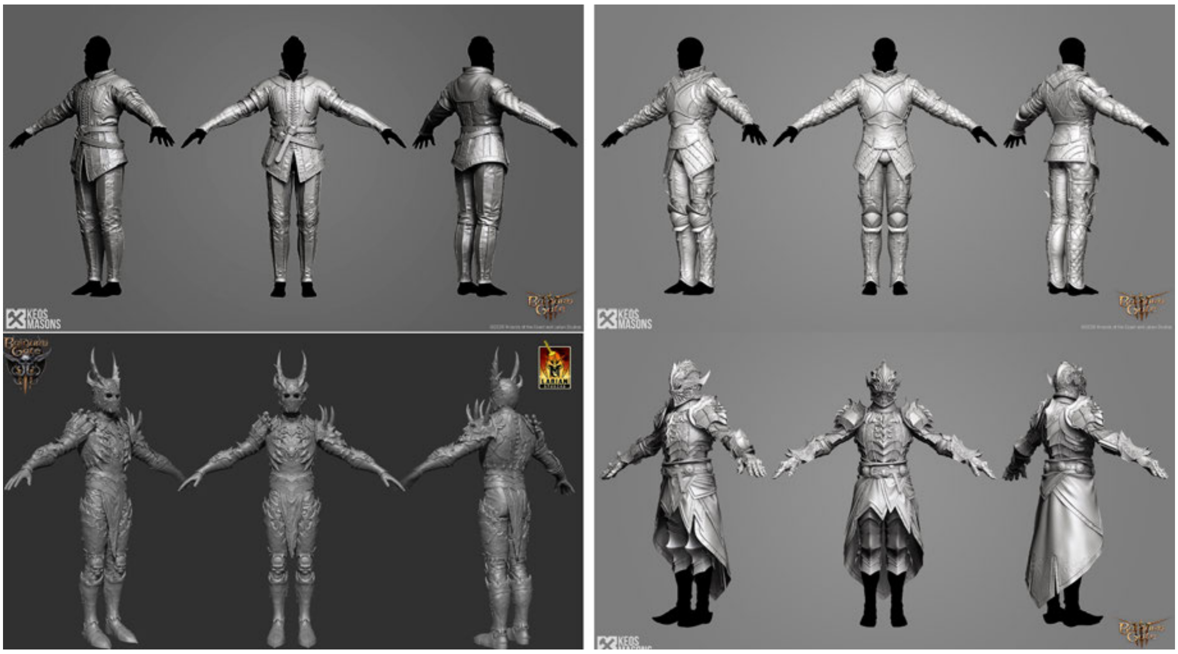

Figure A.7: Further examples of armour 3D models from the game Baldur's Gate 3.
68

[68] https://www.artstation.com/corbax

A.4 Diablo 4

Figure A.8: Examples of armour 3D models from the game Diablo 4.
[69][70]

[70] https://www.artstation.com/bekkiwilkinson
[70] https://www.artstation.com/artbullyproductions

Figure A.9: Further examples of armour 3D models from the game Diablo 4. [71][72]

[72] https://www.artstation.com/artbullyproductions
[72] https://www.artstation.com/entgld

Figure A.10: Further examples of armour 3D models from the game Diablo 4. [73][74]

[74] https://www.artstation.com/artbullyproductions
[74] https://www.artstation.com/entgld

A.5 For Honor

Figure A.11: Examples of armour 3D models from the game For Honor. [75]

[75] https://www.artstation.com/annikc

Figure A.12: Further examples of armour 3D models from the game For Honor.
[76][77]

[77] https://www.artstation.com/saucerain
[77] https://www.artstation.com/artwork/L3YdGP

Figure A.13: Further examples of armour 3D models from the game For Honor.
78

B Games represented in Armour Amount

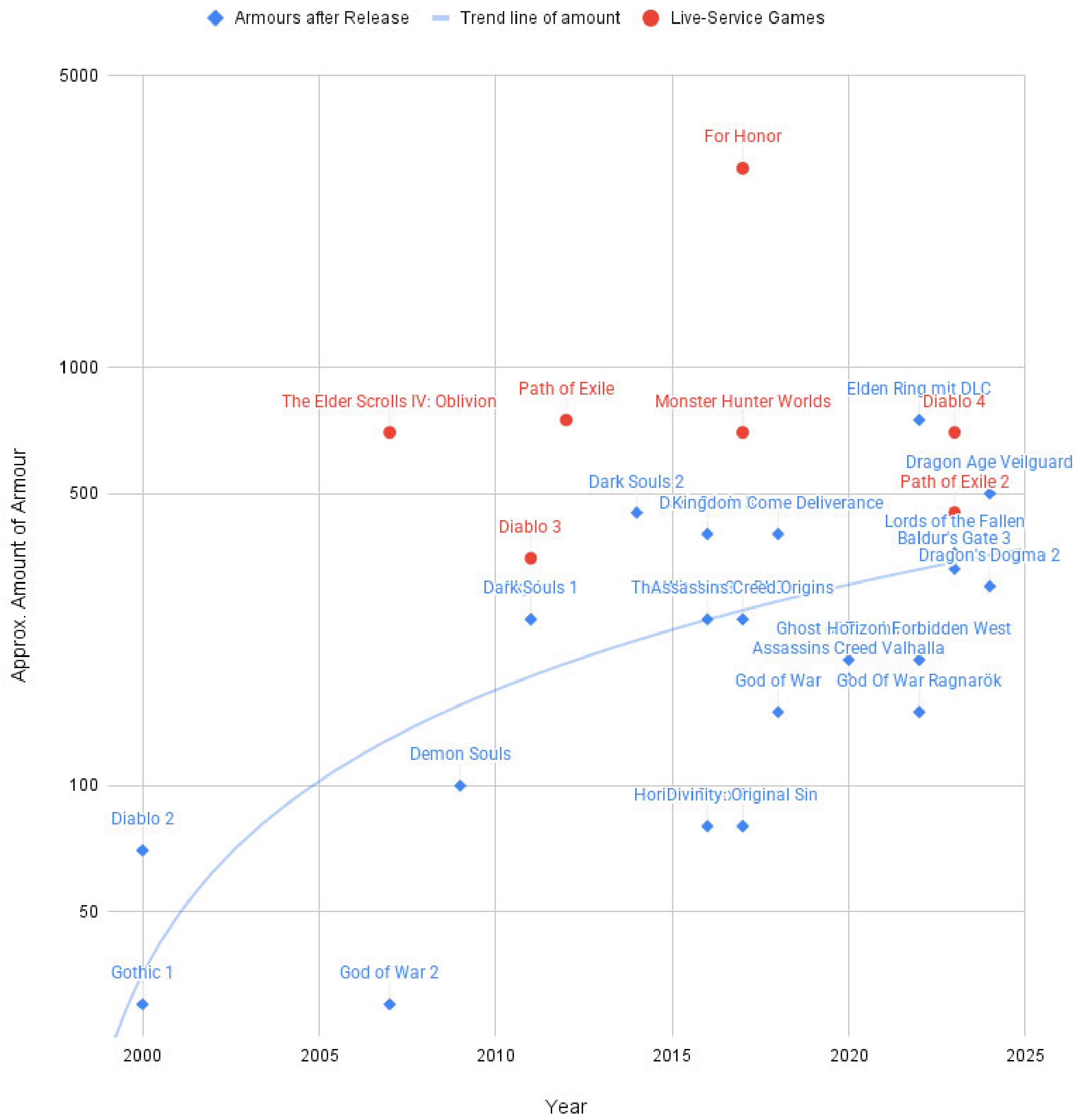

Figure B.1: Example Armour Sets from Blizzard's Diablo 4

Game Name	Year	Equip Slots	Approx. Armour Pieces	Source
Baldur's Gate 2	2000	1	150	https://baldursgate.fandom.com/wiki/Armor
Diablo 2	2000	5	70	https://diablo2.diablowiki.net/Body_Armor
Gothic 1	2000	1	30	https://mikesrpgcenter.com/gothic/armor/armor.html
God of War 2	2007	1	30	https://godofwar.fandom.com/wiki/Category:God_of_War_II_Armor
The Elder Scrolls IV: Oblivion	2006	9	700	https://elderscrolls.fandom.com/wiki/Armor_Sets_(Oblivion)
Demon Souls	2009	4	100	https://demonssouls.wiki.fextralife.com/Armor
Skyrim	2011	4	250	https://elderscrolls.fandom.com/wiki/Armor_(Skyrim)
Dark Souls 1	2011	4	250	https://darksouls.wiki.fextralife.com/Armor
Diablo 3	2012	5	350	https://diablo.fandom.com/wiki/List_of_Sets_(Diablo_III)
Dark Souls 2	2014	4	450	https://darksouls2.wiki.fextralife.com/Armor
Path of Exile	2014	4	750	https://pathofexile.fandom.com/wiki/Body_Armour
Dark Souls 3	2016	4	400	https://darksouls3.wiki.fextralife.com/Armor
The Witcher 3	2016	4	250	https://witcher.fandom.com/wiki/The_Witcher_3_armor
Horizon Zero Dawn	2016	1	80	https://horizon.fandom.com/wiki/Outfit
For Honor	2017	4	3000	https://forfashion.draghmar.pl/
Divinity: Original Sin	2017	1	80	https://divinityoriginalsin2.wiki.fextralife.com/Armor
Assassins Creed Origins	2017	1	250	https://assassinscreed.fandom.com/wiki/Wardrobe_of_Bayek
Monster Hunter Worlds	2018	5	700	https://monsterhunterworld.wiki.fextralife.com/Armor
God of War	2018	3	150	https://godofwar.fandom.com/wiki/Armor_Sets

Game Name	Year	Equip Slots	Approx. Armour Pieces	Source
Kingdom Come Deliverance	2018	20	400	https://kingdomcomedeliverance-archive.fandom.com/wiki/Armour
Assassins Creed Valhalla	2020	1	180	https://assassinscreedvalhalla.wiki.fextralife.com/Armor
Ghost of Tsushima	2020	1	200	https://ghostoftsushima.fandom.com/wiki/Armor
God Of War Ragnarök	2022	3	150	https://godofwar.fandom.com/wiki/Armor_Sets_(Ragnar%C3%B6k)
Horizon Forbidden West	2022	1	200	https://www.ign.com/wikis/horizon-2-forbidden-west/All_Outfits_and_Armors
Elden Ring	2022	4	750	https://eldenring.wiki.fextralife.com/Armor
Diablo 4	2023	5	700	https://diablo4.wiki.fextralife.com/Wardrobe
Baldur's Gate 3	2023	1	330	https://baldursgate3.wiki.fextralife.com/Armor
Lords of the Fallen	2023	4	360	https://thelordsofthefallen.wiki.fextralife.com/Armor
Path of Exile 2	2024	4	450	https://www.poe2wiki.net/wiki/Armour_(equipment)
Dragon's Dogma 2	2024	4	300	https://dragonsdogma2.wiki.fextralife.com/Armor
Dragon Age Veilguard	2024	2	500	https://dragonageveilguard.wiki.fextralife.com/Armor

C Games Referenced

Bandai Namco, FromSoftware, 2022. Elden Ring. PlayStation 4, PlayStation 5, Windows, Xbox One, Xbox Series X/S.

Blizzard Entertainment, 2004. World of Warcraft. Mac OS, Microsoft Windows.

Blizzard Entertainment, 2023. Diablo 4. PlayStation 4, PlayStation 5, Microsoft Windows, Xbox One, Xbox Series X/S.

CI Games, Hexworks, 2023. Lords of the Fallen. Microsoft Windows, PlayStation 5, Xbox Series X/S.

Electronic Arts, 2024. Dragon Age: The Veilguard. Microsoft Windows, PlayStation 5, Xbox Series X/S.

Larian Studios, 2023. Baldur's Gate 3. Microsoft Windows, PlayStation 5, macOS, Xbox Series X/S.

Sony Computer Entertainment, Guerrilla Games 2017. Horizon Zero Dawn. PlayStation 4.

Sony Computer Entertainment, Guerrilla Games 2022. Horizon Forbidden West. PlayStation 4, Playstation 5.

Ubisoft, 2017. For Honor. PlayStation 4, Microsoft Windows, Xbox One.

Ubisoft, 2020. Assassin's Creed Valhalla. PlayStation 4, PlayStation 5, Microsoft Windows, Xbox One, Xbox Series X/S, Stadia.

Warhorse Studios, 2025. Kingdom Come Deliverance 2. PlayStation 5, Xbox Series, Microsoft Windows.

Grinding Gear Games, 2024. Path of Exile 2. PlayStation 5, Xbox Series, Microsoft Windows.

D Additional Sources

D.1 Picture Sources to Armour Timeline

Source	Name	Accessed
https://www.youtube.com/watch?v=MSd72mST7-M&ab_channel=MajorThriftwood	“Lands of Lore: The Throne of Chaos (PC/DOS) PART-2, 1993, Westwood Studios "CD"” by Major Thriftwood	15.02.25
https://diablo.fandom.com/wiki/Sorcerer_(Diablo_I)	“Sorcerer (Diablo I)” in Diablo Fandom Wiki	15.02.25
https://lilura1.blogspot.com/2022/06/Shadow-Dragon-Scale-Armor-Baldurs-Gate-2.html	“Shadow Dragon Scale Armor Baldur’s Gate 2”	15.02.25
https://www.wowhead.com/cata/gallery=1542/armor-sets	“World of Warcraft Armor Sets” Database	15.02.25
http://darksouls.wikidot.com/elite-knight-set	“Elite Knight Set” on Dark Souls Wiki	15.02.25
https://pathofexile.fandom.com/wiki/Twilight_Body_Armour	“Twilight Body Armour” on the Path of Exile Wiki	15.02.25

Source	Name	Accessed
https://forfashion.draghmar.pl/	For Honor Fashion: Armour and Weapon Database	15.02.25
https://www.powerpyx.com/god-of-war-2018-all-armor-sets/	"God of War (2018): All Armor Sets" by PowerPyx	15.02.25
https://www.artstation.com/artwork/xJnGaW	"Chivalry 2 - Agatha Plate Knight" by Chris Nirenberg	15.02.25
https://www.reddit.com/r/Eldenring/comments/t9t1k4/banished_knight_armor_variants/	"Banished Knight Armor Variants" on Reddit	15.02.25
https://gamerant.com/diablo-4-best-barbarian-armor-sets-ranked/	"Diablo 4: 8 Best Barbarian Armor Sets, Ranked" by Megan Smith	15.02.25
https://www.youtube.com/watch?v=movmBGA3gUg&ab_channel=DareToGame	"Dragon's Dogma 2: Top 5 Fighter Armor Sets Ranked" by Dare To Game	15.02.25

D.2 List of Articles and Sources contributing to the Workflow Chapter

Source	Name	Accessed
https://gamesartist.co.uk/ottoman-ghazi/	"Ottoman Ghazi" by Sougato Majumder	05.12.24
https://80.lv/articles/creature-character-art-pipeline/	"Creature & Character Art Pipeline" by Sergeev, A., & Loginova, D.	05.12.24
https://80.lv/articles/thief-texturing-workflow-for-characters-in-substance-and-marmoset/	"Thief: Texturing Workflow for Characters in Substance and Marmoset" by Marleen Vijgen	05.12.24
https://80.lv/articles/learn-how-to-craft-intricate-clothing-historic-chainmail-in-3d/	"Learn How To Craft Intricate Clothing & Historic Chainmail In 3D" by Ian McOran-Campbell	05.12.24
https://www.artstation.com/artwork/8eGgBq	"Tlaloc (The Rookies Breakdown)" by Kaloyan Kalamov	05.12.24
https://80.lv/articles/making-an-armored-orc-from-world-of-warcraft-with-zbrush-substance/	"Making an Armored Orc from World of Warcraft with ZBrush & Substance" by Aaron Hunwick	05.12.24
https://80.lv/articles/don-t-nod-s-banishers-character-art-workflows-digital-outfits-creation/	"Don't Nod's Banishers: Character Art Workflows & Digital Outfits Creation" by Victorien Berry	05.12.24
https://discover.therookies.co/2024/04/28/3d-character-creation-pipeline-for-games/	"3D Character Creation Pipeline for Games" by Abbas Nikmanesh Sigaroudi	05.12.24
https://80.lv/articles/001agt-002mrs-004adk-knight-production-with-valery-klishevich/	"Knight Production with Valery Klishevich" by Valery Klishevich	05.12.24

Source	Name	Accessed
https://80.lv/articles/shardbound-s-character-art-pipeline-explained/	“Shardbound’s Character Art Pipeline Explained” by Ira Goeddel	05.12.24
https://www.artstation.com/artwork/D5EBqG	“Engraving Guide” by Yuriy Porubov	05.12.24
https://www.proquest.com/openview/d5a075ba4e14d2bde71f0881045db6d4/1?pq-origsite=gscholar&cbl=18750	“World warrior” by Dan Farruggia	05.12.24
https://www.artstation.com/artwork/28BG8x	“Exp-Points: Wonder Park foliage and samurai armor” by Jesus Merino	05.12.24

E Additional Implementation Visualisation

E.1 System Architecture Charts

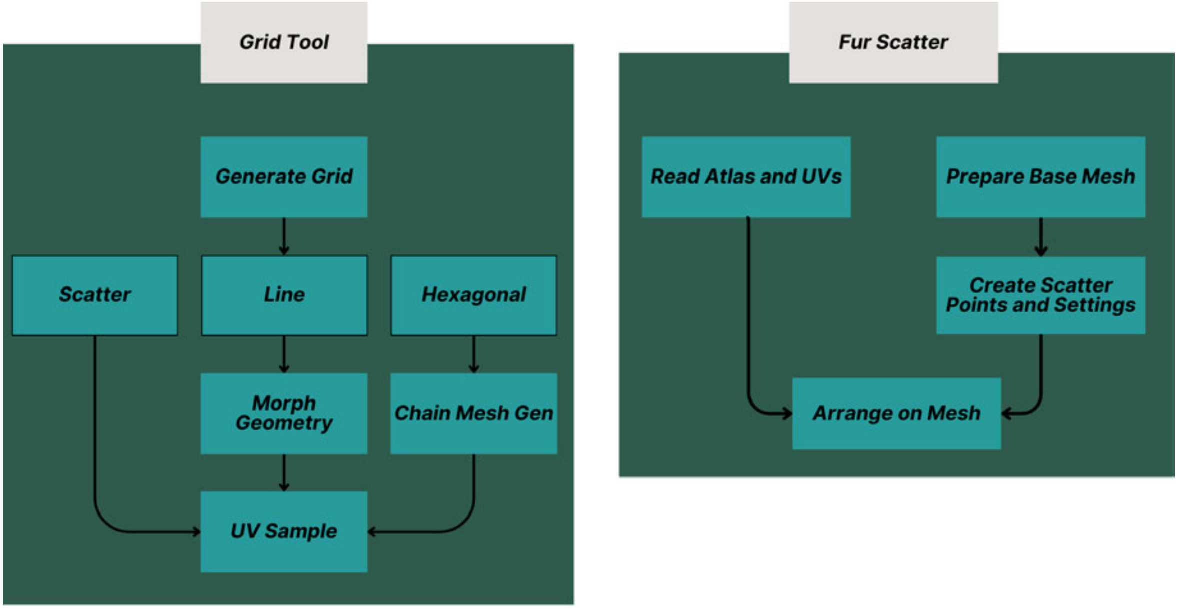

Figure E.1: Abstracted chart of node network within the Grid and Fur Scattering Tool sorted by sub-networks.

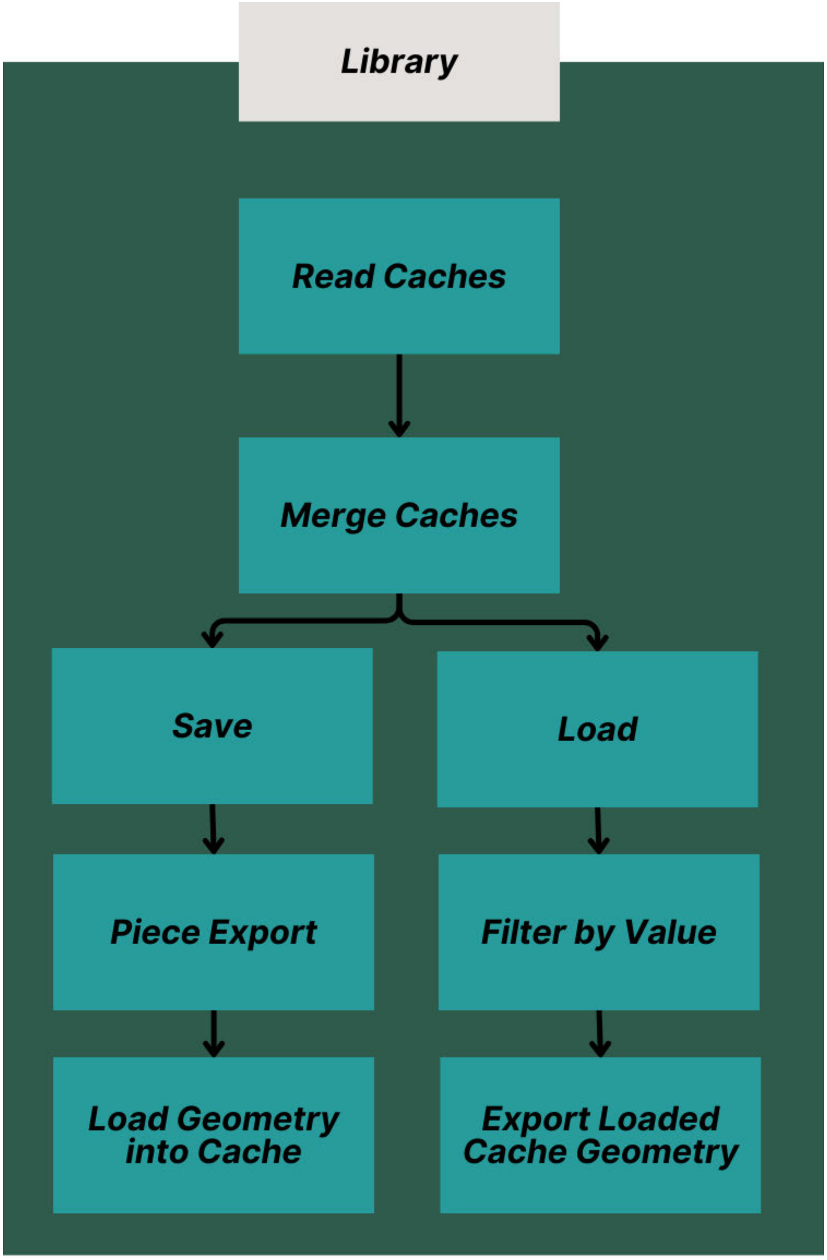

Figure E.2: Abstracted chart of node network within the Library PDG sorted by task steps.

E.2 Node Network of Fur Scatter Tool

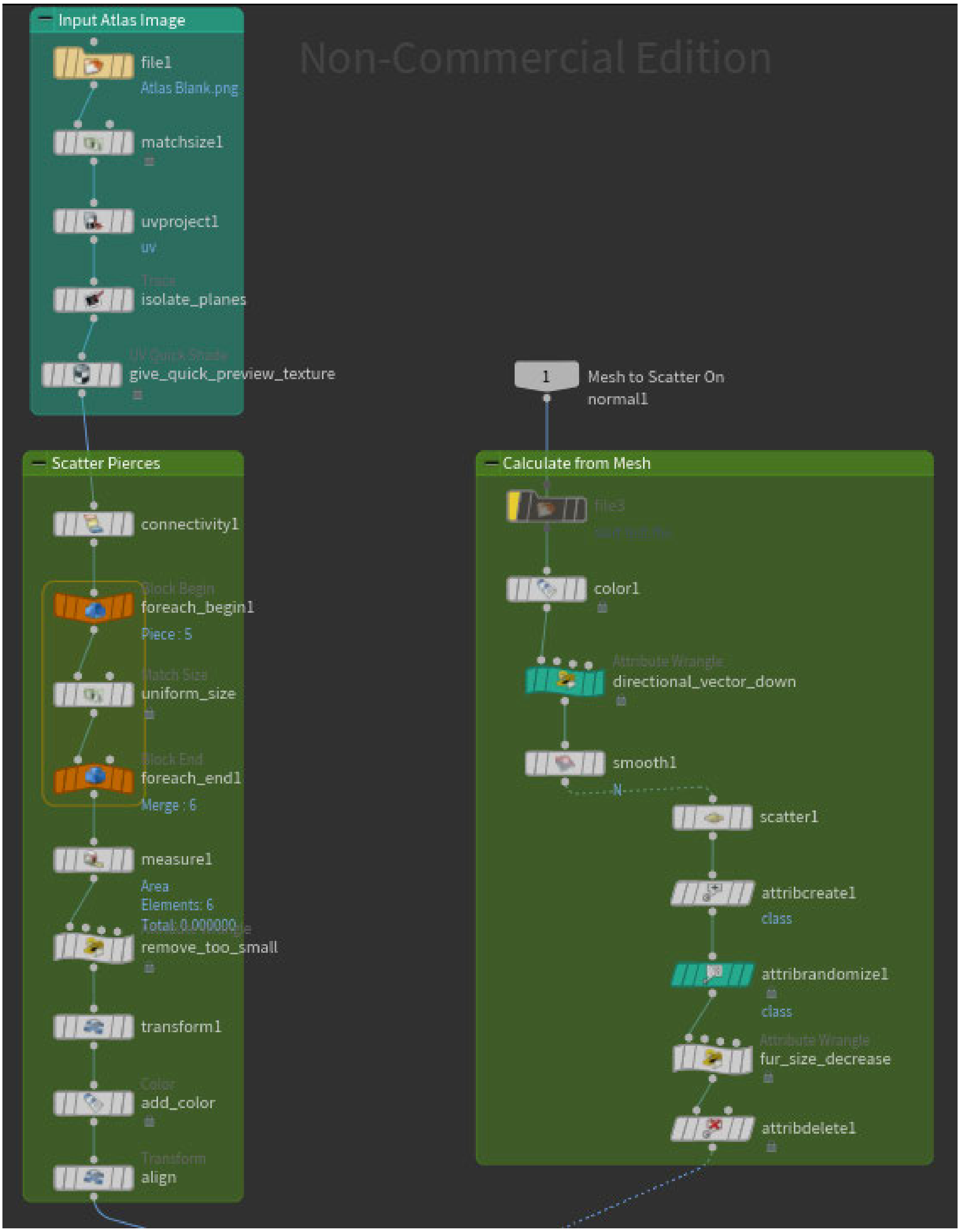

Figure E.3: Grouped Node Graph of Fur Scattering Tool

E.3 Tool Use-Case Examples

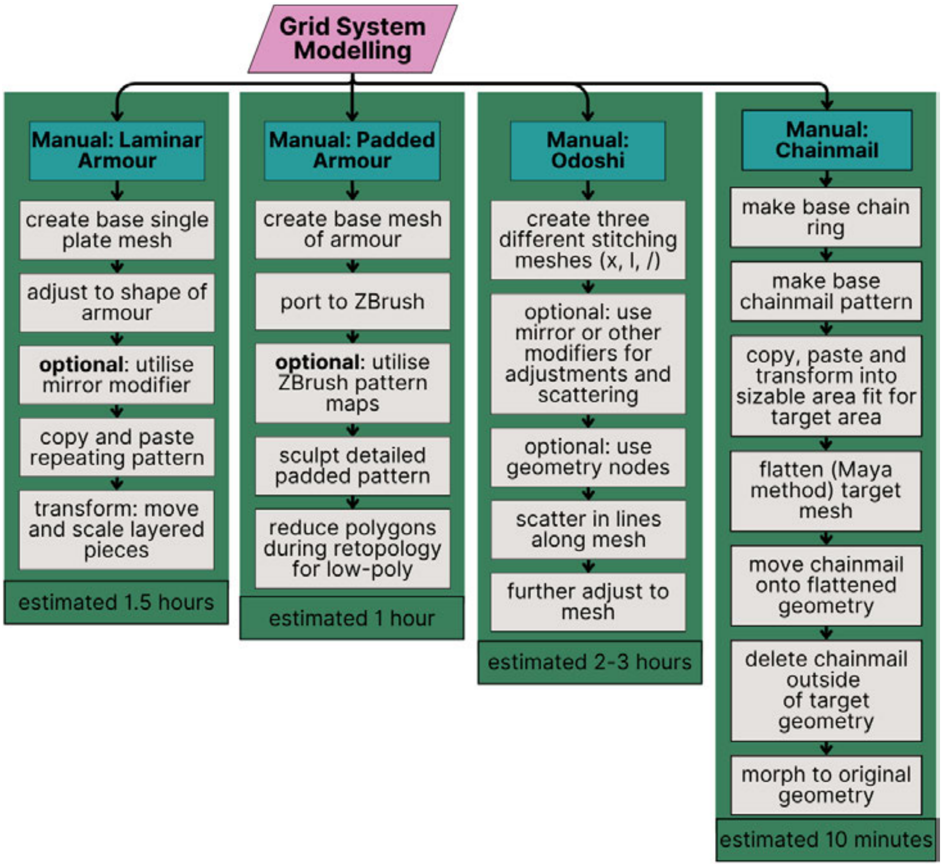

Figure E.4: Workflow Diagram: Manual Processes of Armour parts (laminar, padded, odoshi lacing, chainmail), to be replaced by Grid Tool

Figure E.5: Colour-Coded Armour Toolkit Use-Cases on examples of For Honor Armour Sets [79]

[79] https://www.artstation.com/artwork/036595, visited 16.01.2025

Area Marked	Suggested Tool
Orange/Yellow Plate Armour	1. Anatomical Masking with choice of chest, body based on rig-based grid method and 2. Anatomical Masking with custom grid with isolating the arm mesh.
Red Shoulder and Arm Layered Plates	1. Anatomical masking with a custom grid on the arms and shoulder. 2. A grid system utilising one line, bend and split to shape. 3. The provision of input or adjustment of shape automatically. 4. Mirroring.
Green Layered Leg Plates	1. Anatomical masking with the Voronoi method utilising a custom directional grid or rig grid on the legs. 2. A grid system utilising one line, bend and split to shape. 3. The input shape or the adjusted shape must be given. 4. Mirror.
Turqouise Back Leg Plates and Light pink additional plates	The following two methods of anatomical masking are proposed: 1. Anatomical masking with Voronoi grid masking applied to the legs. 2. Anatomical masking with a cluster with a directional-based custom grid to isolate the back of the legs.
Light Green Padded Armour	The grid tool can be configured to display padded armour with a rotated block grid.
Purple Belts	The Belt Wrap Around tool is an effective solution for this application.
Pink sash	The utilisation of the Random Shape and Belt Wrap Around tool is recommended.

F Evaluation Age Data

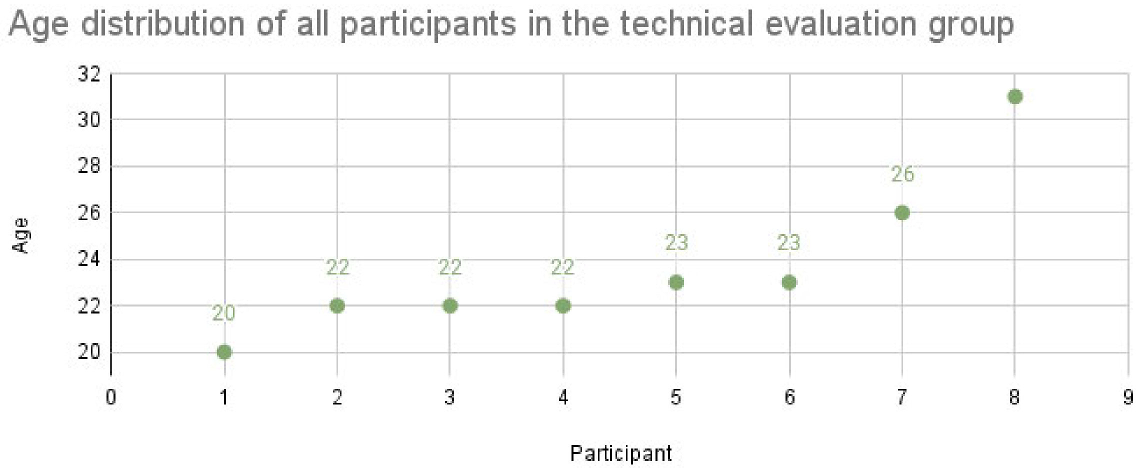

Figure F.1: Age distribution of all male participants in the technical evaluation group. Median is 22,5. Oldest participant and outlier is 31.

Bibliography

[AD19] D. Aversa and C. Dickinson. *Unity Game Optimization: Enhance and Extend the Performance of All Aspects of Your Unity Games, 3rd Edition*. 3rd ed. Vol. 1. 1. Packt Publishing, Limited, Nov. 2019. 404 pp. ISBN: 9781838556518.

[Bus21] Philipp Buschauer. “Building An Artist Friendly CFX Pipeline”. In: *Houdini HIVE SIGGRAPH 2021*. URL: https://www.sidefx.com/houdini-hive/houdini-hive-siggraph/, visited 02.02.2025, Presented at the Houdini HIVE SIGGRAPH, held on August 10, 2021. Axis Studios. SideFX: SideFX, Aug. 2021.

[Dom+24] José Miguel Domingues et al. “Understanding the Impact of Perceived Challenge on Narrative Immersion in Video Games: The Role-Playing Game Genre as a Case Study”. In: *Information* 15.6 (2024). ISSN: 2078-2489. DOI: 10.3390/info15060294.

[Ebe+02] David S. Ebert et al. *Texturing and Modeling: A Procedural Approach*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. 721 pp. ISBN: 1558608486.

[Gol14] Matthias Goll. “Iron Documents. Interdisciplinary studies on the technology of late medieval european plate armour production between 1350 and 1500”. PhD thesis. Universität Heidelberg, Jan. 2014. 232 pp.

[Hol20] Paulina Hollstrand. “Supporting Pre-Production in Game Development : Process Mapping and Principles for a Procedural Prototyping Tool”. MA thesis. KTH, School of Electrical Engineering and Computer Science (EECS), 2020. 17 pp.

[KSS13] Jussi Kasurinen, Jukka-Pekka Strandén, and Kari Smolander. “What do game developers expect from development and design tools?” In: *ACM International Conference Proceeding Series* (Apr. 2013), pp. 36–41. DOI: 10.1145/2460999.2461004.

[Kov15] Simo Kovanen. “Functional Workflow in 3D Character Design”. Bachelor’s thesis. Karelia University of Applied Sciences, Oct. 2015. 28 pp.

[Mcm03] Alison Mcmahan. “Immersion, engagement, and presence: A method for analyzing 3-D video games”. In: *The Video Game Theory Reader* (Jan. 2003), pp. 67–86.

[Met19] Petra Mether. “Character Customization in Video Games”. Bachelor’s thesis. The address of the publisher: Tampere University of Applied Sciences, July 2019. 57 pp.

[Mit24] Anita Mitchell. *Advances in Sewn Product Technology*. 1st ed. Vol. 1. 1. CRC Press, May 2024. 244 pp. ISBN: 9781003858546.

[NM09] Mark Nelson and Michael Mateas. “A requirements analysis for videogame design support tools”. In: *Proceedings of the 4th International Conference on Foundations of Digital Games, FDG 2009, Orlando, Florida, USA, April 26-30, 2009*. Apr. 2009, pp. 137–144. DOI: 10.1145/1536513.1536543.

[Rod24] Chris Roda. *Essential Programming for the Technical Artist*. 1st ed. Vol. 1. 1. CRC Press, May 2024. 501 pp. ISBN: 0367820404.

[SA17] Tanya Short and Tarn Adams. *Procedural Generation in Game Design*. 1st ed. Vol. 1. 1. CRC Press, June 2017. 336 pp. ISBN: 1498799205.

[Sid24] SideFX. *Houdini – 3D Procedural Software for Film, Game, and VFX Production*. URL: https://www.sidefx.com/, visited 05.02.2025. Feb. 2024.

[Smi14] Gillian Smith. "Understanding procedural content generation: a design-centric analysis of the role of PCG in games". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 917–926. ISBN: 9781450324731. DOI: 10.1145/2556288.2557341. URL: https://doi.org/10.1145/2556288.2557341.

[Smi15] Gillian Smith. "Procedural Content Generation: An Overview". In: vol. 2. CRC Press, Apr. 2015. Chap. 40, pp. 501–518. ISBN: 978-1-4822-5479-2. DOI: 10.1201/b18373-48.

[Smi21] Gillian Smith. "The Future of Procedural Content Generation in Games". In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 10.3 (June 2021), pp. 53–57. DOI: 10.1609/aiide.v10i3.12748. URL: https://ojs.aaai.org/index.php/AIIDE/article/view/12748.

[Ter17] Tapio Terävä. "Workflows for Creating 3D Game Characters". Bachelor's thesis. Kajaanin University of Applied Sciences, 2017. 154 pp.

[Tra08] Oliver M. Traxel. "Medieval and Pseudo-Medieval Elements in Computer Role-Playing Games: Use and Interactivity". In: *Studies in Medievalism XVI: Medievalism in Technology Old and New*. Ed. by Karl Fugelso and Carol L.Editors Robinson. Studies in Medievalism. Boydell 38; Brewer, 2008, pp. 125–142.

[TA15] Selen Turkay and Sonam Adinolf. "The effects of customization on motivation in an extended study with a massively multiplayer online roleplaying game". In: *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* 9.3 (Oct. 2015), Article 2. DOI: 10.5817/CP2015-3-2.

[USF10] Torsten Ullrich, Christoph Schinko, and Dieter Fellner. "Procedural modeling in theory and practice". In: *Fraunhofer IGD* 18 (Jan. 2010), pp. 5–8.

[Yee07] Nick Yee. "Motivations for Play in Online Games". In: *Cyberpsychology behavior : the impact of the Internet, multimedia and virtual reality on behavior and society* 9 (Jan. 2007), pp. 772–775. DOI: 10.1089/cpb.2006.9.772.

Declaration of Authorship

I do hereby declare that I have composed the presented work independently on my own and without any other resources than the ones given.

All thoughts taken directly or indirectly from external sources are correctly acknowledged.

This work has neither been previously submitted to another authority nor has it been published yet.

Mittweida, 11. April 2025 [redacted]

Location, Date Weidner Fanny