More Than Killmonger Locs:
A Style Guide for Black Hair (in Computer Graphics)

A.M. DARKE, University of California, Santa Cruz, USA
ISAAC OLANDER, Tallgran, Sweden
THEODORE KIM, Yale University, USA

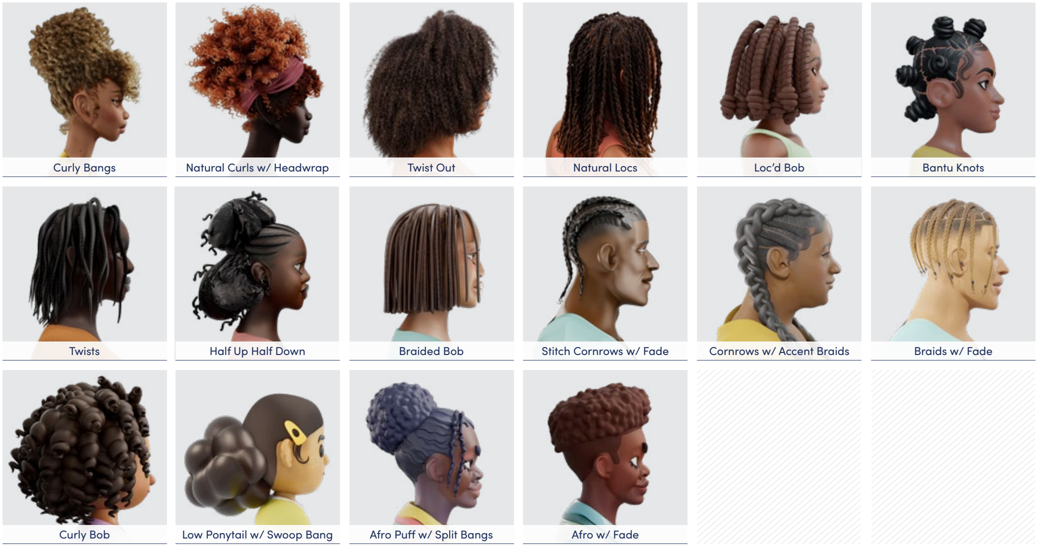


Fig. 1. Hair types from *Code My Crown*, a guide to building digital Black hair co-authored by one of the presenters. In addition to these digital styles, real-life photographic references for each will also be presented.

We will cover recent advances and ongoing challenges in the depiction of Black hair, otherwise known as kinky, or Afro-textured hair. In computer graphics, the majority hair research has been in the depiction straight or wavy hair. As a result, many aspects of the aesthetics and mechanics of Black hair remain poorly understood. To help fill this gap, we will present *Code My Crown*, a free guide to creating Black digital hairstyles that we co-authored in collaboration with a community of game artists and Dove. We also cover styling guidelines for 3D models in the *Open Source Afro Hair Library*, and present *Lifted Curls*, our strand simulation technique specifically designed for Afro-textured hair. Finally, we will suggest future directions for hair research.

ACM Reference Format:
A.M. Darke, Isaac Olander, and Theodore Kim. 2024. More Than Killmonger Locs: A Style Guide for Black Hair (in Computer Graphics). 1, 1 (July 2024), 251 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: A.M. Darke, darke@ucsc.edu, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA, 95064, USA; Isaac Olander, isaacolander@tallgran.com, Tallgran, Stockholm, Sweden; Theodore Kim, theodore.kim@yale.edu, Yale University, USA.

SIGGRAPH Courses '24, July 27 - August 01, 2024, Denver, CO, USA
ACM 979-8-4007-0683-7/24/07.
10.1145/3664475.3664535

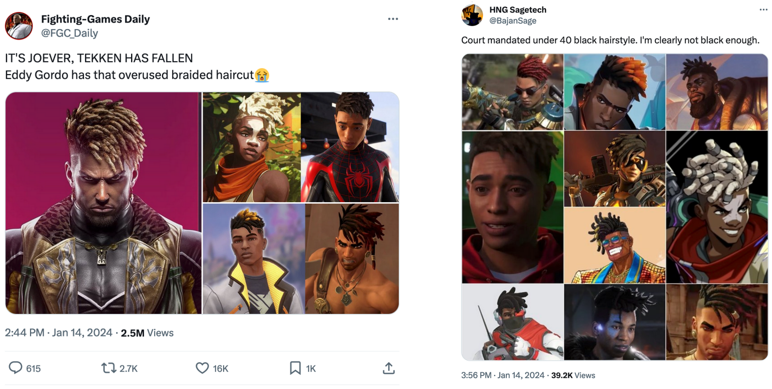
1 INTRODUCTION

According to a recent survey from Edelman DXI, 85% of Black gamers believe that Black hair, otherwise known as kinky or Afro-textured hair, is poorly represented in video games. [Darke et al. 2023] This dissatisfaction extends to game developers, where nearly 9 in 10 developers agree more resources are needed to enable them to create more realistic Black hair styles.

This lack of Black hair assets, and the lack of tools to support them, is not surprising when viewed in terms of the fundamental computer graphics research conducted on human hair. In one of our recent Talks [Kim et al. 2022], we surveyed the technical papers published at *SIGGRAPH* and other high-profile computer graphics venues, and found that 41 of the foundational papers on the topic only contained images of straight or wavy hair. Black hair has not been given the level of basic scientific attention of straighter hair types, resulting in a downstream lack of tools, and consequently, a dearth of assets and models.

As of 2021, we only found two technical papers that showed Afro-textured hair [Bertails et al. 2005; Patrick et al. 2004]. While some technical papers containing Black hair have started to appear since we published our findings [Hsu et al. 2023; Wang et al. 2023], these usually represent relatively unstyled hair. Some styles have started to appear in short talks [Ogunseitan 2022], but a wide range of intricate, sophisticated, and very common hair styles still lie outside the visual language of computer graphics research.

For example [IGN 2024], the hairstyle of the Michael B. Jordan character *Killmonger* from *Black Panther* (2018) has been widely and successfully imitated in games since the success of that movie (Figure 2). Once the visual language is understood, wider representation of hair styles is possible. At the same time, the success of that specific hairstyle has reached a saturation point, and if it becomes interpreted as the *only* Black hairstyle that needs to bring inclusion games, then the wider goal of full representation remains unfulfilled.



Fighting-Games Daily
@FGC_Daily

IT'S JOEVER, TEKKEN HAS FALLEN
Eddy Gordo has that overused braided haircut😭

2:44 PM · Jan 14, 2024 · 2.5M Views

615 2.7K 16K 1K

HNG Sagetech
@BajanSage

Court mandated under 40 black hairstyle. I'm clearly not black enough.

3:56 PM · Jan 14, 2024 · 39.2K Views

Fig. 2. The *Killmonger* style from *Black Panther* (2018), has been so successfully transferred to games that it has become a clichè. A wider range of other hair styles is possible.

2 THE *CODE MY CROWN* GUIDE

To address this gap in knowledge, two of us collaborated with a community of digital artists and the cosmetics company *Dove* to assemble the *Code My Crown* guide, a free guide to digitally constructing a wide range of Black hairstyles. [Darke et al. 2023] The assets we generated for this guide are also

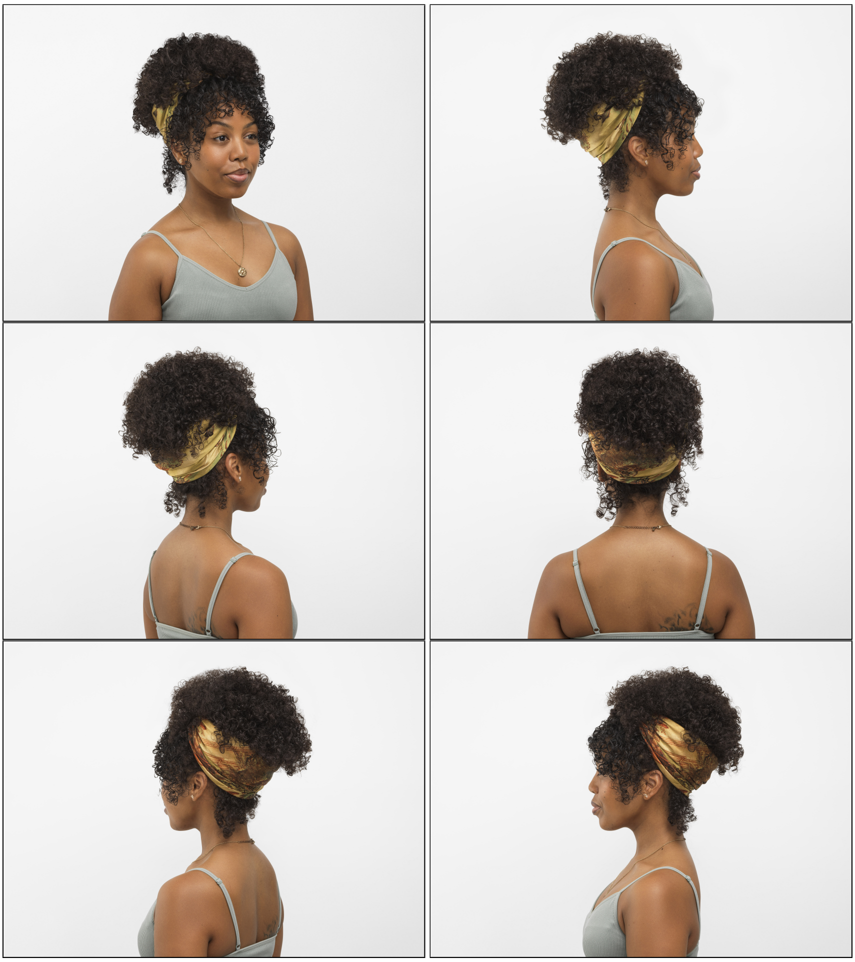


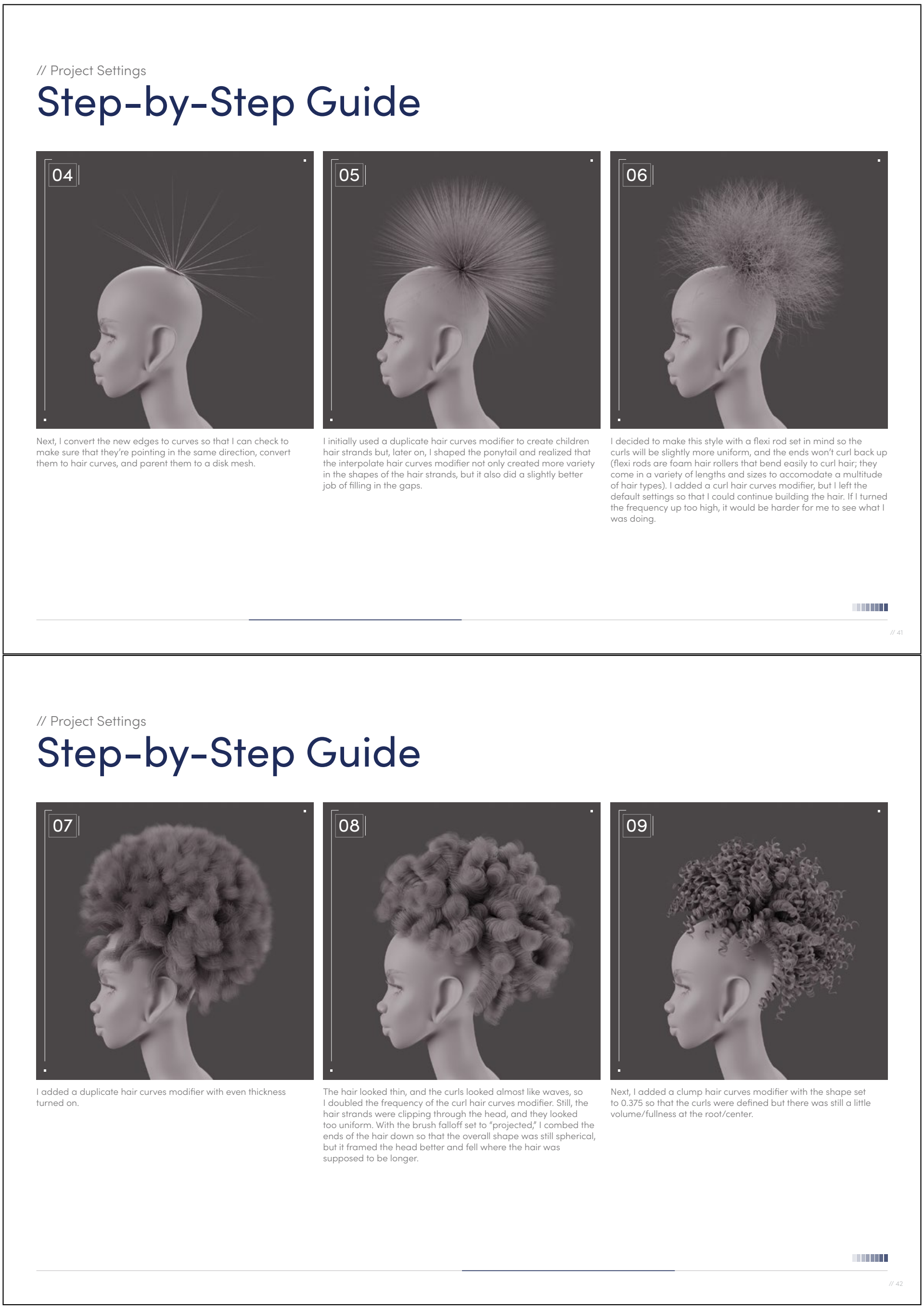
Fig. 3. Reference photos for *Natural Curls with Headwrap* example from *Code My Crown*.

freely available, and provided as part of the public release at https://github.com/dove-us/code-my-crown. In the course presentation, we will step through one of the examples provided in this guide, and also place it all in a wider context.

We will specifically cover the *Natural Curls with Headwrap* example from the *Code My Crown* guide, starting from reference photos (Figure 3), through step-by-step construction (Figure 4), and concluding with a final, downloadable 3D asset (Figure 5). The materials cover pages 32-44 in the guide. This is only one of 16 hair styles described by the guide, and is intended to serve as an introduction to the wider material. For completeness, we have included the entire *Code My Crown* guide at the end of this document.s

3 *LIFTED CURLS:* A MODEL FOR TIGHTLY COILED HAIR SIMULATION

We will cover the *Lifted Curls* hair simulation model [Shi et al. 2023], which received the *Best Paper* award at the *Symposium on Computer Animation* in 2023. This is the first hair simulation method in computer graphics that was specifically designed for the simulation of tightly coiled hair, and is



// Project Settings

Step-by-Step Guide

04

Next, I convert the new edges to curves so that I can check to make sure that they're pointing in the same direction, convert them to hair curves, and parent them to a disk mesh.

05

I initially used a duplicate hair curves modifier to create children hair strands but, later on, I shaped the ponytail and realized that the interpolate hair curves modifier not only created more variety in the shapes of the hair strands, but it also did a slightly better job of filling in the gaps.

06

I decided to make this style with a flexi rod set in mind so the curls will be slightly more uniform, and the ends won't curl back up (flexi rods are foam hair rollers that bend easily to curl hair; they come in a variety of lengths and sizes to accomodate a multitude of hair types). I added a curl hair curves modifier, but I left the default settings so that I could continue building the hair. If I turned the frequency up too high, it would be harder for me to see what I was doing.

// 41

// Project Settings

Step-by-Step Guide

07

I added a duplicate hair curves modifier with even thickness turned on.

08

The hair looked thin, and the curls looked almost like waves, so I doubled the frequency of the curl hair curves modifier. Still, the hair strands were clipping through the head, and they looked too uniform. With the brush falloff set to "projected," I combed the ends of the hair down so that the overall shape was still spherical, but it framed the head better and fell where the hair was supposed to be longer.

09

Next, I added a clump hair curves modifier with the shape set to 0.375 so that the curls were defined but there was still a little volume/fullness at the root/center.

// 42

Fig. 4. Step-by-step instructions from the *Natural Curls with Headwrap* example from *Code My Crown*.

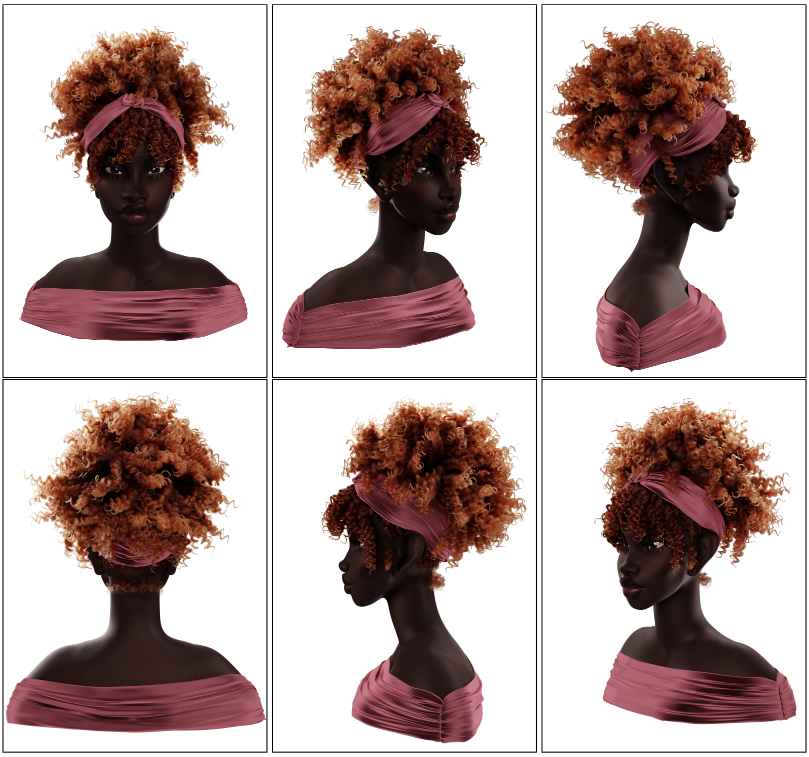


Fig. 5. Final 3D renders of *Natural Curls with Headwrap* example from *Code My Crown*.

capable of producing a variety of looks for curly hair that had not previously been demonstrated in the graphics literature (Figure 6).

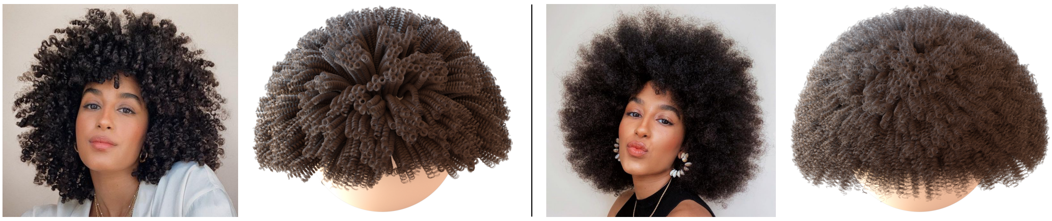


Fig. 6. **Left pair:** Comparison to real-world "clumped" look. **Right pair:** Comparison to real-world "picked out" look. Images are from Alexander [2023].

One of the main features of the technique is a new formulation for *torsion* energies. When the underlying strand being simulated can be assumed to be non-straight, then it allows a quadratic energy to be formulated that does not need the parallel transport operators of previous methods. Thus, a significant source of numerical complexity is sidestepped, resulting in a more robust simulation method.

In particular, we have found that the asymmetries that result in the Hessians of the conventional *Discrete Elastic Rods* (DER) [Bergou et al. 2008] model are not widely known, as it only appeared in the supplement to Panetta et al. [2019]. This means that the DER model is not quite hyperelastic, as it contains plasticity-like dependence on its parallel transport frame from the previous timestep. This complicates Hessian construction, as a symmetrizing approximation must be introduced, and solver convergence, as the matrix is no longer exact. Examining the non-straight case underscores research challenges in existing models.

For completeness, we have included the complete paper from Shi et al. [2023] at the end of this document.

REFERENCES

Leal Alexander. 2023. *Instagram.* https://www.instagram.com/curlygallal/

TF Banchoff. 1982. Global geometry of polygons. III. Frenet frames and theorems of Jacobi and Milnor for space polygons. *Rad Jugoslav. Akad. Znan. Umjet* 396 (1982), 101–108.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. *ACM Trans. Graph.* 27, 3 (2008), 1–12.

F. Bertails, B. Audoly, B. Querleux, F. Leroy, JL Lévêque, and MP Cani. 2005. Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics short papers*. Eurographics.

A.M. Darke, H.D. Harris, Nai'Vasha Grace, Isaac Olander, Malika Sajhmori Dunn, Diandra Rose, Adésayo Adéoyé, and Chrissy Powell. 2023. *Code My Crown: A guide to the creation and celebration of Black hair in video games.* Dove.

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. 2023. Sag-Free Initialization for Strand-Based Hybrid Hair Simulation. *ACM Trans. Graph.* 42, 4, Article 74 (jul 2023), 14 pages.

IGN. 2024. Why Are Game Devs Obsessed With THIS Hairstyle? https://youtu.be/mZfjhxYlg4E?feature=shared.

T. Kim, H. Rushmeier, J. Dorsey, D. Nowrouzezahrai, R. Syed, W. Jarosz, and A.M. Darke. 2022. Countering Racial Bias in Computer Graphics Research. In *ACM SIGGRAPH Talks* (Vancouver, BC, Canada).

Sofya Ogunseitan. 2022. Space Rangers with Cornrows: Methods for Modeling Braids and Curls in Pixar's Groom Pipeline. In *SIGGRAPH Talks* (Vancouver, BC, Canada). Association for Computing Machinery, New York, NY, USA, Article 49, 2 pages. https://doi.org/10.1145/3532836.3536277

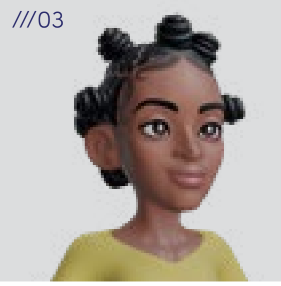
Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.* 38, 4 (2019), 1–15.

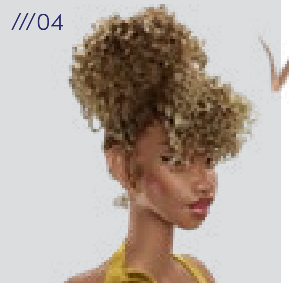
Deborah Patrick, Shaun Bangay, and Adele Lobb. 2004. Modelling and rendering techniques for african hairstyles. In *Proc. of Afrigraph.* 115–124.

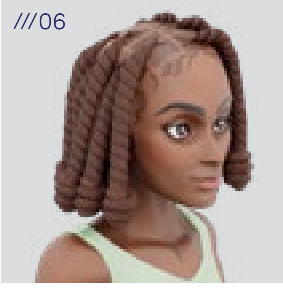
Alvin Shi, Haomiao Wu, Jarred Parr, A. M. Darke, and Theodore Kim. 2023. Lifted Curls: A Model for Tightly Coiled Hair Simulation. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 42 (aug 2023), 19 pages. https://doi.org/10.1145/3606920

B. Smith, F. De Goes, and T. Kim. 2018. Stable neo-hookean flesh simulation. *ACM Trans. Graph.* 37, 2 (2018), 1–15.

Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. 2023. Adaptive Shells for Efficient Neural Radiance Field Rendering. *ACM Trans. Graph.* 42, 6, Article 259 (2023), 15 pages. https://doi.org/10.1145/3618390







Code My Crown

A guide to the creation and celebration of Black hair in video games.

Navigation

> Black hair is as dynamic, alive, and ever-changing as Black culture itself.

A.M. Darke - Lead Guide Contributor and Founder of the Open Source Afro Hair Library.

Foreword

This is a guide to the creation and celebration of Black hair textures and protective styles in video games.

It features instructions, personal insights, and detailed references, created by people within the Black community, for anyone to use to create characters in the digital world.

No single document can capture the history and beauty of Black hair textures and protective styles. But through this body of work, our goal is to help create the change we want to see.

Introduction

Introduction
Introduction

At Dove, we've always stood for beauty that's unique, diverse, and inclusive. We want a world where beauty is a positive experience for everyone, including in virtual spaces.

Our experience of beauty is intrinsically linked with our freedom to express and represent ourselves as we choose. It is important to be the architects of our own sense of beauty, and that's why we've brought together some of the most visionary Black 3D artists to share their mastery in crafting virtual Black hair textures and protective styles.

// Nearly 20 years ago, we pushed back against stereotypes perpetuated by the beauty industry to feature the bodies of real women in global advertising campaigns.

// We launched the Dove Self-Esteem Project, which promises to support a quarter billion young people by 2030 by using an evidence-based curriculum to build self-esteem and body confidence.

// We co-founded the CROWN Coalition to help advocate for the passage of legislation (The CROWN Act) to prohibit race-based hair discrimination based on hair texture and protective styles.

// We launched the Real Virtual Beauty Initiative, a commitment to educate and empower game developers to create more diverse and inclusive representation of girls and women in games.

// Introduction

Introducing Code My Crown, the world's first comprehensive guide to the creation and celebration of Black hair textures and protective styles in video games. Millions of Black people play video games for fun, a sense of community and a unique platform for self-expression.

Yet for all the technological advances in the gaming industry, the depiction of Black hair textures and protective styles continues to be limited and inaccurate. It is often highly stereotypical, unrealistic and does not faithfully represent the beauty or cultural nuance of its real-world inspiration.

> In the real world, there is an incredible variety of Black hairstyles. But this is rarely reflected in the gaming world. When Black hair is absent from the games we play or is consistently low-quality, it communicates that Black players and our culture are an afterthought, that our stories aren't worth telling. How else can we explain the ubiquity of matted Cornrows, bald patches instead of parts, giant disco Afros, and messy, unstyled Locs? Why is a common Fade or Twist Out rarely an option?

// A.M. Darke - Lead Guide Contributor
and Founder of the Open Source Afro Hair Library.

// Introduction

New research shows 85% of Black gamers believe Black hair textures are poorly represented in video games. And approximately 8 in 10 Black gamers feel excluded, discriminated against, or not good enough when they don't see themselves represented accurately*.

It's not just the players who feel this way. Nearly 9 in 10 developers agree there should be more resources available to enable them to code Black hair textures that realistically represent real-world styles.*

85%

of Black gamers believe natural hair is poorly represented in video games.

Nearly

9 in 10

developers agree there should be more resources available to enable them to code natural hair that realistically represents real-world styles.*

Code My Crown was created to help fight this injustice.

*Source: Online survey conducted by Edelman DXI, a global, multidisciplinary research, analytics, and data consultancy, in the USA and the UK in March/April 2023. Surveyed 1,002 adult gamers (aged 18+, 502 in the USA and 500 in the UK), 300 Black gamers (men and women aged 18-55 who self-identify as Black gamers playing video games, 175 in the USA and 125 in the UK) and 160 video game developers (men and women aged 18-55 who work as developers in the gaming industry, 80 in each of the USA and the UK).

> In the Black community, our hairstyles are reflections of our history and culture, showcasing our creativity, innovation, and resilience. Like the games we play, Black hair tells a story.

A.M. Darke - Lead Guide Contributor and Founder of the Open Source Afro Hair Library.

// Introduction

Created in partnership with Black artists, academics, and stylists, Code My Crown exists to make sure Black hair textures and protective styles are recreated digitally with the care and respect they deserve. This guide gives developers and designers the tools needed to better represent real beauty in the digital world.

Dove

open source afro hair library

A.M. Darke, Lead Guide Contributor and Founder of the Open Source Afro Hair Library, further explains:

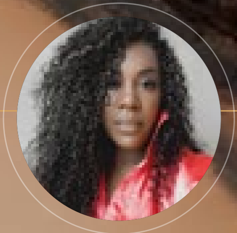
"Open Source Afro Hair Library (OSAHL) and Dove have partnered to create Code My Crown, a guide created by Black artists, with images and step-by-step instructions for creating Black hairstyles in 3D. OSAHL's contributing team includes 3D artists from around the world and across the Black diaspora, bringing together the technical and cultural knowledge necessary to create authentic Black hair textures for use in games and virtual media. Many OSAHL artists are self-taught, knowing first-hand how difficult it is to find resources and reference material for learning how to model Black hair textures and styles. In the absence of any formal materials covering the intricacies of Black hair textures, the examples developed for Code My Crown are a result of trial, error and personal innovation."

Clearly, there is so much more to be done to ensure Black gamers see themselves in the games they play. And that's exactly why we created Code My Crown. We have partnered with Black game developers to create a resource that will help others in the industry to create more diverse, true-to-life depictions of Black hair in their games.

> Too often, as a game developer, I have created designs that authentically represent Black culture, only to be told that they don't align to the studio's vision or that they're "too cliché." Again and again, I have been tasked to strive for authenticity but when presented with true authentic representation, non-Black leaders have shot me down just because they personally don't resonate with what I've created. This only allows for a very narrow definition of representation. A big first step to truly create change is to cultivate trust between studios, Black game developers and the Black community at large.
>
> // Chrissy Powell, 2D/3D artist

While this guide isn't a definitive solution, it can help set a new standard for the importance of representation of Black hair textures and protective styles in video games. It serves as a powerful example of the change you can make when you let people lead in the representation of their own cultures and identity.

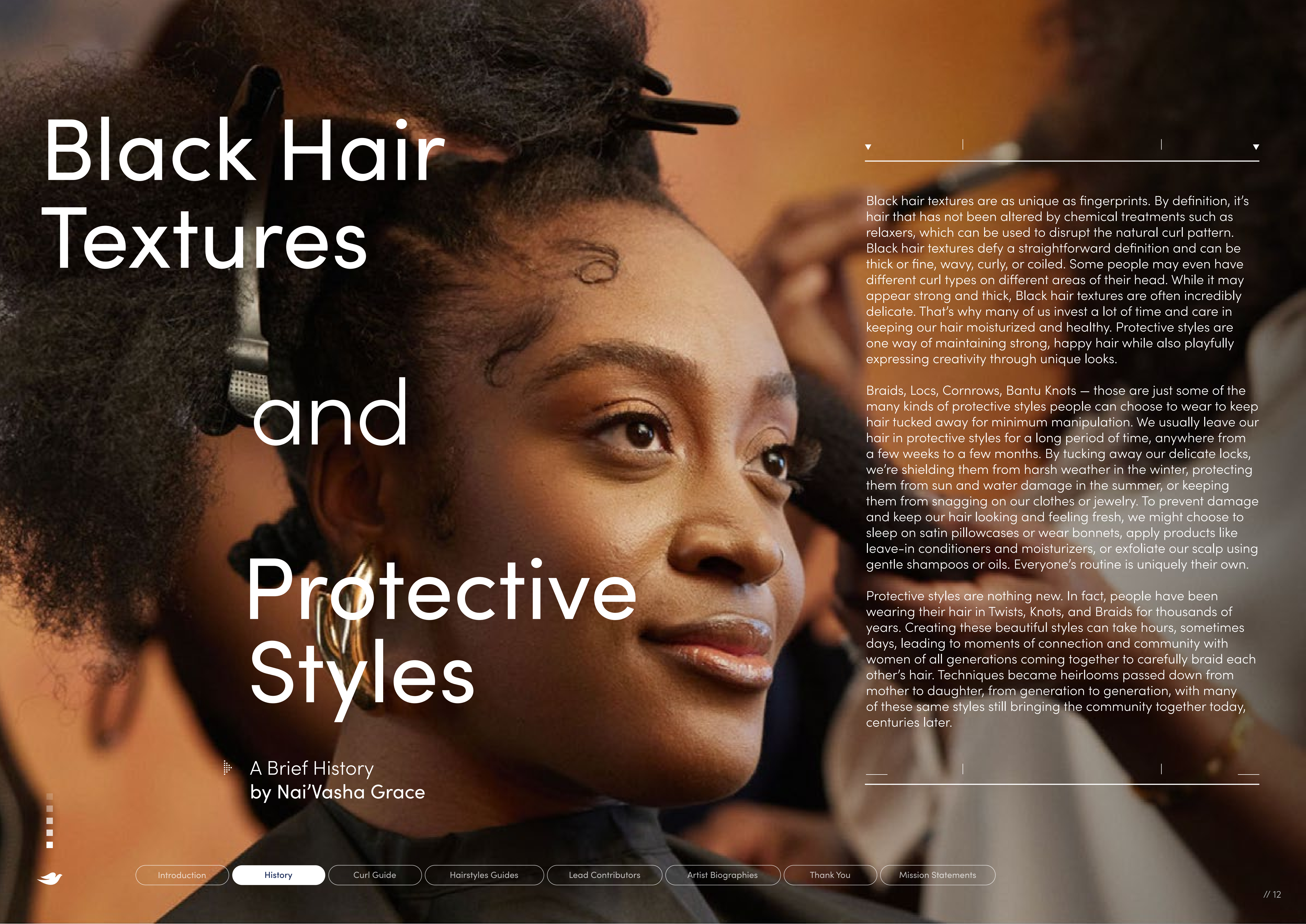
> When we see ourselves in the media we consume, we feel seen and celebrated. Everyone should be able to create an avatar that really looks like them in the games they play. We all deserve our chance to be heroes in the fictional worlds we create.



// Nai'Vasha Grace - Celebrity Stylist

A source of inspiration and best practices for approaching your own designs.

A.M. Darke - Lead Guide Contributor and Founder of the Open Source Afro Hair Library.



Black Hair Textures and Protective Styles

A Brief History
by Nai'Vasha Grace

Black hair textures are as unique as fingerprints. By definition, it's hair that has not been altered by chemical treatments such as relaxers, which can be used to disrupt the natural curl pattern. Black hair textures defy a straightforward definition and can be thick or fine, wavy, curly, or coiled. Some people may even have different curl types on different areas of their head. While it may appear strong and thick, Black hair textures are often incredibly delicate. That's why many of us invest a lot of time and care in keeping our hair moisturized and healthy. Protective styles are one way of maintaining strong, happy hair while also playfully expressing creativity through unique looks.

Braids, Locs, Cornrows, Bantu Knots — those are just some of the many kinds of protective styles people can choose to wear to keep hair tucked away for minimum manipulation. We usually leave our hair in protective styles for a long period of time, anywhere from a few weeks to a few months. By tucking away our delicate locks, we're shielding them from harsh weather in the winter, protecting them from sun and water damage in the summer, or keeping them from snagging on our clothes or jewelry. To prevent damage and keep our hair looking and feeling fresh, we might choose to sleep on satin pillowcases or wear bonnets, apply products like leave-in conditioners and moisturizers, or exfoliate our scalp using gentle shampoos or oils. Everyone's routine is uniquely their own.

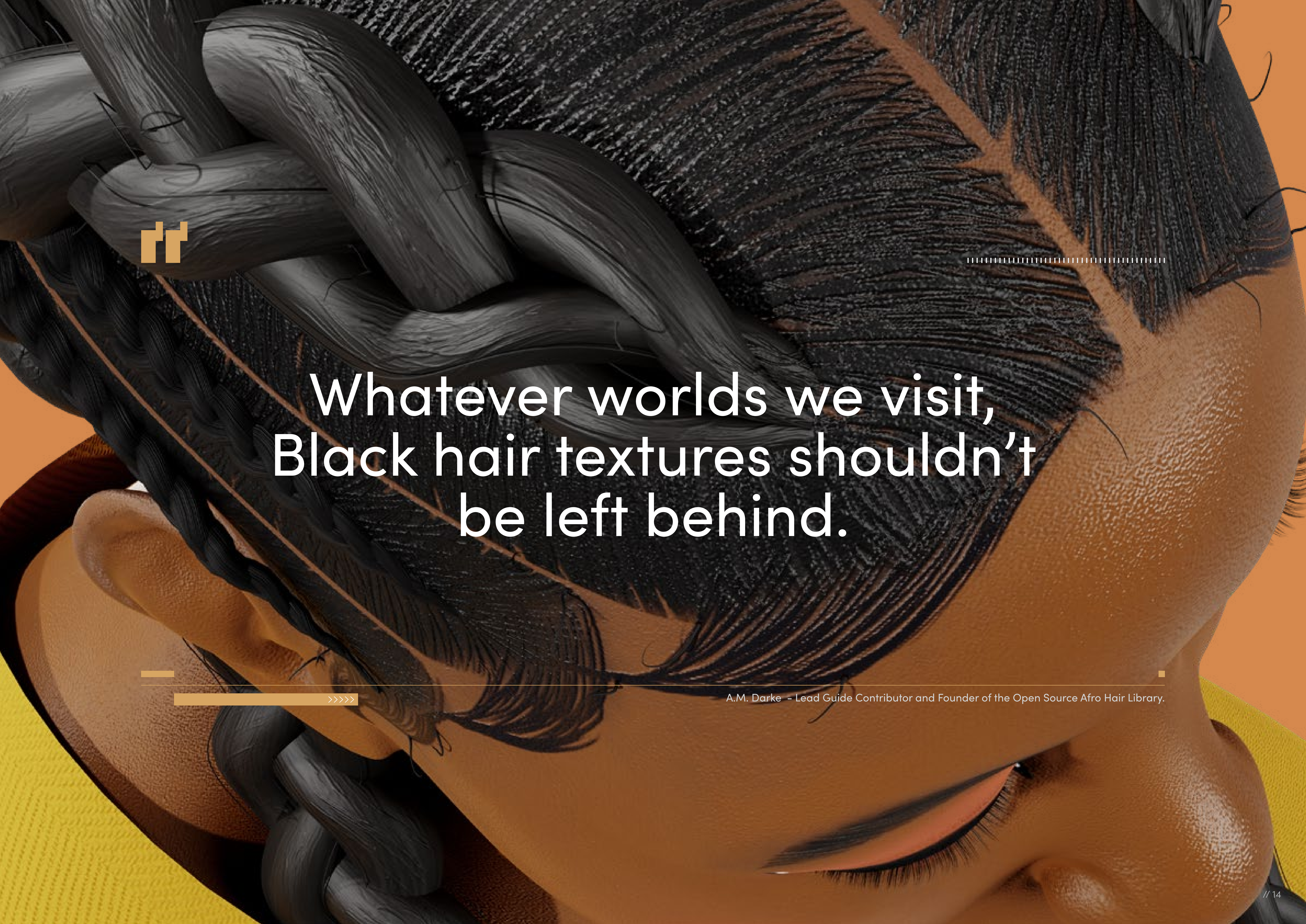
Protective styles are nothing new. In fact, people have been wearing their hair in Twists, Knots, and Braids for thousands of years. Creating these beautiful styles can take hours, sometimes days, leading to moments of connection and community with women of all generations coming together to carefully braid each other's hair. Techniques became heirlooms passed down from mother to daughter, from generation to generation, with many of these same styles still bringing the community together today, centuries later.

Black hair textures and protective styles have remained a symbol of Black identity, but throughout history, some would attempt to strip us of this identity. In the 15th century, Black women were made to cover their hair with headdresses or to emulate European styles, to fit a standard that told them their natural hair wasn't beautiful. Styles like the Afro would become symbolic of African Americans fighting to change the course of history during liberation movements in the United States in the '60s and '70s. Wearing hair natural or in protective styles became a protest in itself, a brave rejection of Eurocentric beauty standards.

Eventually, natural hair products for Black hair textures began appearing on local supermarket shelves, we no longer had to trek to specialist stores to find the right shampoos and conditioners to care for our hair. Still, echoes of those Eurocentric beliefs — that Black hair textures are less beautiful, less tidy, less professional — endure to this day. Around the world, schoolchildren can still be punished for wearing their natural hair, adults going for job interviews still worry if their Afro will cost them the role, and female athletes are forced to take out the beads in their braids in order to compete. Reclaiming our natural hair has become a process of relearning, with many of us only now beginning to untangle those beliefs and reconnect with our natural hair.

Transitioning and reconnecting with our natural hair can take time. Some women will choose to do the "big chop," cutting off hair that has been permanently straightened by chemicals so they can regrow their natural tresses. Some will begin exploring different protective styles for the first time, learning how to express their personal style through complex, beautiful looks. How people choose to style their hair is personal; what matters is ensuring that every person's style has a chance to be celebrated.

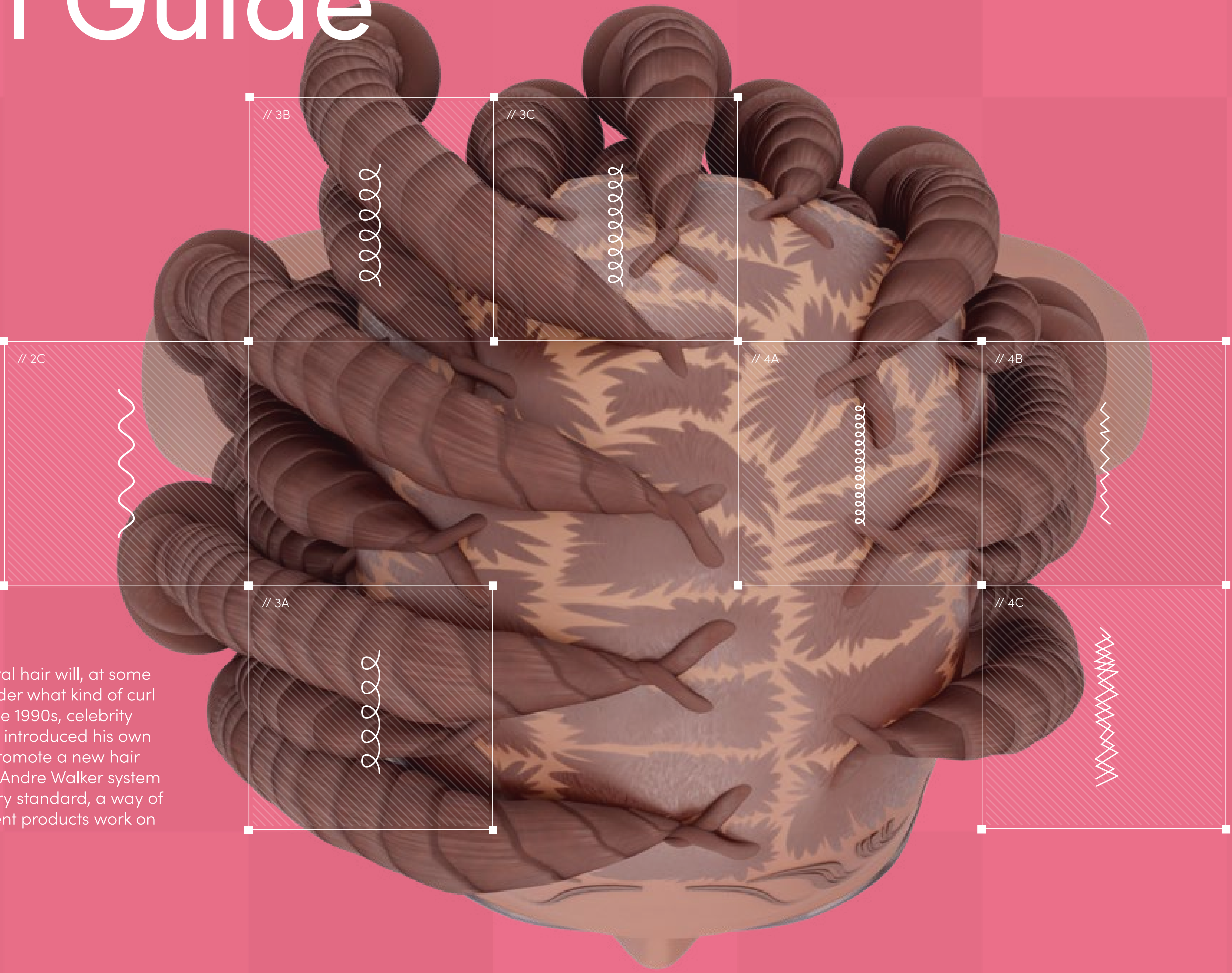
How people choose to style their hair is personal; what matters is ensuring everyone has the freedom to make that choice.



> Whatever worlds we visit, Black hair textures shouldn’t be left behind.

A.M. Darke - Lead Guide Contributor and Founder of the Open Source Afro Hair Library.

Curl Guide



Most people with natural hair will, at some point in their lives, wonder what kind of curl pattern they have. In the 1990s, celebrity hairstylist Andre Walker introduced his own hair typing system to promote a new hair product collection. The Andre Walker system has become the industry standard, a way of articulating how different products work on different hair types.

by Nai'Vasha Grace

Curl Guide

The system is broken down into numbers that represent whether hair is straight, wavy, curly, or coily, and the letters A, B, C, which represent how curly, coarse, or fine the hair is, with A being the least curly and C being the curliest. This system can help people choose the right products for their hair or the best protective styles to wear. In the era of social media, it can be a useful tool for having more productive conversations about Black hair textures as people may follow certain social media influencers who have the same hair type as them, so they can get more relevant haircare advice.

The Andre Walker system isn’t perfect. People rarely fit neatly into categories and this is especially true for those of us with natural hair. No two heads are the same and our hair type can even change over time. Some people transition between relaxed and natural hair, or their pattern may be altered after bleaching or dyeing. Age, weather, and the plethora of styling options can result in someone having different hair types at different points in their hair journey.

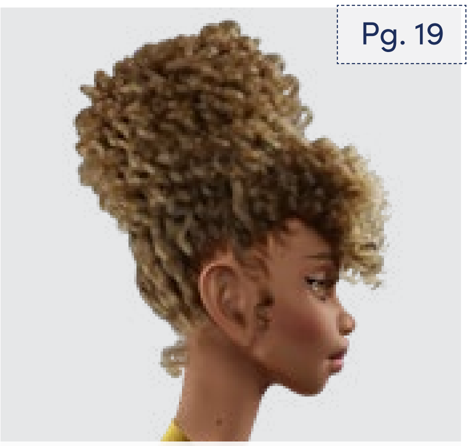
Hair type charts can be a powerful framework for accurately recreating Black hair textures digitally, but any system is best approached as a starting point. By referencing the Andre Walker system, creators can gain a better understanding of the nuances of Black hair textures, examine different types of curls, coils, and kinks, and ensure the breadth of this diversity is faithfully represented in the characters they create.

Andre Walker Hair Typing System Breakdown:		
Type 1 Straight hair	1A – thin and fine hair 1B – medium thickness hair 1C – thick hair	↓
Type 2 Wavy hair	2A – fine, loose S-shaped waves 2B – defined S-shaped waves 2C – thick, defined S-shaped waves	↓
Type 3 Curly hair	3A – large S-shaped loose curls 3B – tight, springy curls 3C – densely packed corkscrew curls	↓
Type 4 Coily hair	4A – densely packed S-shaped coils 4B – tight Z-shaped coils 4C – very tight zigzag coils	↓

> Most people have more than one texture on their head, so any system meant to categorize human features is best used as a guide, not a rule.

A.M. Darke - Lead Guide Contributor and Founder of the Open Source Afro Hair Library.

Select a character


Pg. 19

Curly Bangs

Pg. 32

Natural Curls w/ Headwrap

Pg. 45

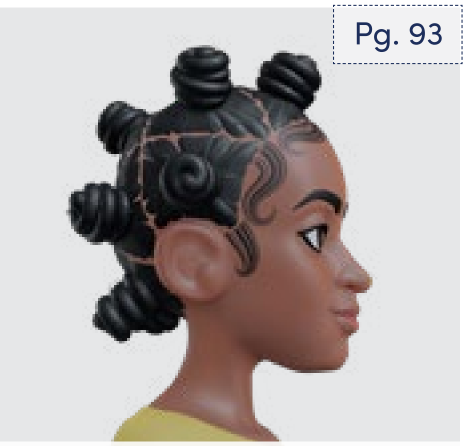
Twist Out

Pg. 63

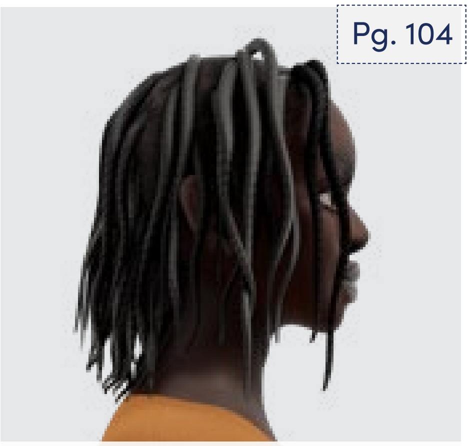
Natural Locs

Pg. 81

Loc'd Bob


Pg. 93

Bantu Knots


Pg. 104

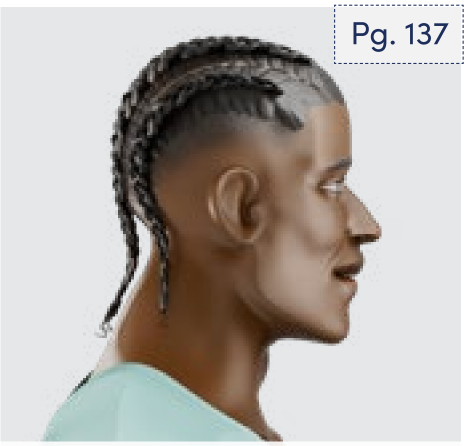
Twists

Pg. 115

Half Up Half Down

Pg. 125

Braided Bob


Pg. 137

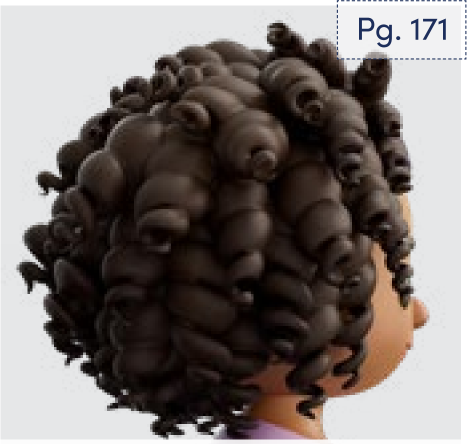
Stitch Cornrows w/ Fade

Pg. 148

Cornrows w/ Accent Braids

Pg. 160

Braids w/ Fade


Pg. 171

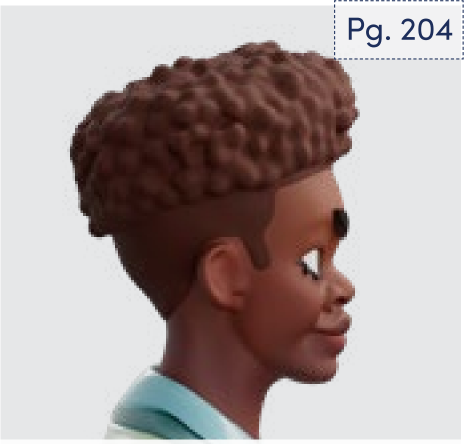
Curly Bob

Pg. 184

Low Ponytail w/ Swoop Bang

Pg. 192

Afro Puff w/ Split Bangs


Pg. 204

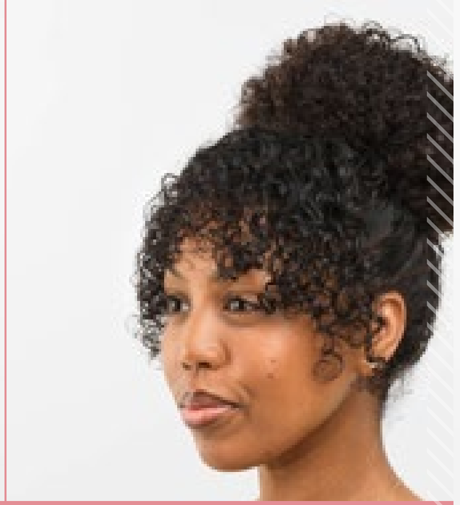
Afro w/ Fade

>>>

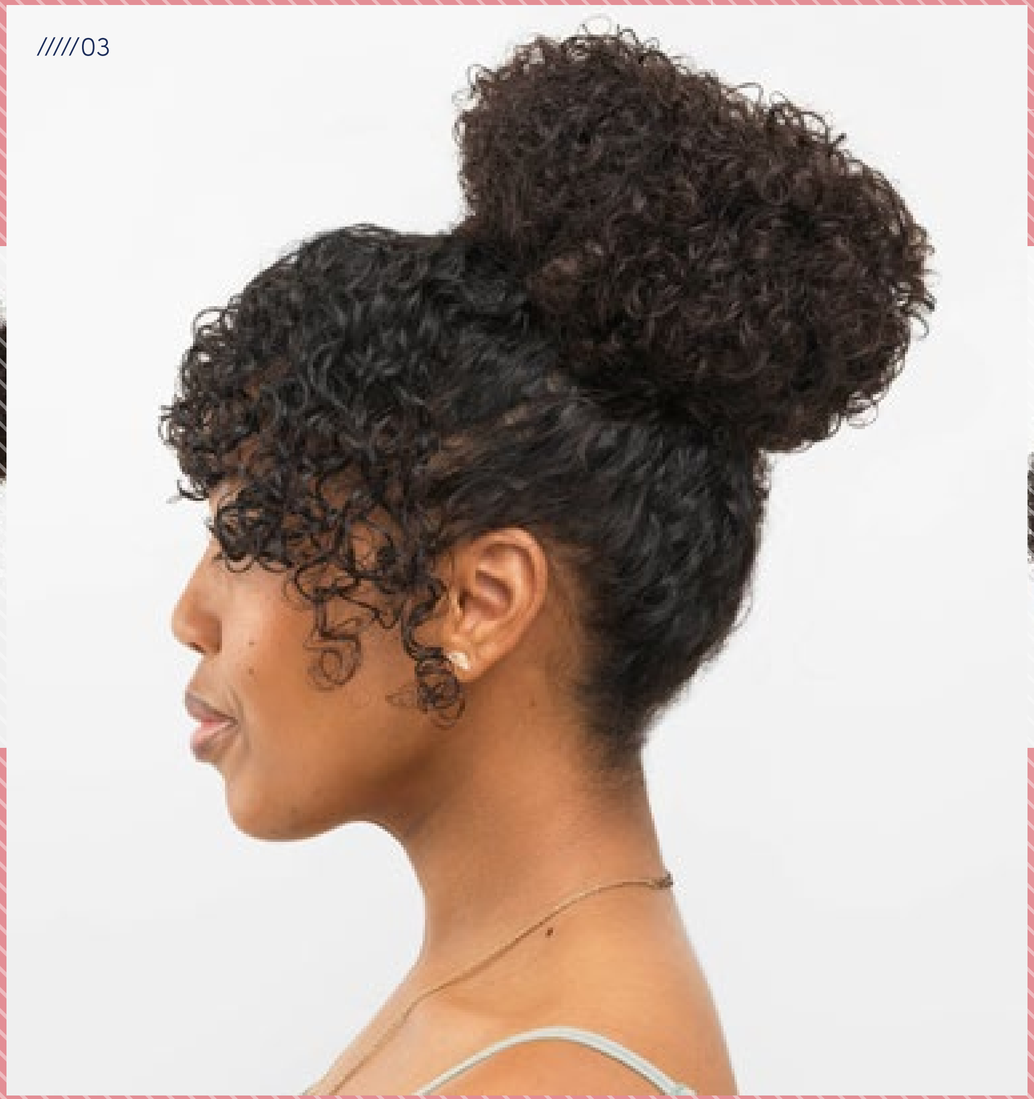
Introduction | History | Curl Guide | Hairstyles Guides | Lead Contributors | Artist Biographies | Thank You | Mission Statements

i **Click on** the character to see more details.

Curly
Curly
Curly



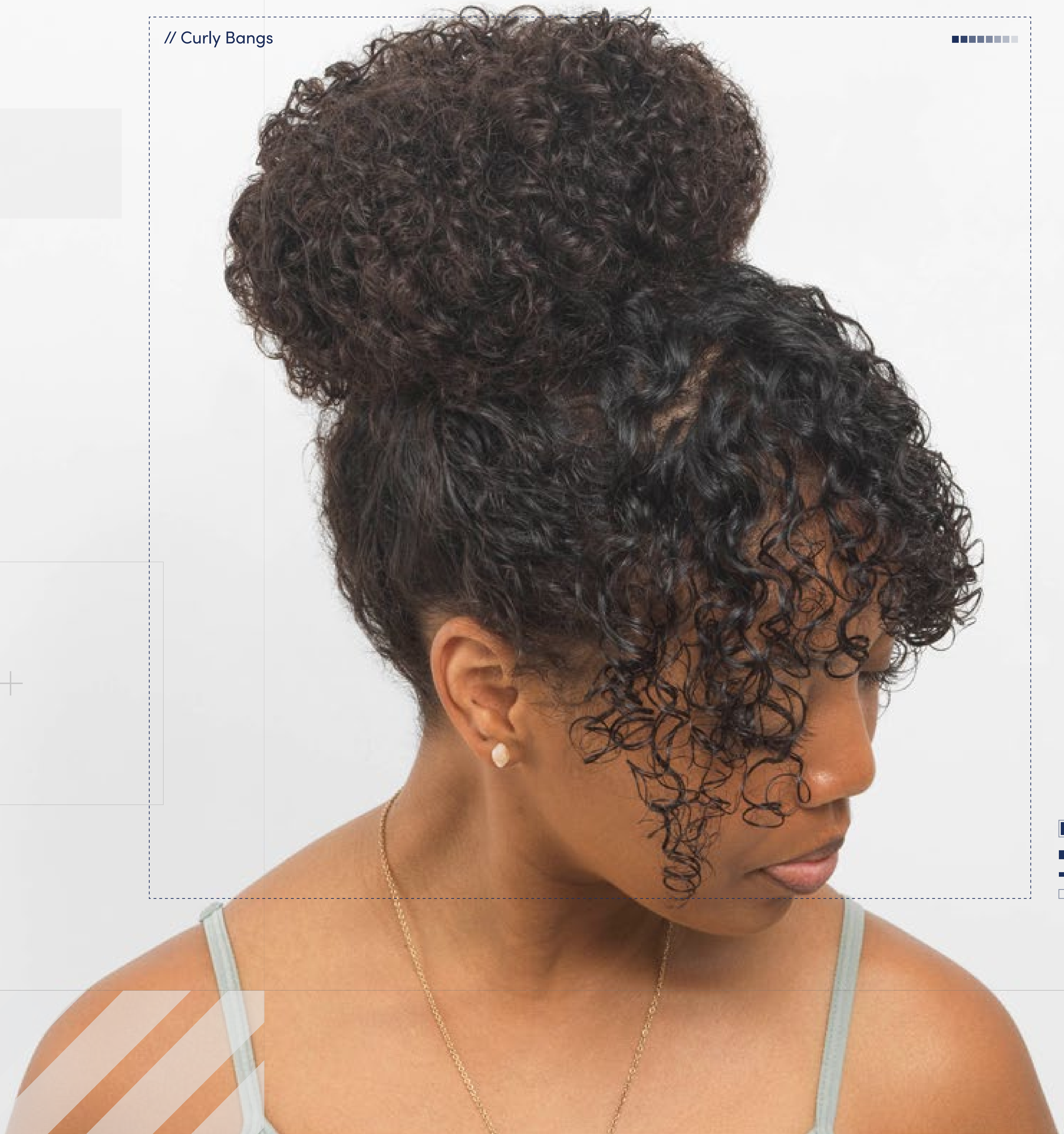
/////03



Bangs

Download Images

// Curly Bangs

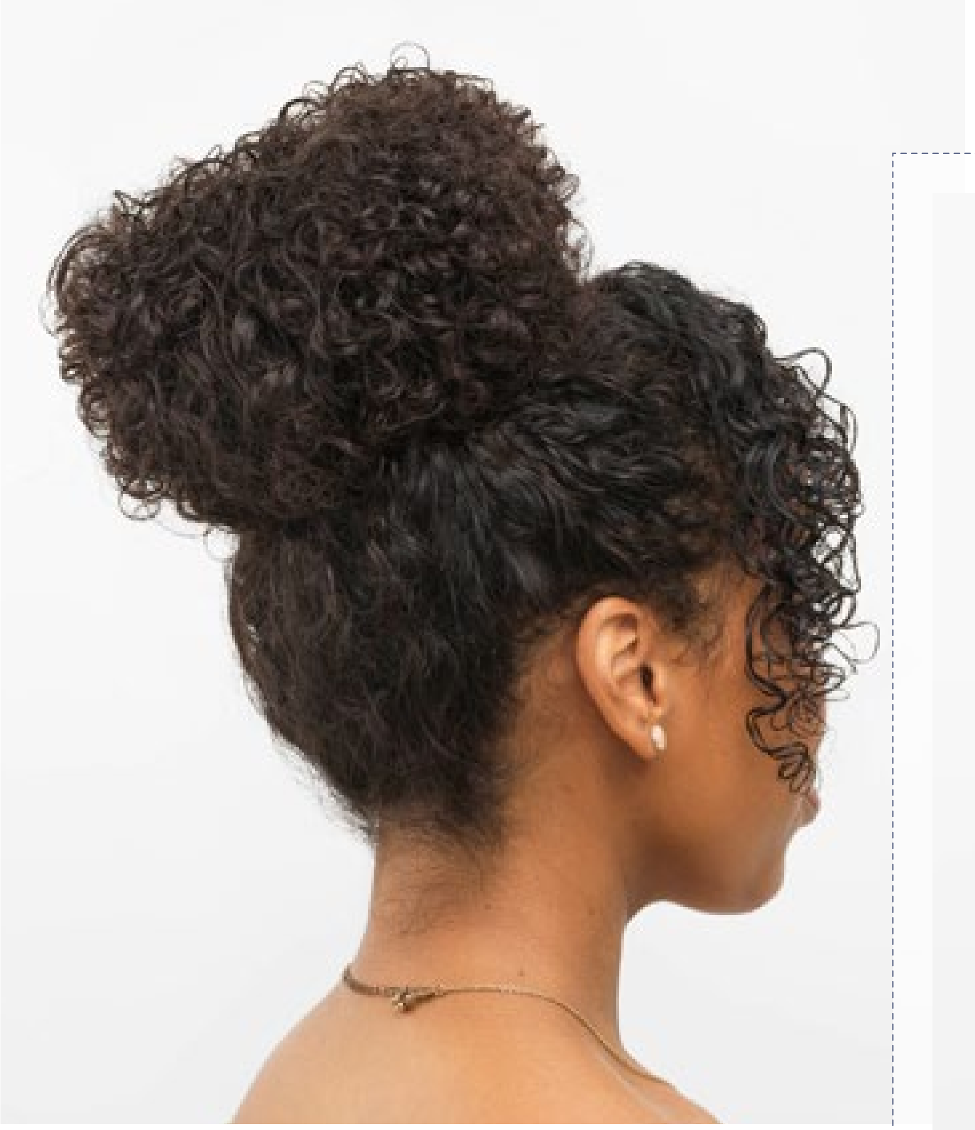


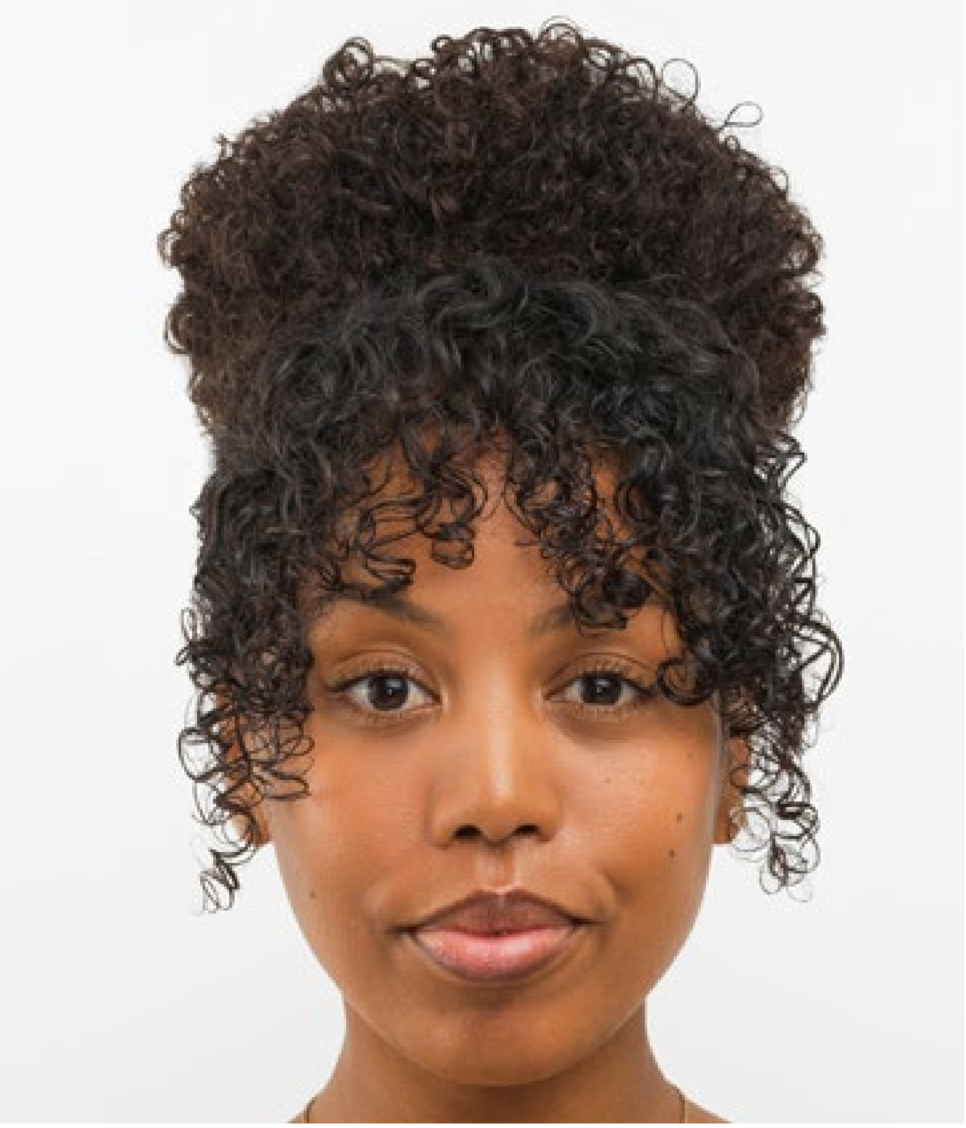
// Hairstyle

Curly Bangs

Style Background: Bangs can transform your entire look. People often associate full bangs with straight hair, but in recent years, curly bangs have become a hot trend, with several celebrities wearing their curly or wavy hair with full bangs.

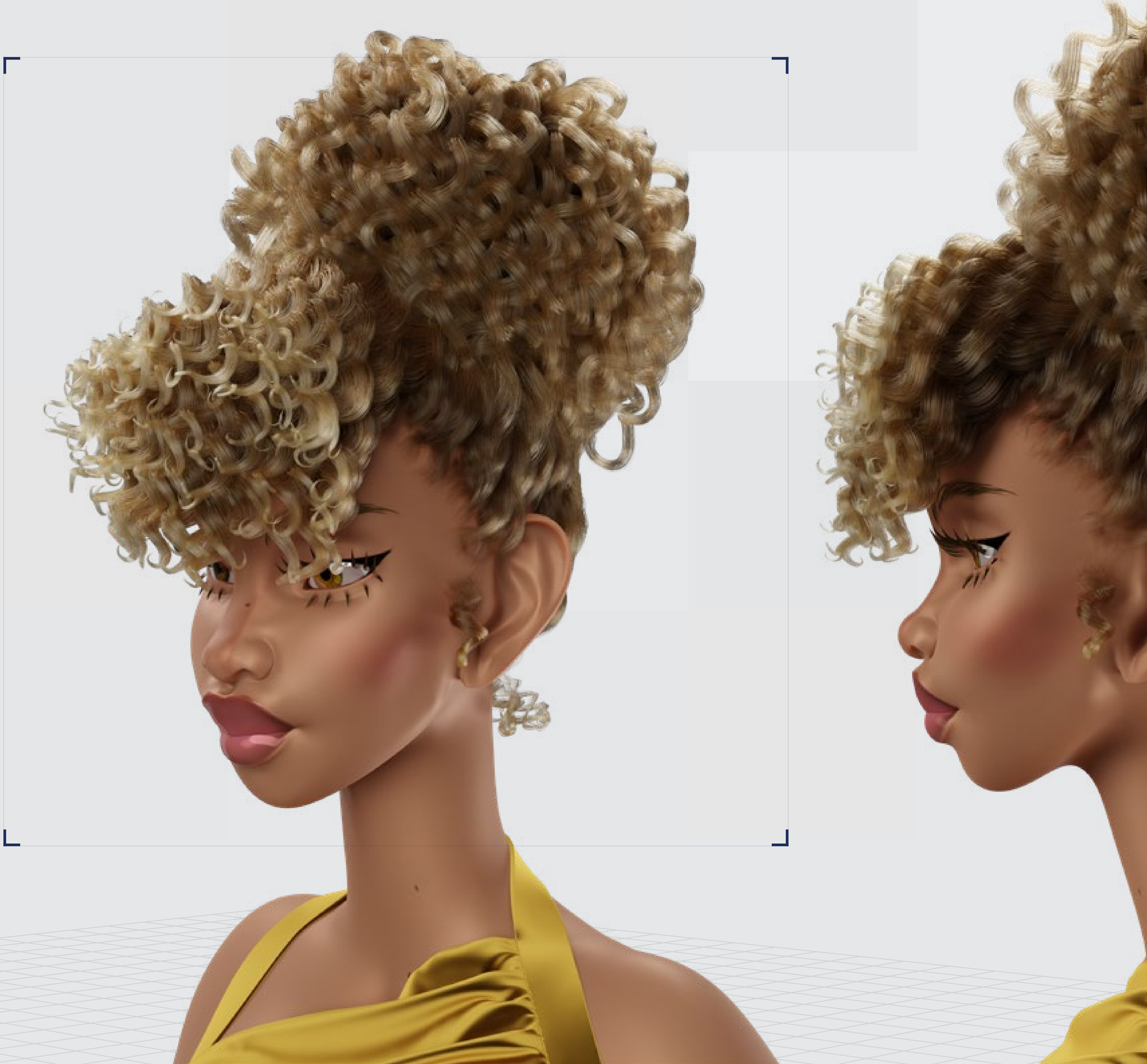
Download Images





3D Design

Curly Bangs



Digital Artist:
Diandra Rose

// Curly Bangs

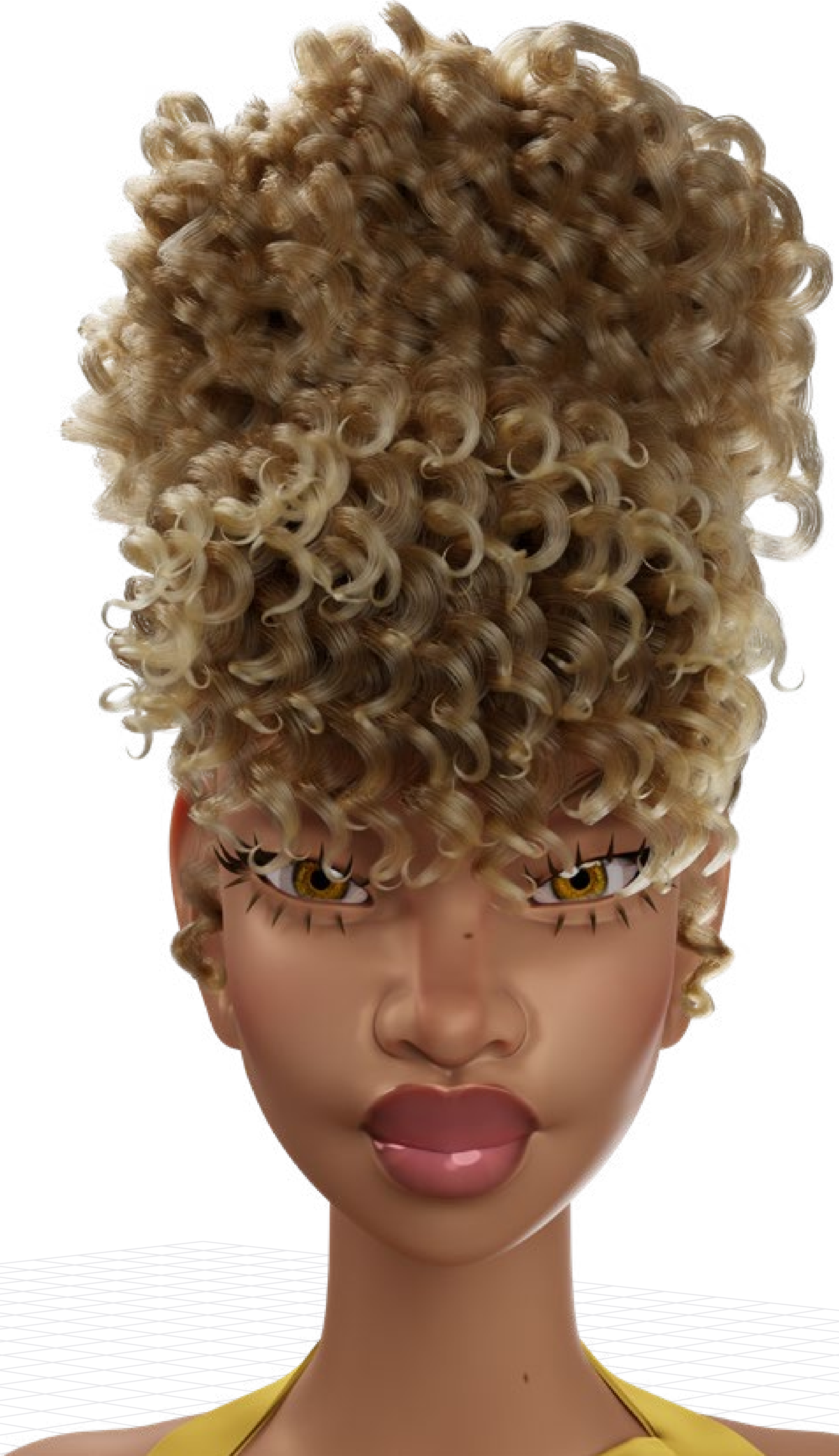
3D Design

Curl Type Modeled: **3A/3B**

// 01

// 02

Download 3D Render



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

Project

// Curly Bangs

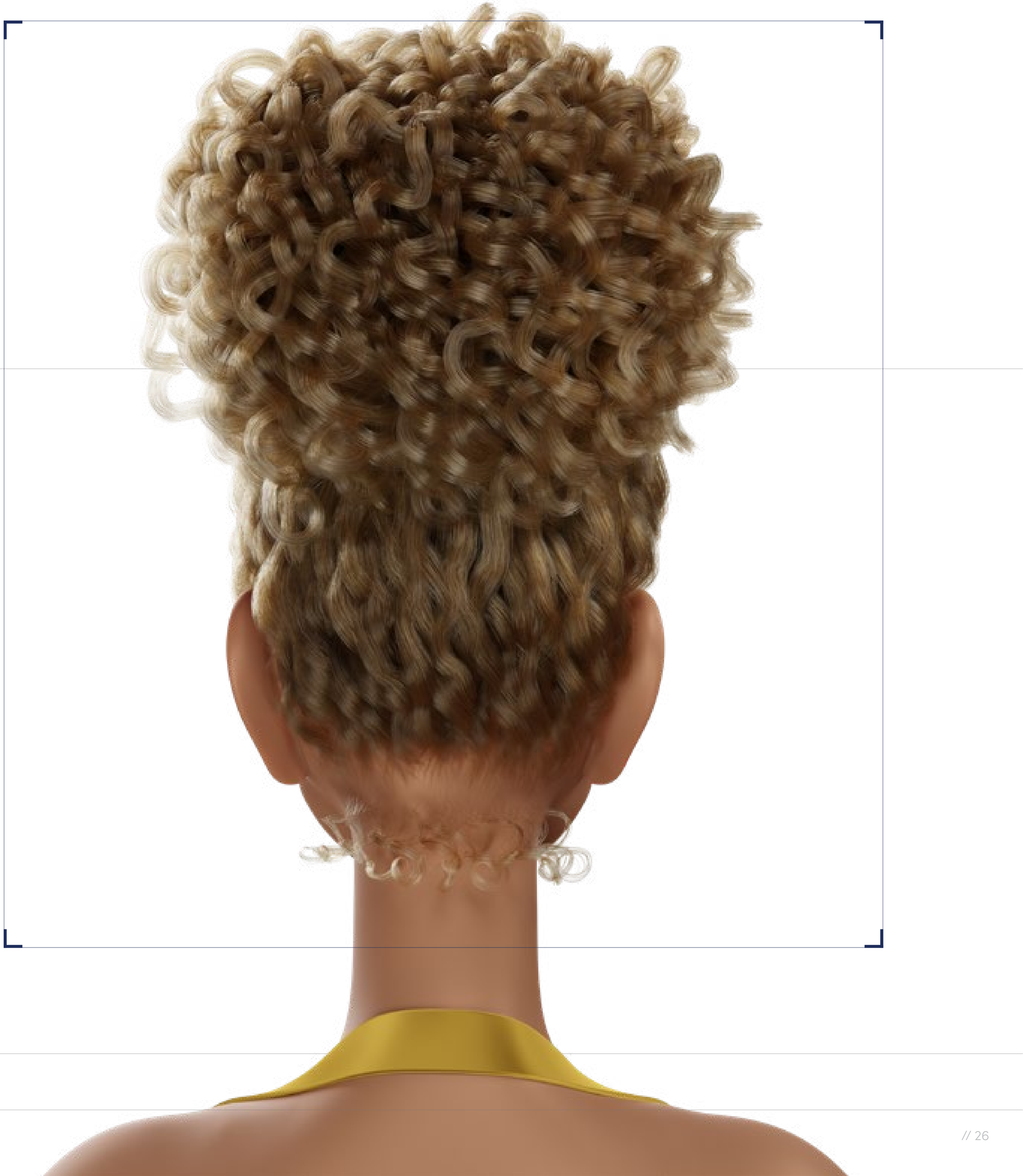
Settings

// Project Settings

Program Guide

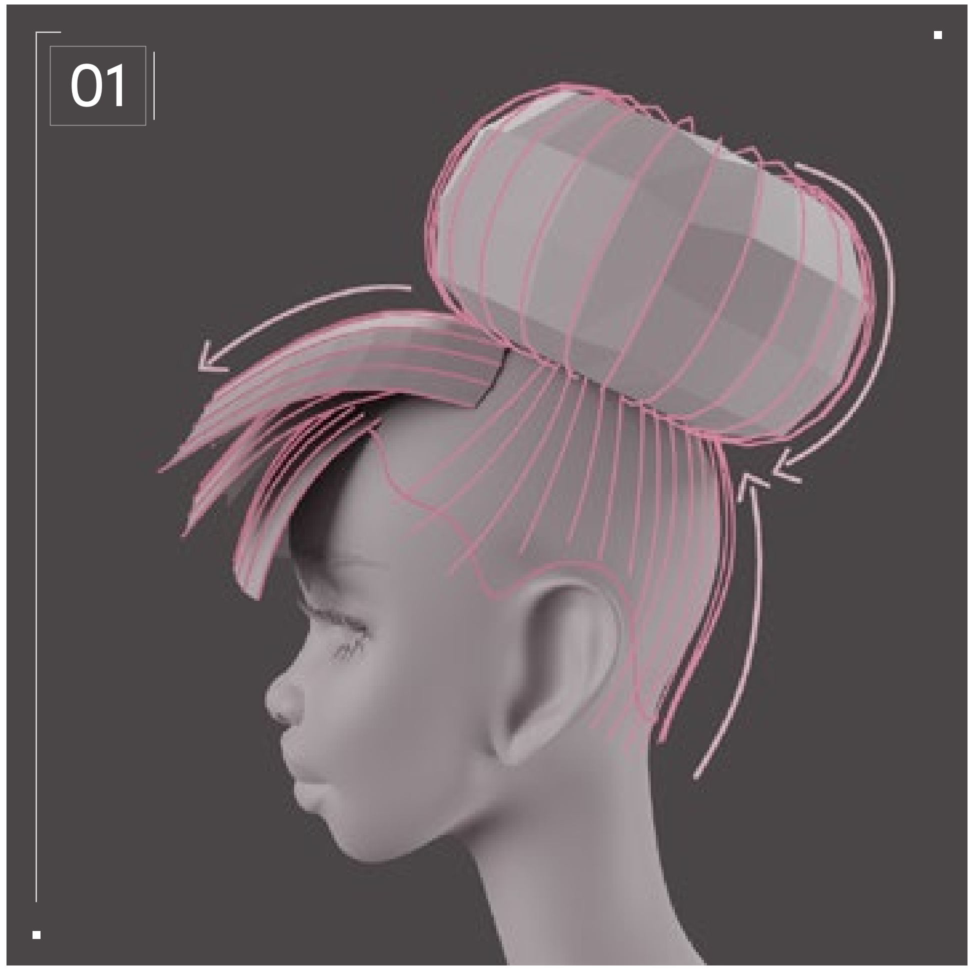
Software: Geometry node hair system in Blender 3.5

I didn't have to "break" anything to create this style, but if I wanted to make it "game-ready" at any fidelity, it would be difficult to achieve in its current form because many game engines aren't equipped to handle large quantities of hair particles. I could reduce the number of baby hairs and convert them to flat ribbons instead of rounded ribbons/tubes, but unfortunately there are still limitations with current game engine capabilities. I could remove most of the baby hairs, convert the main hair strands to hair cards, and convert the remaining baby hairs to smaller hair cards for stray hairs. The issue with this method is that there's a lot of overlap/clipping. You might be able to get away with that for dark hair colors, but it would be much more noticeable with lighter colors.



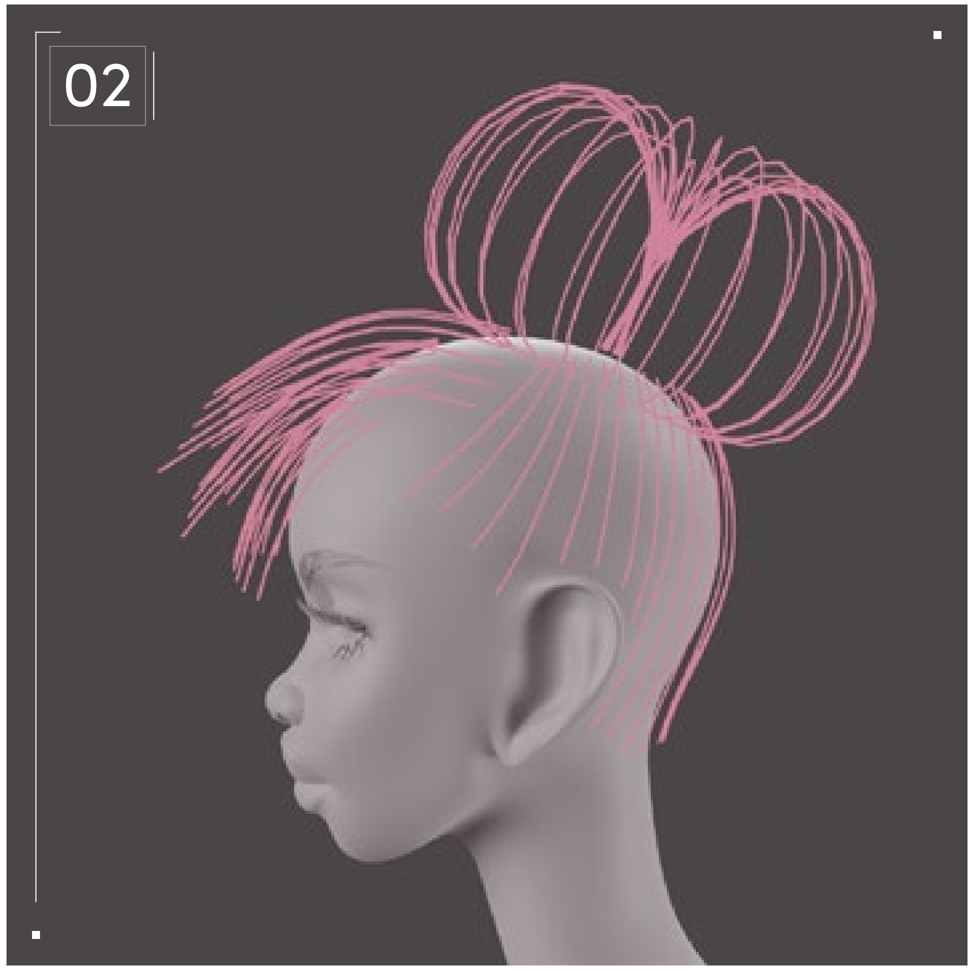
// Project Settings

Step-by-Step Guide

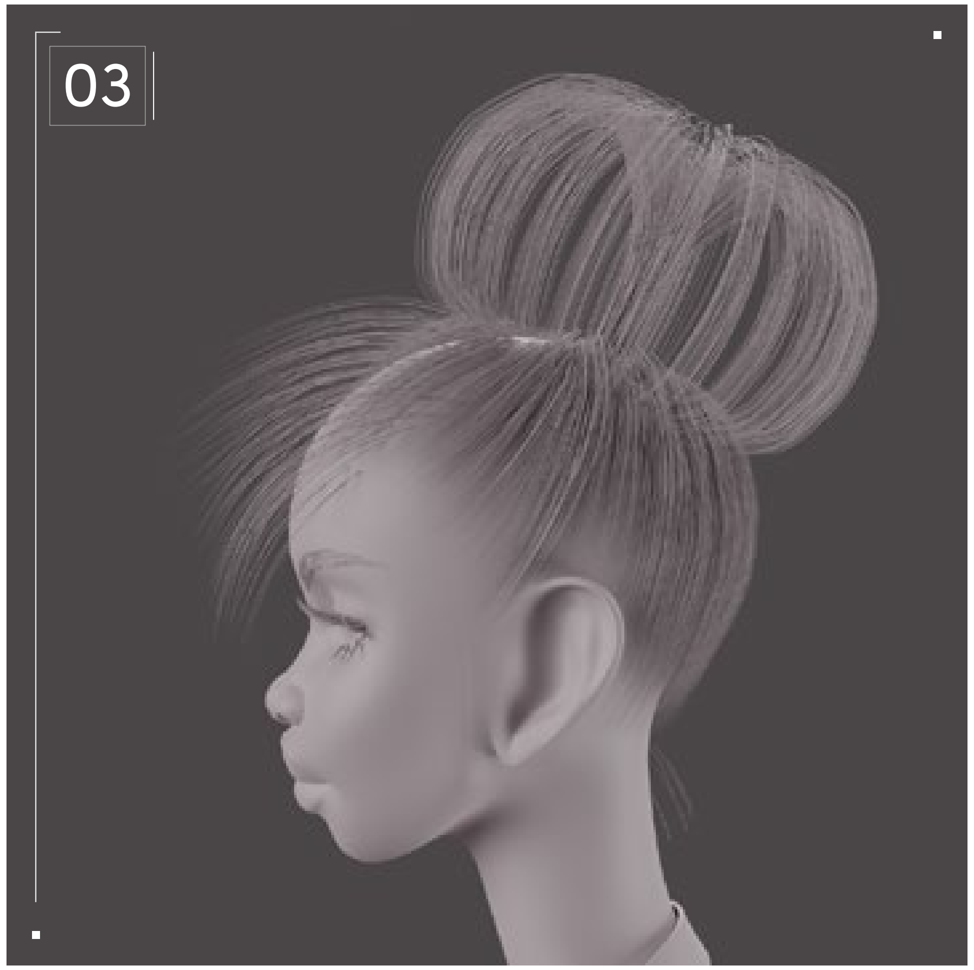


I made my style with a faux bang in mind. This style is created by putting the hair in a bun with the ends pulled out in the front to create the illusion of a bang.

I start by blocking out the basic shapes of the style. For sections like the bun or the bangs, I use mesh objects. The edges/lines of the mesh have to follow the flow of the hairstyle that I'm creating. For sections like the root of the hair, or anything that's meant to stay close to the scalp, I use the grease pencil to draw the strokes. The grease pencil is easy to modify and the shrink-wrap modifier has features that aren't normally available.

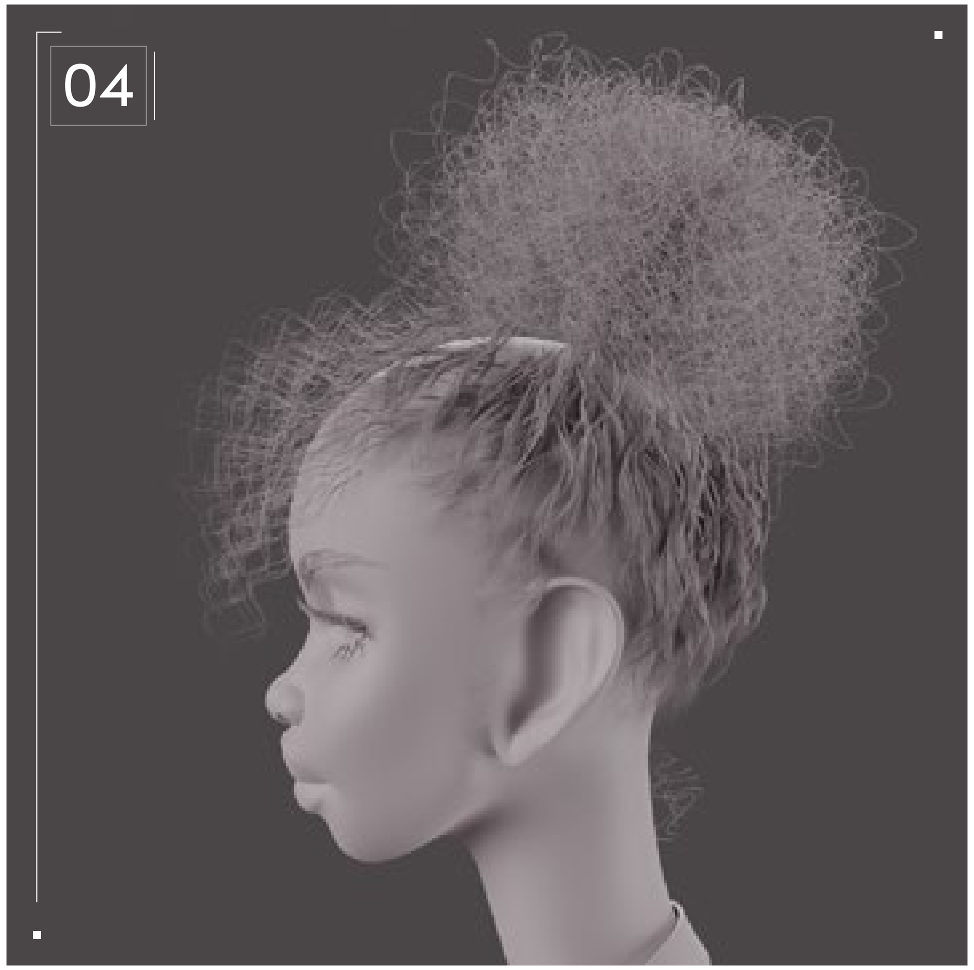


The faces are then deleted, and the remaining edges and grease pencil layers are converted to curves so that I can make sure the direction of the roots/ends are consistent for all the hairs. After that, they can be converted to hair curves. The bangs and roots are parented to an (invisible) scalp mesh and the bun is parented to a disk mesh.

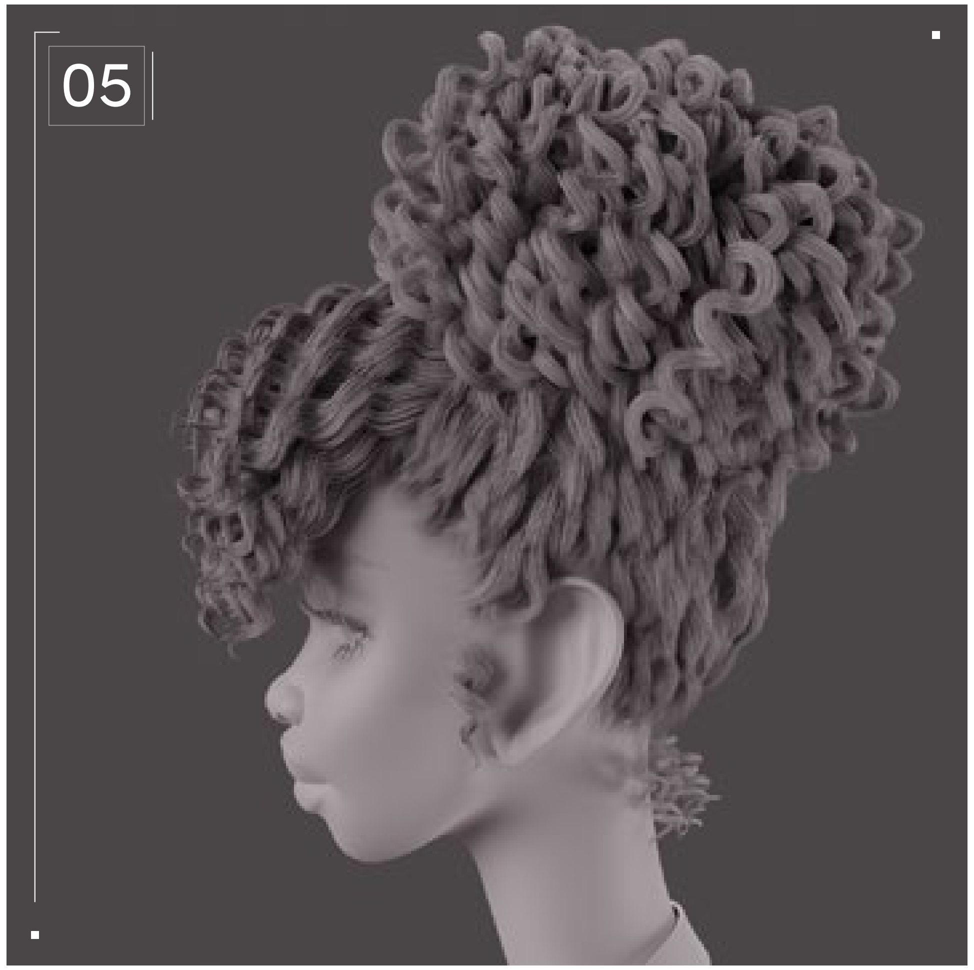


While there is an interpolate hair curves modifier, for this style I prefer to create more guide hairs and use the duplicate hair curves modifier instead. The purpose of this is to add volume to the hair so I applied a larger radius. This is similar to using the simple children hair option in the old particle hair system.

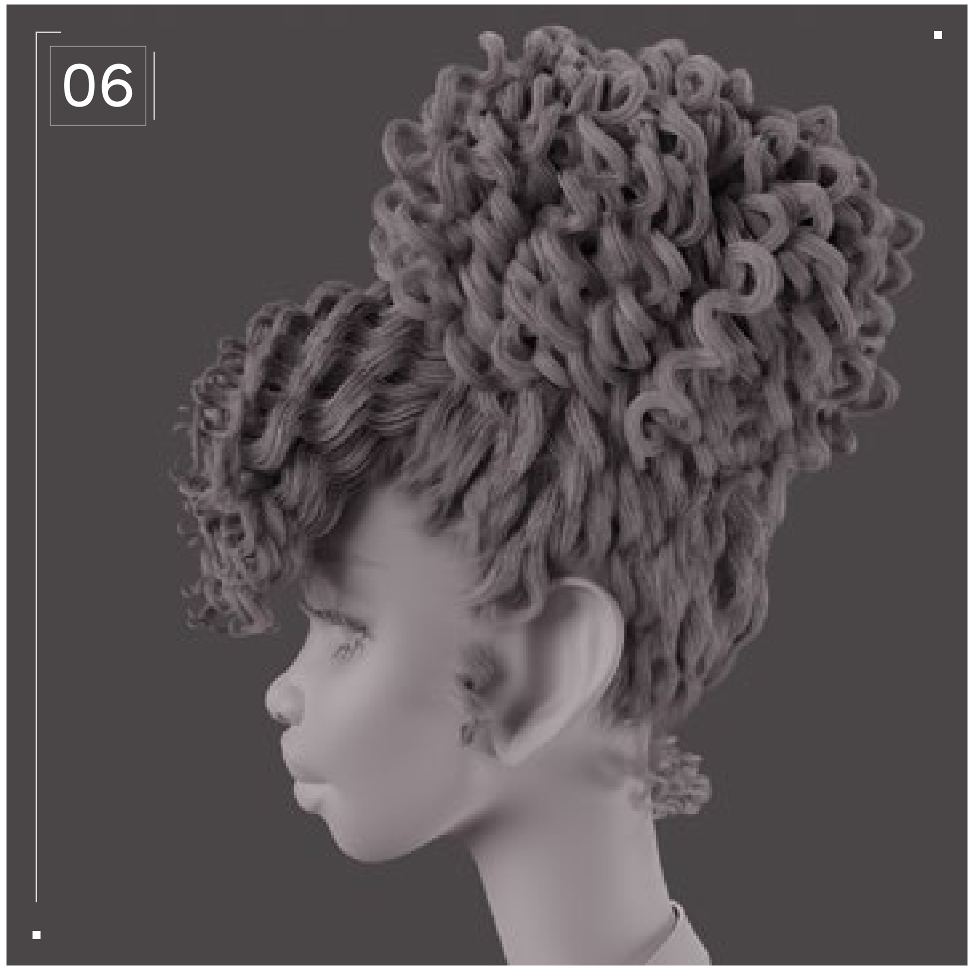
Step-by-Step Guide



I added a curl hair curves modifier to each section and adjusted the frequency and radius of the curls based on tension. The hair coming from the root is being pulled into a scrunchie, so the curls are stretched. Because I went for a more relaxed style, it isn't pulled tight. That would make the hair look more wavy, almost straight. The rest of the hair has the same or similar radii with different frequencies relative to the length of the hair. I also added a frizz hair curves modifier to break up the uniformity.



I added a duplicate hair curves modifier and a clump hair curves modifier to create curl clumps.

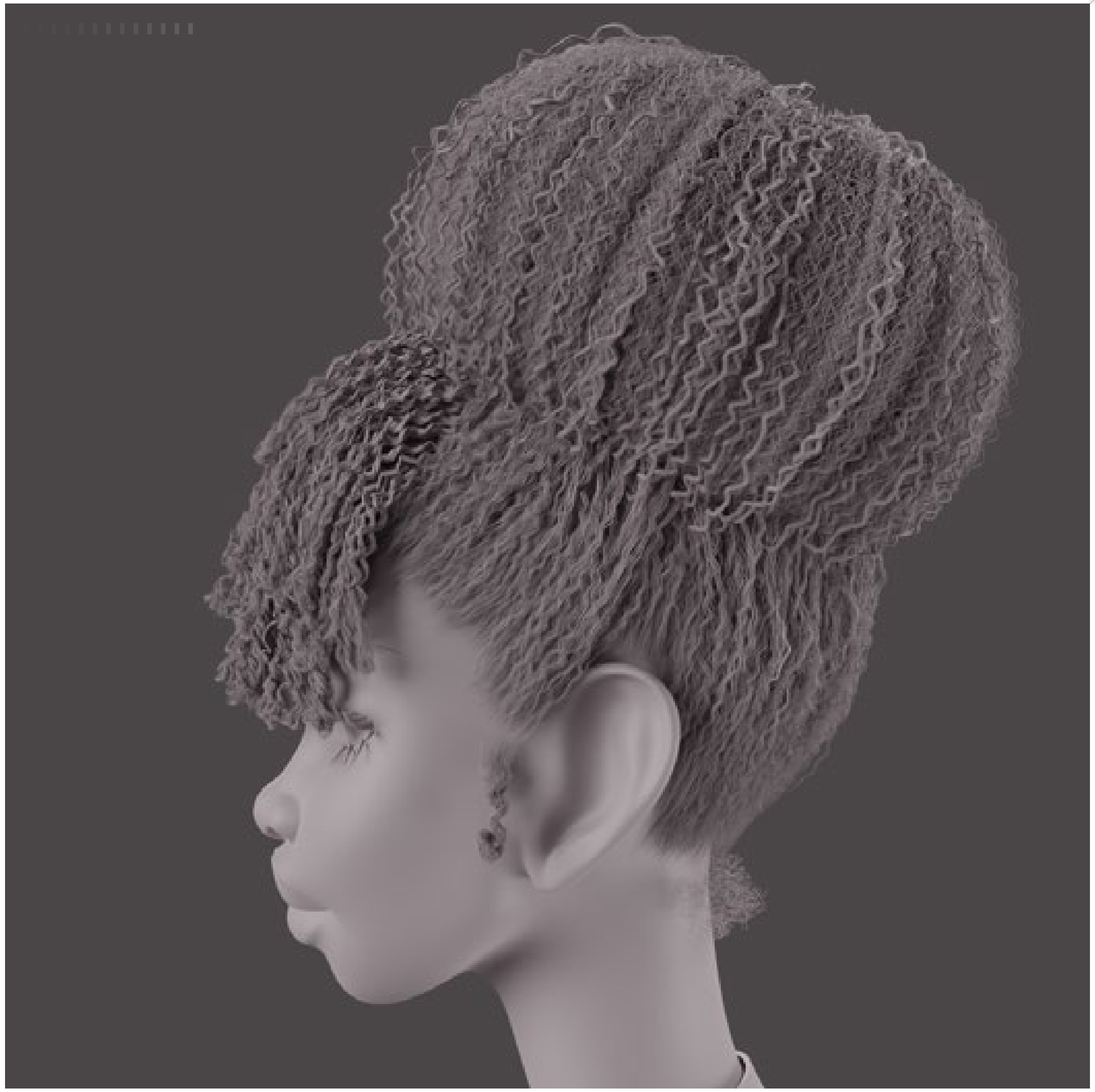


I added a roll curves modifier to give my style a more natural look. Natural hair sometimes coils more tightly or curls back on itself at the ends.

Alternative Hair Texture

For tighter curls, or type 4 hair, the individual spirals or ringlets are smaller in volume/radius and higher in frequency. The hair clumps together differently, creating bigger forms that look almost solid. You could add more ringlets or create a base with thicker hair strands or even hair cards underneath.

I would also leave out a lot less hair along the perimeter for the sideburns and nape hair. For a more voluminous bang, the coils would be less defined, while for a flatter bang, the coils would be more defined. For my example, the only change I made to the groom was combing the bang down a little and duplicating the bun so that I could add some defined curls.

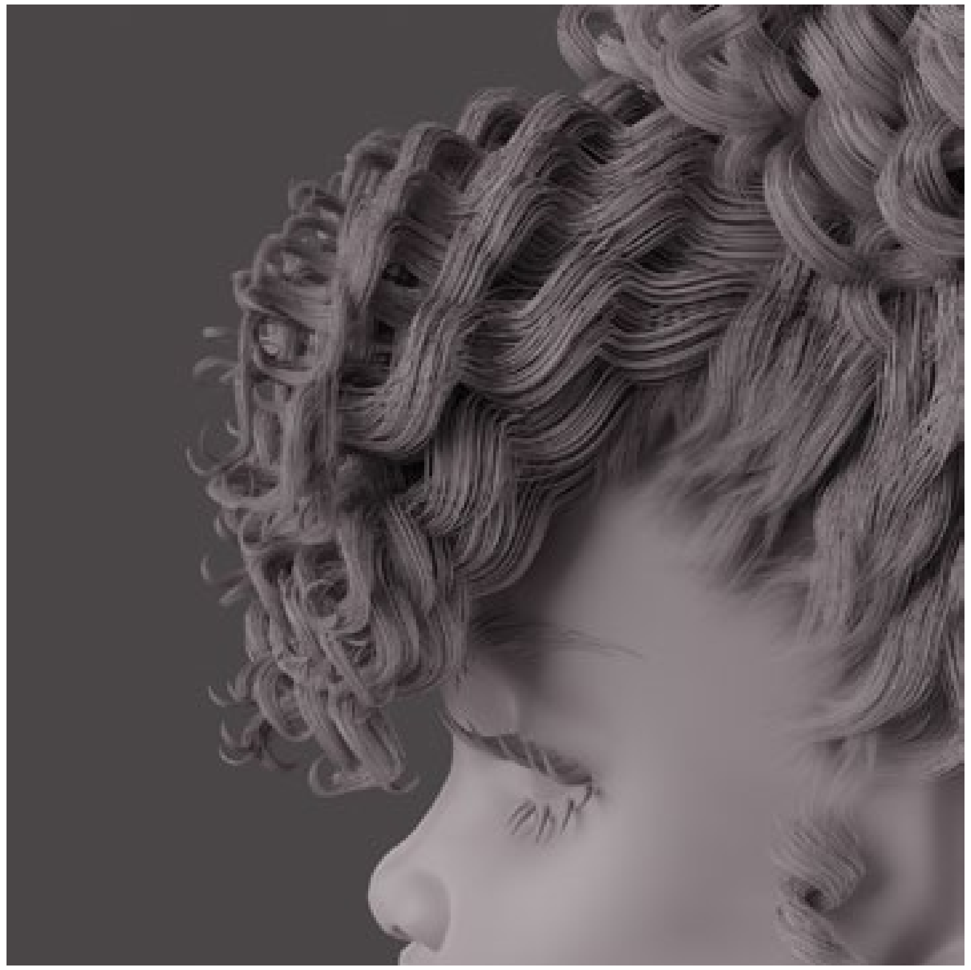


// Project Settings

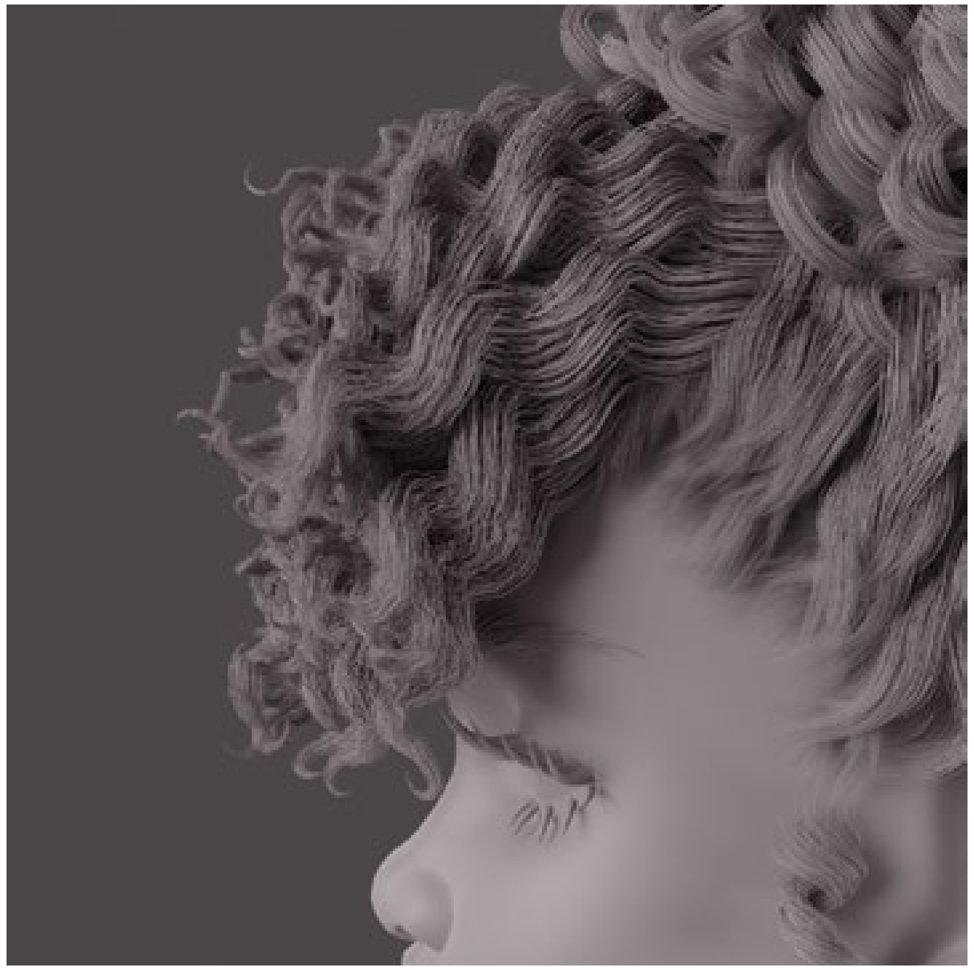
Technical Watchouts

You don't need to turn your subdivision level way up to get perfectly smooth curls. I didn't notice a significant increase in quality/smoothness when I had my subdivision turned up to 5, but it did slow Blender down to the point that I was waiting upwards of 30 seconds every time I made a change to my sculpt.

Even when experimenting with smaller coils/curls, I didn't have to increase my subdivision beyond 2 or 3. The results are similar to increasing the path steps setting in the old particle hair system, but much slower. This setting might be more useful for longer hairstyles, but don't forget to turn up the Curve Subdivision level under the Render Properties tab!



// Subdivision Level: 2



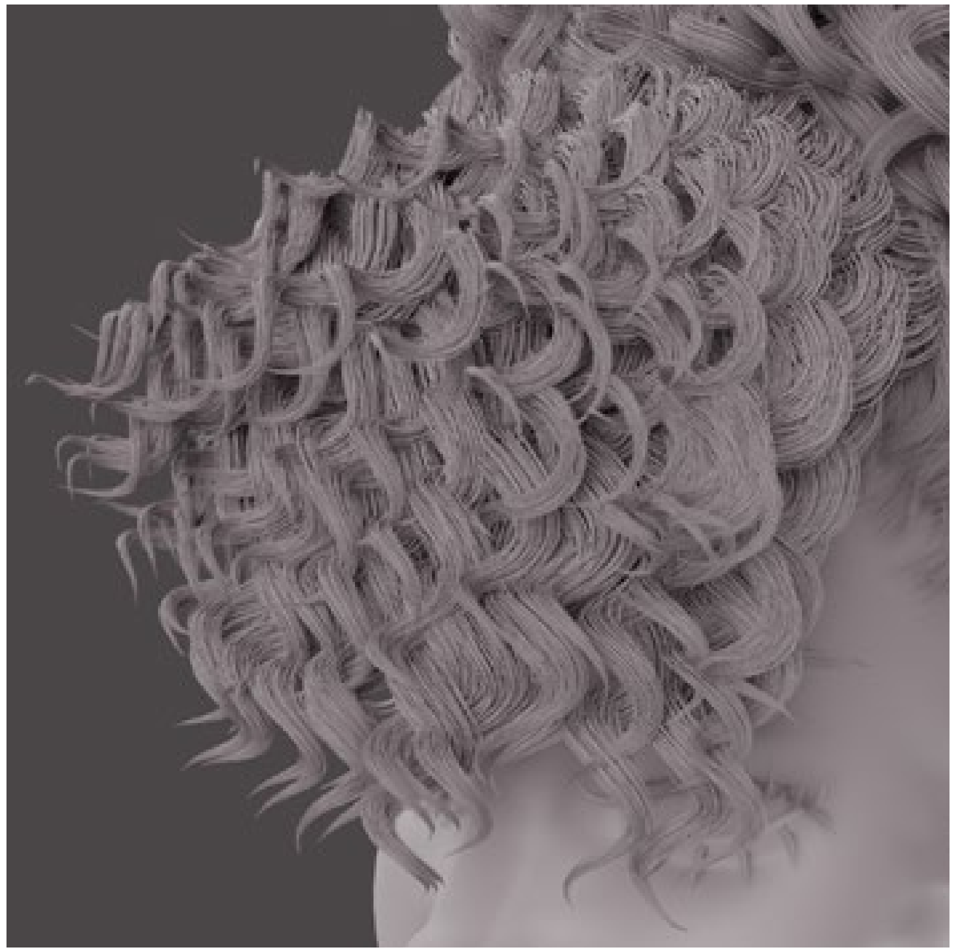
// Subdivision Level: 5

50%

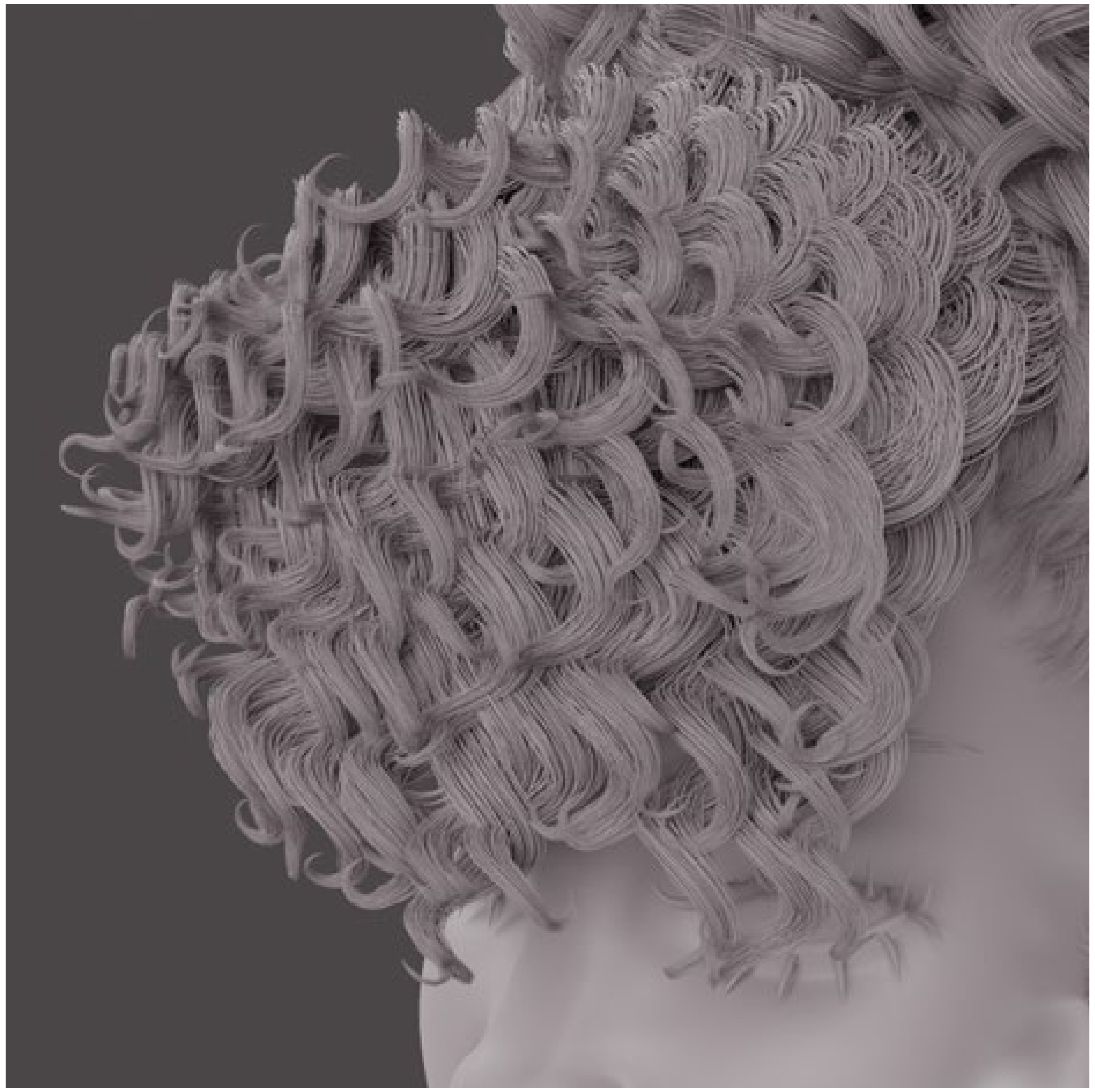
Top Tip

Adding the roll curves node to my bangs and sideburn added a bit of realism.

Without it, the ends of the hair continued spiraling downward, which made the style look more like a flexi rod set. Natural curls tend to coil more tightly or sometimes even curl back up on themselves.

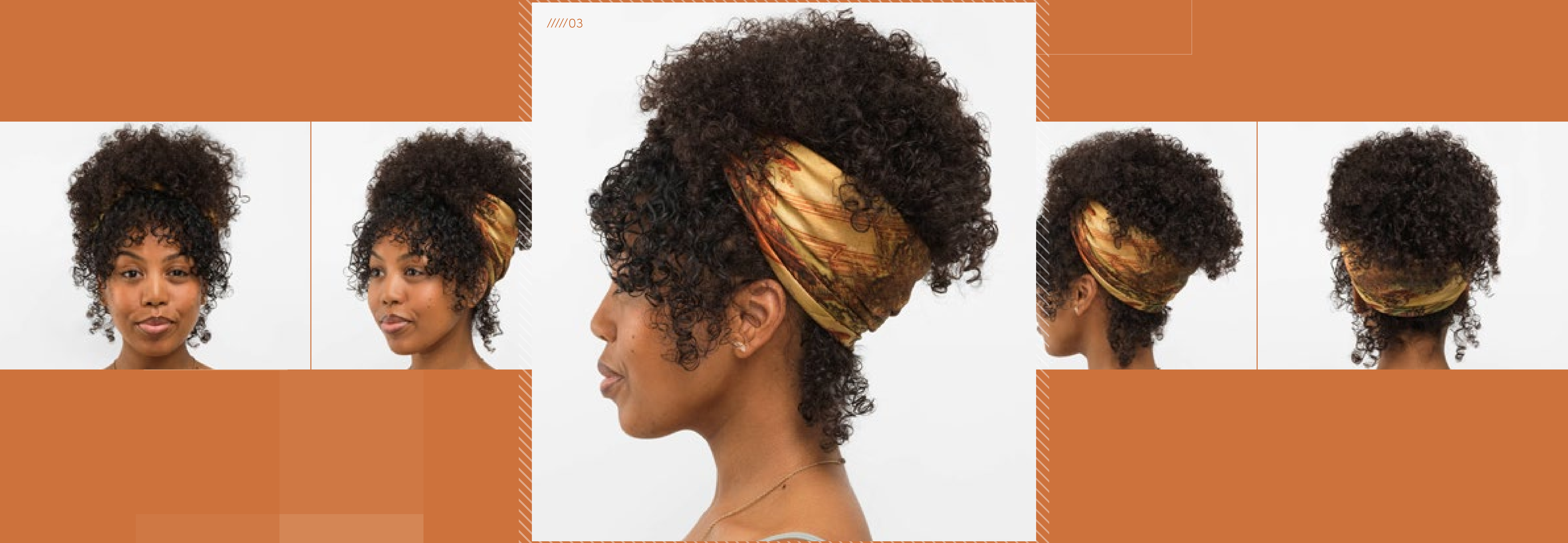


// Subdivision Level: Off



// Subdivision Level: On

Natural Curls with Headwrap



Download Images

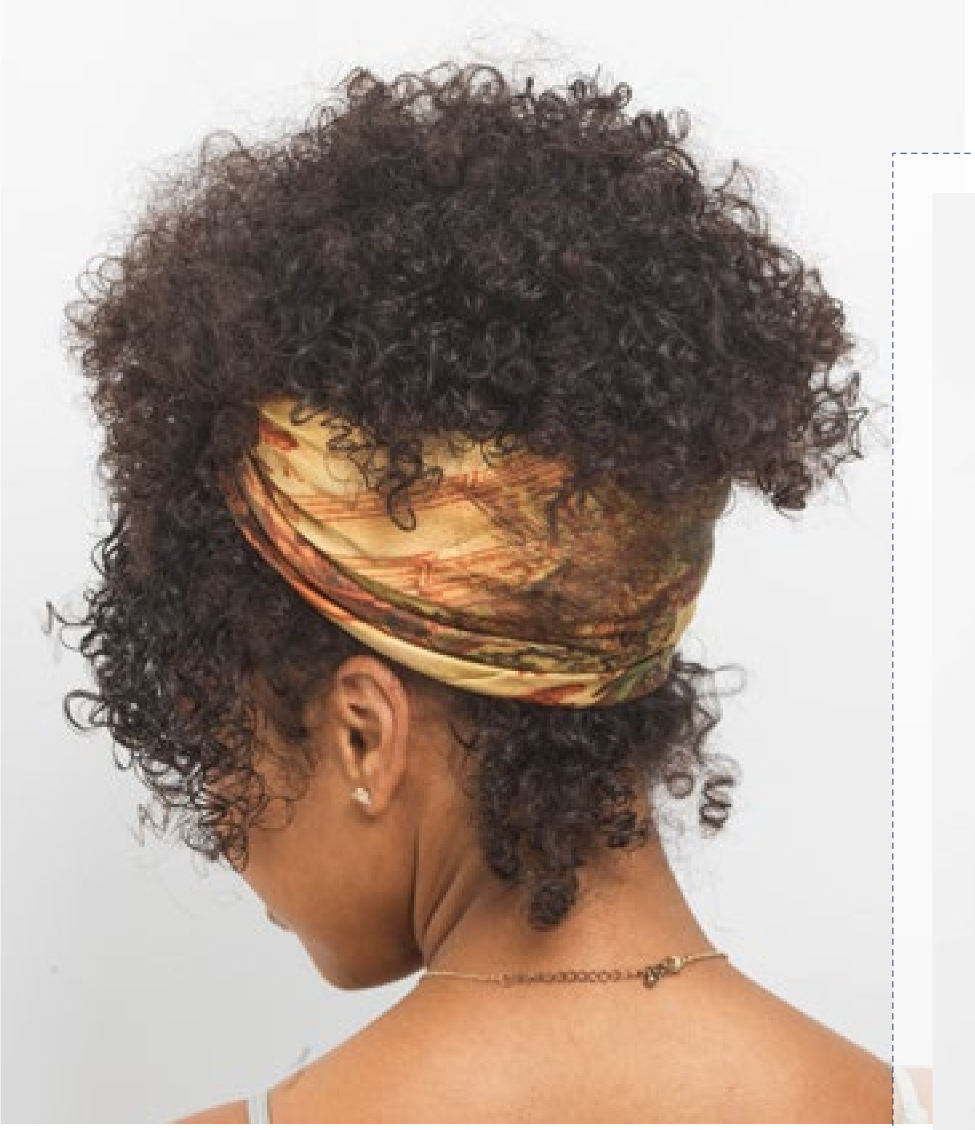
// Natural Curls w/ Headwrap

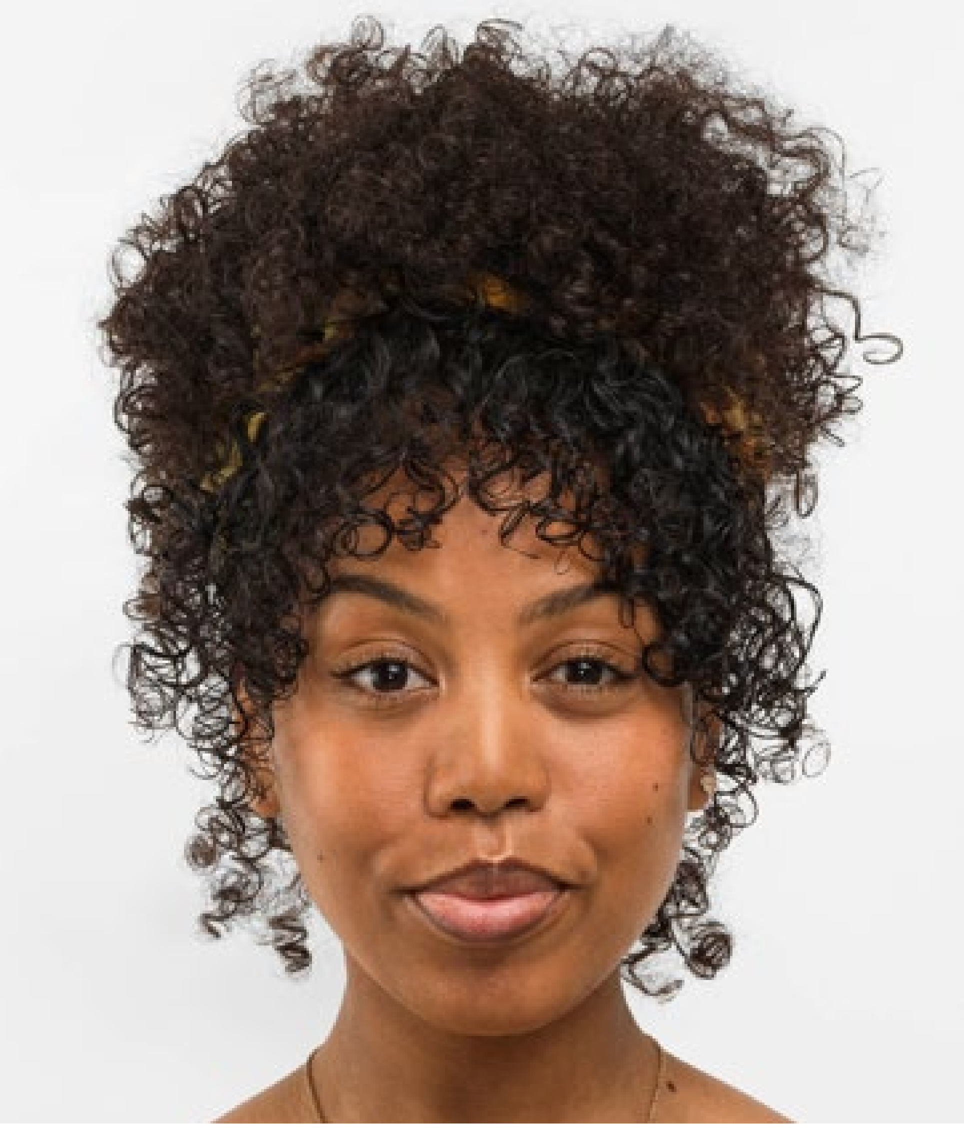
// Hairstyle

Natural Curls with Headwrap

Style Background: Headwraps have long been worn as traditional attire in many African cultures, going by different names depending on the region. They're incredibly versatile, coming in a vast variety of colors, patterns, and materials. They can be styled in many different ways to create different looks, and are worn both for formal occasions and for everyday wear. The headwrap not only serves as a fashion statement, but it also functions as protection against environmental elements to ensure desired style lasts longer.

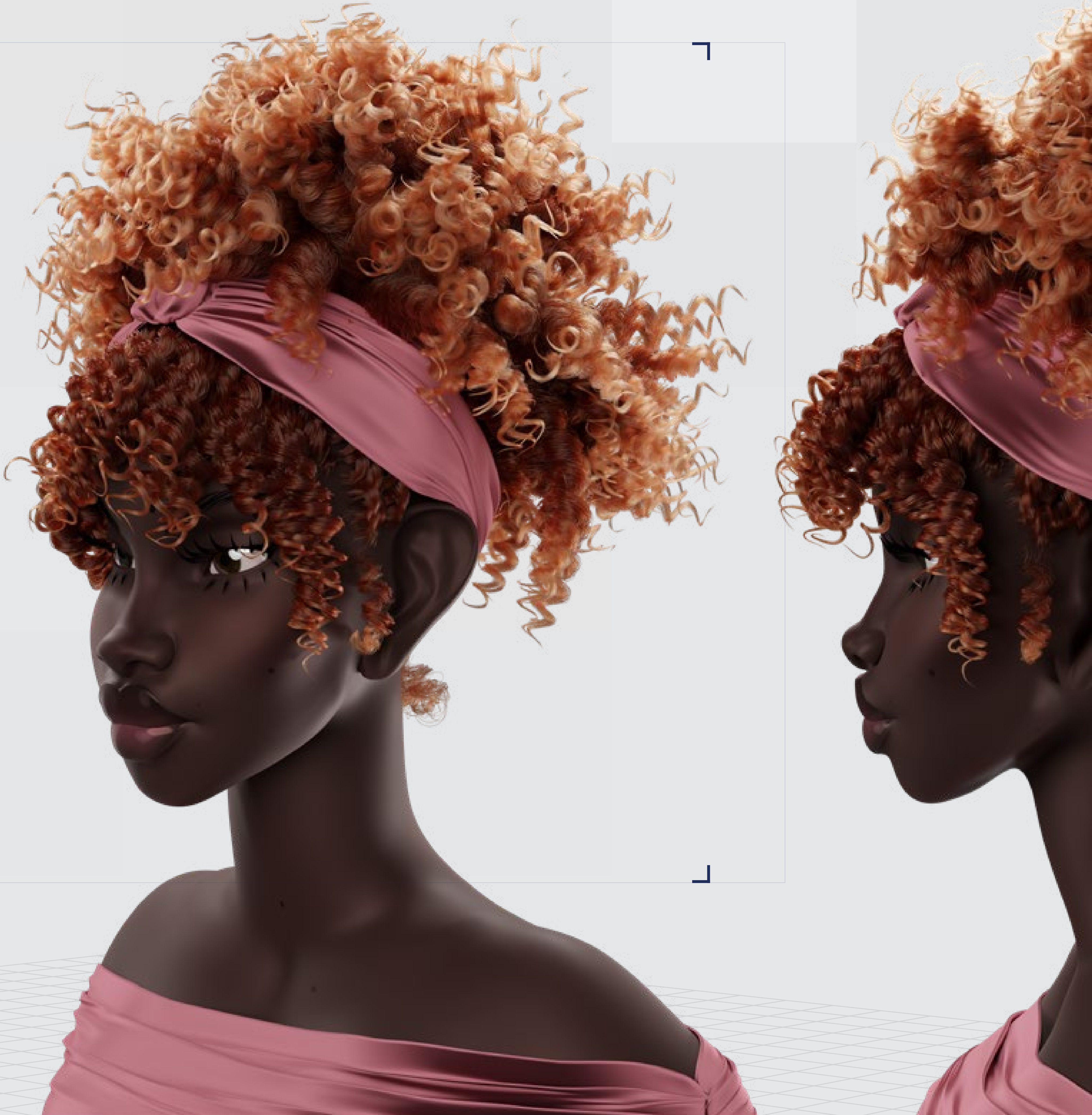
Download Images





3D Design

Natural Curls with Headwrap



Digital Artist:
Diandra Rose

// Natural Curls with Headwrap

3D Design

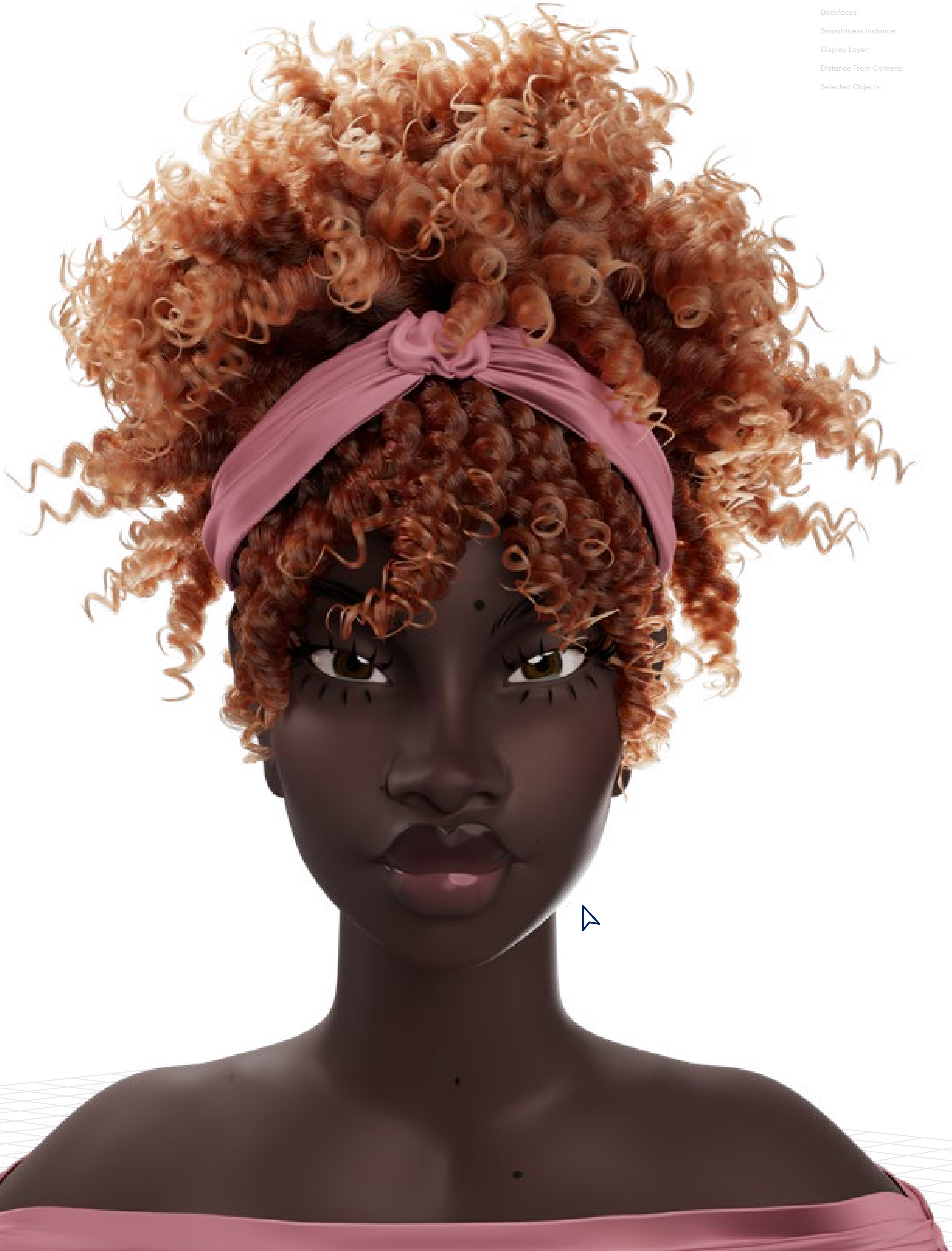
Curl Type Modeled: **3A/3B**

// 01

// 02

Download 3D Render

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

Project

Settings

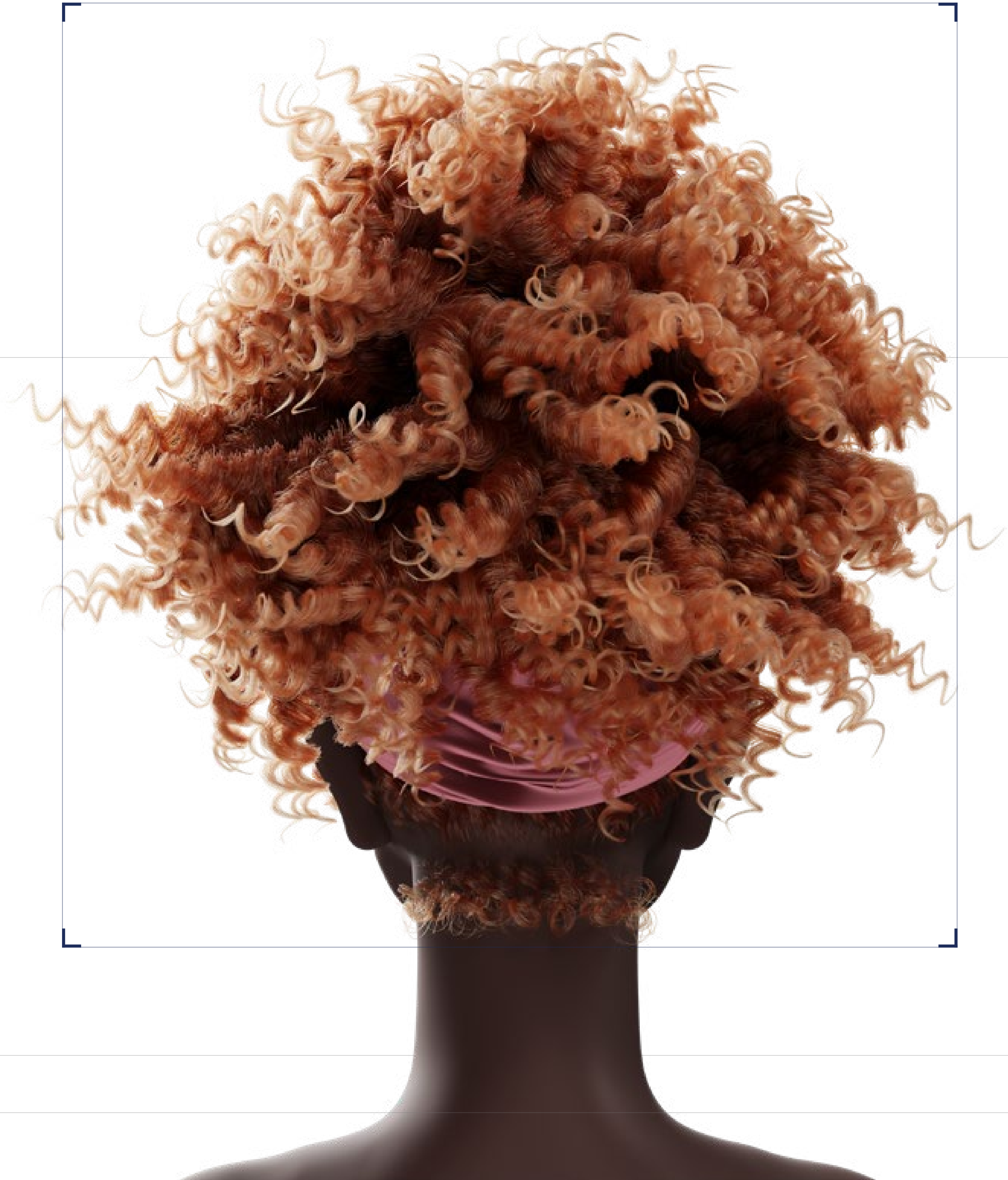
// Natural Curls
with Headwrap

// Project Settings

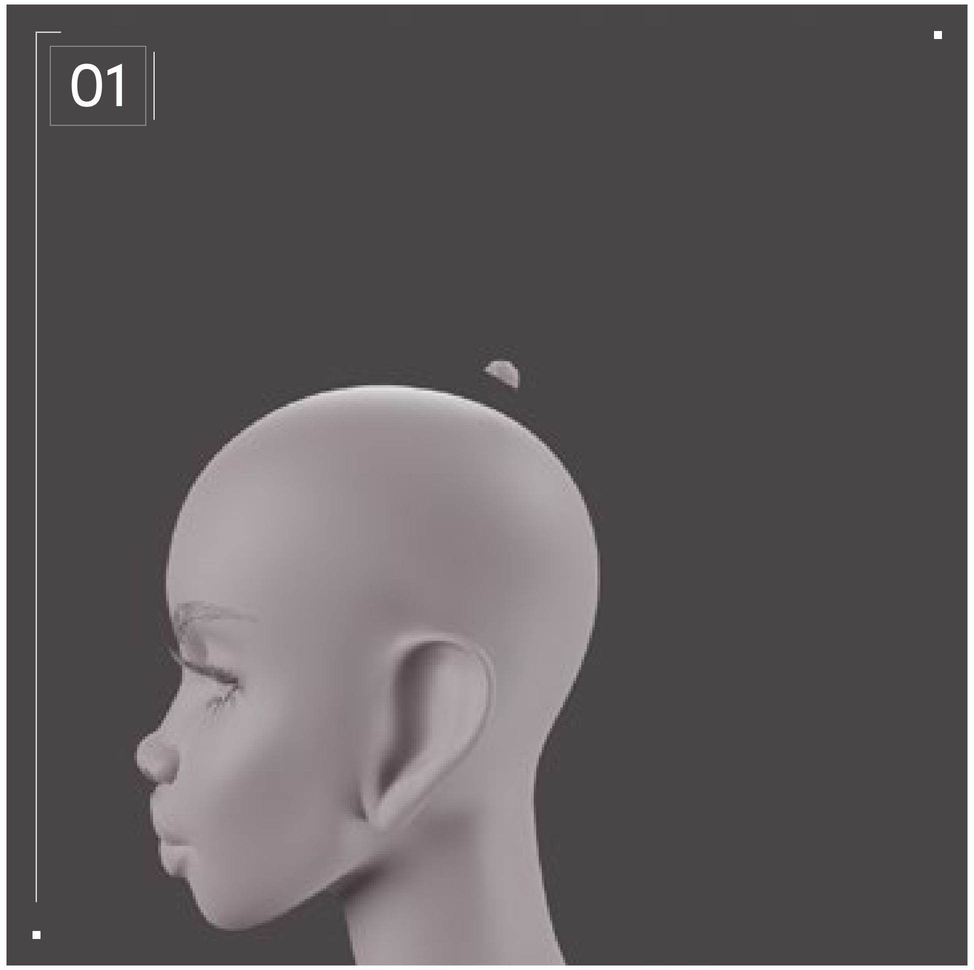
Program Guide

Software: Geometry node hair system in Blender 3.5

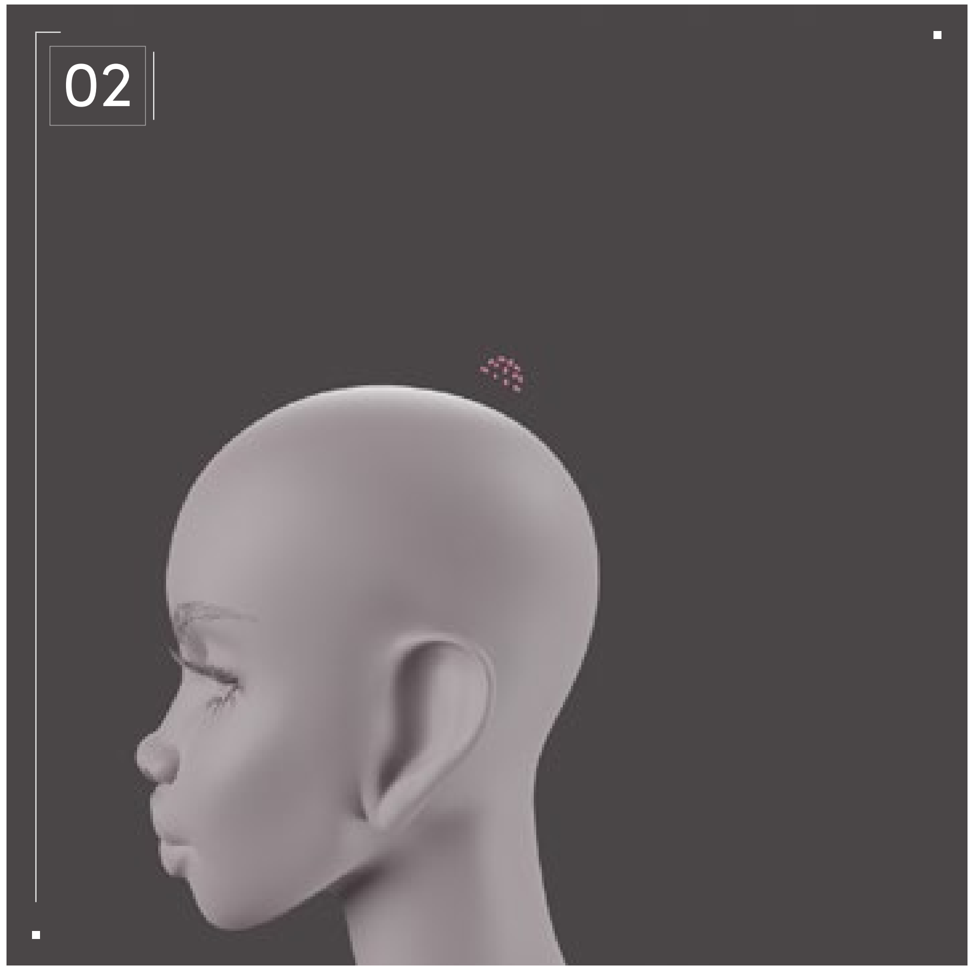
Quick note: The bangs, roots, sideburns, and nape hair here are all made in the exact same way as they are within the guide to creating the Curly Bangs style. With this in mind, please refer to page 19 for the base of the hair, returning here to add the all-important ponytail.



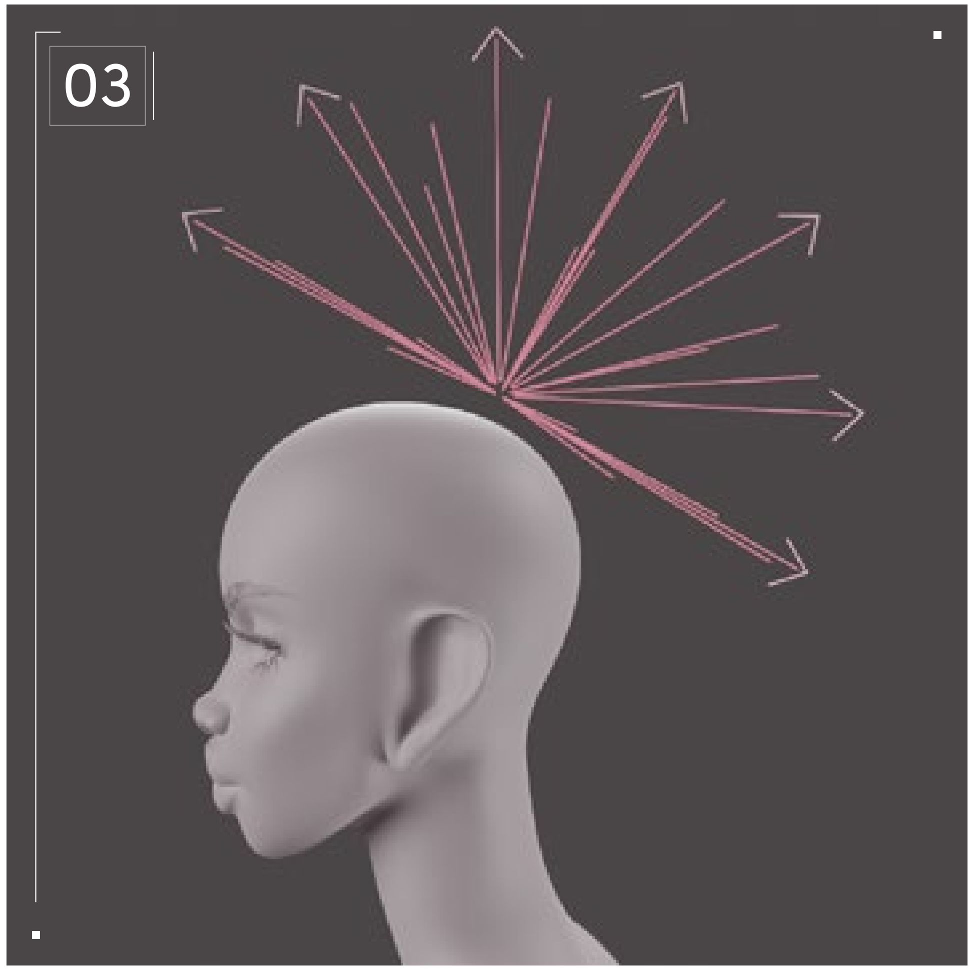
Step-by-Step Guide



I start by adding an icosphere, cutting it in half, and aligning it with the head.

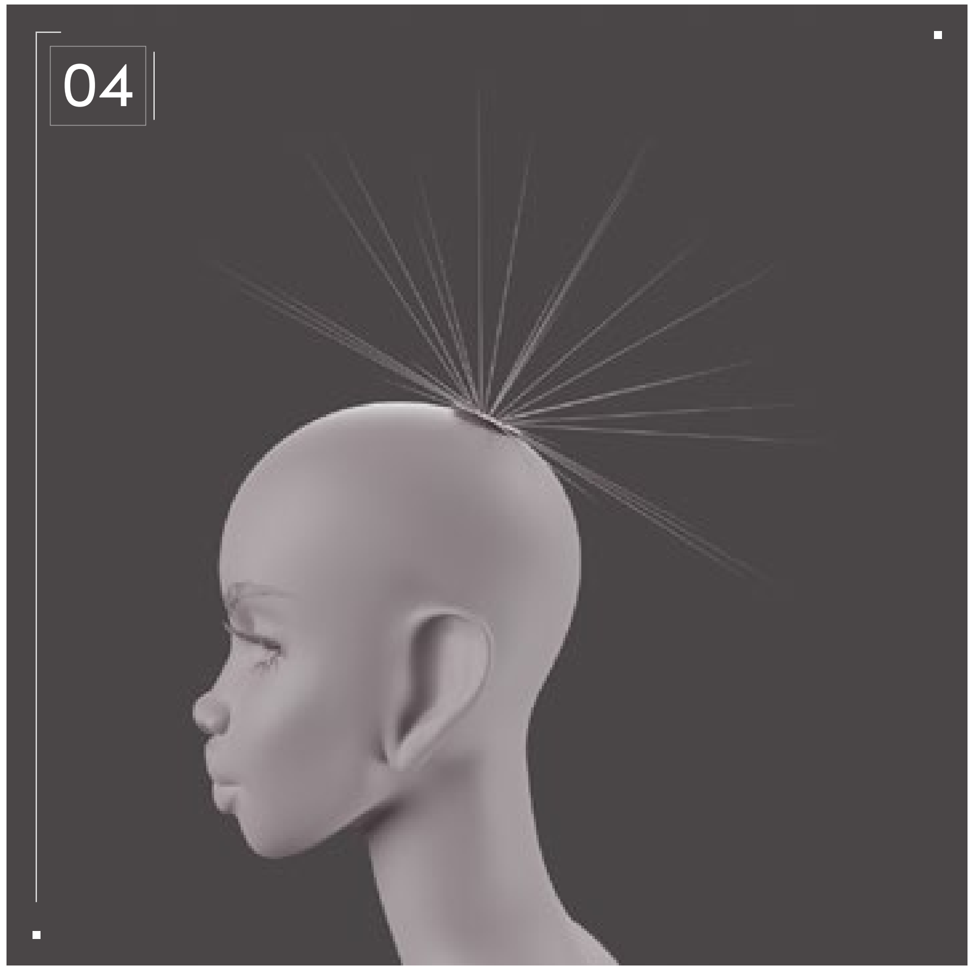


I delete the edges and faces, leaving only the vertices.

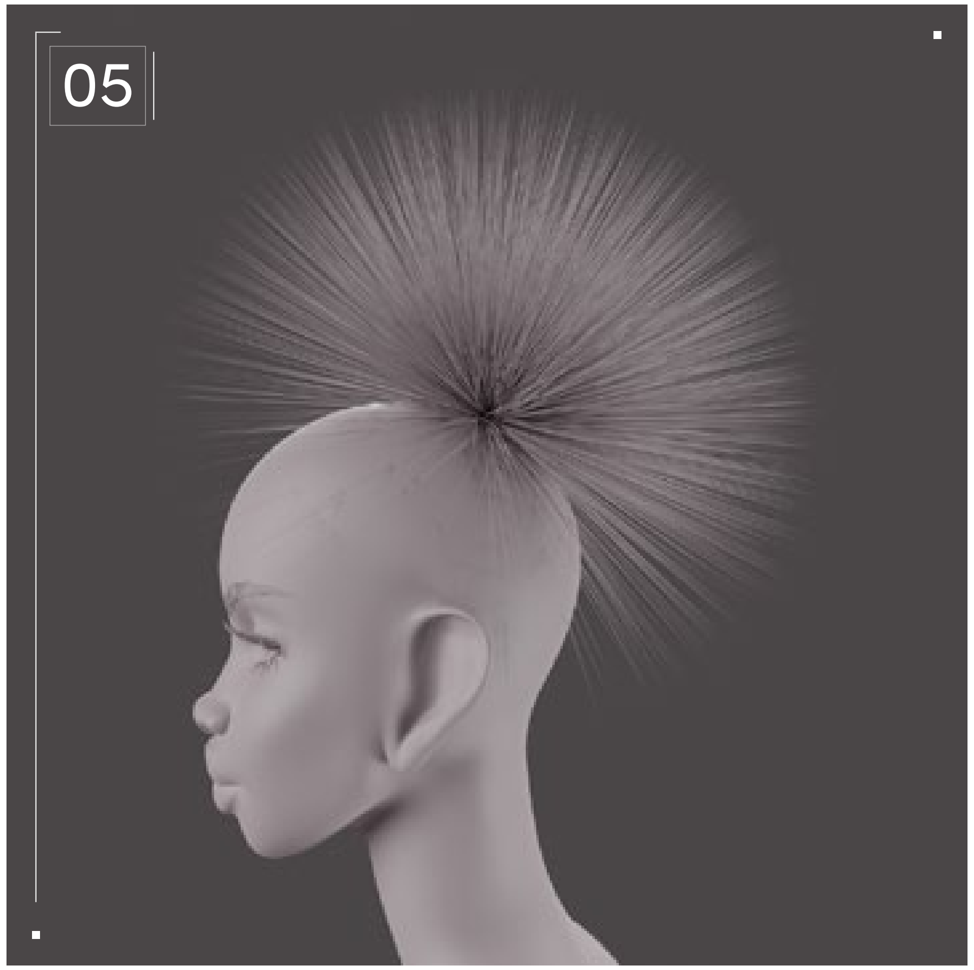


With the transform pivot point set to the cursor or the bottom edge of the halved icosphere, I extrude the vertices.

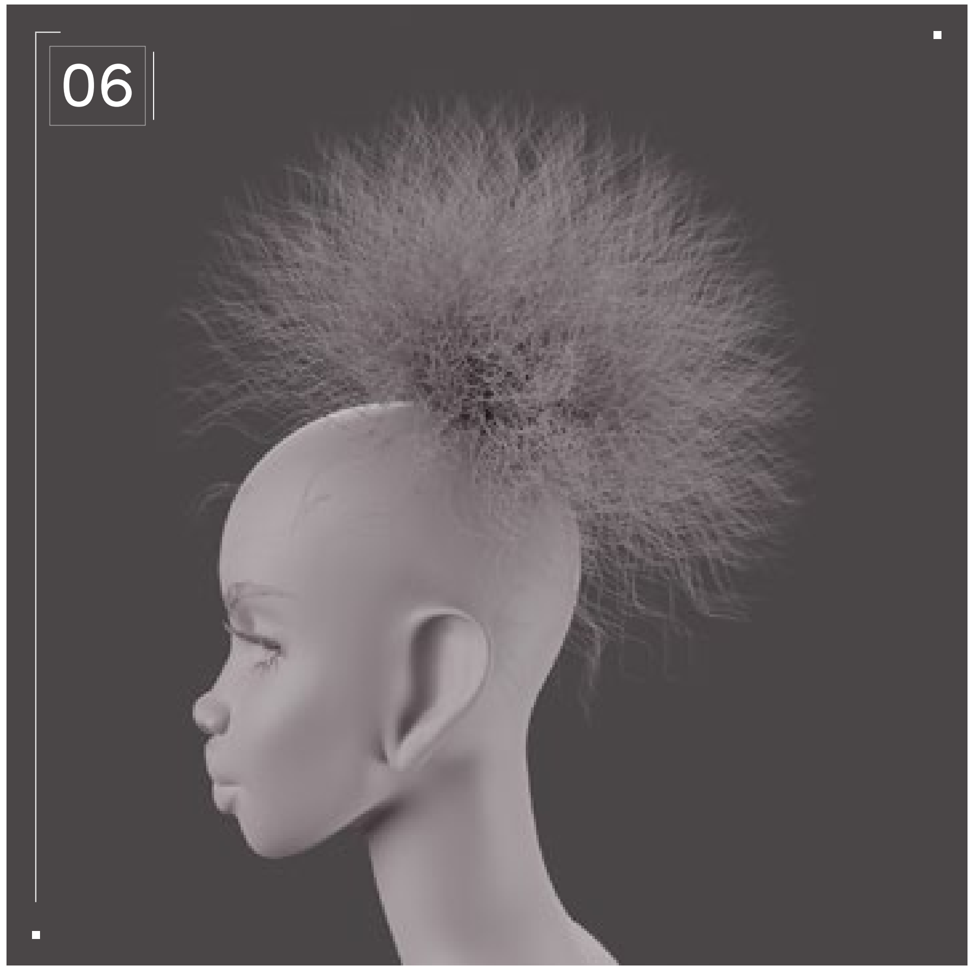
Step-by-Step Guide



Next, I convert the new edges to curves so that I can check to make sure that they're pointing in the same direction, convert them to hair curves, and parent them to a disk mesh.



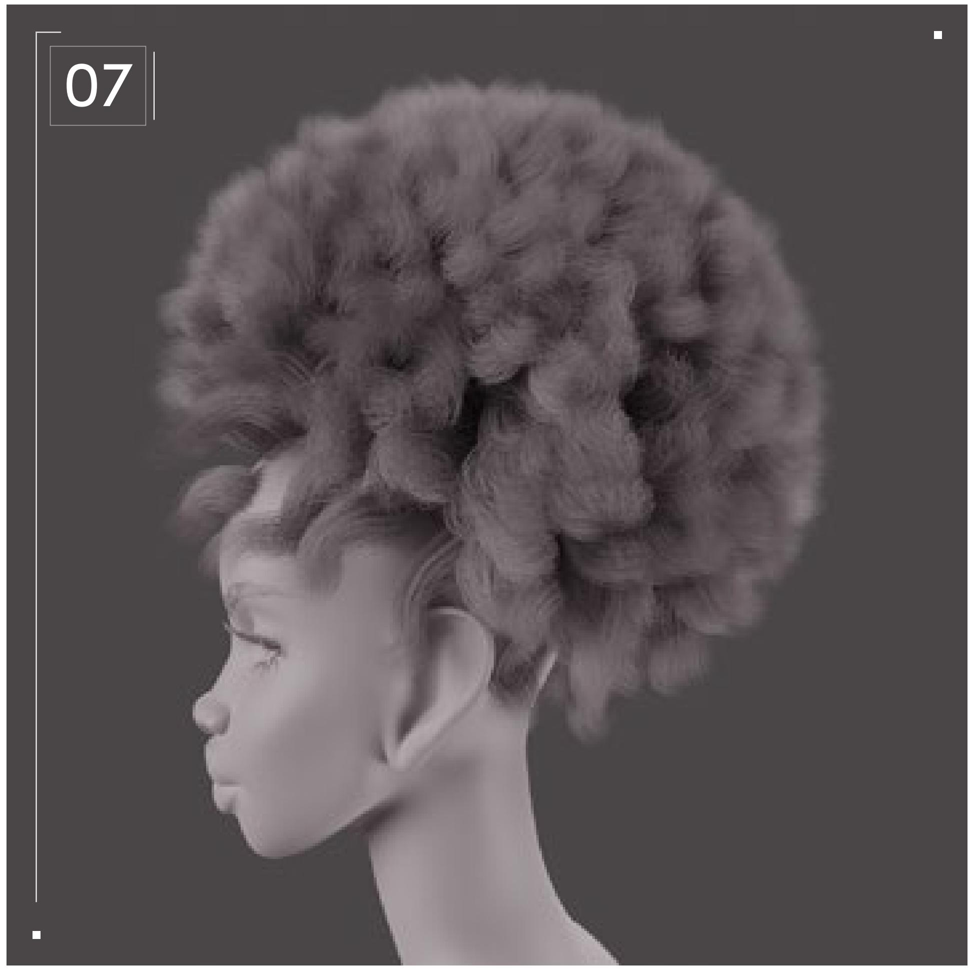
I initially used a duplicate hair curves modifier to create children hair strands but, later on, I shaped the ponytail and realized that the interpolate hair curves modifier not only created more variety in the shapes of the hair strands, but it also did a slightly better job of filling in the gaps.



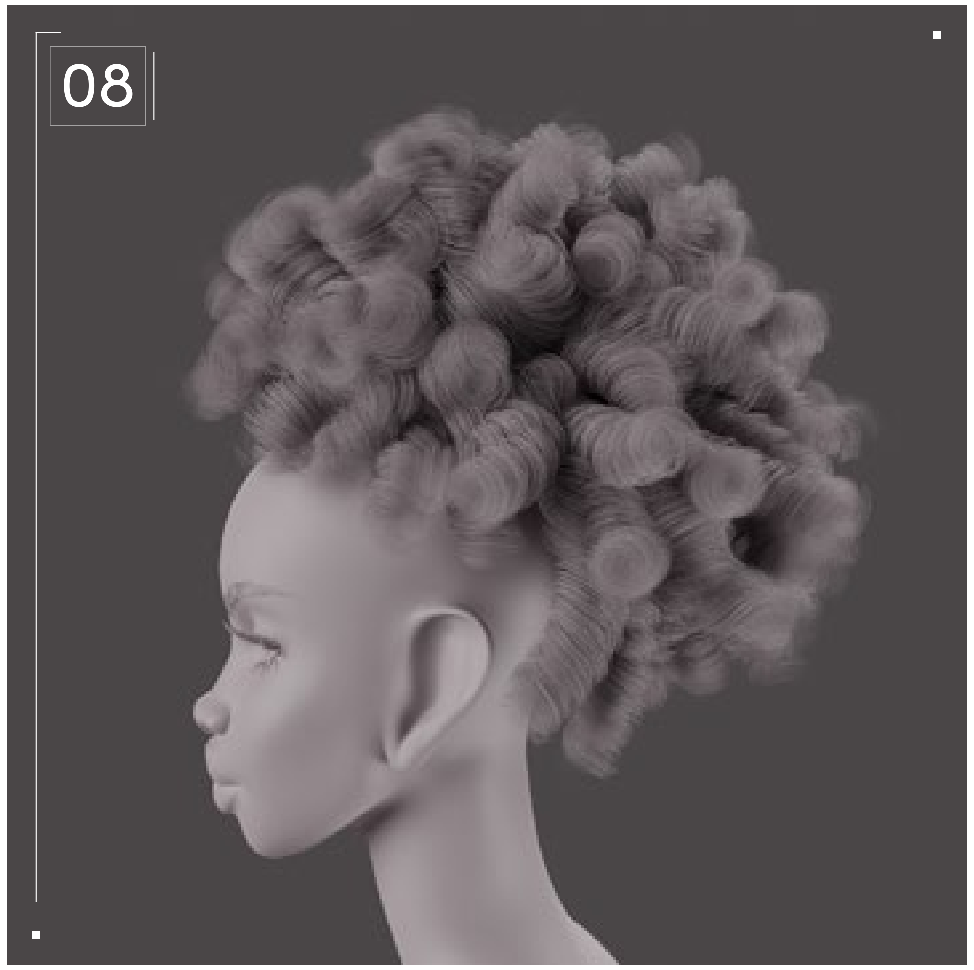
I decided to make this style with a flexi rod set in mind so the curls will be slightly more uniform, and the ends won't curl back up (flexi rods are foam hair rollers that bend easily to curl hair; they come in a variety of lengths and sizes to accomodate a multitude of hair types). I added a curl hair curves modifier, but I left the default settings so that I could continue building the hair. If I turned the frequency up too high, it would be harder for me to see what I was doing.

// Project Settings

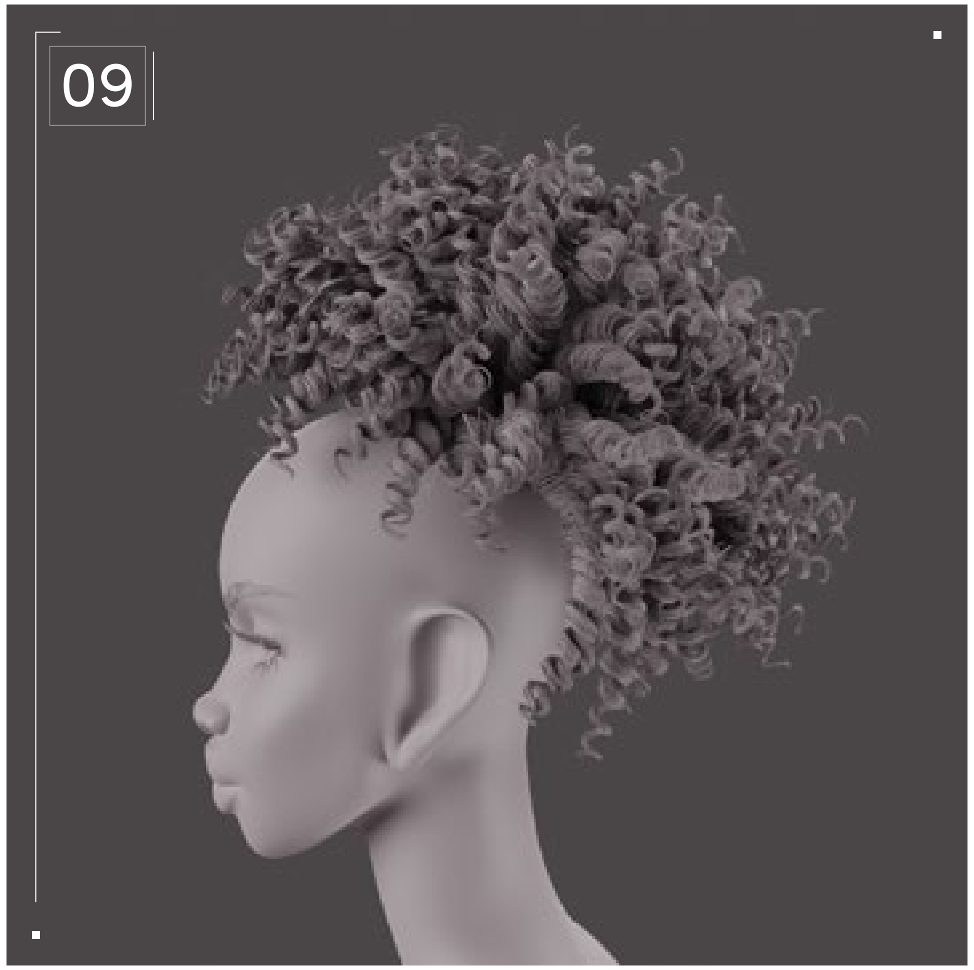
Step-by-Step Guide



I added a duplicate hair curves modifier with even thickness turned on.

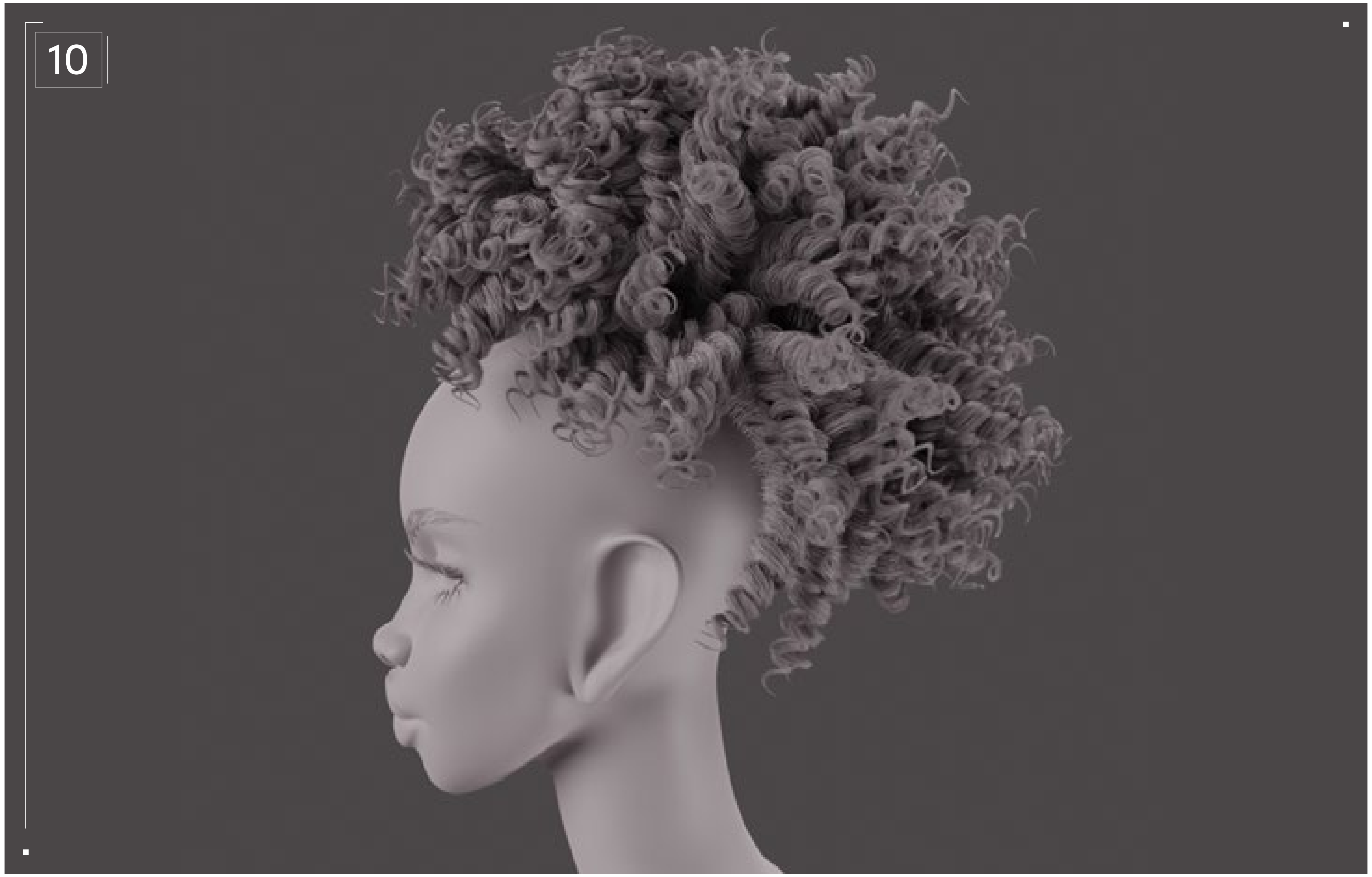


The hair looked thin, and the curls looked almost like waves, so I doubled the frequency of the curl hair curves modifier. Still, the hair strands were clipping through the head, and they looked too uniform. With the brush falloff set to “projected,” I combed the ends of the hair down so that the overall shape was still spherical, but it framed the head better and fell where the hair was supposed to be longer.



Next, I added a clump hair curves modifier with the shape set to 0.375 so that the curls were defined but there was still a little volume/fullness at the root/center.

Step-by-Step Guide

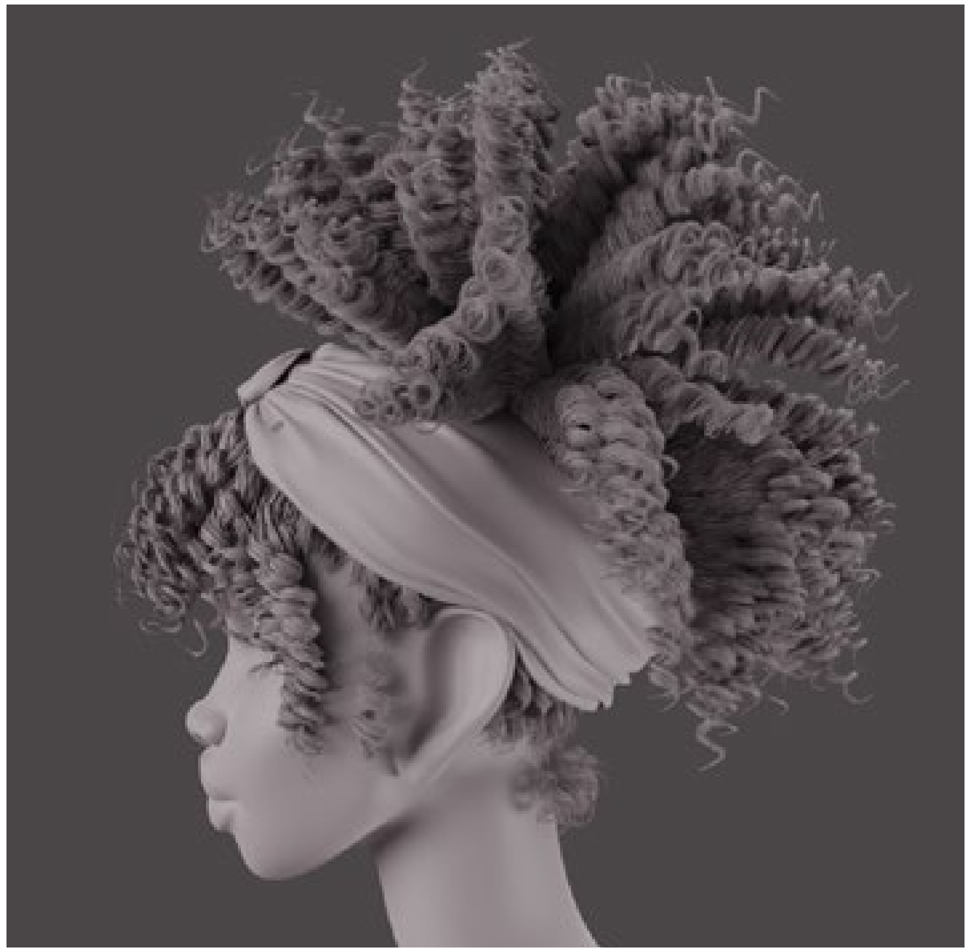


The ponytail still looked a little thin, which is not a desirable look, so I added a frizz hair curves modifier above the last clump hair curves modifier — and that's it!

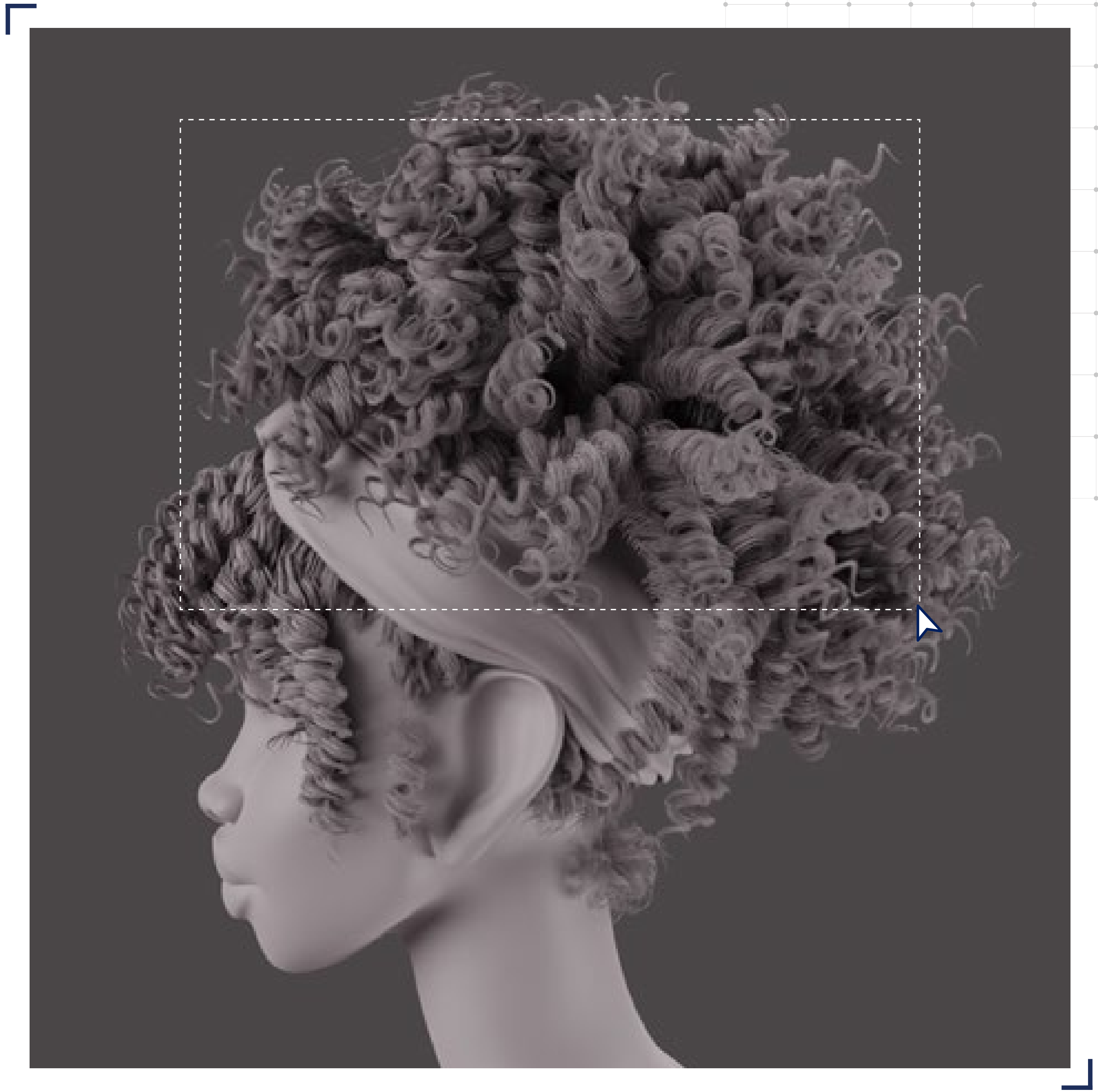
Alternate Hair Texture

I covered most of this in the Curly Bang section, so I'll just show how this style would look with type 4 hair. I decided to keep the style fluffier this time. Because the curls are no longer "manipulated," I added a roll hair curves modifier to the ends. This could be the same hair without a flexi rod set!

Top Tip



// Clump Hair Curves: Off

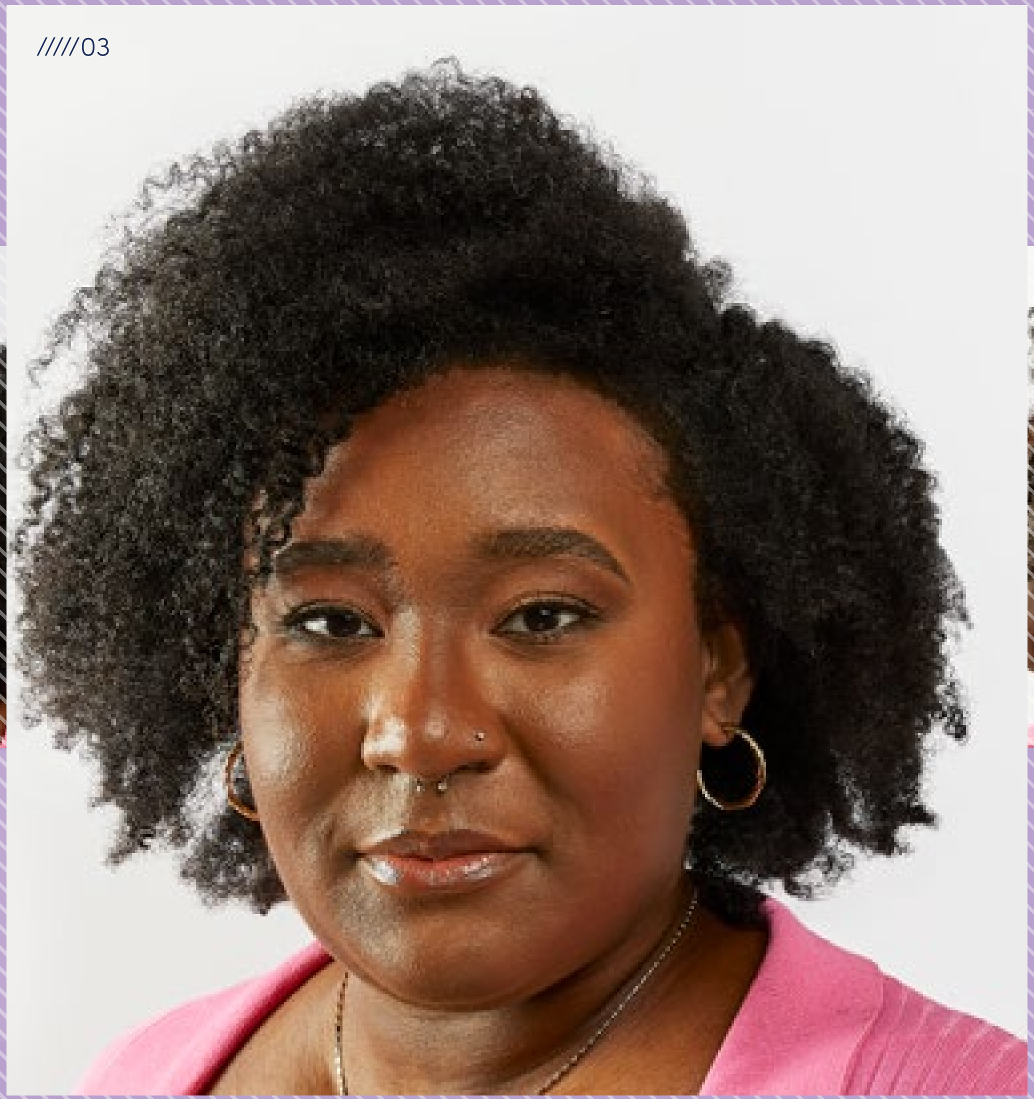


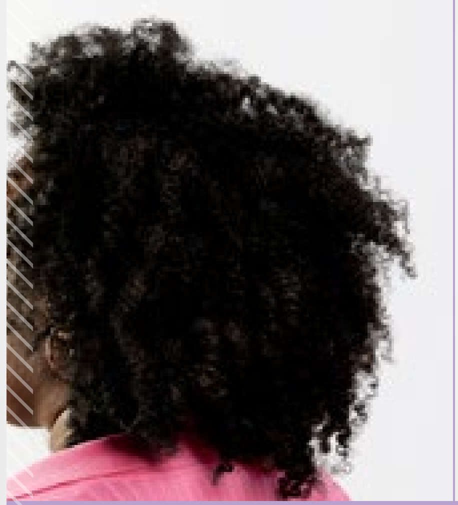
// Clump Hair Curves: On

In order to achieve the volume of the ponytail, I added an interpolate hair curves modifier this time, but it wasn't enough. Adding a clump hair curves modifier with the tip spread set to 2 was the key. This is obviously not the purpose of the clump hair curves modifier, but it worked perfectly.

Twist

/////03





Out

Out
Out

Download Images

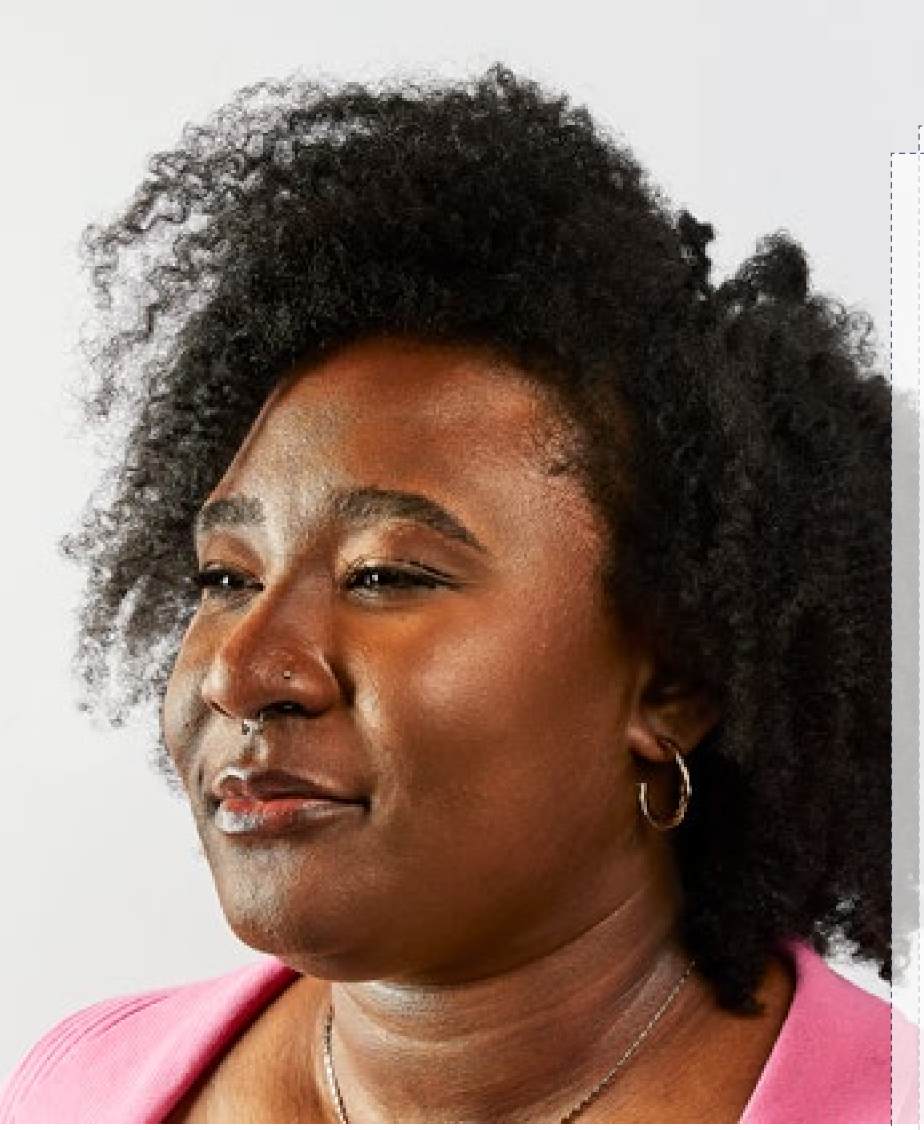
// Twist Out

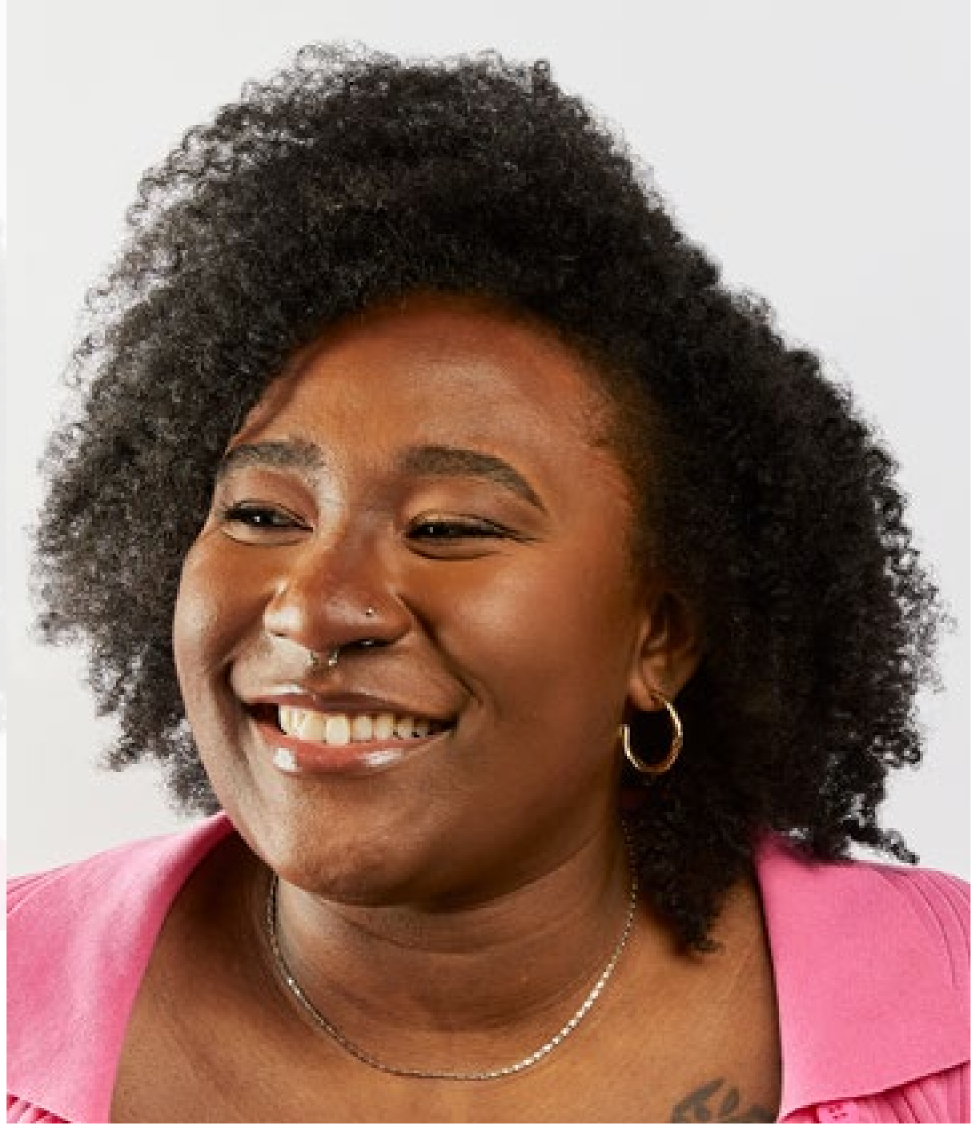
// Hairstyle

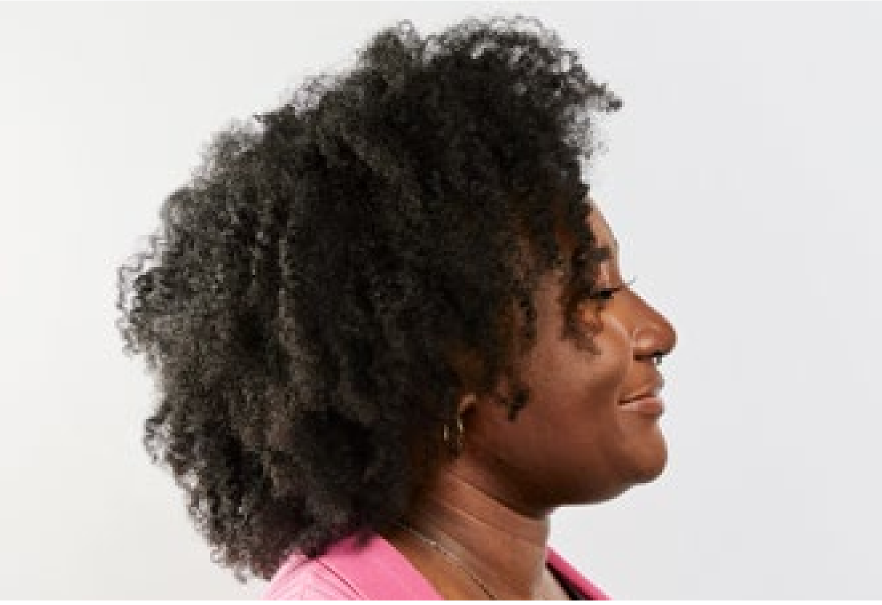
Twist Out

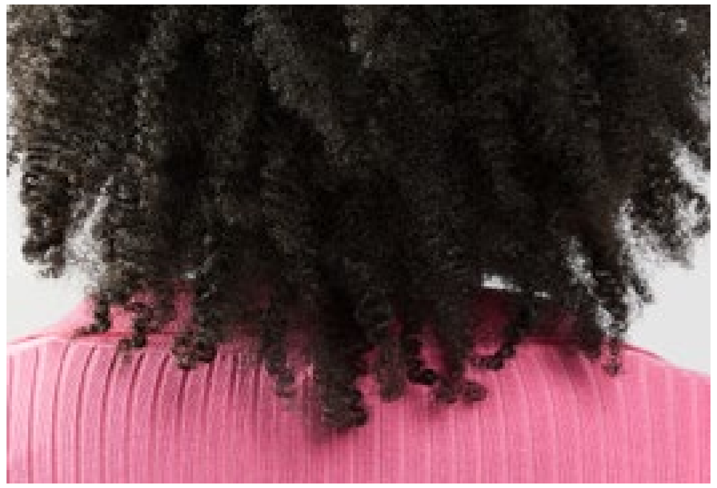
Style Background: The Twist Out is a hairstyle that is achieved by taking hair strands in their natural state, whether in curls, coils, or kinks, and styling them into Twists, letting them out after the hair sets, hence the name Twist Out. The shape of these Twists are still imprinted in the shape of the hair strands when the Twists are undone. This style is often worn to help define curls without heat styling, and to create a longer or fuller look.

Download Images

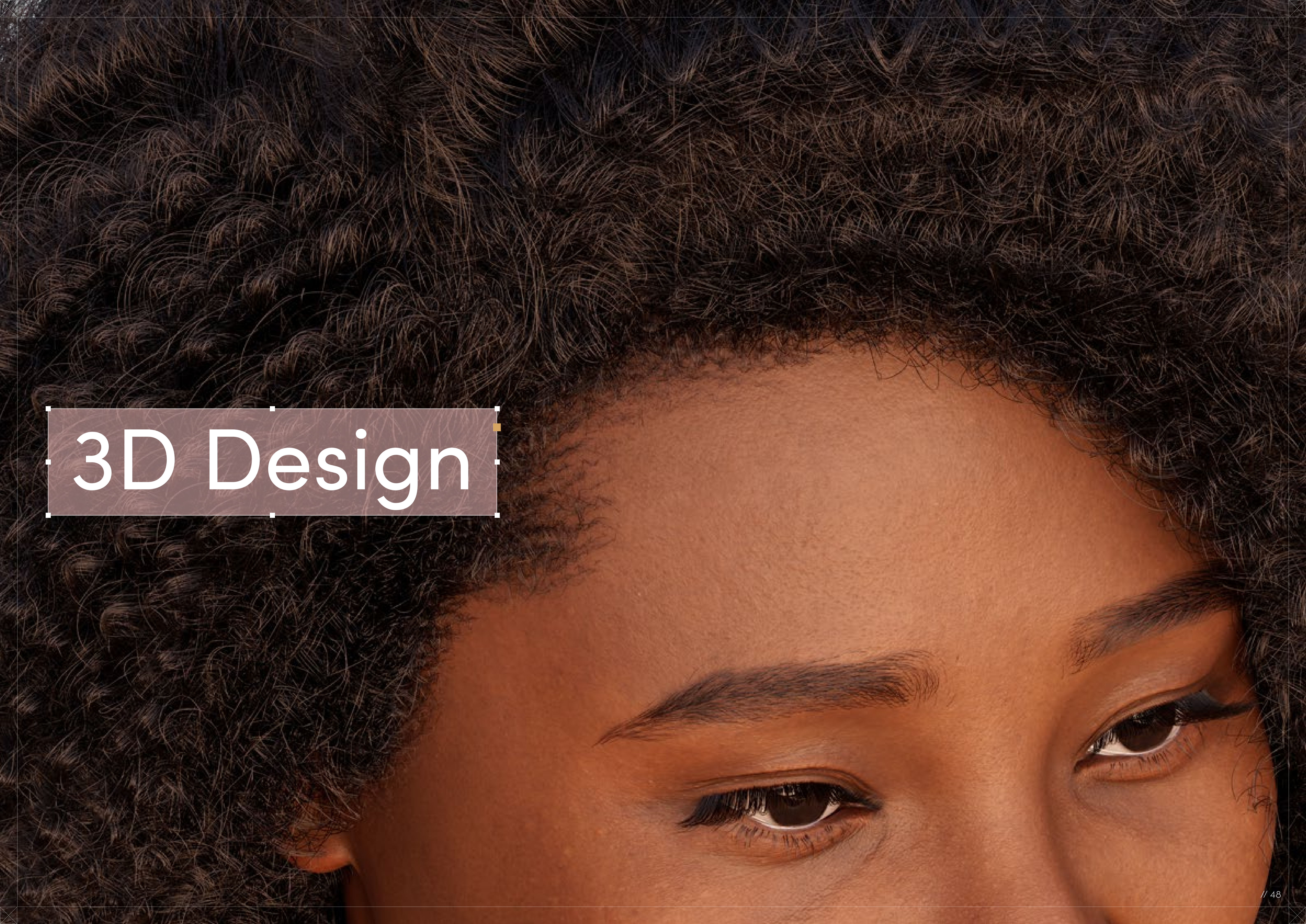




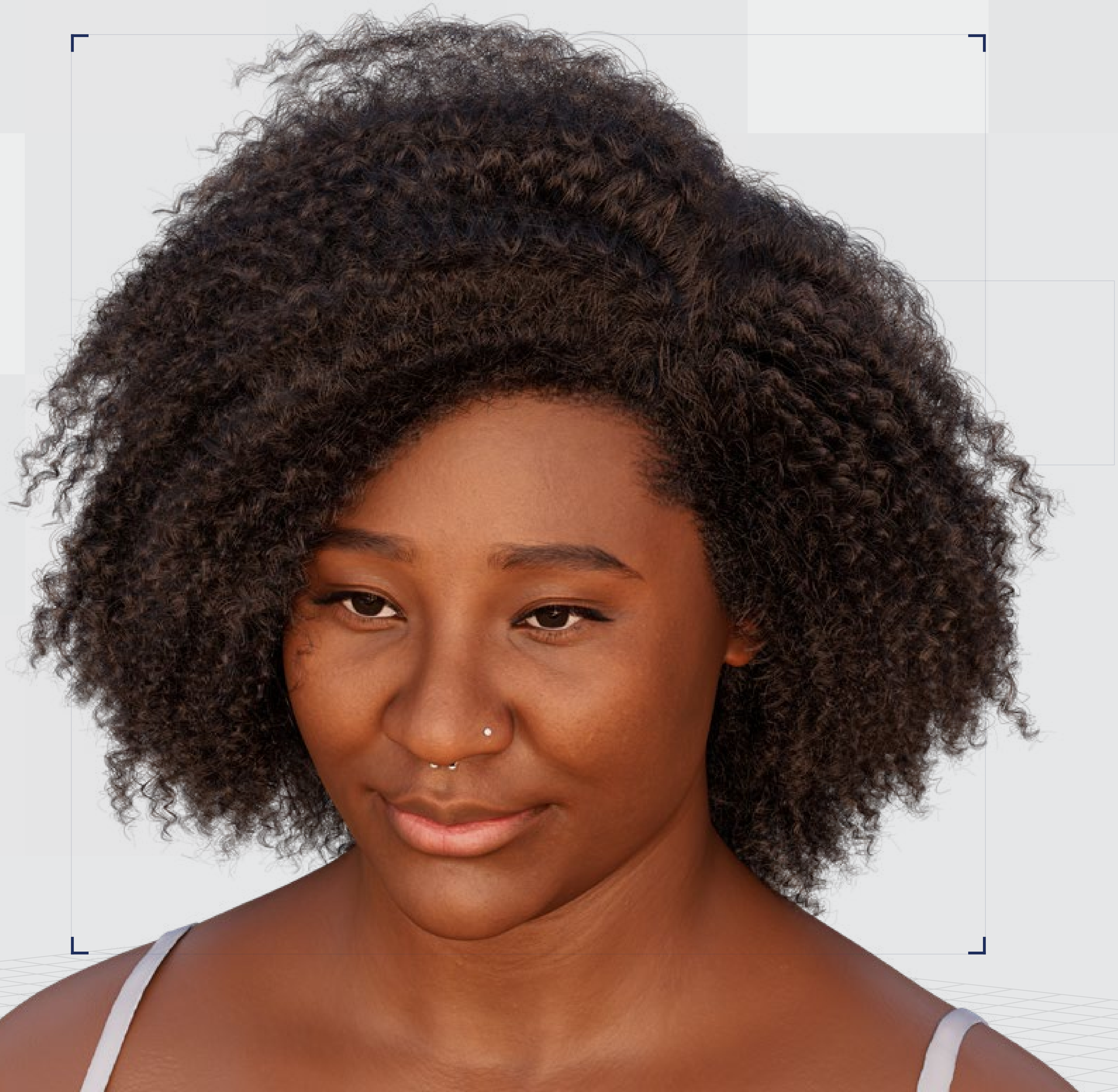




3D Design



Twist Out

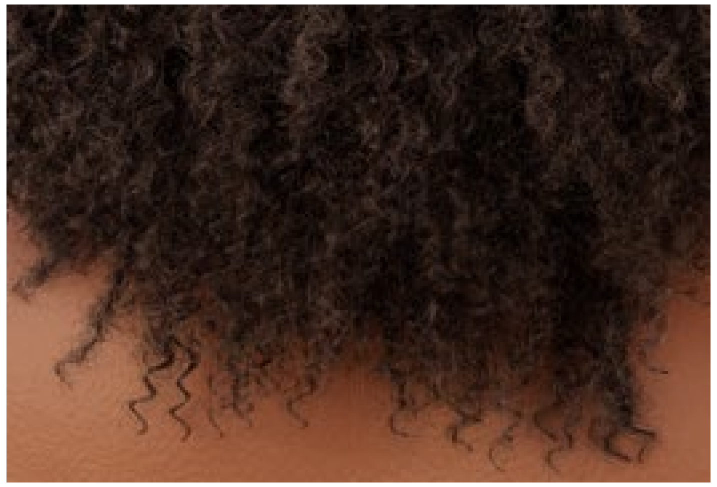


Digital Artist:
Isaac Olander

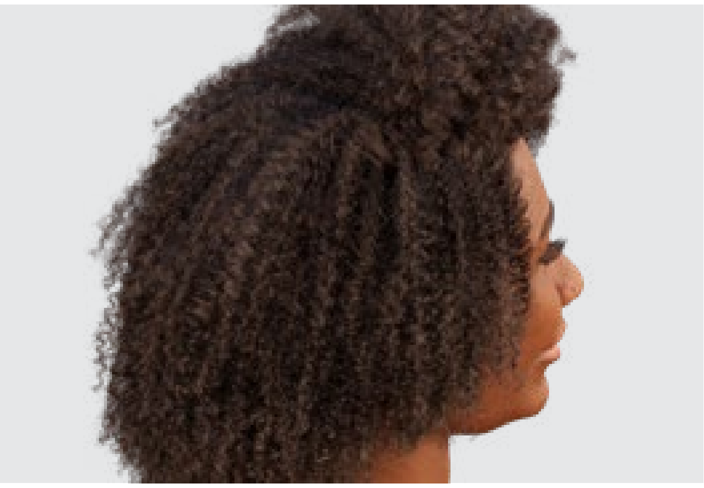
// Twist Out

3D Design

Curl Type Modeled: **4B**

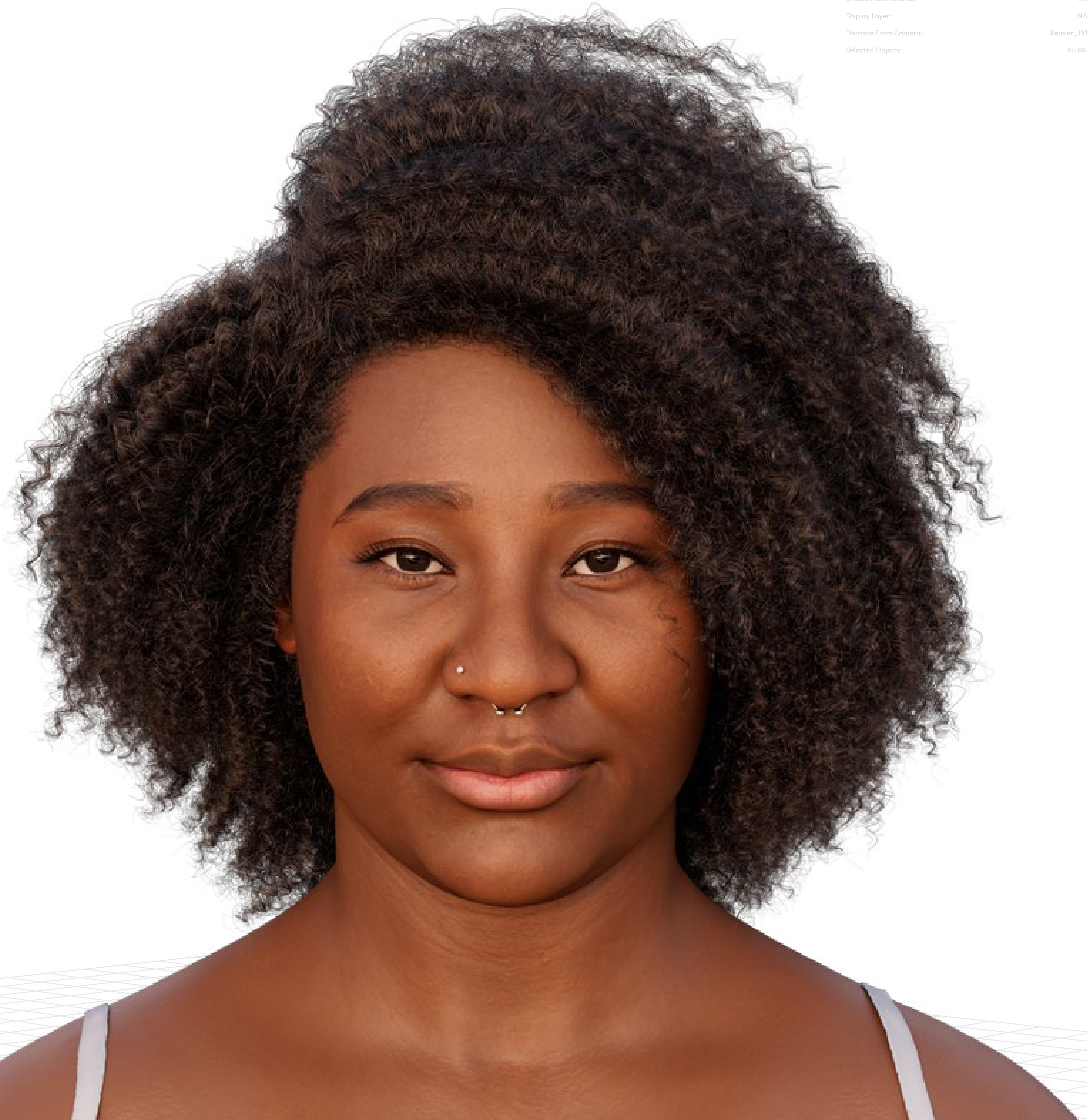


// 01



// 02

Download 3D Render



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

Project Settings

// Twist Out

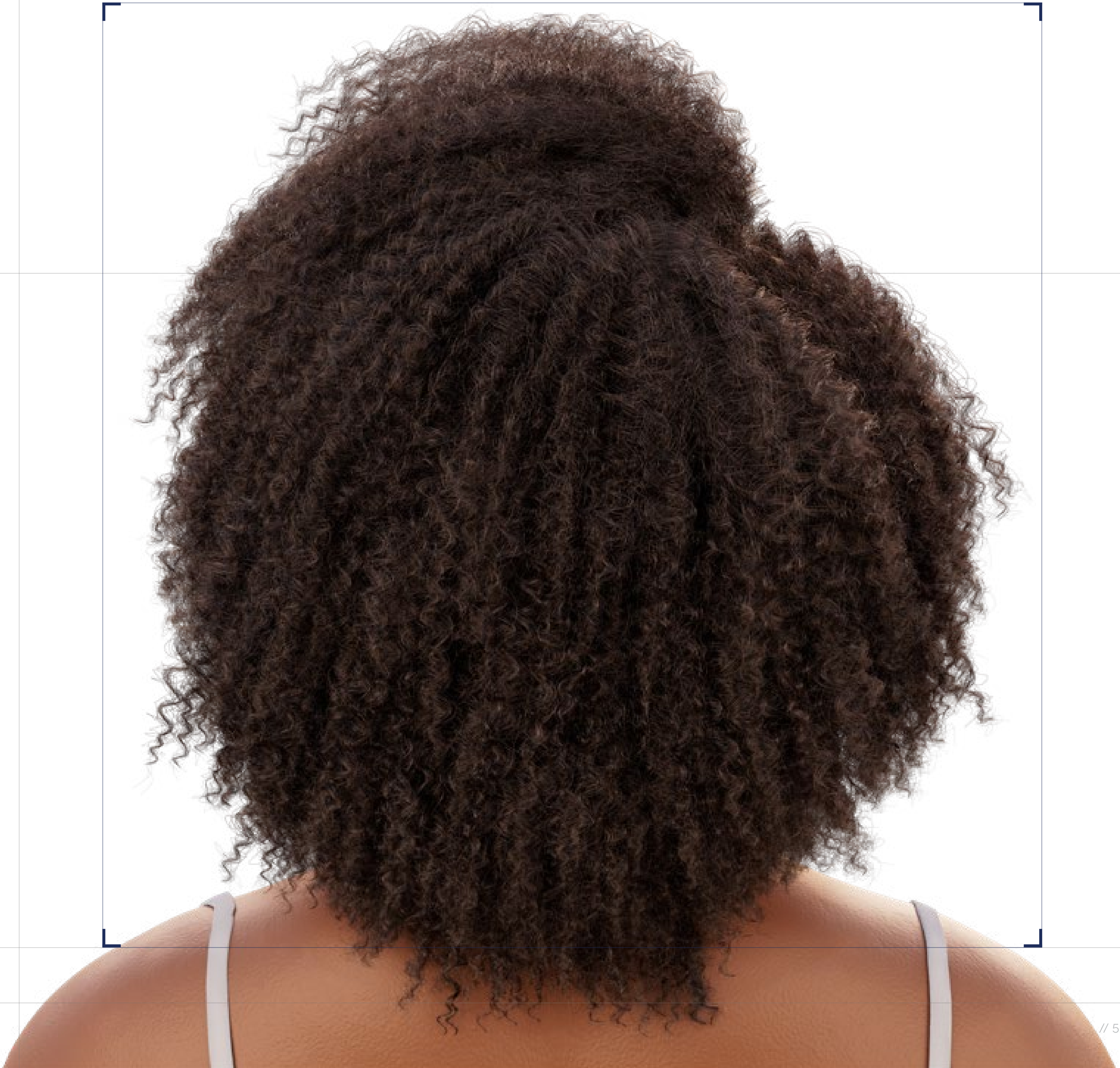
// Project Settings

Program Guide

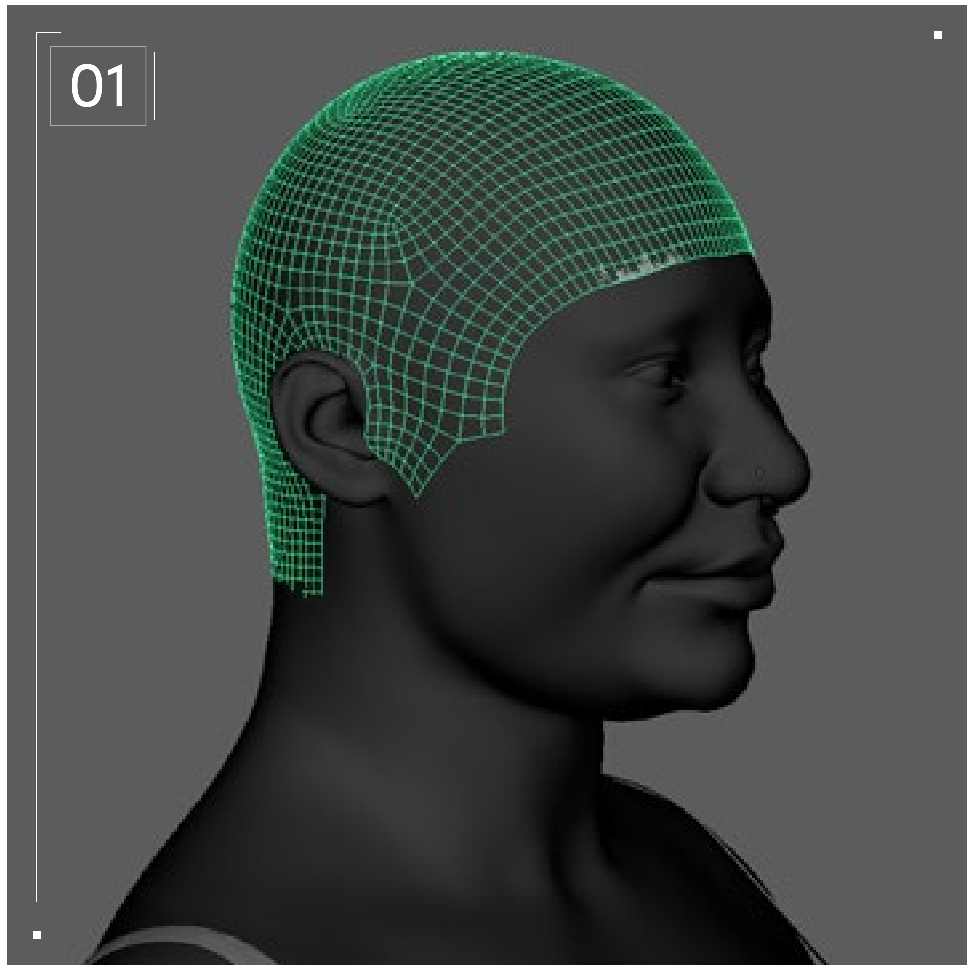
Software: Autodesk Maya 2022
Render Engine: Arnold
Additional Plug-Ins: XGen Interactive Grooming Toolset

Spirals are a common occurrence in nature. We see them in the growth of most organic things, whether it's plants, shells, bones, or hair. Natural hair is hair that is composed of many spiraling strands of hair. That's why Afros are often represented with cylindrical spirals in low poly games, and the reason why most 3D hair groom software has features that allow for the creation of spirals.

Natural curly hair creation is rather straightforward in hair creation programs like XGen, which already has a curl modifier that produces natural-looking spirals without much tweaking. For the Twist Out hairstyle, I had to "break" the clump modifier in order to achieve a flat wave look that showcases the hair strands in the shape of an "s."

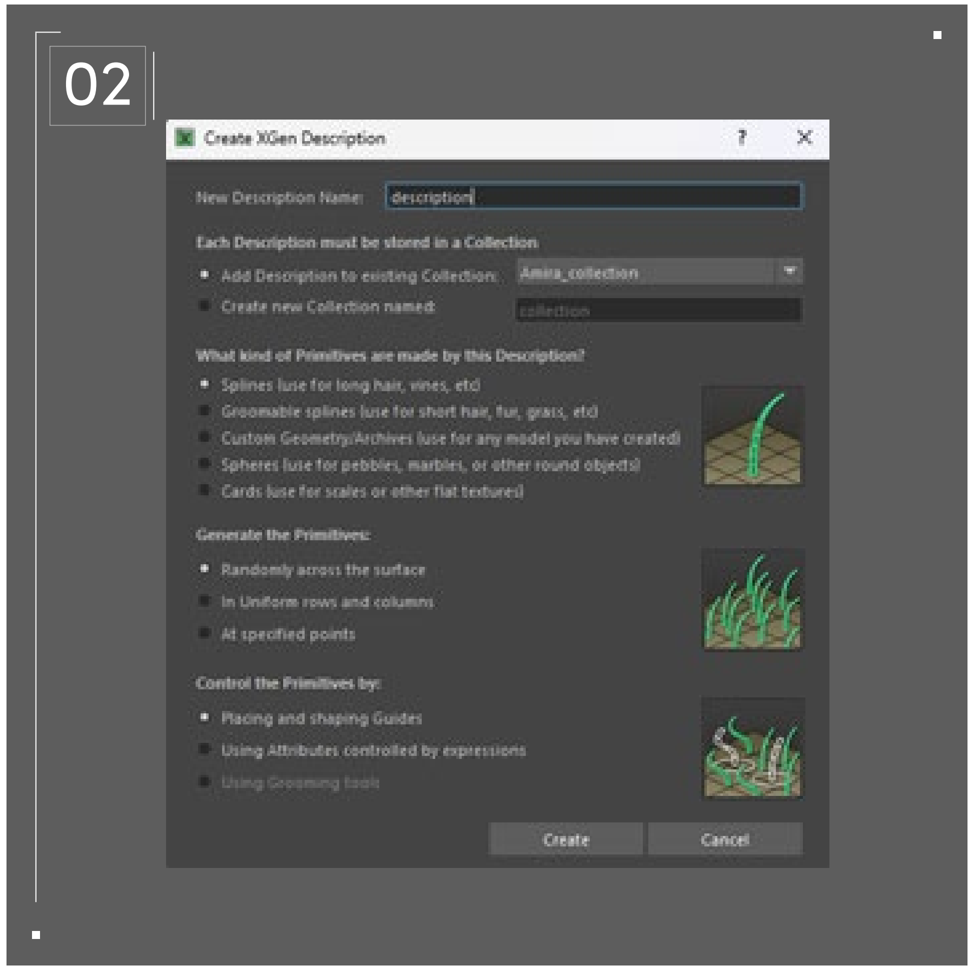


Step-by-Step Guide



// Geometry preparation

First, duplicate the head model, and manually select all the polygons from which you want the hair to grow. Select "inverse" and delete the remaining polygons. Make sure to keep the scalp geometry bigger than necessary, as you might underestimate how much space you need to not only grow the hair, but also, for example, to add baby hairs, which are short, wispy hairs that sit around the hairline and are sculpted into curlicues or other designs to add detail to a style.



Create XGen Description

New Description Name: description

Each Description must be stored in a Collection

- ● Add Description to existing Collection: Amira_collection
- ○ Create new Collection named: collection

What kind of Primitives are made by this Description?

- ● Splines (use for long hair, vines, etc)
- ○ Groomable splines (use for short hair, fur, grass, etc)
- ○ Custom Geometry/Archives (use for any model you have created)
- ○ Spheres (use for pebbles, marbles, or other round objects)
- ○ Cards (use for scales or other flat textures)

Generate the Primitives:

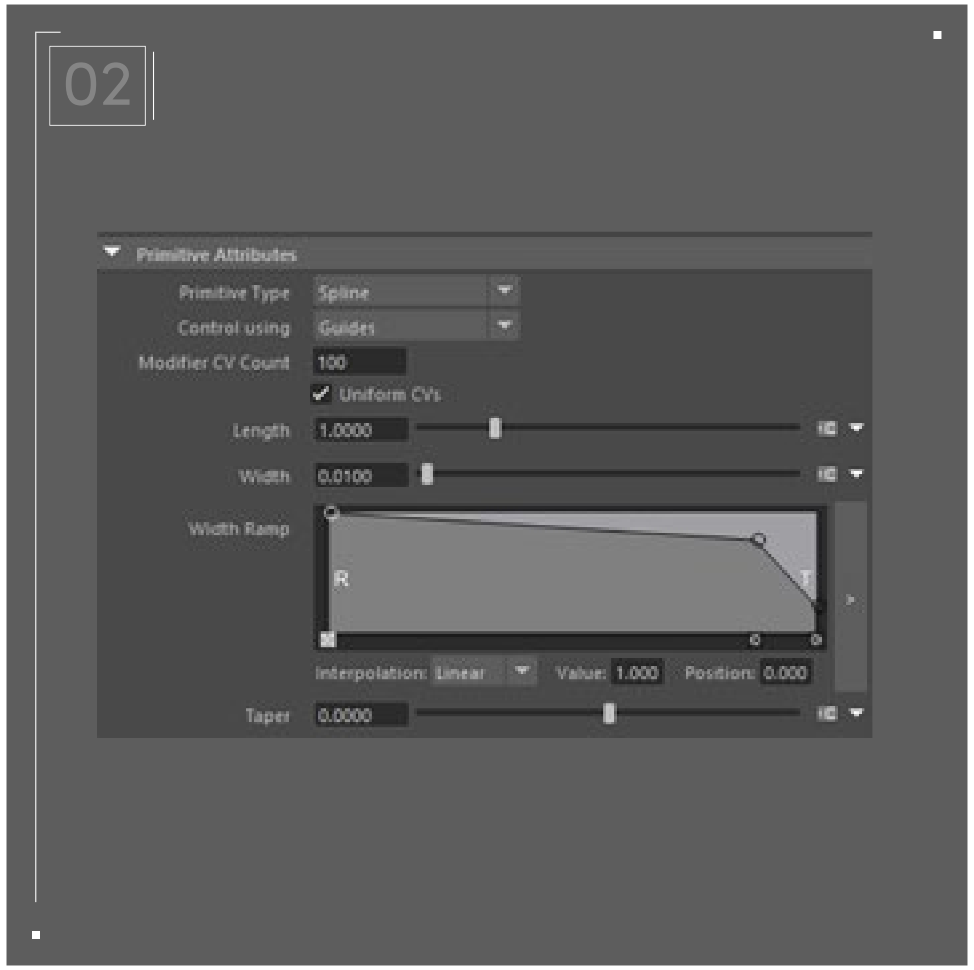
- ● Randomly across the surface
- ○ In Uniform rows and columns
- ○ At specified points

Control the Primitives by:

- ● Placing and shaping Guides
- ○ Using Attributes controlled by expressions
- ○ Using Grooming tools

// XGen description

Next, apply a new lambert material to the scalp geo. Open XGen editor and create a new description and a collection. Under "Control the Primitives by," select "Placing and shaping guides." This ensures that you will be placing and shaping guides manually, which gives you enough control to follow your reference.



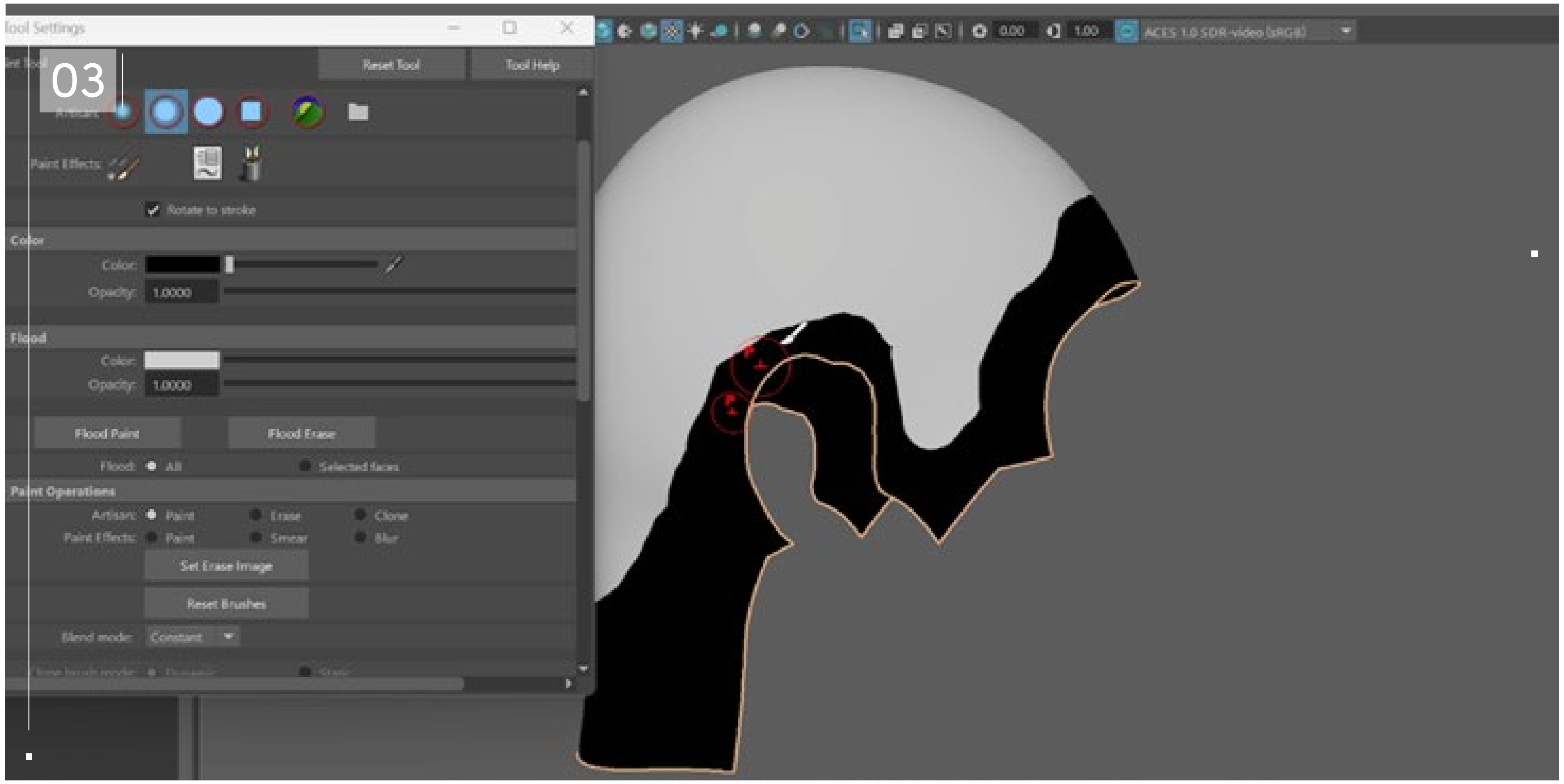
Primitive Attributes

Primitive Type: Spline
Control using: Guides
Modifier CV Count: 100
☑ Uniform CVs
Length: 1.0000
Width: 0.0100
Width Ramp
Interpolation: Linear   Value: 1.000   Position: 0.000
Taper: 0.0000

Then, under "Primitive Attributes," go to the width ramp and lower the value of the "Tip" side of the graph, so that the hair strands taper off towards the tip. Also set the "modifier CV count" to something like 50, to ensure there are enough vertices on the hair strands in order for them to curve.

Set Density to 100. You can increase and decrease after the modifiers are applied to find the density that you want.

Step-by-Step Guide



03

Tool Settings

☑ Rotate to stroke

Color

Color: ■

Opacity: 1.0000

Flood

Color: □

Opacity: 1.0000

Flood: ● All ○ Selected faces

Paint Operations

Artisan: ● Paint ○ Erase ○ Clone

Paint Effects: ○ Paint ○ Smear ○ Blur

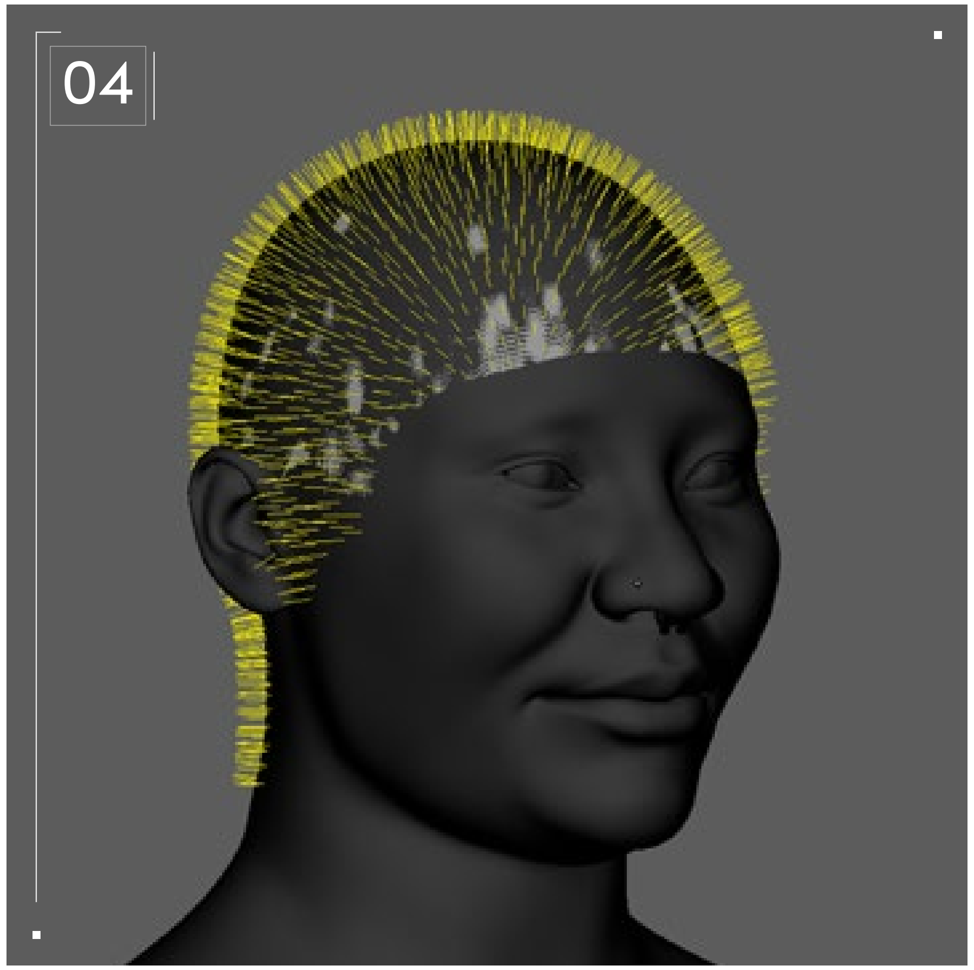
Blend mode: Constant

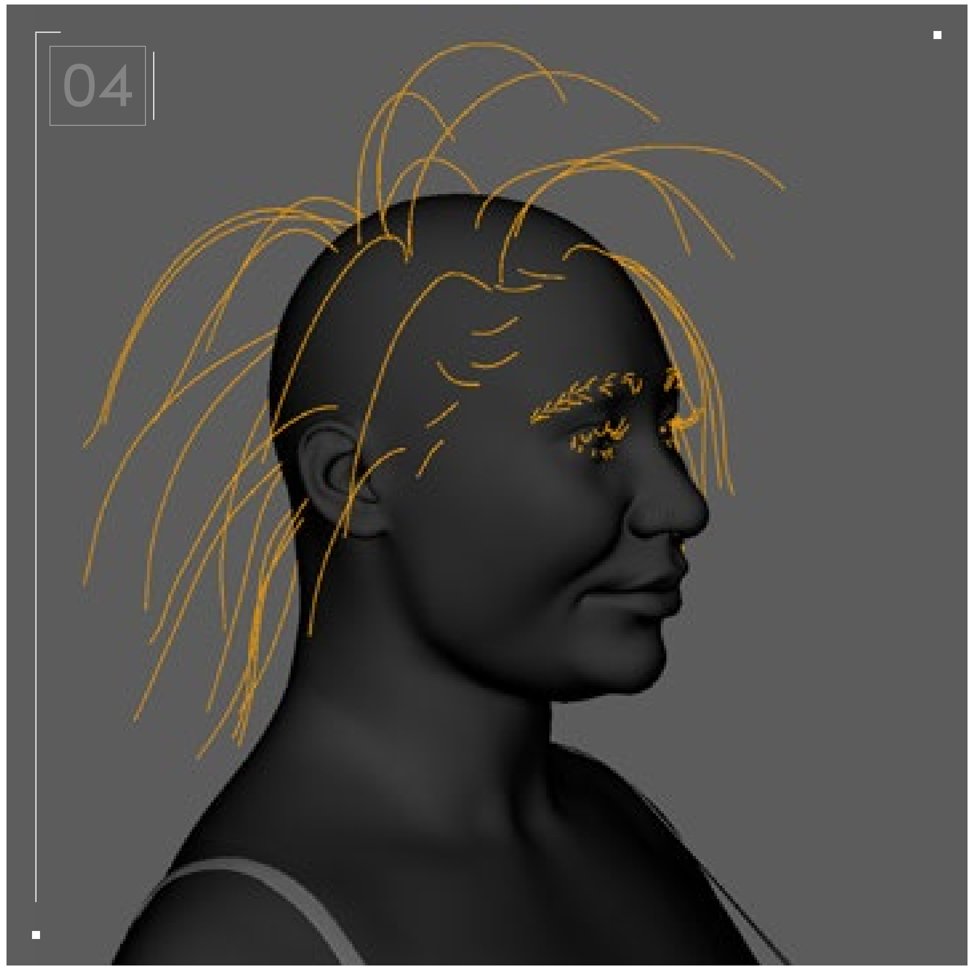
// Paint mask

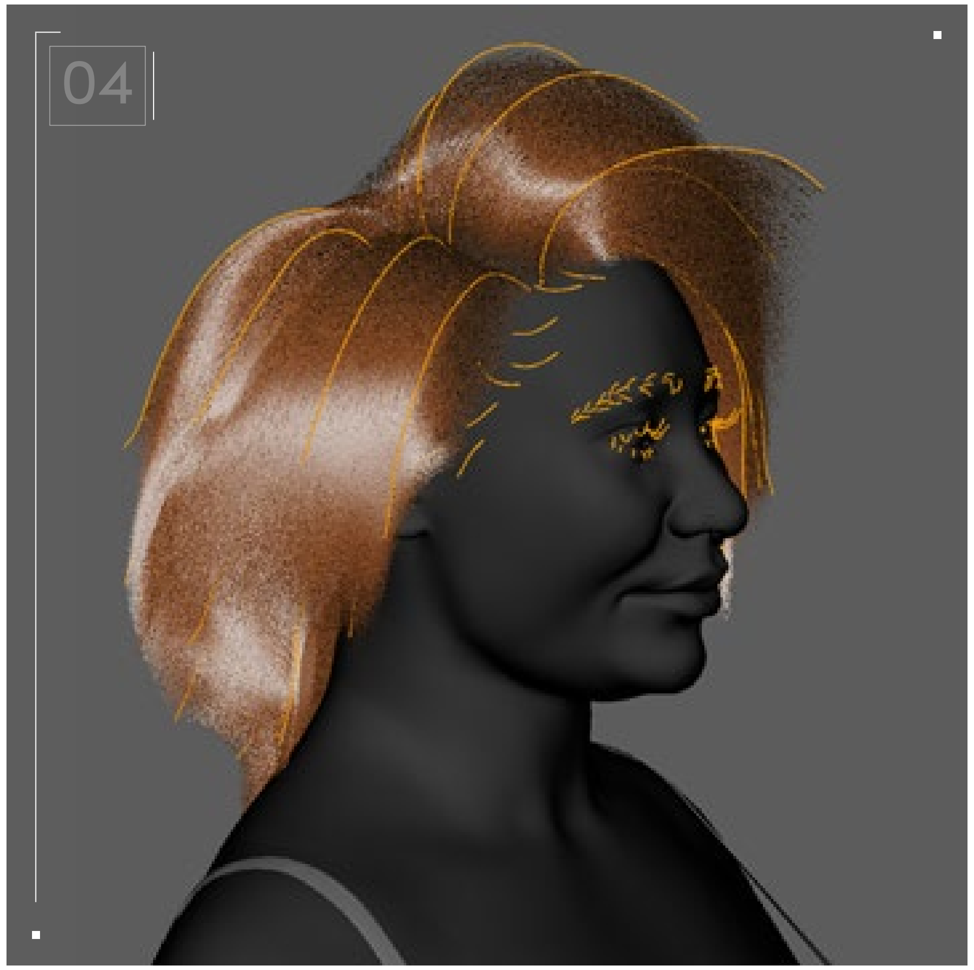
Go to the XGen editor for your hair description. Next to the mask, click "Create Map..." This allows you to create a paint mask to determine where the hair strands will grow from the geometry. Set the map resolution to 25 and keep it at white.

Open the brush tool settings and set the blend mode to "constant," then scroll down to "stroke" and select "reflection" to apply brush strokes symmetrically. The XGen will grow from all the areas that are white, so pick a black value for the brush color and paint all the areas where you don't want the hair to grow from. Come back and iterate on this mask throughout the groom process; you might need to paint the mask more unsymmetrically or soften the transitions between black and white.

Step-by-Step







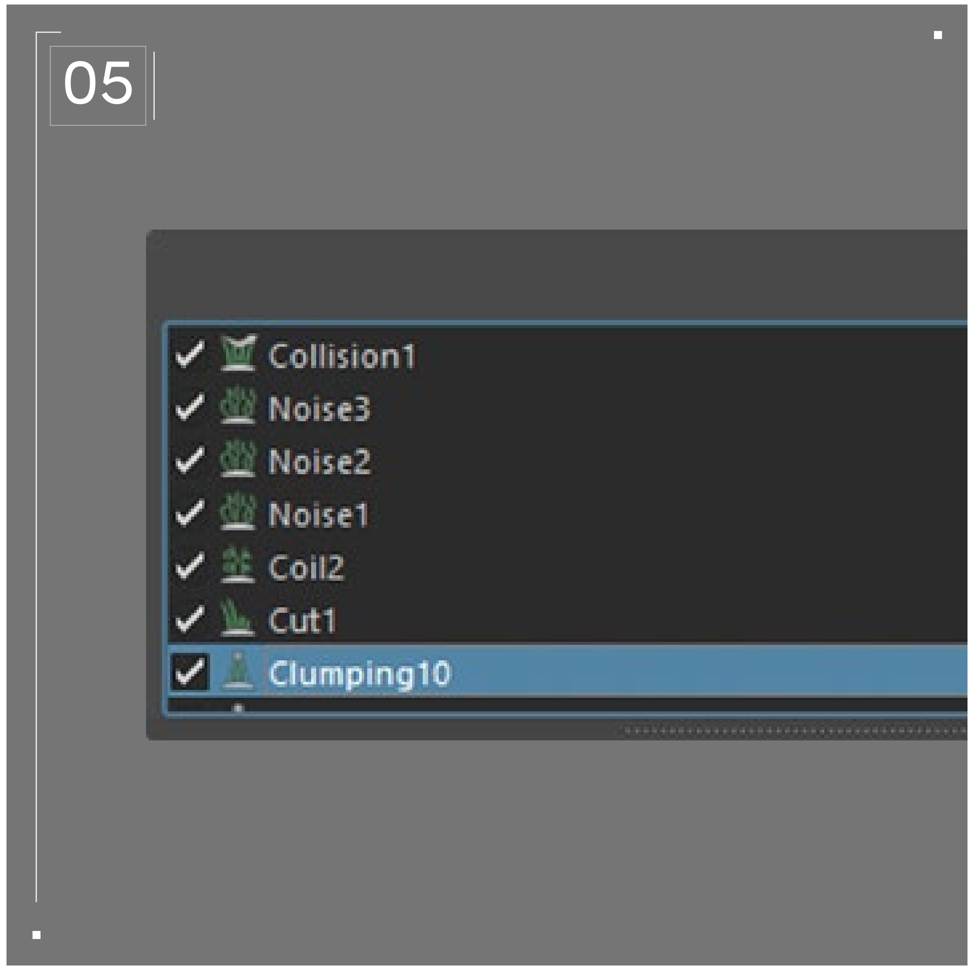
// Guides

Before starting to place the guides, I recommend that you create a mesh that reflects the overall volume and shape of the hairstyle you're going for.

Once that's done, start by placing one guide on the scalp geo. Scale and modify the shape of this guide with the available XGen tools and rebuild the guide with higher CV count to allow for more detail/blend. Once the guide is placed and shaped correctly, select it and go to the XGen utilities window and click "copy guides." Now you can place new guides and paste the first one into them and adjust accordingly to fit the shape of the hair.

Increase the hair density in "Primitive Attributes" to preview the volume of the hair.

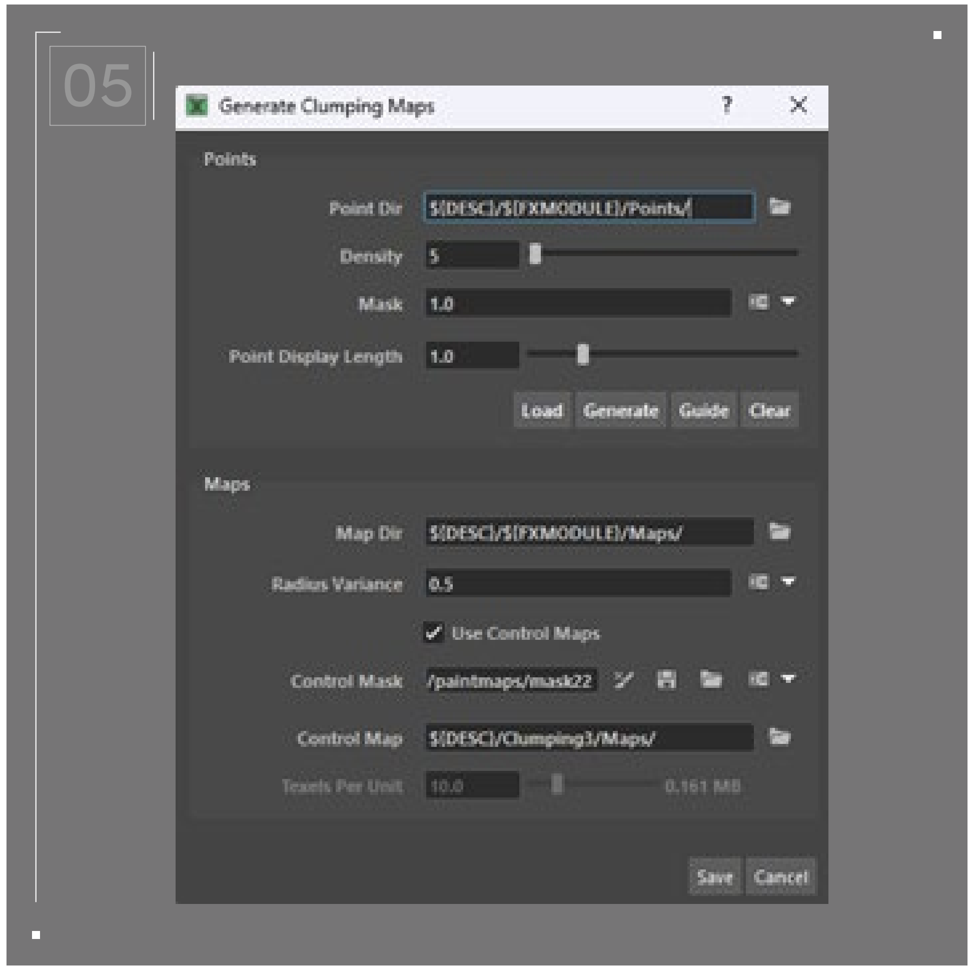
Step-by-Step



✓ Collision1
✓ Noise3
✓ Noise2
✓ Noise1
✓ Coil2
✓ Cut1
✓ Clumping10

// Modifiers

Next, you're going to want to add the following modifiers, in this order:



Generate Clumping Maps

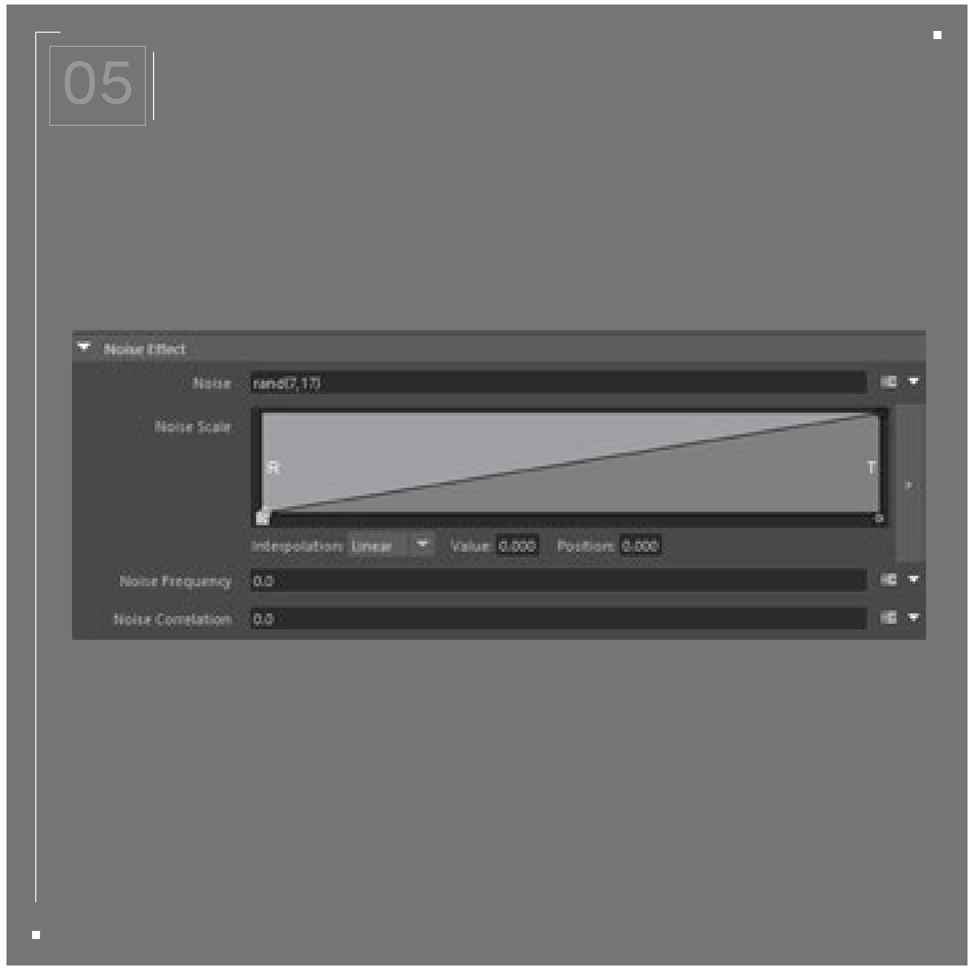
Points

- Point Dir: ${DESC}/${FXMODULE}/Points/
- Density: 5
- Mask: 1.0
- Point Display Length: 1.0

Maps

- Map Dir: ${DESC}/${FXMODULE}/Maps/
- Radius Variance: 0.5
- ☑ Use Control Maps
- Control Mask: /paintmaps/mask22
- Control Map: ${DESC}/Clumping3/Maps/
- Texels Per Unit: 10.0 0.161 MB

Clumping: In the clumping modifier, scroll down to "setup maps," set "Density" to something like 5, click "Generate" to generate the clump guides and then save.

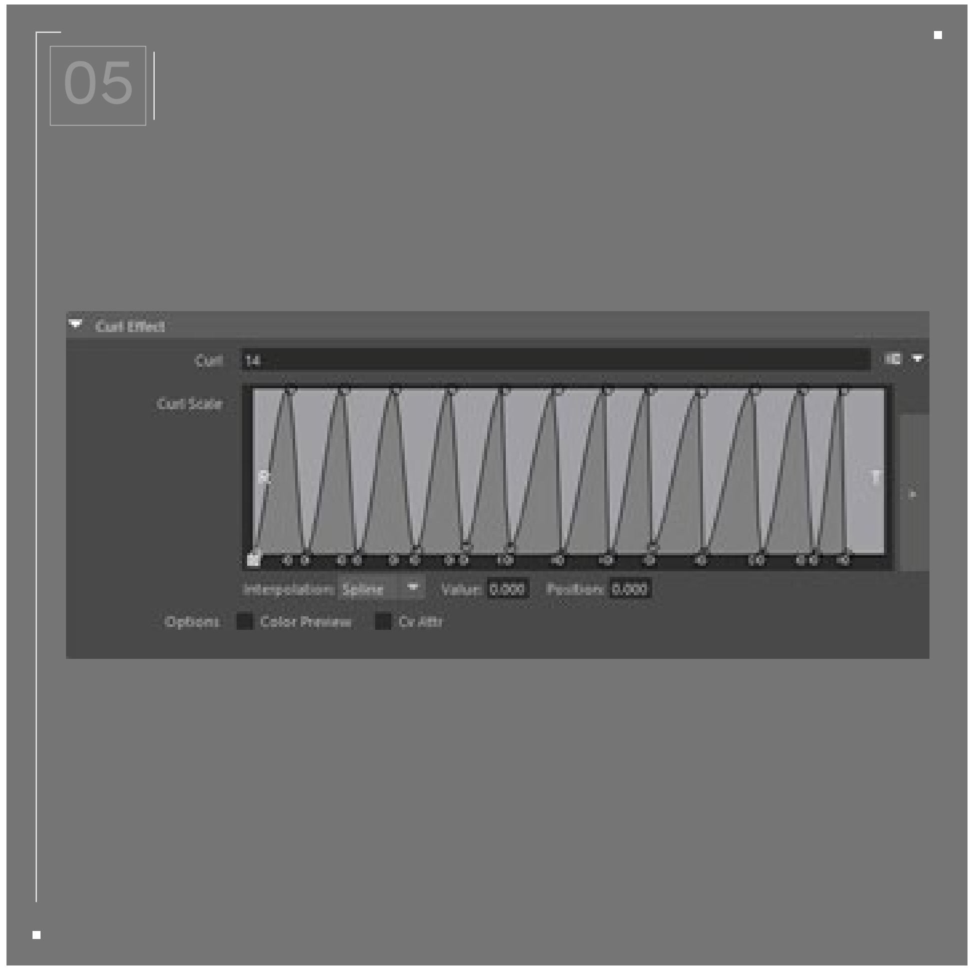


Noise Effect

- Noise: rand(7,17)
- Noise Scale
- Interpolation: Linear Value: 0.000 Position: 0.000
- Noise Frequency: 0.0
- Noise Correlation: 0.0

Set the Noise Effect to rand (7,17).

Step-by-Step



Curl Effect

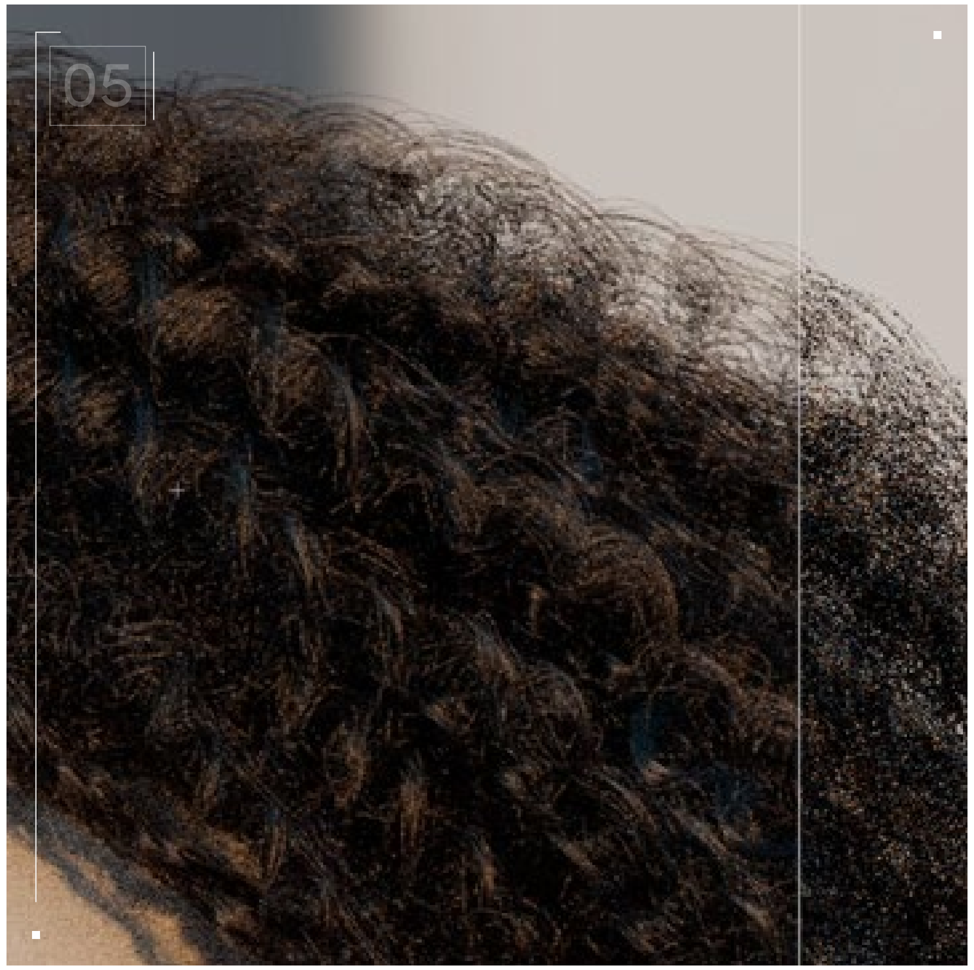
Curl: 14

Curl Scale

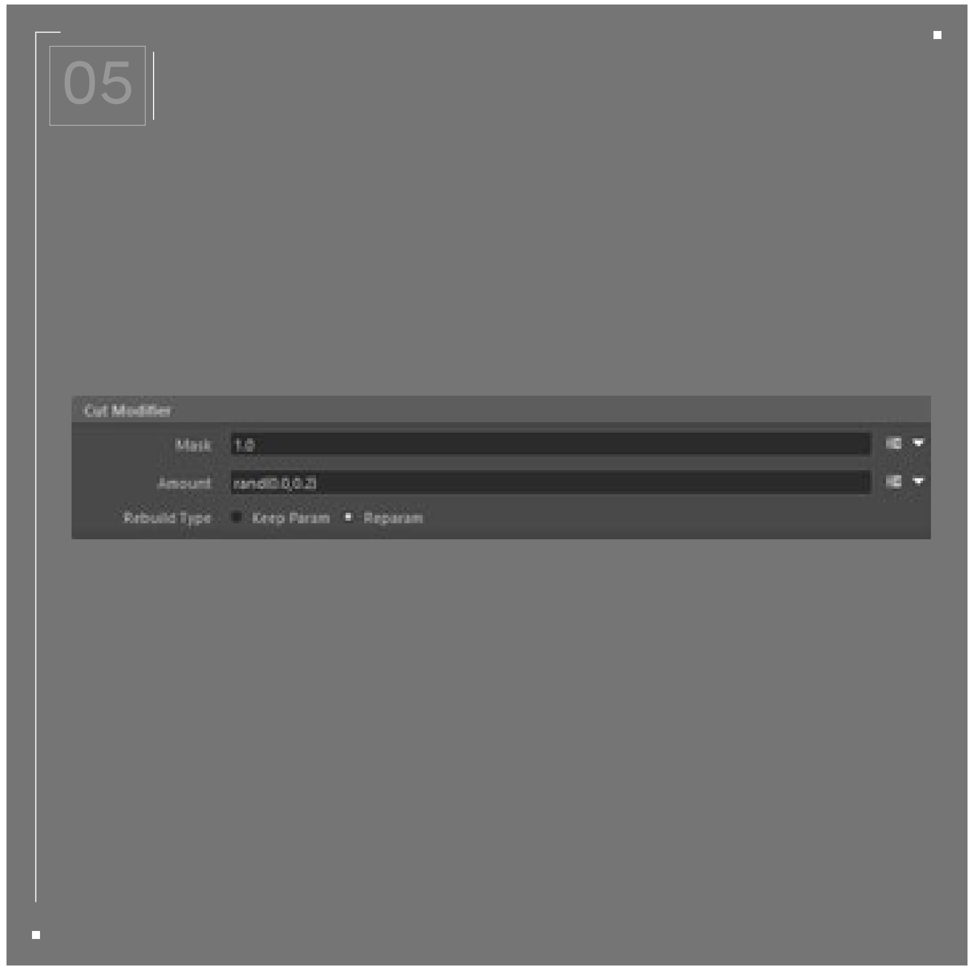
Interpolation: Spline Value: 0.000 Position: 0.000

Options □ Color Preview □ Cv Attr

We can tweak the clump modifier by removing the information that causes the hair spirals to move circularly. This will make the hair strand shape look flat and wavy along its length. In the curl modifier, I set a high number of curls, like 14. Then in the curl expression graph editor, I created 25 points that influence the strength of the spirals. Since the frequency of curls and the number of points are equal, each point should influence one curl. By changing the value of these influence points, we can counteract the hair spirals, in order to achieve a flat wavy look.



Once this modifier is applied, the hair, which was once curly, should now look like a collection of waves.



Cut Modifier

Mask: 1.0

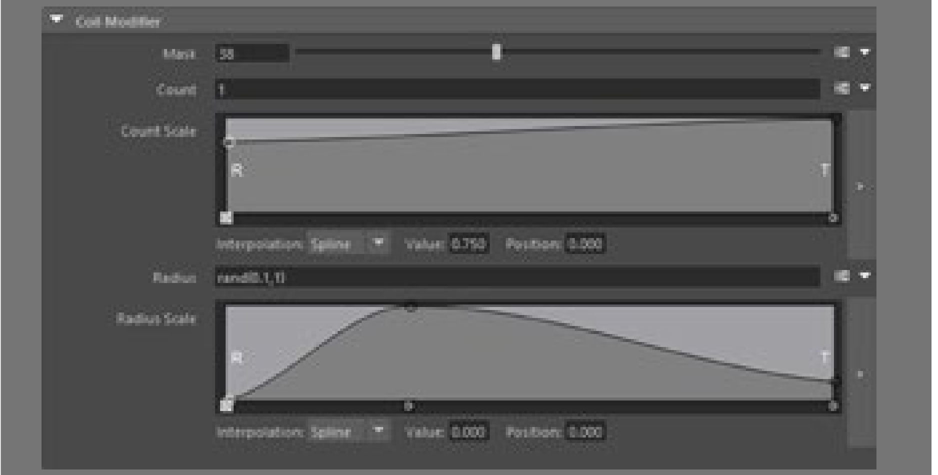
Amount: rand(0.0,0.2)

Rebuild Type ○ Keep Param ● Reparam

Cut.

Step-by-Step

05



Coil Modifier

Mask 58

Count 1

Count Scale
Interpolation: Spline Value: 0.750 Position: 0.000

Radius rand(0.1,1)

Radius Scale
Interpolation: Spline Value: 0.000 Position: 0.000

You can keep the default settings or modify to your liking. Coil.

05

Make a noise modifier and click on the expression editor next to the mask. Add the following:

```
$a=map('${DESC}/paintmaps/Main_Clump');#3dpaint,5.0
$percentStray=87; #0,100
$b=rand() < $percentStray/100.0 ? 1 : 0;
$b
```

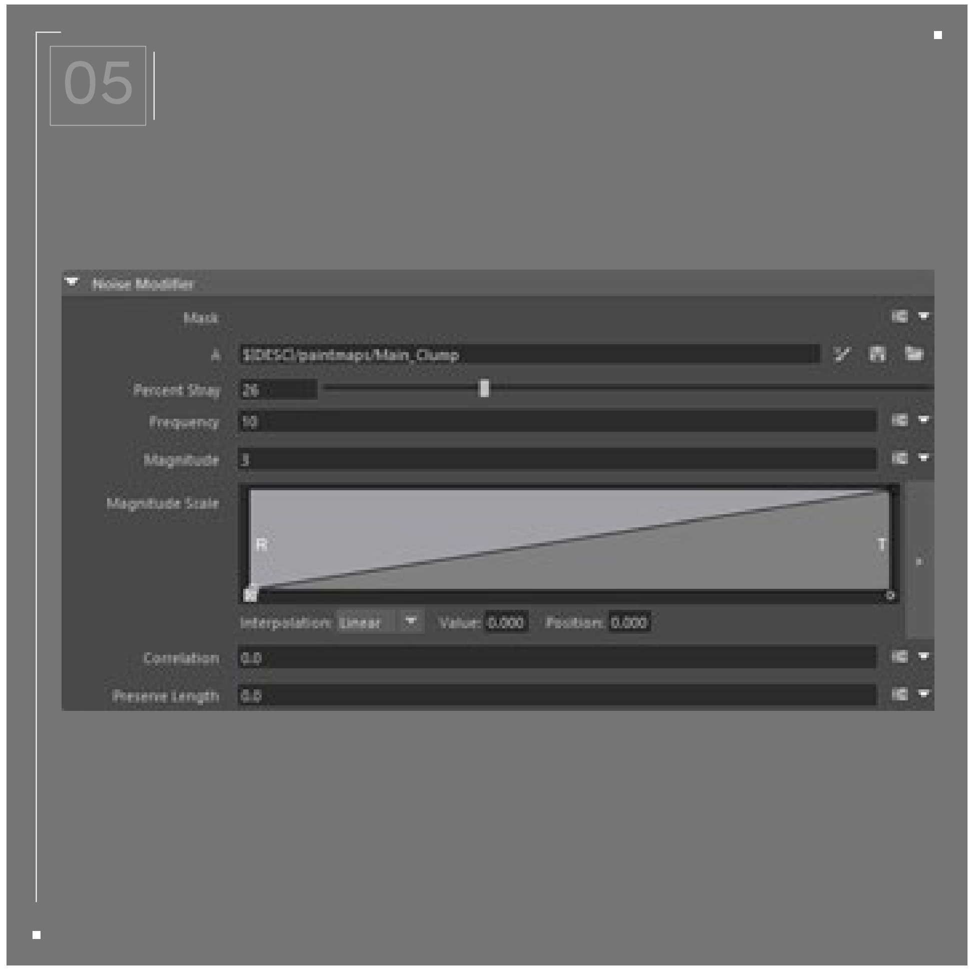
This adds noise to your mask, which adds natural variation to the noise strength. Use the mask slider to decide the percentage of hairs that are influenced by the noise modifier.

05

Duplicate this noise modifier so you have three in total. Each modifier will represent disorder in the hair root, hair tip, and the hair overall. Use the Magnitude Scale to dictate where each respective noise modifier will influence the hair.

// Project Settings

Step-by-Step



Mask

A $(DESC)/paintmaps/Main_Clump

Percent Stray 26

Frequency 10

Magnitude 3

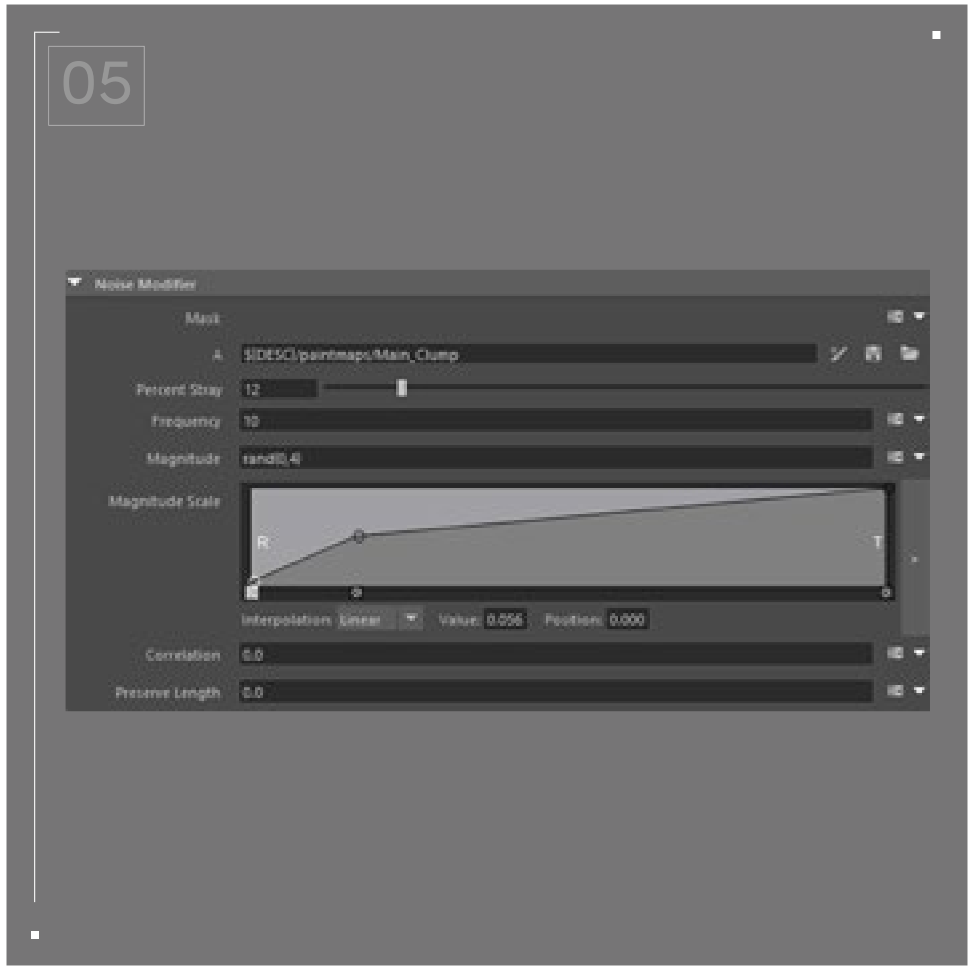
Magnitude Scale

Interpolation: Linear Value: 0.000 Position: 0.000

Correlation 0.0

Preserve Length 0.0

Noise 1: Tip/root



Mask

A $(DESC)/paintmaps/Main_Clump

Percent Stray 12

Frequency 10

Magnitude rand(0,4)

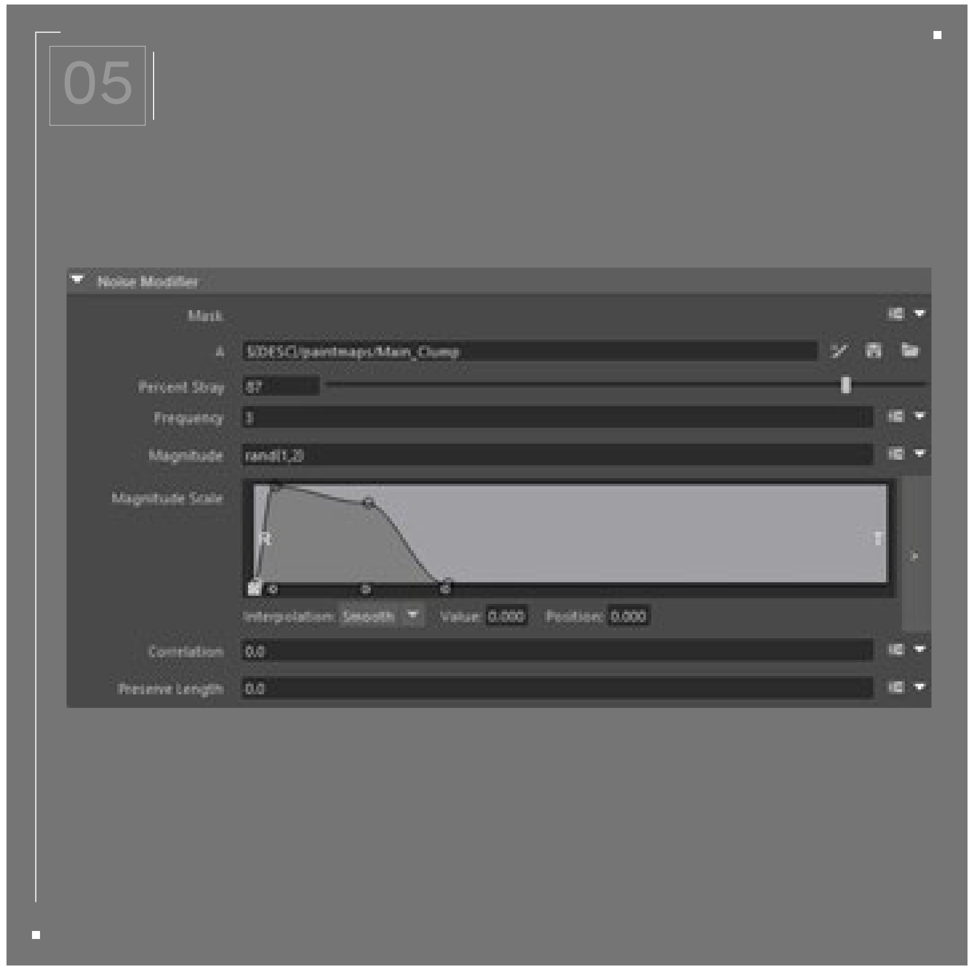
Magnitude Scale

Interpolation: Linear Value: 0.056 Position: 0.000

Correlation 0.0

Preserve Length 0.0

Noise 2: General



Mask

A $(DESC)/paintmaps/Main_Clump

Percent Stray 87

Frequency 3

Magnitude rand(1,2)

Magnitude Scale

Interpolation: Smooth Value: 0.000 Position: 0.000

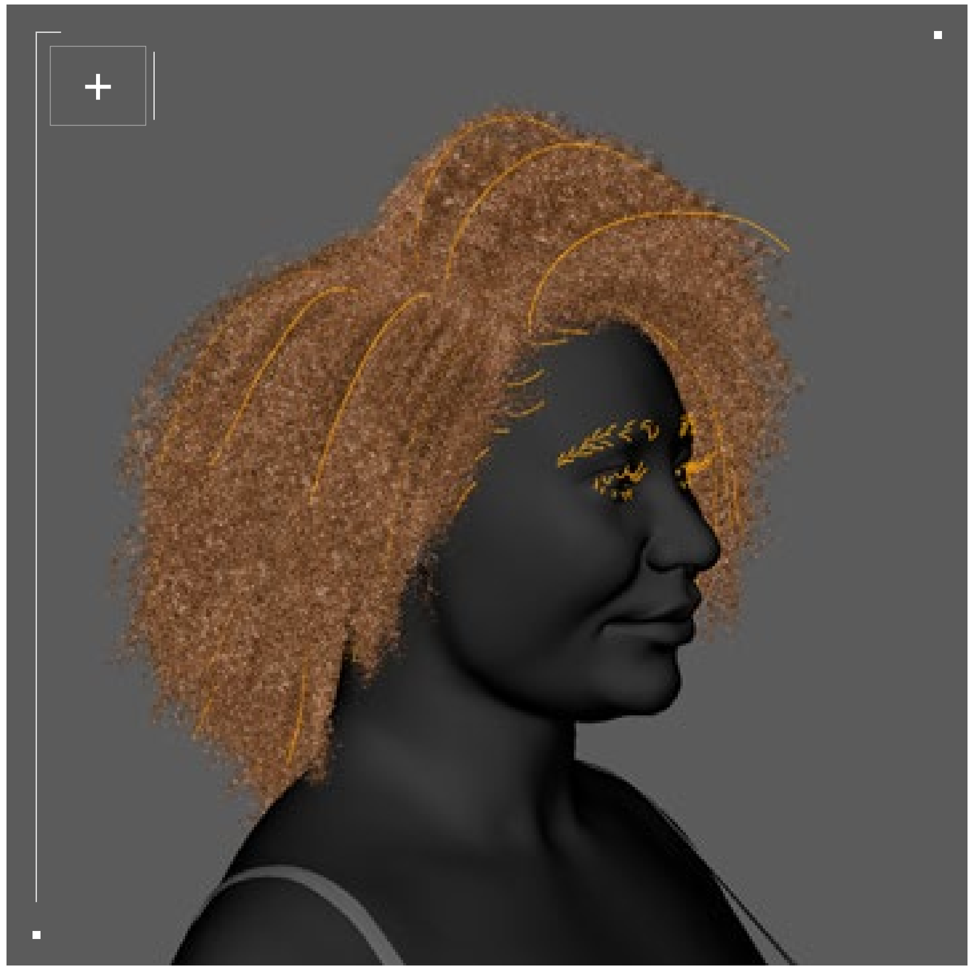
Correlation 0.0

Preserve Length 0.0

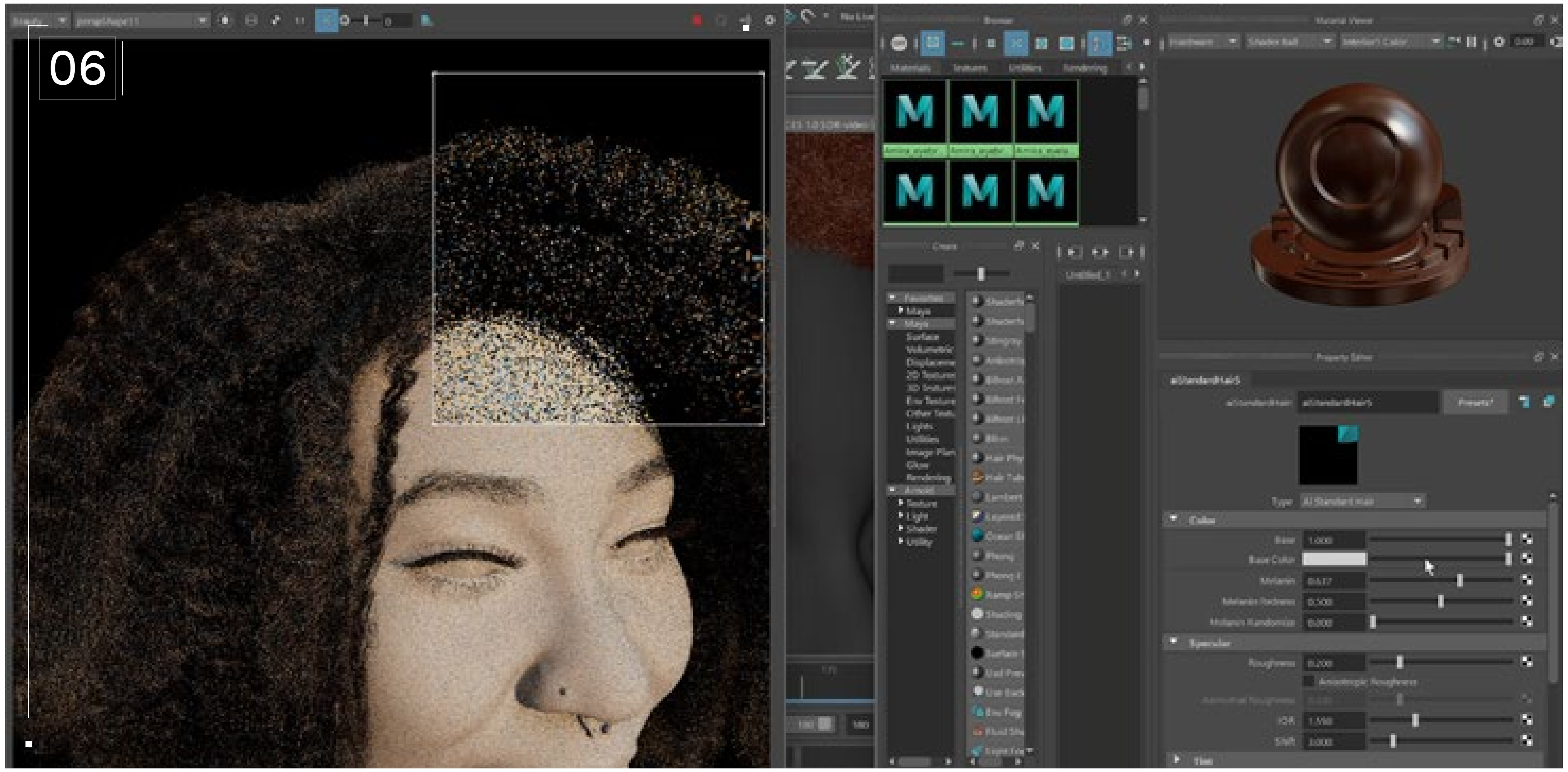
Noise 3: Root

// Project Settings

Step-by-Step



Collision: Export the head mesh as an alembic file, then import it as the collision mesh in the collision modifier. This ensures the hair won't clip through the character.



aiStandardHair: aiStandardHair5
Type: AI Standard Hair
Base: 1.000
Melanin: 0.617
Melanin Redness: 0.500
Melanin Randomize: 0.000
Roughness: 0.200
IOR: 1.550
Shift: 3.000

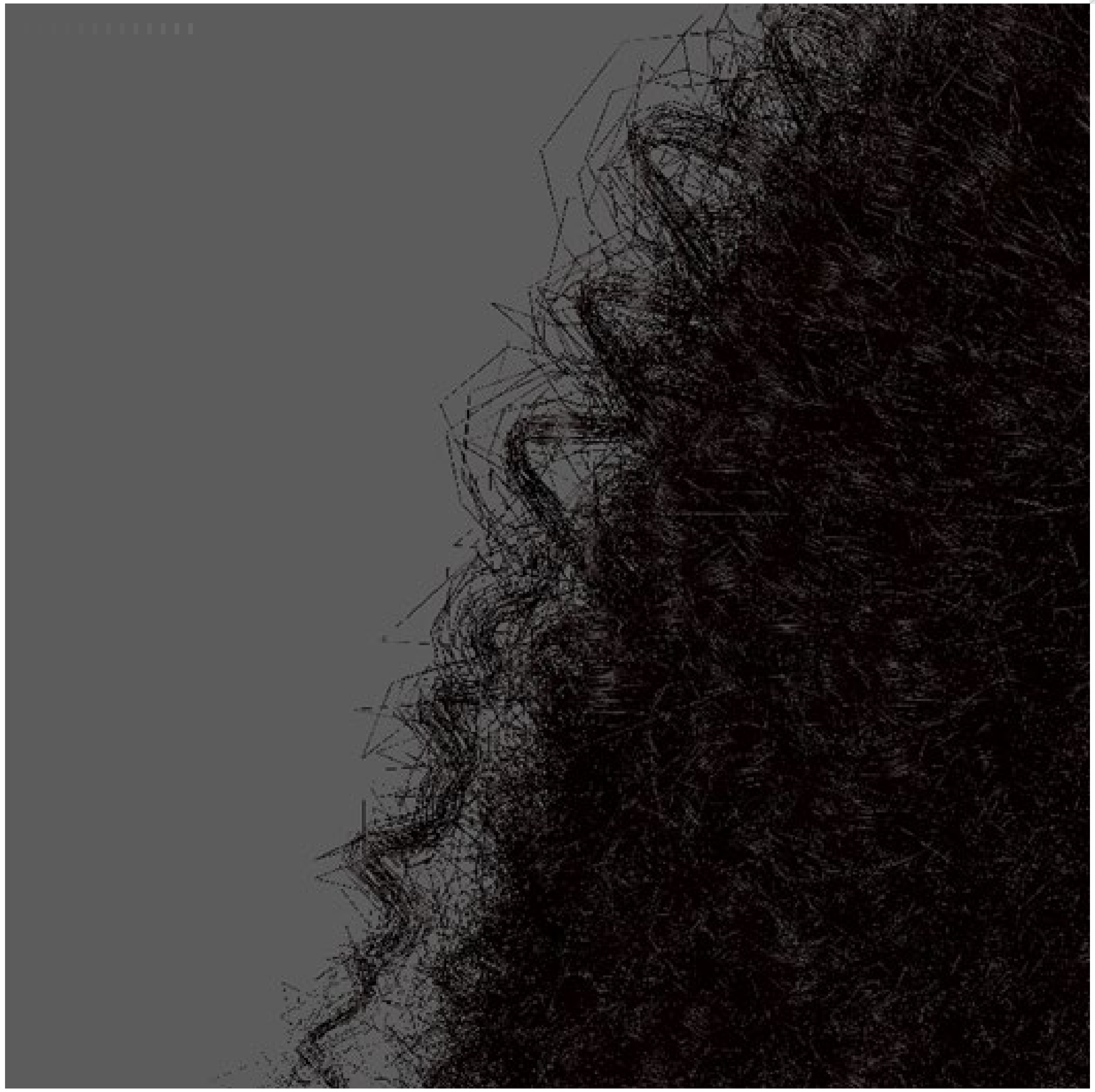
// Hair material

Select the guides in the outliner, go to "Hypergraph" and assign an Ai_standard_hair material to the XGen description. This should apply the material to the hair strands. Next, adjust the melanin count, but don't touch the base color as we want to be working with realistic and natural colors for the hair.

Natural hair has a more coarse texture to it; this is partly due to the fact that the bends in natural hair create a coil or curl and have a different angle when compared to other hair types. This difference is apparent enough that it is visible to the human eye. We can present this difference in bends through our Aistandard_hair material by going to the IOR section and setting the "Shift" value to 2.3. Continuously test your material by rendering the scene and iterate until it matches your hair reference.

Alternative Hair Texture

Most of the effects that we see as indicative of natural hair in XGen come from the clumping modifiers and their respective "Curl Effects." To adapt to other hair textures, play around with the curl frequency and the noise modifier parameters. Also try increasing the number of curl guides for smaller, tighter curls. This may require you to increase your strand CV count in the XGen "Primitive Attributes."



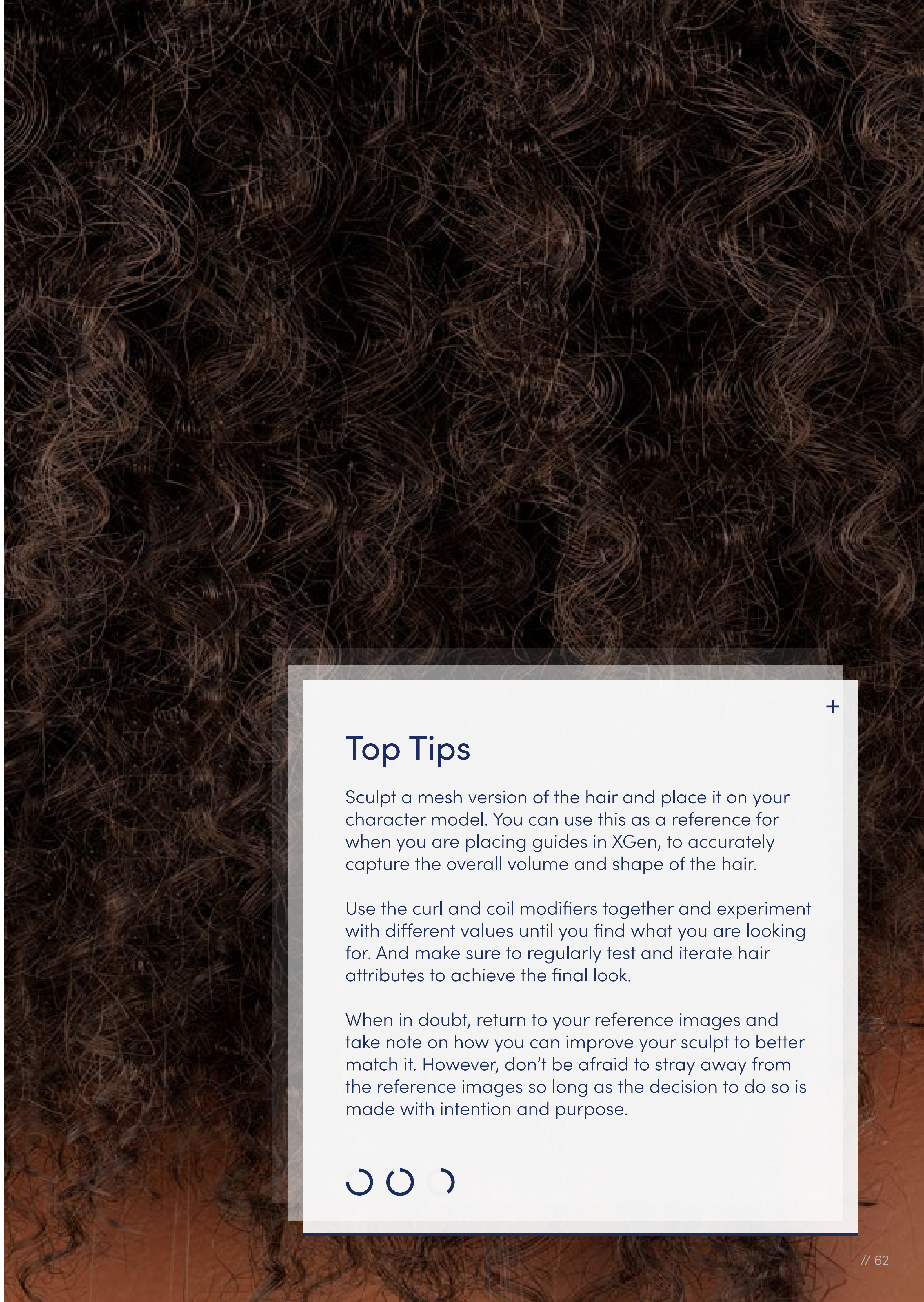
// Project Settings

Technical Watchouts

Set the Maya project file before creating anything XGen related. Never use spaces in your naming conventions as it will lead to the loss of your XGen project. XGen will fail to reference any resource that has a space in its location directory or name. Just use underscore "_" instead.

Attaching your XGen description directly to a character model with a rig can result in problems when the character is animated. To avoid this, use a scalp geometry that is either parented to the head joint or proximity wrapped to the character model. Ensure the sculpt geometry has a lambert material attached to it before an XGen description is attached — it's necessary for XGen to generate hair splines.

Finally, curly strand-based hair requires high vertex counts (VCs) and can be demanding for your workstation. Consider the performance impact of high VC on your system resources and save frequently.



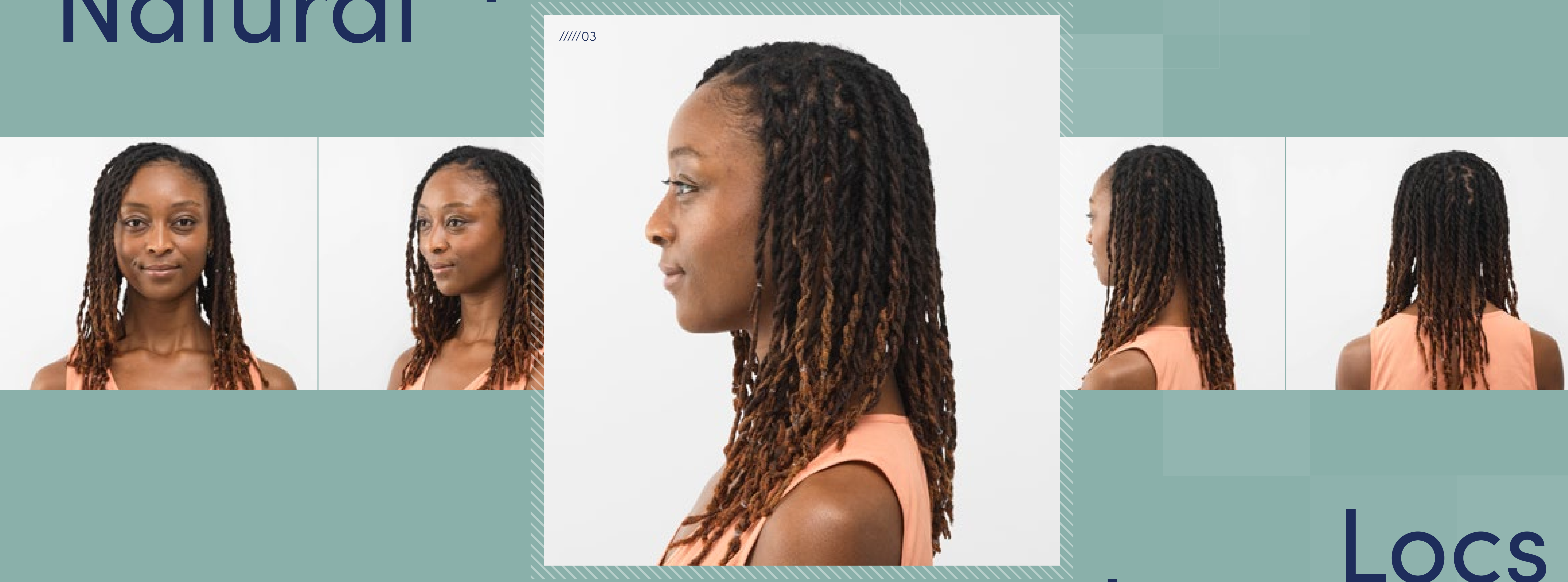
Top Tips

Sculpt a mesh version of the hair and place it on your character model. You can use this as a reference for when you are placing guides in XGen, to accurately capture the overall volume and shape of the hair.

Use the curl and coil modifiers together and experiment with different values until you find what you are looking for. And make sure to regularly test and iterate hair attributes to achieve the final look.

When in doubt, return to your reference images and take note on how you can improve your sculpt to better match it. However, don't be afraid to stray away from the reference images so long as the decision to do so is made with intention and purpose.

Natural
Natural
Natural



Locs

Download Images

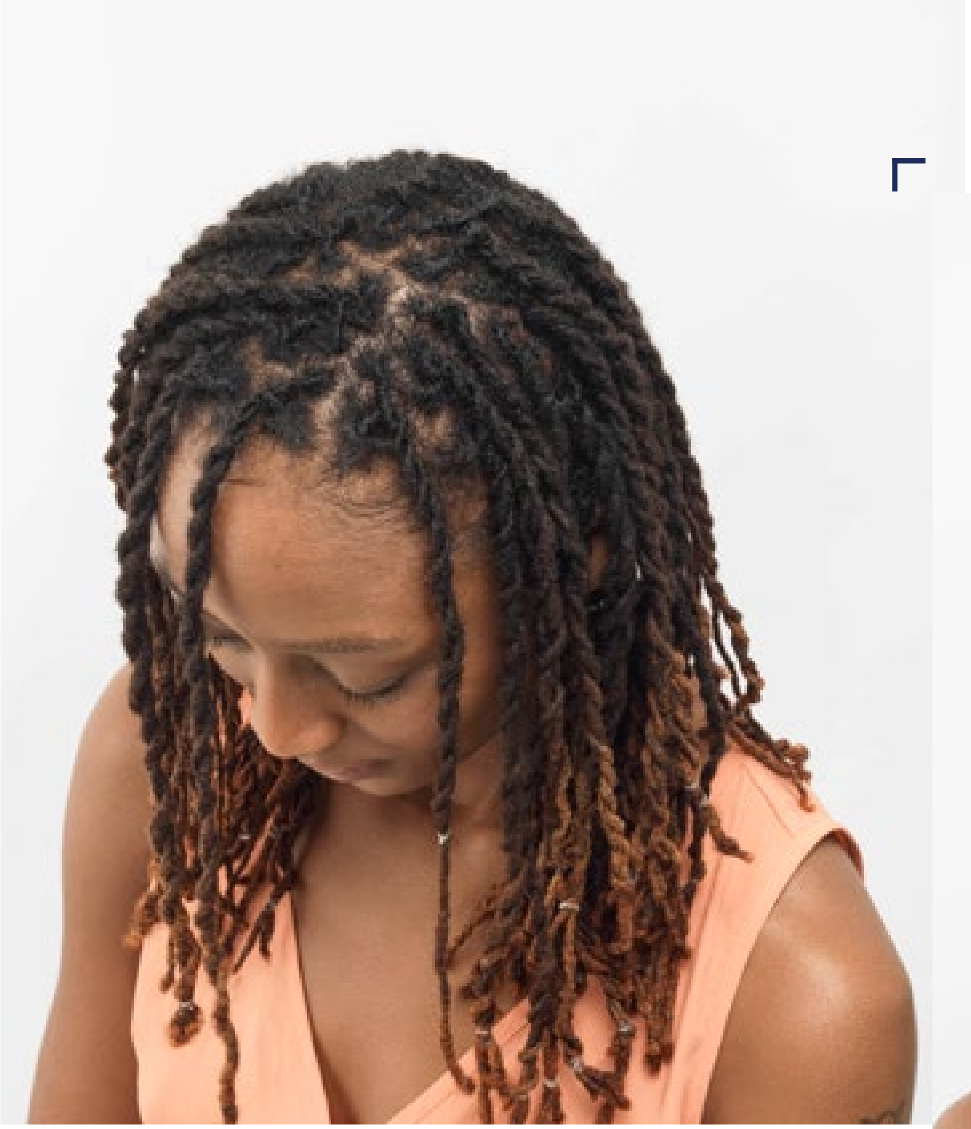
// Natural Locs

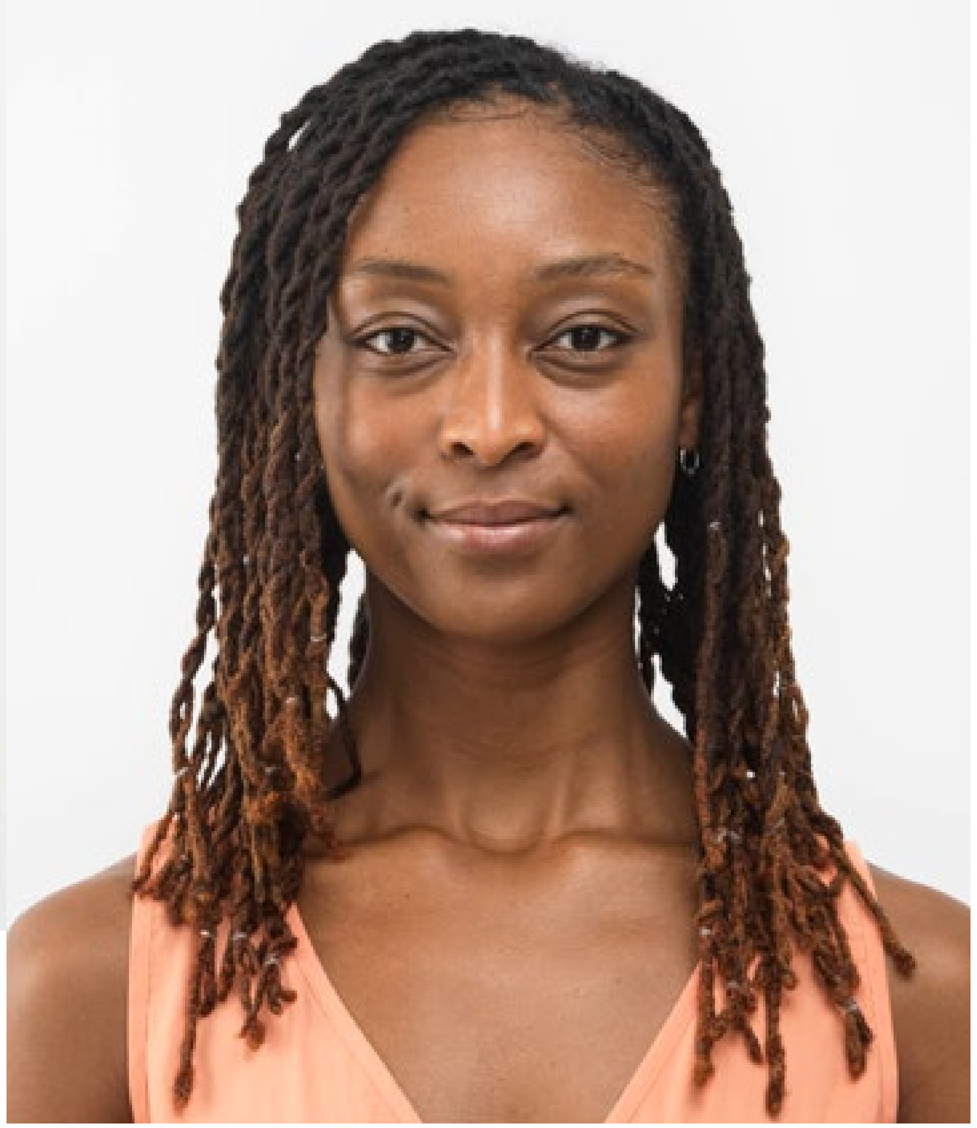
// Hairstyle

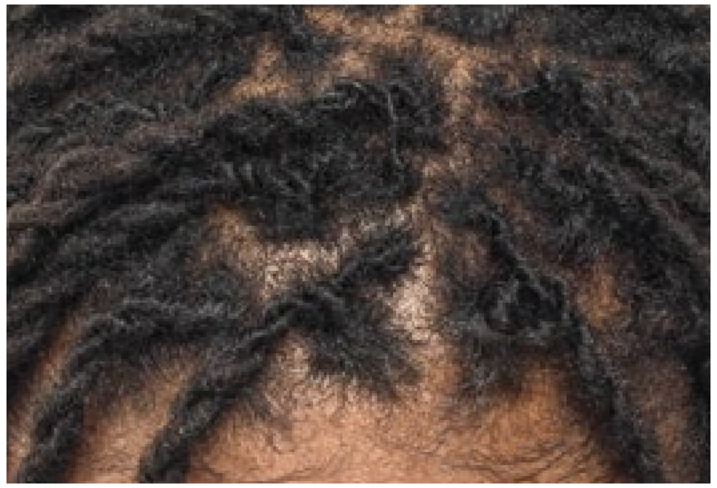
Natural Locs

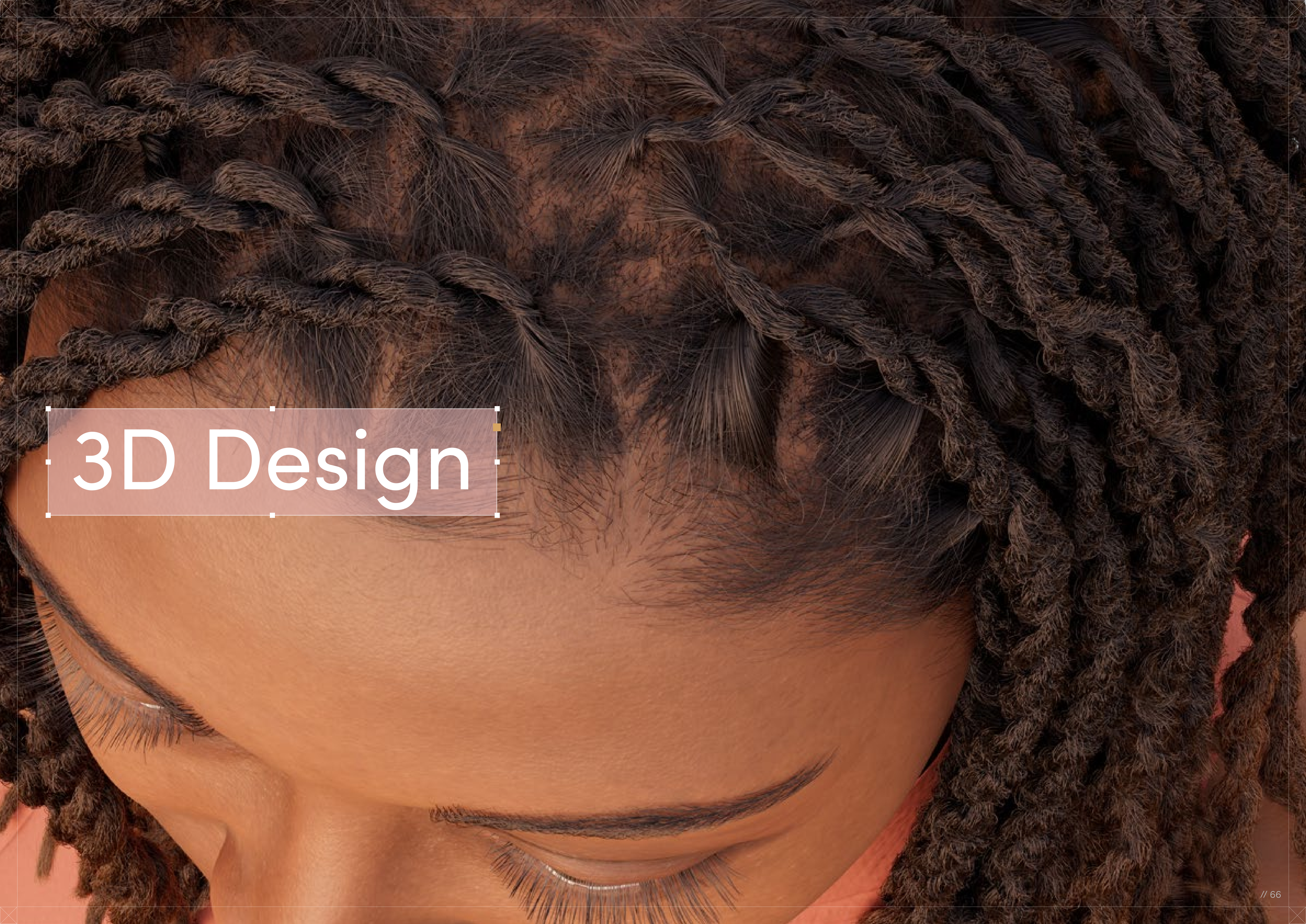
Style Background: Natural Locs are grown without the addition of synthetic hair. While any hair texture is capable of matting, the Locs that most people think of are the result of intentional styling techniques used by Black hairstylists, sometimes called locticians. There has been a resurgence in this popular hairstyle in recent years, including celebrities who are showcasing the versatility of Locs on the red carpet. Locs can be worn loose or in a variety of styles. The style represented in this guide are stylized Locs that are twisted together.

Download Images



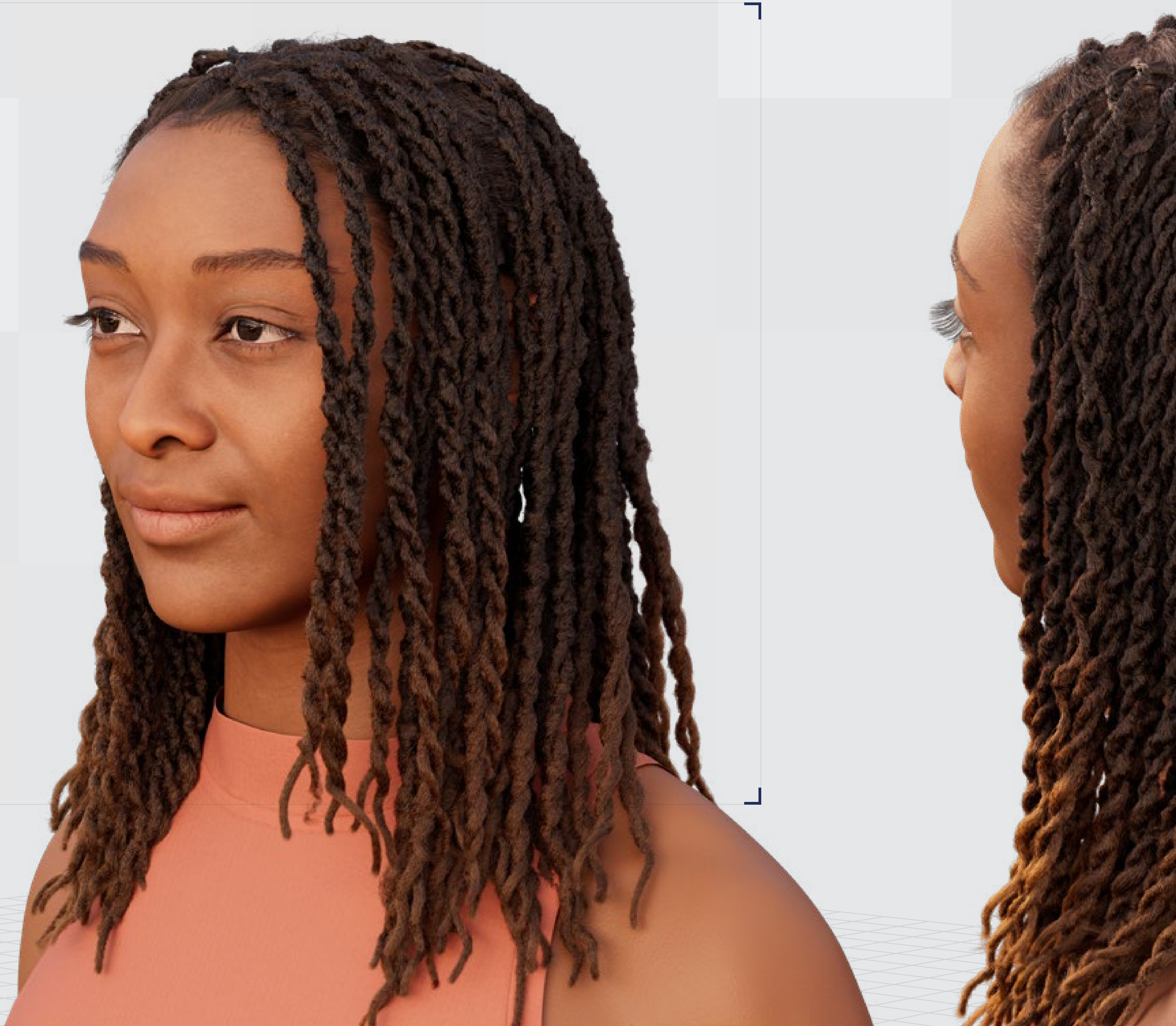






3D Design

Natural Locs



Digital Artist:
Isaac Olander

// Natural Locs

3D Design

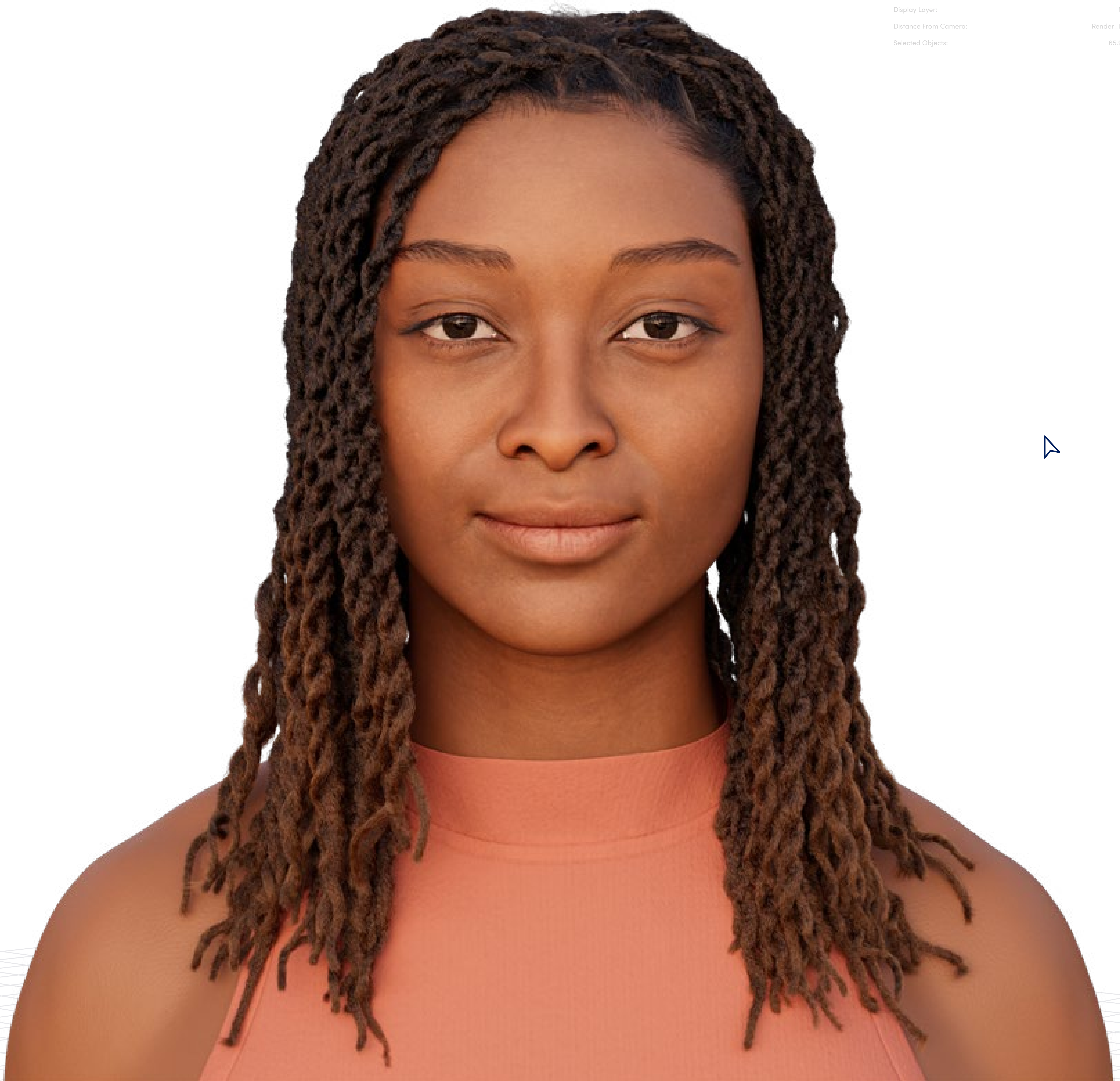
Curl Type Modeled: **4B/4C**

// 01

// 02

Download 3D Render

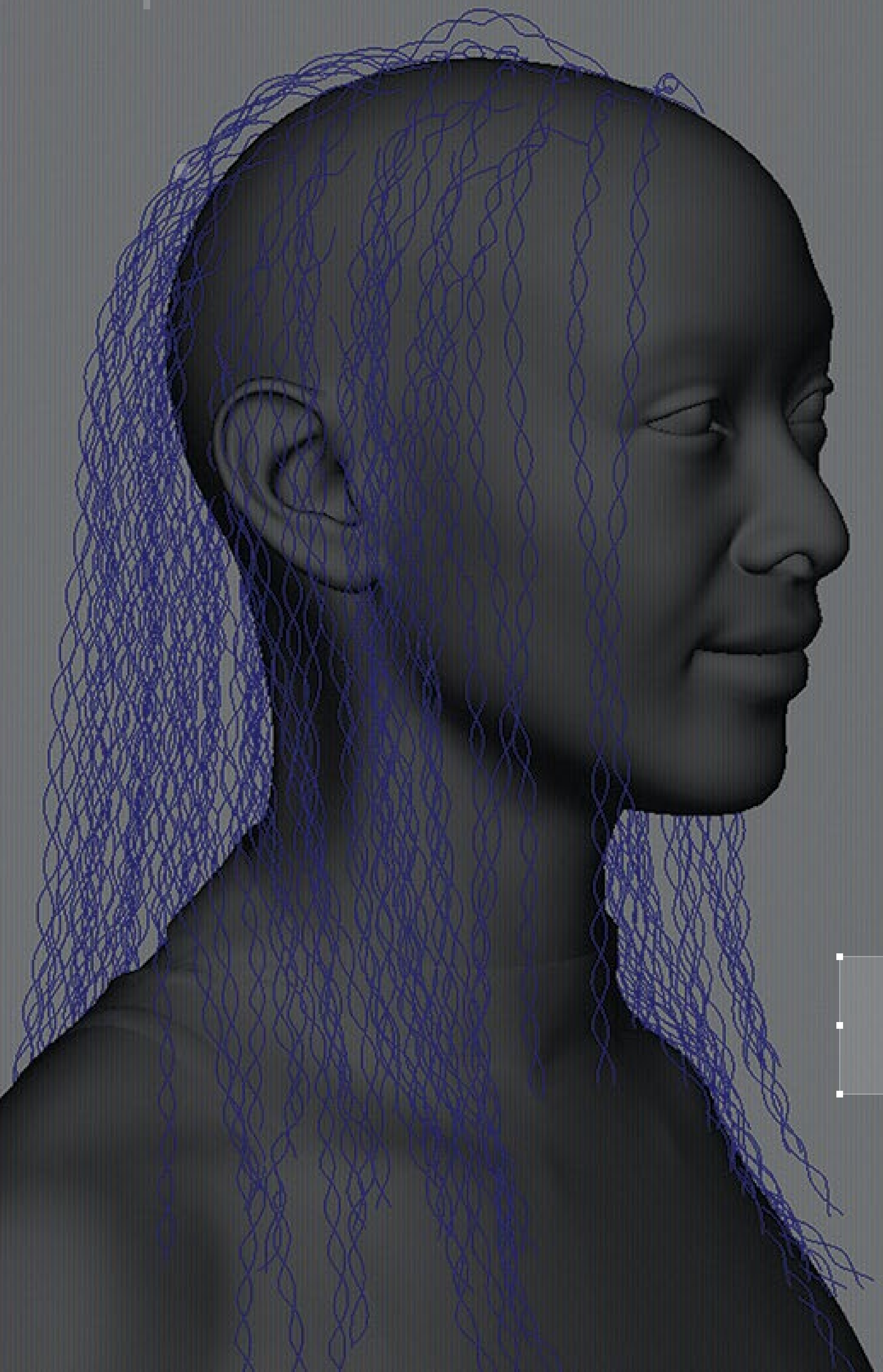
Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

Project Settings

// Natural Locs

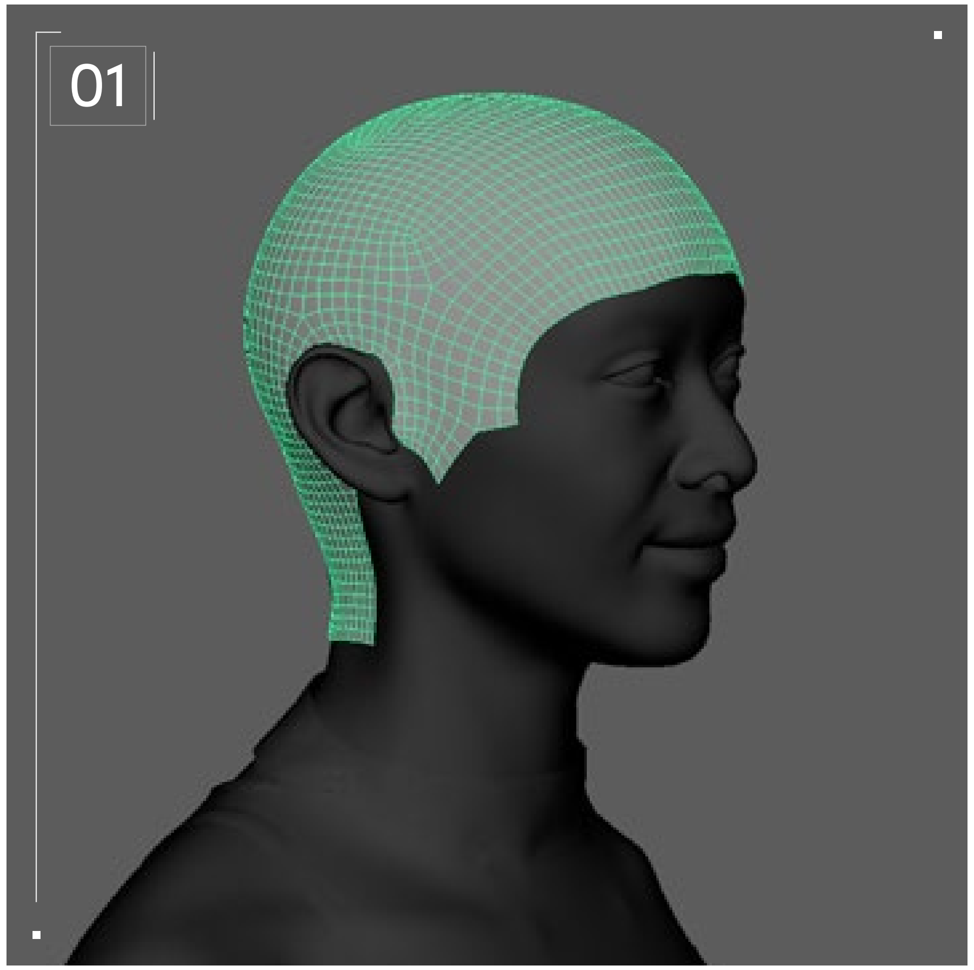


// Project Settings

Program Guide

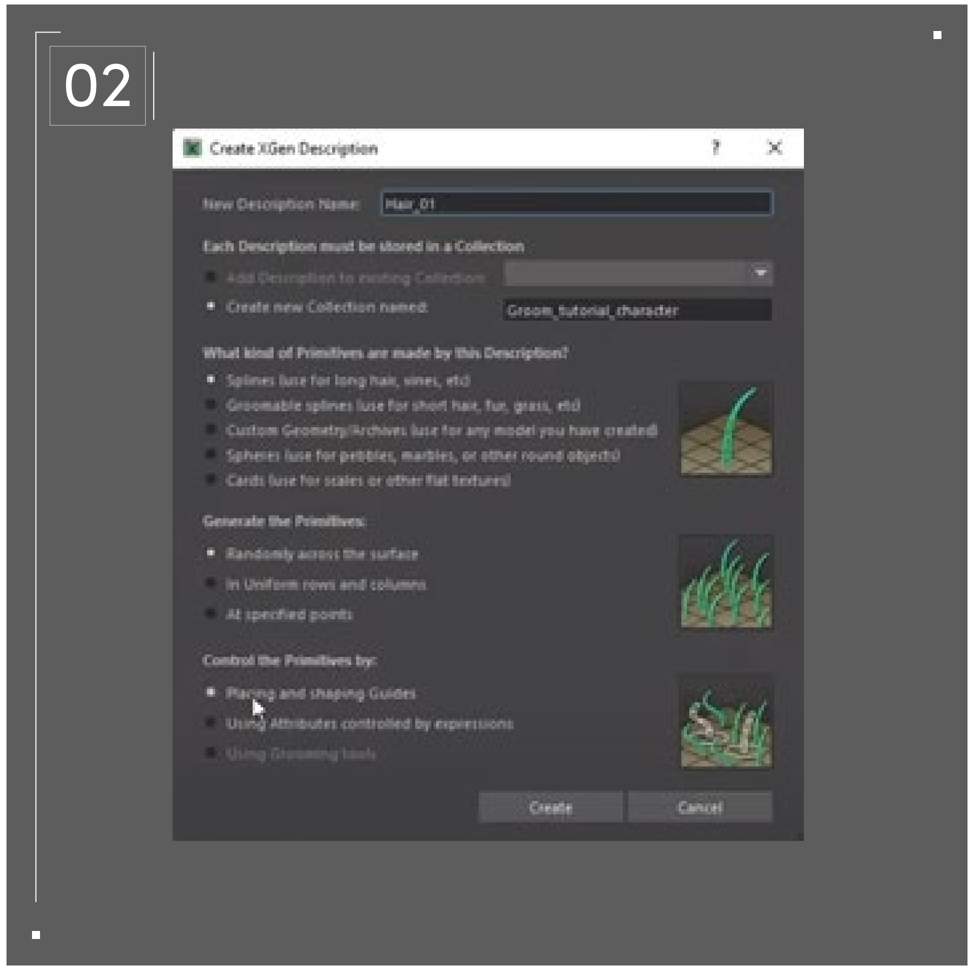
Software: Autodesk Maya 2022
Render Engine: Arnold
Additional Plug-Ins: XGen Interactive Grooming Toolset

Step-by-Step Guide



// Geometry preparation

First, duplicate the head model, and manually select all the polygons which you want the hair to grow from.
Select “inverse” and delete the remaining polygons. Make sure to keep the scalp geometry bigger than necessary, as you might underestimate how much space you need to not only grow the hair, but also, for example, to place baby hairs.



Create XGen Description

New Description Name: Hair_01

Each Description must be stored in a Collection

- ○ Add Description to existing Collection
- ● Create new Collection named: Groom_tutorial_character

What kind of Primitives are made by this Description?

- ● Splines (use for long hair, vines, etc)
- ○ Groomable splines (use for short hair, fur, grass, etc)
- ○ Custom Geometry/Archives (use for any model you have created)
- ○ Spheres (use for pebbles, marbles, or other round objects)
- ○ Cards (use for scales or other flat textures)

Generate the Primitives:

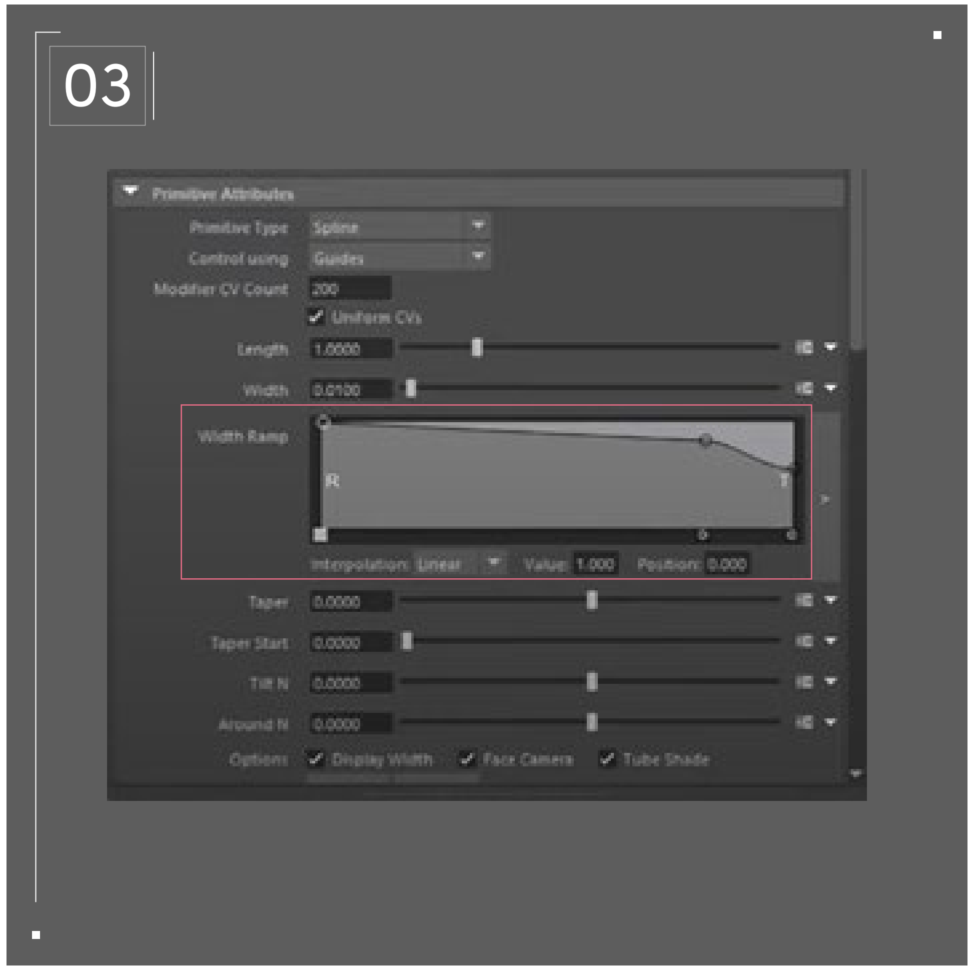
- ● Randomly across the surface
- ○ In Uniform rows and columns
- ○ At specified points

Control the Primitives by:

- ● Placing and shaping Guides
- ○ Using Attributes controlled by expressions
- ○ Using Grooming tools

// XGen description

Apply a new lambert material to the scalp geo. Open XGen editor and create a new description and a collection. Under “Control the Primitives by,” select “Placing and shaping guides.” This will ensure you will be placing and shaping guides manually, which gives you enough control to follow your reference.



Primitive Attributes

Primitive Type: Spline

Control using: Guides

Modifier CV Count: 200

☑ Uniform CVs

Length: 1.0000

Width: 0.0100

Width Ramp

Interpolation: Linear Value: 1.000 Position: 0.000

Taper: 0.0000

Taper Start: 0.0000

Tilt N: 0.0000

Around N: 0.0000

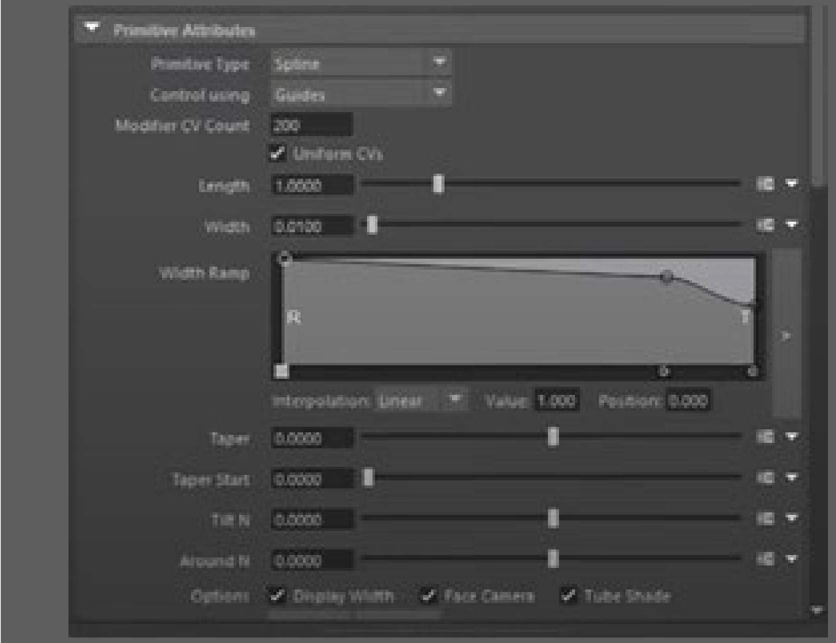
Options: ☑ Display Width ☑ Face Camera ☑ Tube Shade

// Primitive Attributes

Under “Primitive Attributes,” go to the width ramp and lower the value of the “Tip” side of the graph, so that the hair strands taper off towards the tip. Also set the “modifier CV count” to something like 50, to ensure there are enough vertices on the hair strands in order for them to curve.

Step-by-Step Guide

03

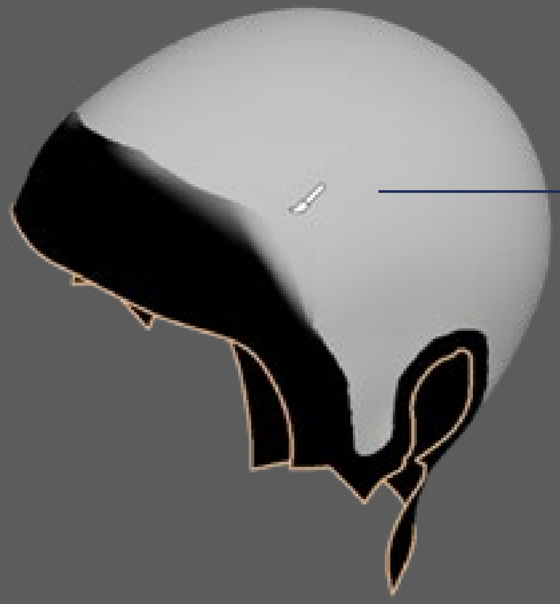


Primitive Attributes

Primitive Type	Spline
Control using	Guides
Modifier CV Count	200
	☑ Uniform CVs
Length	1.0000
Width	0.0100
Width Ramp	Interpolation: Linear Value: 1.000 Position: 0.000
Taper	0.0000
Taper Start	0.0000
Tilt N	0.0000
Around N	0.0000
Options	☑ Display Width ☑ Face Camera ☑ Tube Shade

Set Density to 100. You can increase and decrease after the modifiers are applied to find the density that you want.

04

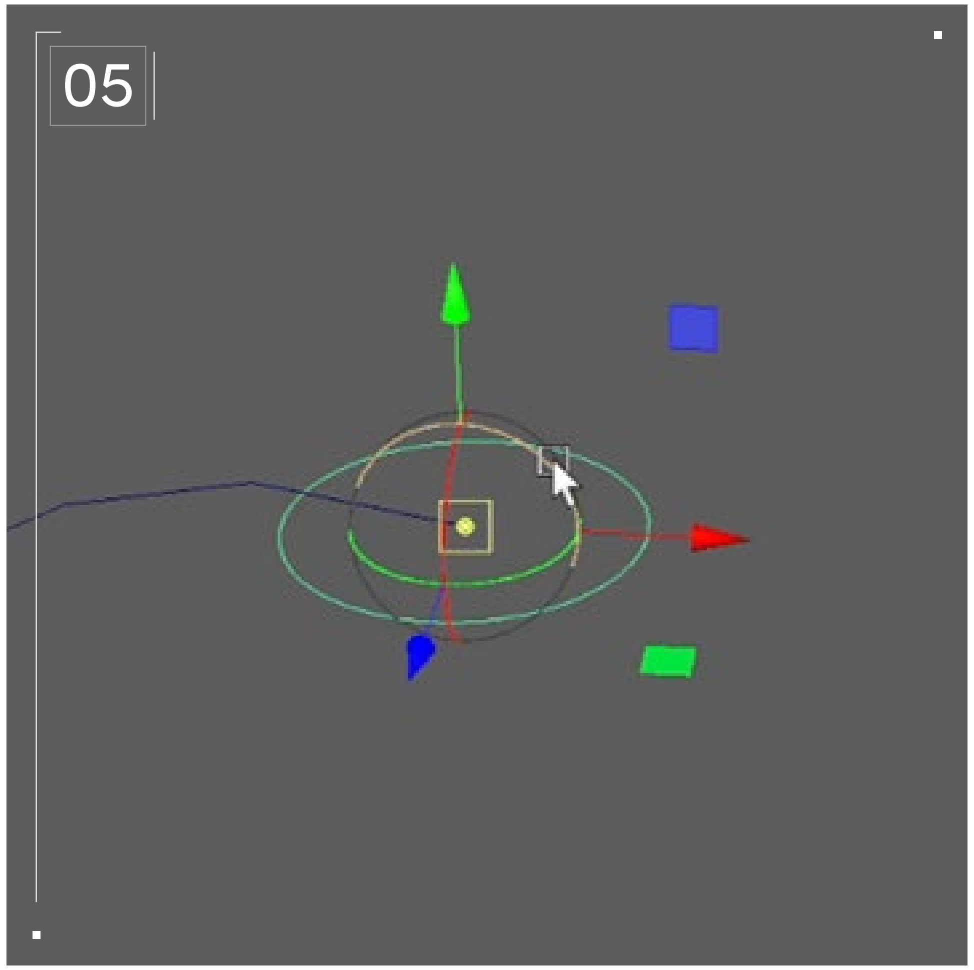


Open the brush tool settings and set the blend mode to “constant,” then scroll down to “stroke” and select “reflection” to apply brush strokes symmetrically. The XGen will grow from all the areas that are white, so pick a black value for the brush color and paint all the areas that you don’t want the hair to grow from. Come back and iterate on this mask throughout the groom process. You might need to paint the mask more unsymmetrically or soften the transitions between black and white.

// Paint mask

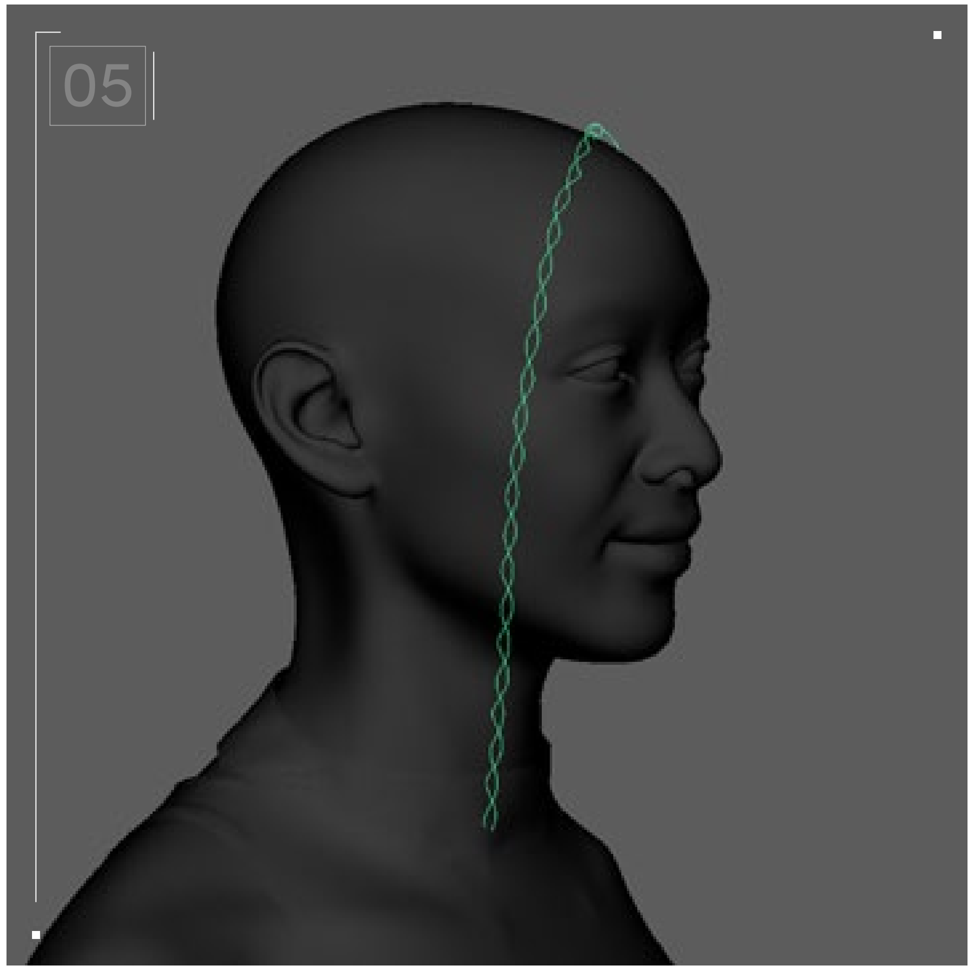
Now, go to the XGen editor for your hair description. Next to the mask, click “Create Map...” This allows you to create a paint mask to determine where the hair strands will grow from the geometry. Set the map resolution to 25 and keep it white.

Step-by-Step Guide

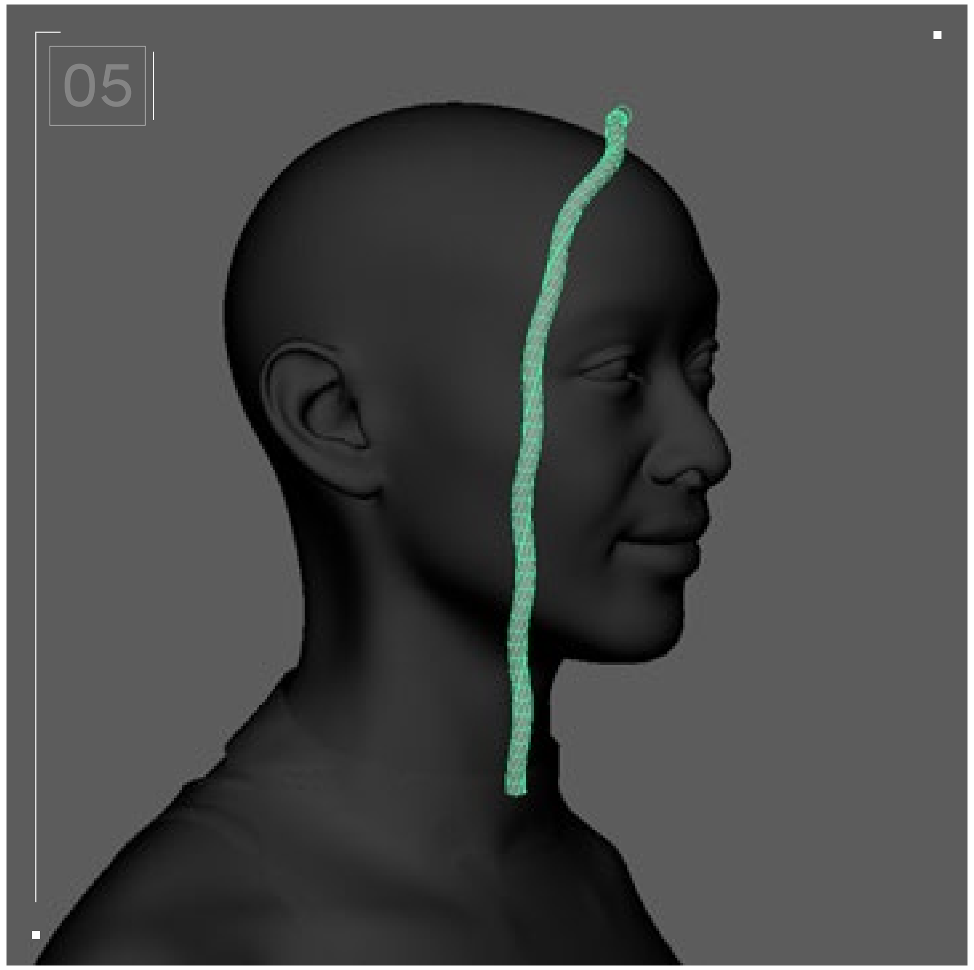


// Guides

Now we will create a numbers curve. This curve will represent the general flow of one single braid. Go to "Curve" then "Curve Tools," and then "CV Curve Tool." Go to an orthographic
view and use the CV Curve Tool to plot points to represent the general shape of the braid you're creating. Move this curve to the scalp.

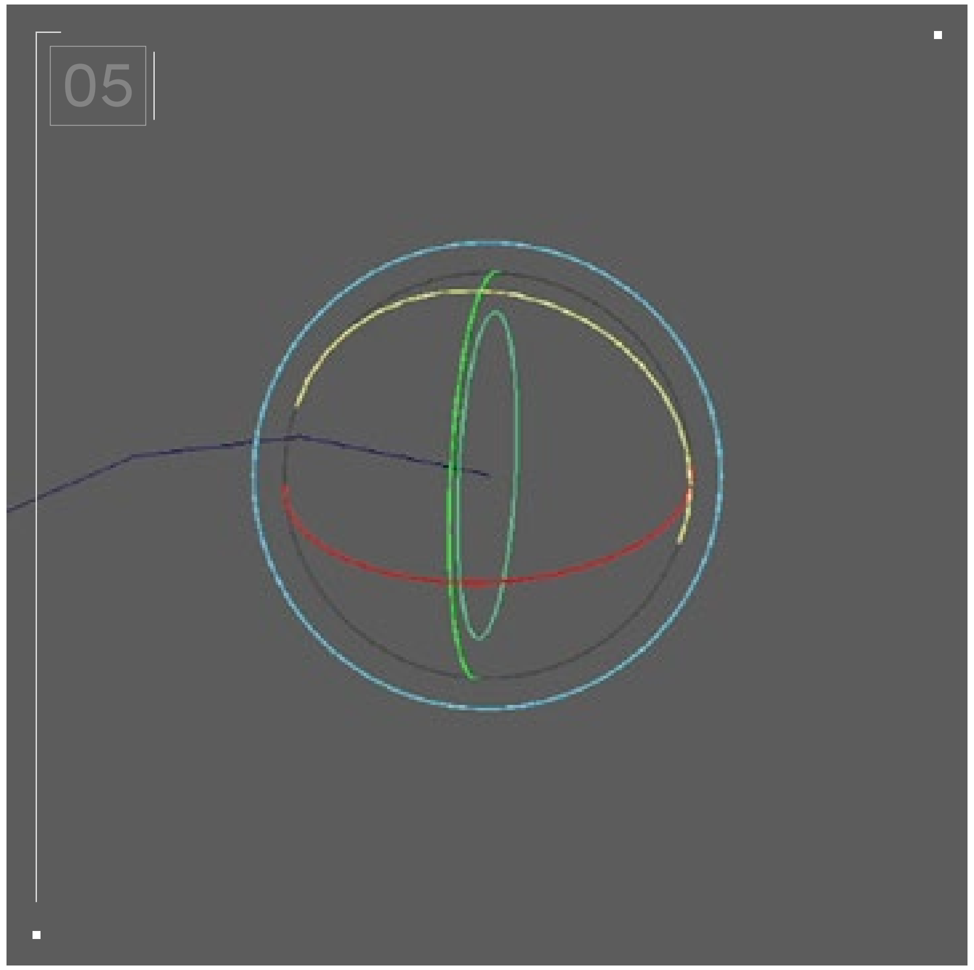


With vertex selection, move the vertex at the root of the curve as close to the scalp surface as possible. Use smooth selection while shaping the curve to keep it smooth and consistent. Now the curve should appear as one single shape that represents the first braid of the groom. We will call this Braid Shape Curve.

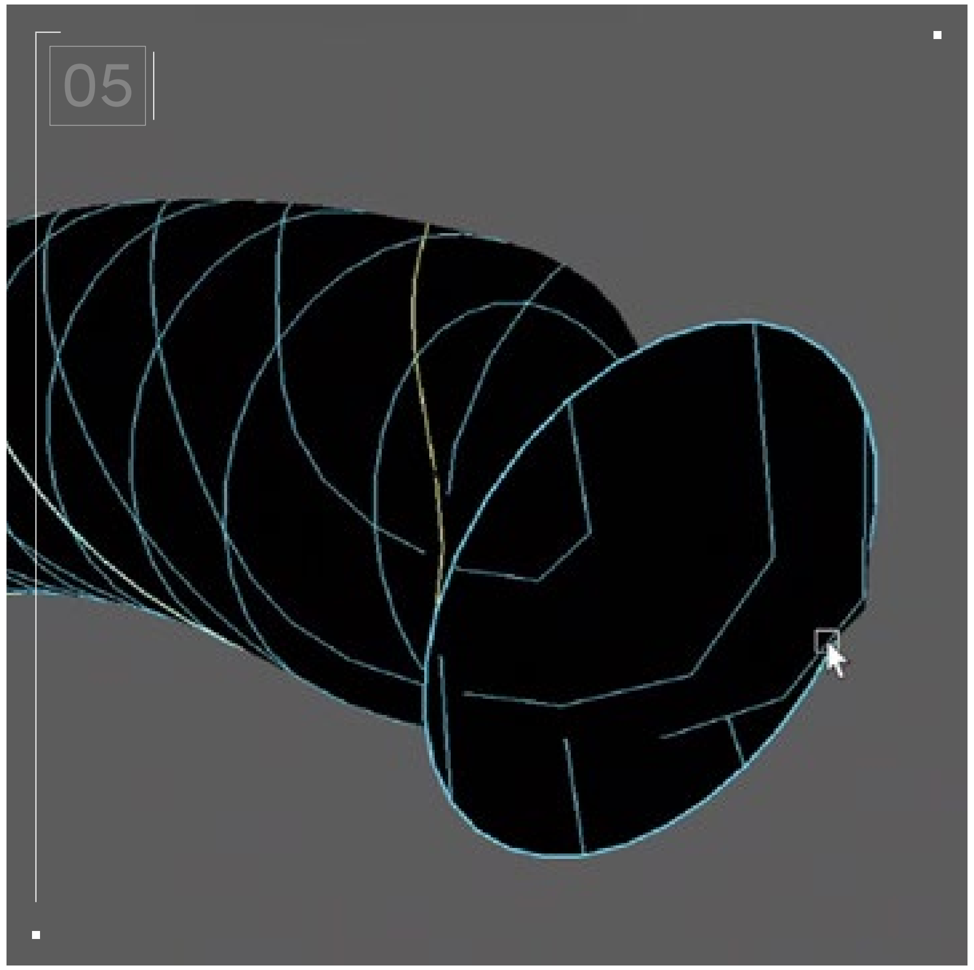


Next, go to the Create tab and create a Numbers Circle. Snap the circles pivot to the end of the curve that is closest to the scalp; this vertex represents the root of your braid. Scale the circle so it represents the desired thickness of the entire braid.

Step-by-Step Guide



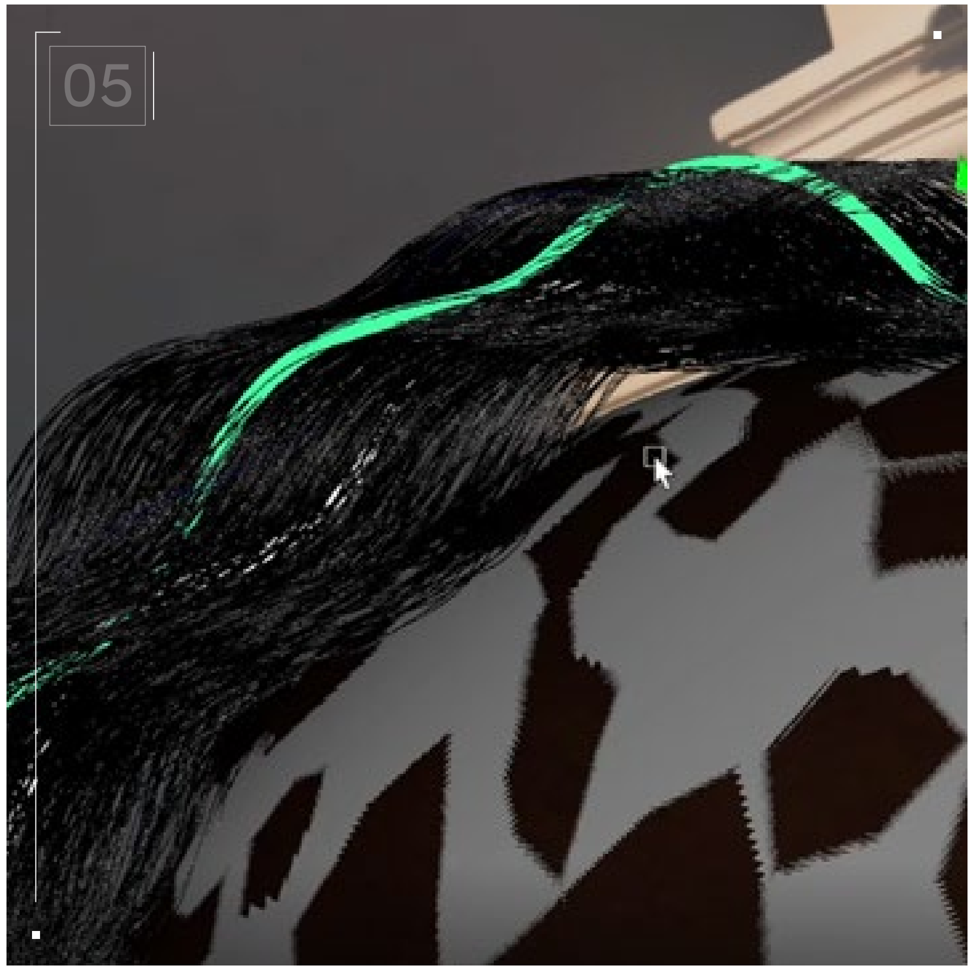
Rotate the circle so it lays perpendicular to the curve's length.



Go to "Curves," then "Rebuild Curve Options." Change the number of pans to 12 and rebuild. Shift select the Braid Shape Curve, open the Surface tab, and select "Extrude." Now you have a cylinder that runs along the curve. Go to the attribute editor and turn "Fixed Path" on.

With the cylinder selected, right-click and select "Isoparm." Select two edges that run the full length of the cylinder. Go to Curves, then click on "Duplicate Selected Curves." These curves are bound to the cylinder and will deform along it; it will be used to create your guides — let's call them twist guides.

Select the cylinder, go to the attribute editor, and in the shapes options, set rotation to a high number like 4,000.

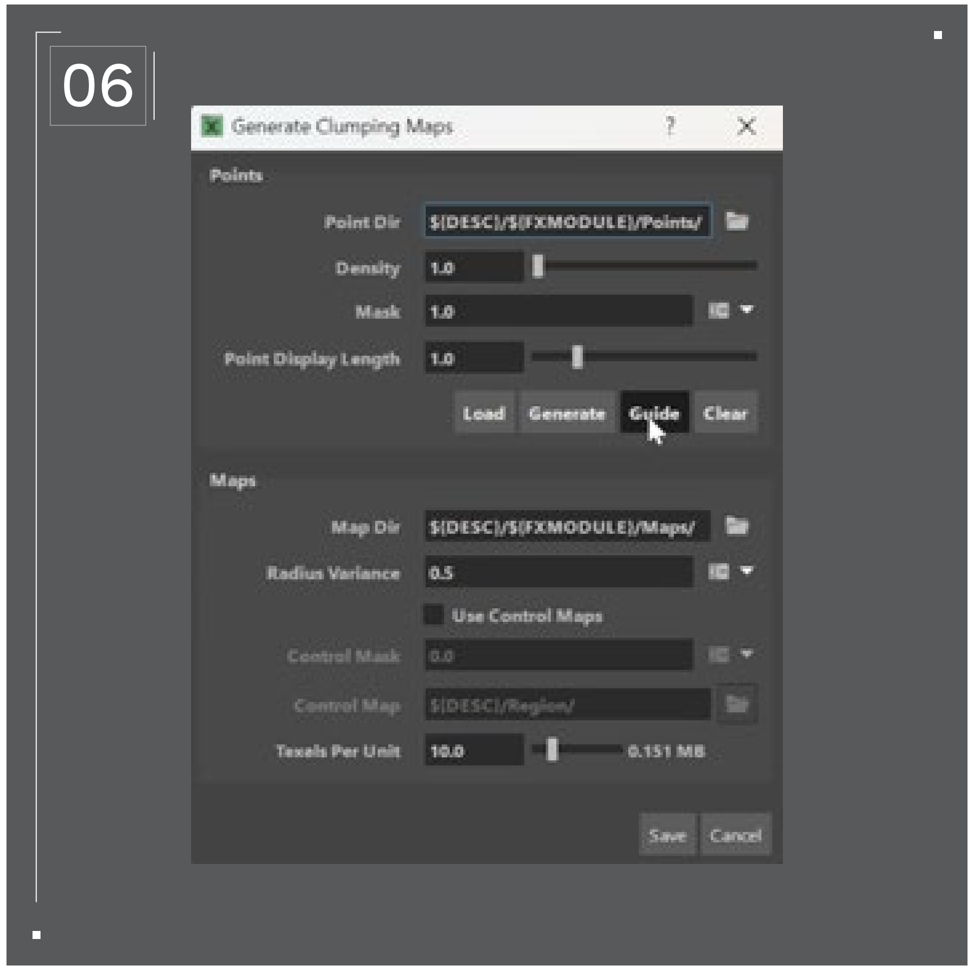


Hide everything except the two twist guides. You can use the vertex selection with soft selection to gently place the edge vertex of both these curves as close to the scalp surface as possible. Your braid will grow from where you've placed these two edges on the scalp geometry.

You can test converting these curves to guides by selecting them, then going to XGen utilities and clicking "Curve to Guide" — make sure to uncheck "Delete Curves."

An important tip: Do not delete the history of any of the steps relating to the curves we've just walked through.

Step-by-Step Guide



06

Generate Clumping Maps

Points

Point Dir: $(DESC)/$(FXMODULE)/Points/

Density: 1.0

Mask: 1.0

Point Display Length: 1.0

Maps

Map Dir: $(DESC)/$(FXMODULE)/Maps/

Radius Variance: 0.5

□ Use Control Maps

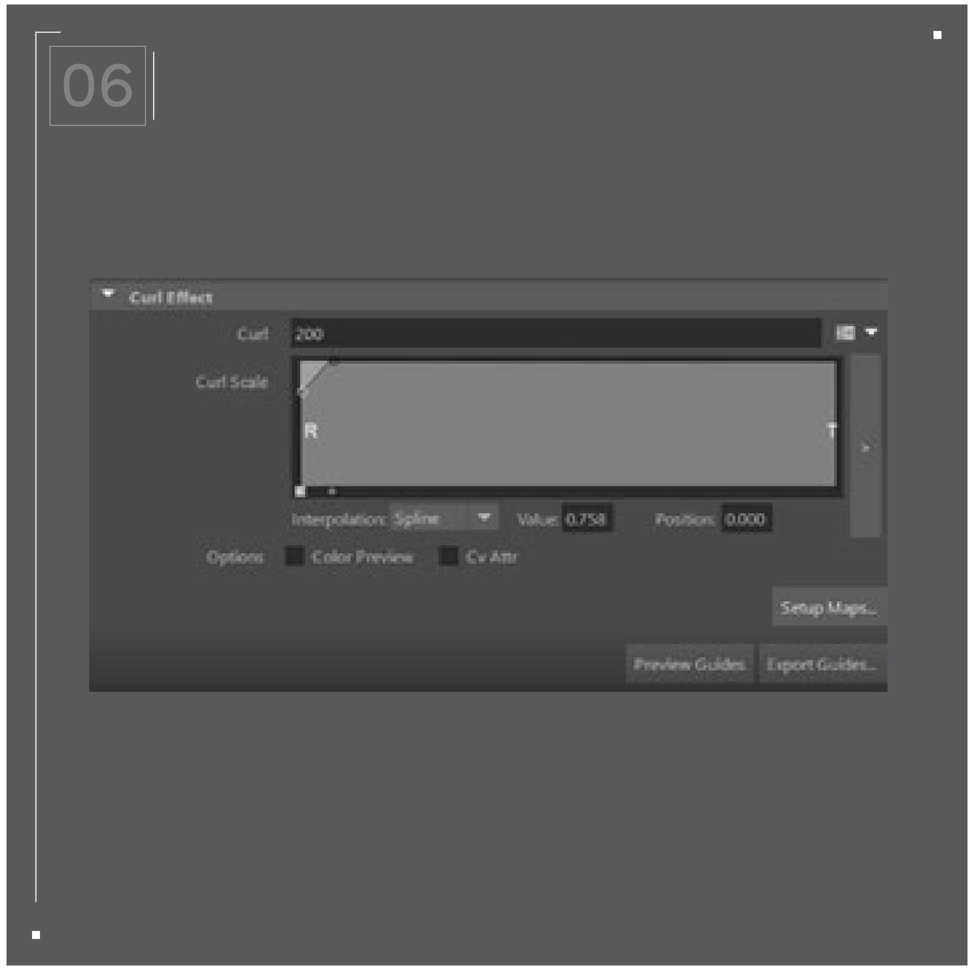
Control Mask: 0.0

Control Map: $(DESC)/Region/

Texels Per Unit: 10.0 0.151 MB

// Locs with modifiers

Add a clump modifier. It's important to set the density to 1. Instead of generating clump guides for the hair to follow, we want the clump to be directly related to the hair guides that we generated with the curves. To do this, click the Guide button and save.



06

Curl Effect

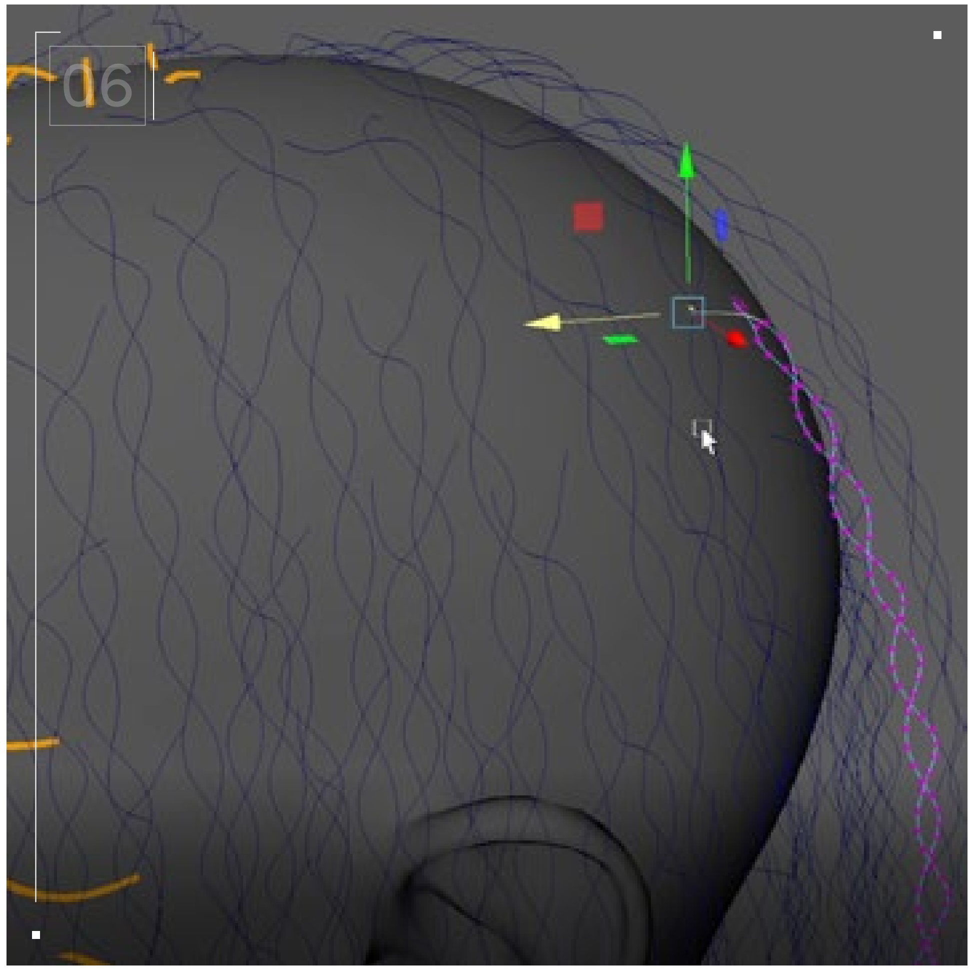
Curl: 200

Curl Scale

Interpolation: Spline Value: 0.758 Position: 0.000

Options □ Color Preview □ Cv Attr

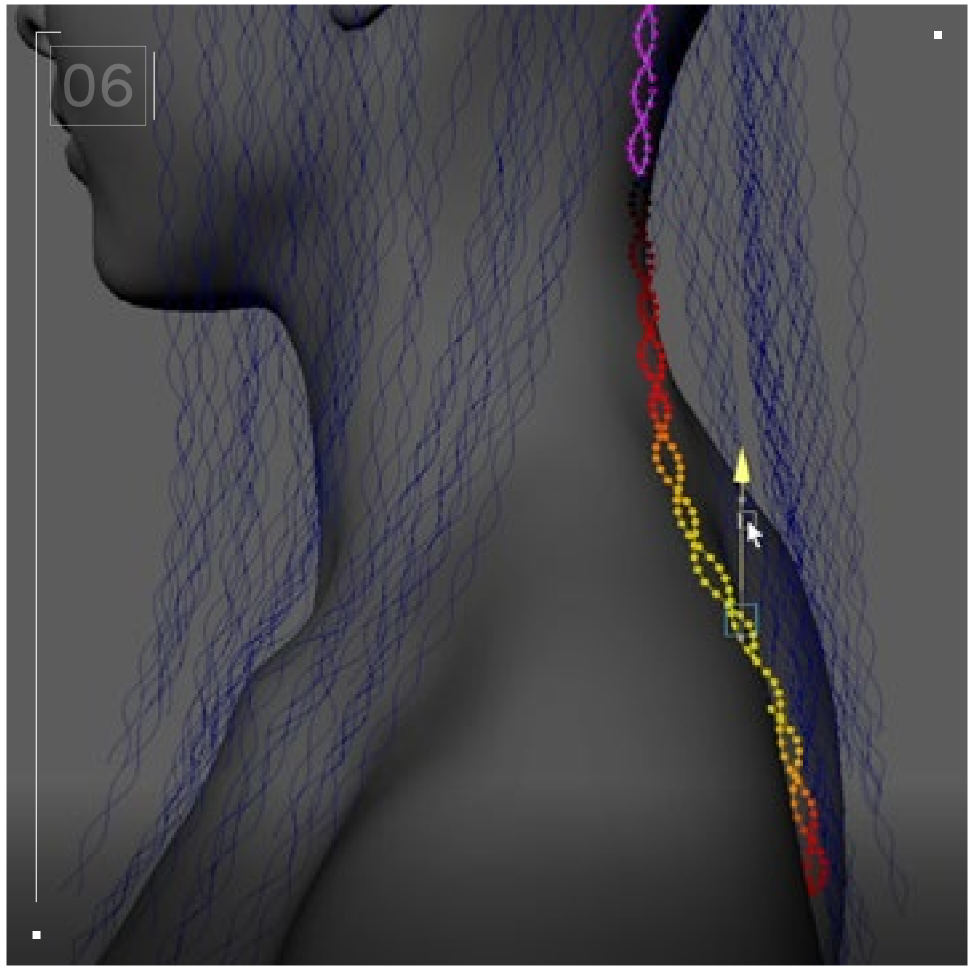
In the "Curl Effect" section of the clump modifier, set the curl to a high value, like 200, and set the influence on the graph to be 1 along the length of the hair. Let the effect taper at the root because we don't want the root of the hair to be straight, so it appears like the hair is tightening into a Loc.

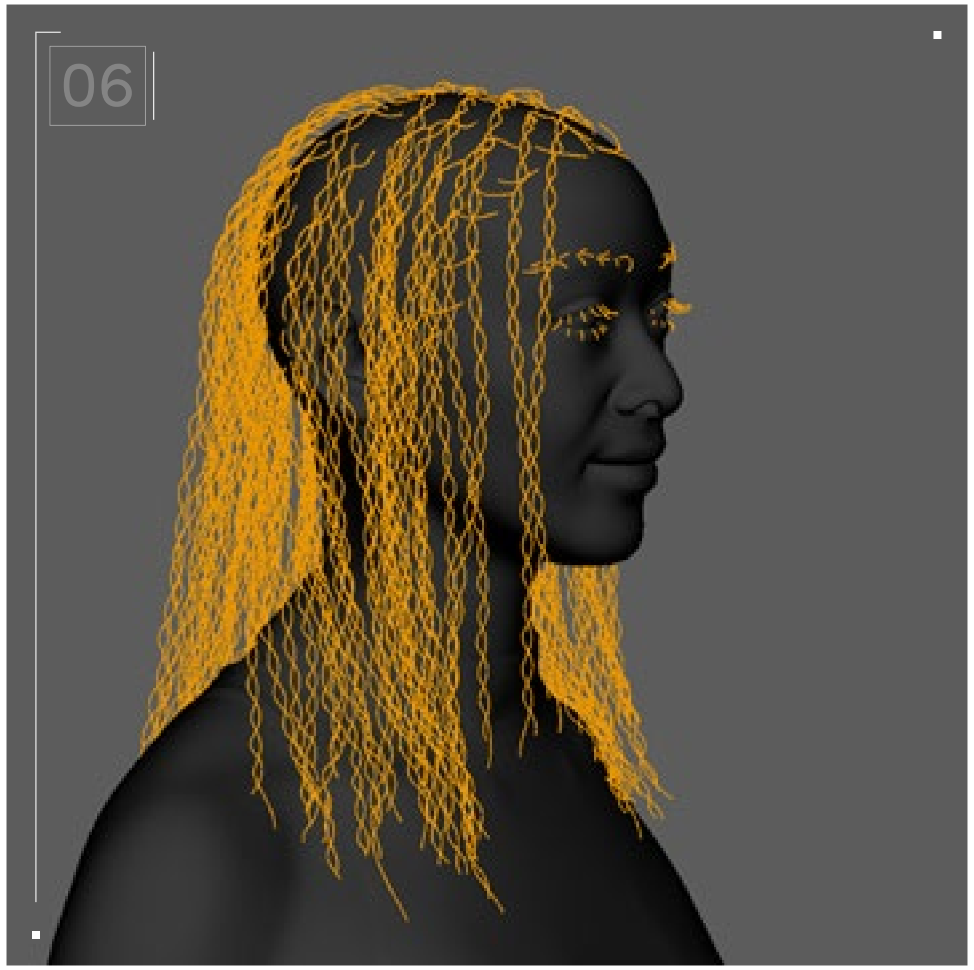


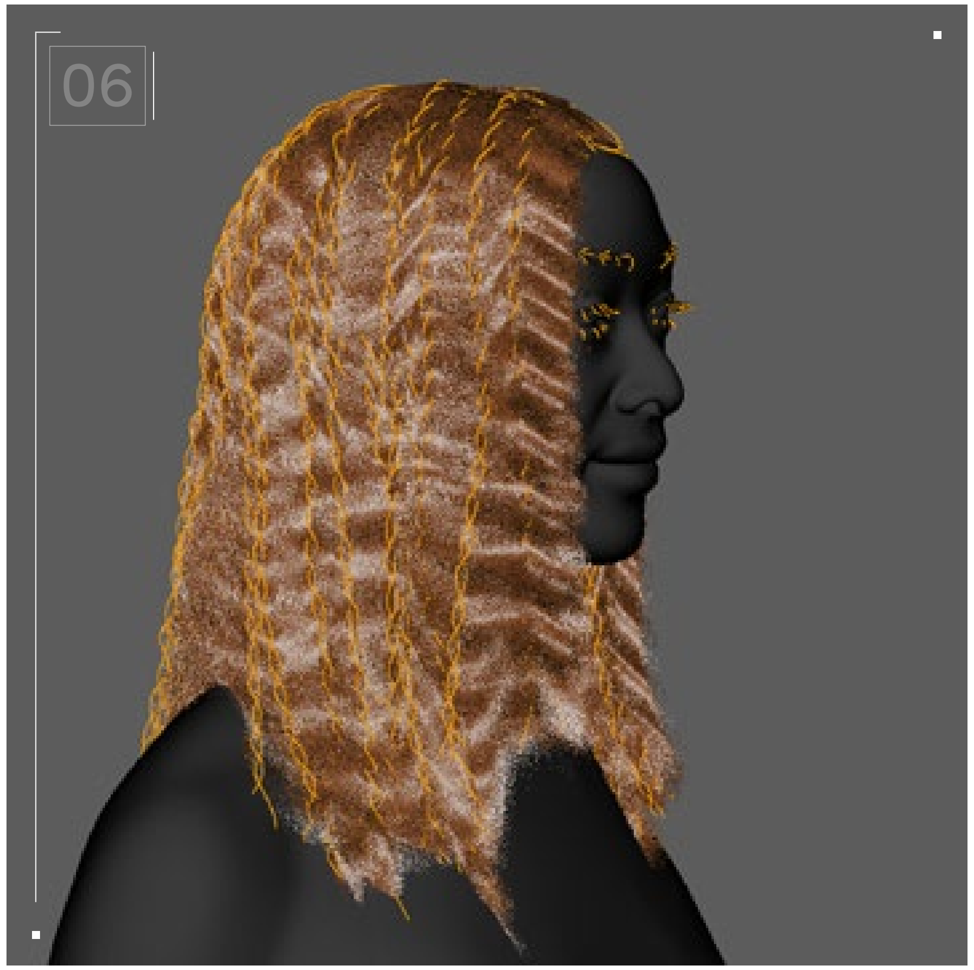
06

Now you can add a coil modifier. This will coil the curls and add a lot of noise. This will create the look of a dreadlock. If the braid looks as you intended, then you can go ahead and group this set of twist curves, then duplicate this group, place them along the scalp until the hair is filled, and shape the root vertices of all the curves as close as possible to the surface.

Step-by-Step Guide



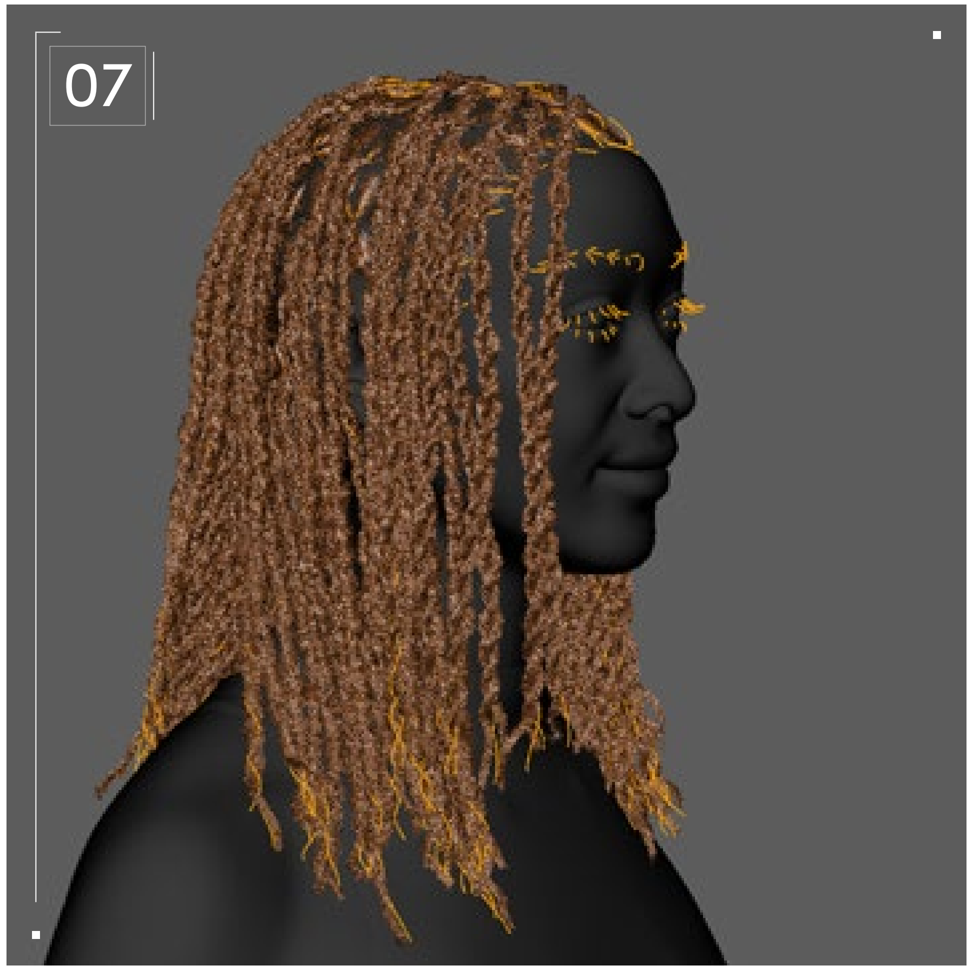




To modify the shape of these curves, vertex select a section that you want to modify, select "Soft Selection," and in the soft selection options, set the influence to be by volume. Now you can move large chunks of hair while retaining the overall structure of the braid.

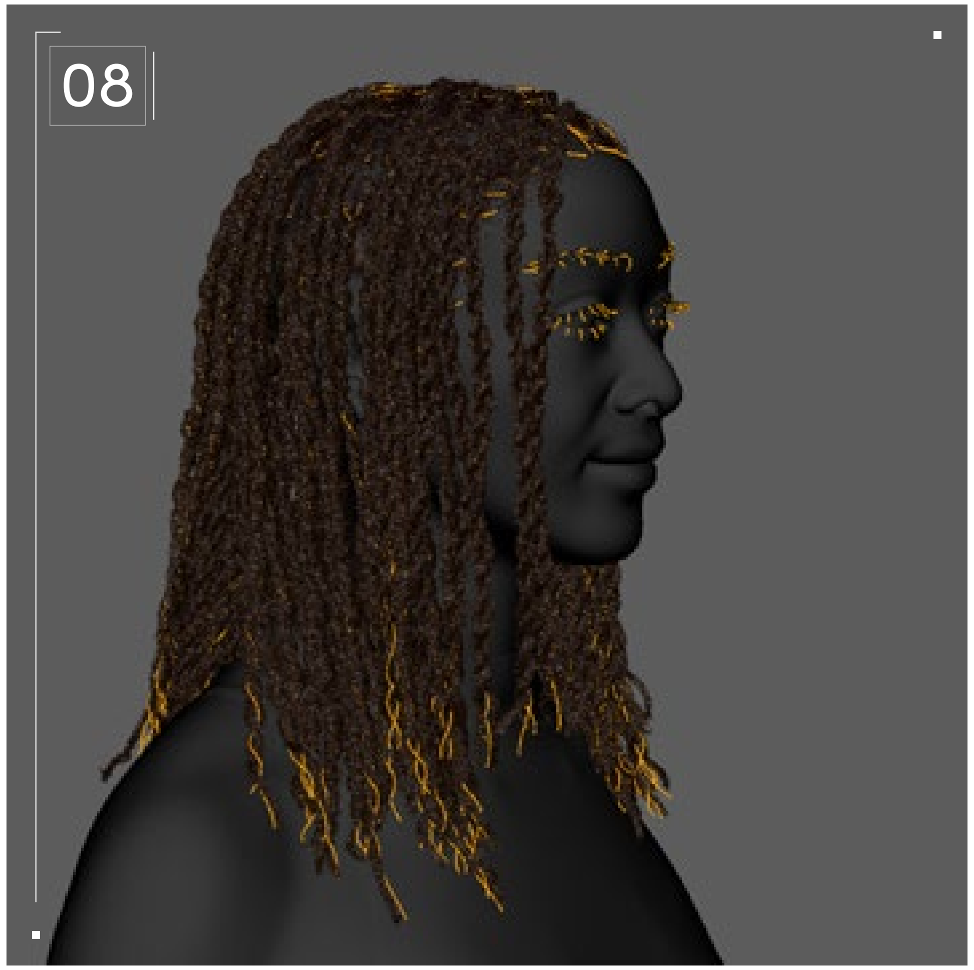
With all the curves selected, go to the Natural Curls XGen description utilities tab and select "Curves to Guides." Make sure that you uncheck "Delete Curves." That way, you will keep your curves in case you want to edit them further to regenerate new sets of guides. Now you should see that the scalp is populated with guides that accurately reflect the curves you made before.

Step-by-Step Guide



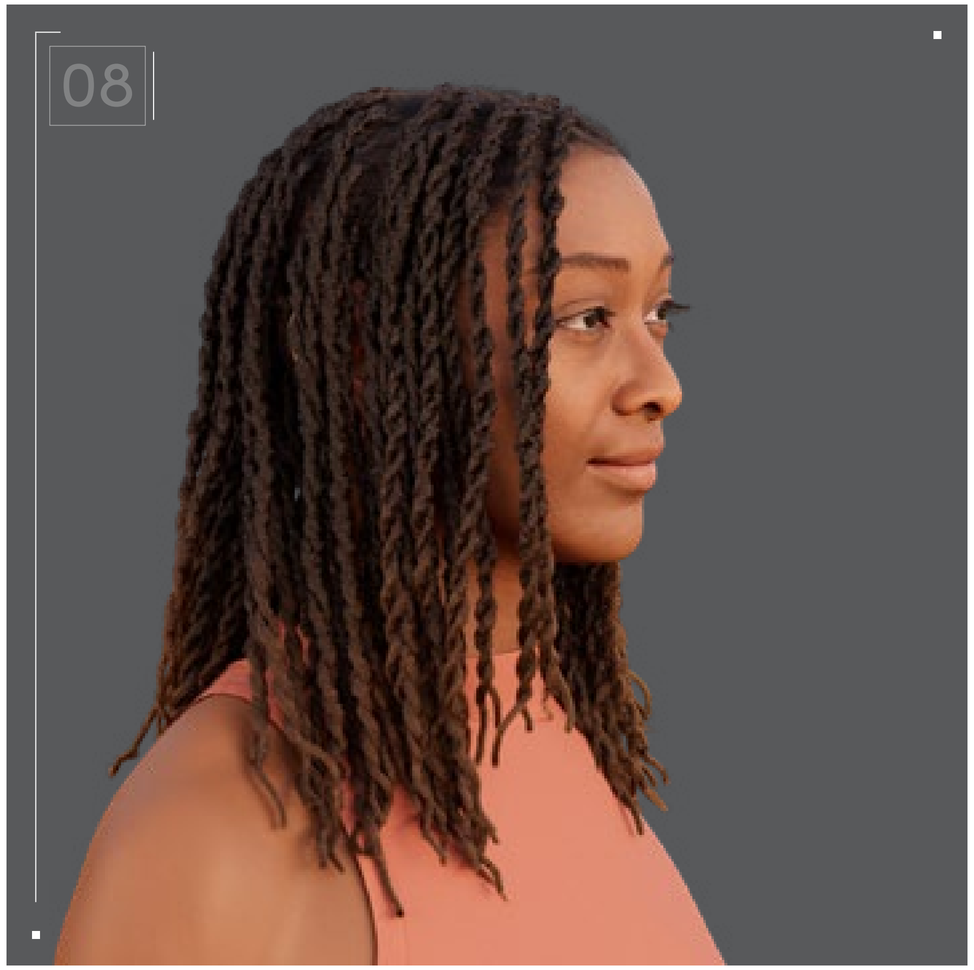
// Modifiers

Add a cut modifier — you can keep default settings or modify to your liking. Make three separate noise modifiers. Each one will affect one of three areas. Hair root, hair tip, and hair overall. Adjust each one in the influence graph editor to influence the different sections accordingly. Experiment with the modifiers until you achieve the look that you want for the Locs.



// Hair material

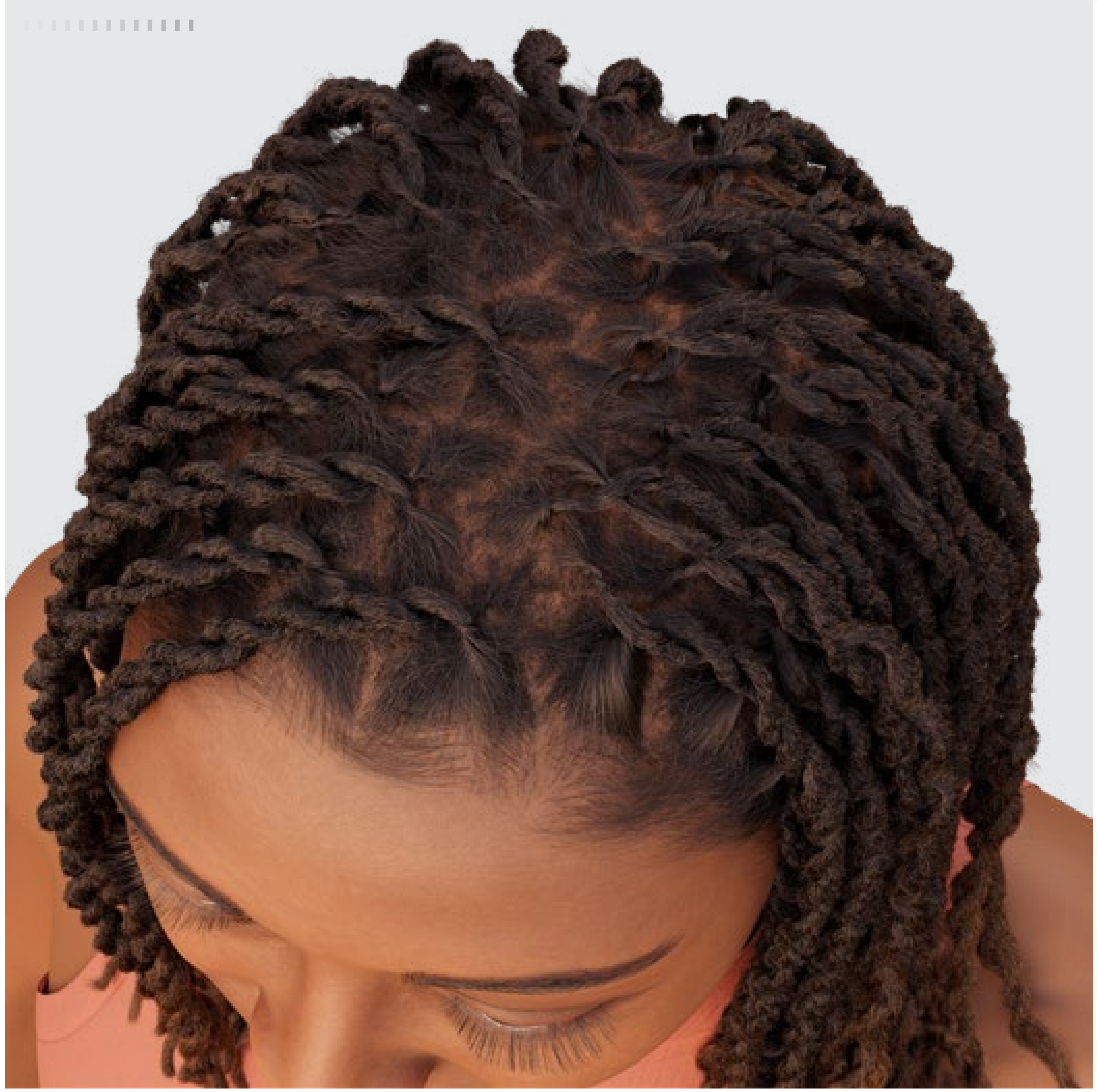
Select the guides in the outliner, go to the Hypergraph and assign an Ai_standard_hairmaterial to the XGen description. This should apply the material to the hair strands. Next adjust the melanin count, but don't touch the base color as we want to be working with realistic and natural colors for the hair.



Natural hair has a coarser texture to it; this is partly due to the fact that the bends in natural hair have a different angle when compared to other hair types. We can show this difference in bends through our Aistandard_hair material by going to the IOR section and setting the "Shift" value to 2.3. Continuously test your material by rendering the scene and iterate until it matches your hair reference.

Alternative Hair Textures

Try experimenting with the Curl Effect in the clump modifier in conjunction with the coil modifier values to look at the Locs. Another useful tip is that you can drive the thickness of the Locs from the Coil Scale value.



// Project Settings

Technical Watchouts

Set the Maya project file before creating anything XGen related. Never use spaces in your naming convention, as it will lead to the loss of your XGen project. XGen will fail to reference any resource that has a space in its location directory or name. Use underscore "_" instead.

Attaching your XGen description directly to a character model with a rig can result in problems when the character is animated. To avoid this, use a scalp geometry that is either parented to the head joint or proximity wrapped to the character model. Make sure the sculpt geometry has a lambert material attached to it before an XGen description is attached, as it is necessary in order for XGen to generate hair splines.

Curly strand-based hair requires high vertex counts (VCs) and can be demanding for your workstation. Consider the performance impact of high VC on your system resources and save frequently.

+++++

Top Tips

Put as much effort as possible into perfecting your guides; they are imperative for achieving a Loc'd hairstyle. Number curves can be a great tool for sculpting guides before attaching them to a scalp.

Overlay your reference image to your 3D software to figure out the position of each braid, but also allow yourself flexibility to strafe when applicable. 3D models do not need to match the reference 100% to look accurate.

The guide creation step is the most important step when creating hairstyles such as Natural Locs. Modifiers, on the other hand, are the most important step when creating loose hairstyles such as a Twist Out. Guides give us the control to influence distinct hair strands with precision, while modifiers give us control to influence all hair strands on a homogeneous level. To better understand the use of guides and modifiers, let's compare the role of guides and modifiers when creating a braided and loose hair style:

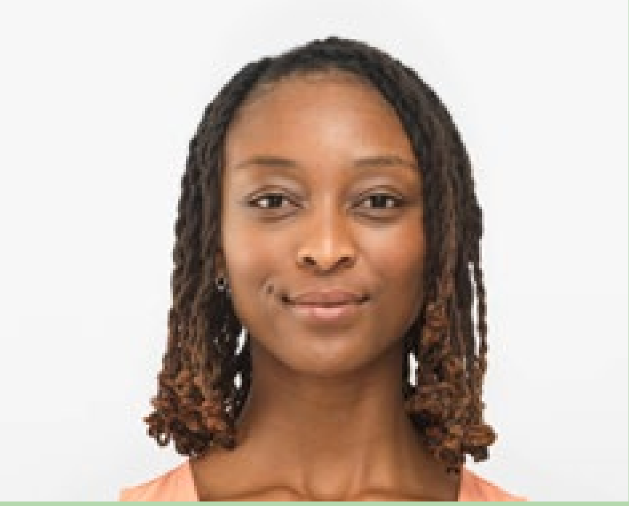
Loose hairstyles: When we approach loose hair types, such as a Twist Out, the direction of individual hair strands is uniform. Therefore, we rely on modifiers to procedurally influence features of the hair on a global level. We also use the influence of guides to drive a uniform direction of the hairs — for instance, we can use guides to part the hair in the middle of the head, to direct the hair away from the face, or to curve the hair around the ears to avoid clipping.

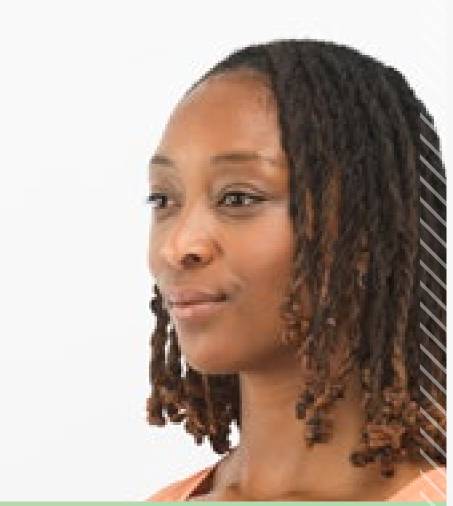
Braided, Twisted, or Loc'd hairstyles: When it comes to braided and Loc'd hairstyles, the direction of individual hair strands is distinct and non-uniform. Therefore, we rely on guides to manually influence the features of the hair on a localized level. The guides allow us to be precise enough to make distinctions between clumps of hairs and braids. We use the influence of modifiers to drive the uniform features of the hair strands that make up the braids.

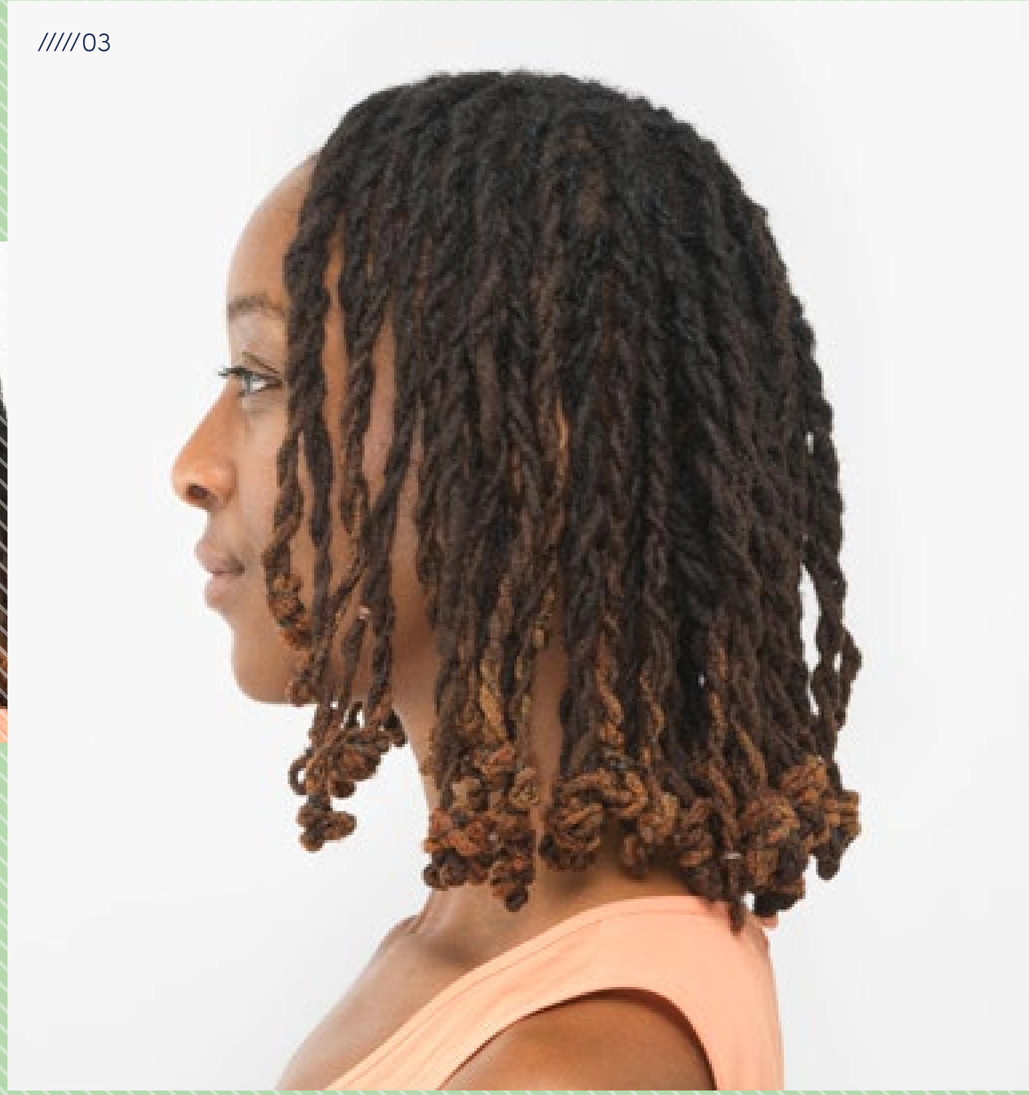
Now we understand that we can replicate braids with the help of guides. The question is, How do we do this in a smart way? It is far too tedious to manually shape and braid our guides together, so we need an alternative workflow. Luckily, the XGen utilities window has a utility for converting numbers curves to guides. We will use this feature to our advantage.

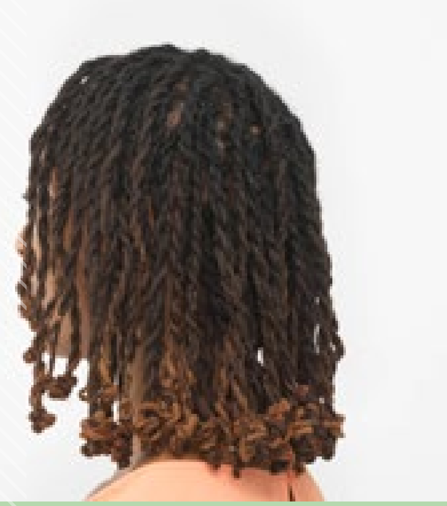
Curves are much easier to work with, as there are far more tools that enable us to easily duplicate, place, and shape curves, as opposed to guides. We will procedurally generate curves following the workflow that is set out in Step 7 of the step-by-step guide.

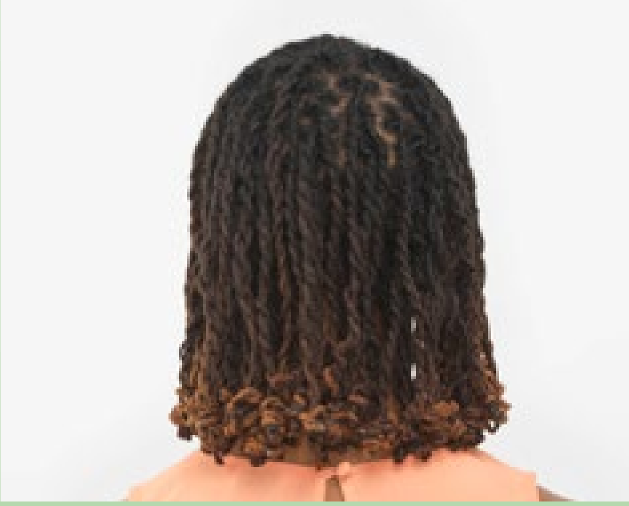
Loc'd









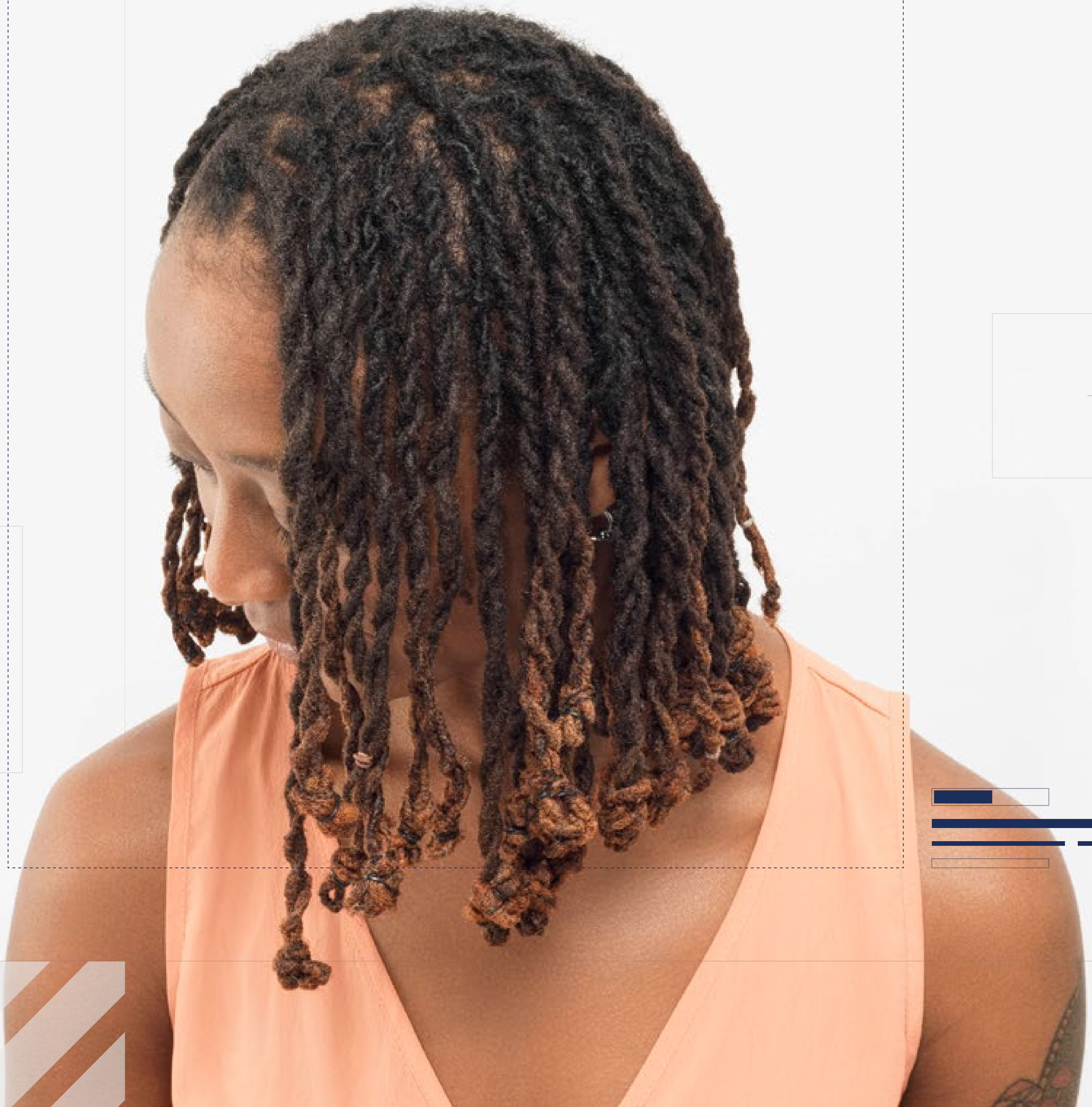


Bob

Bob
Bob

Download Images

// Loc'd Bob

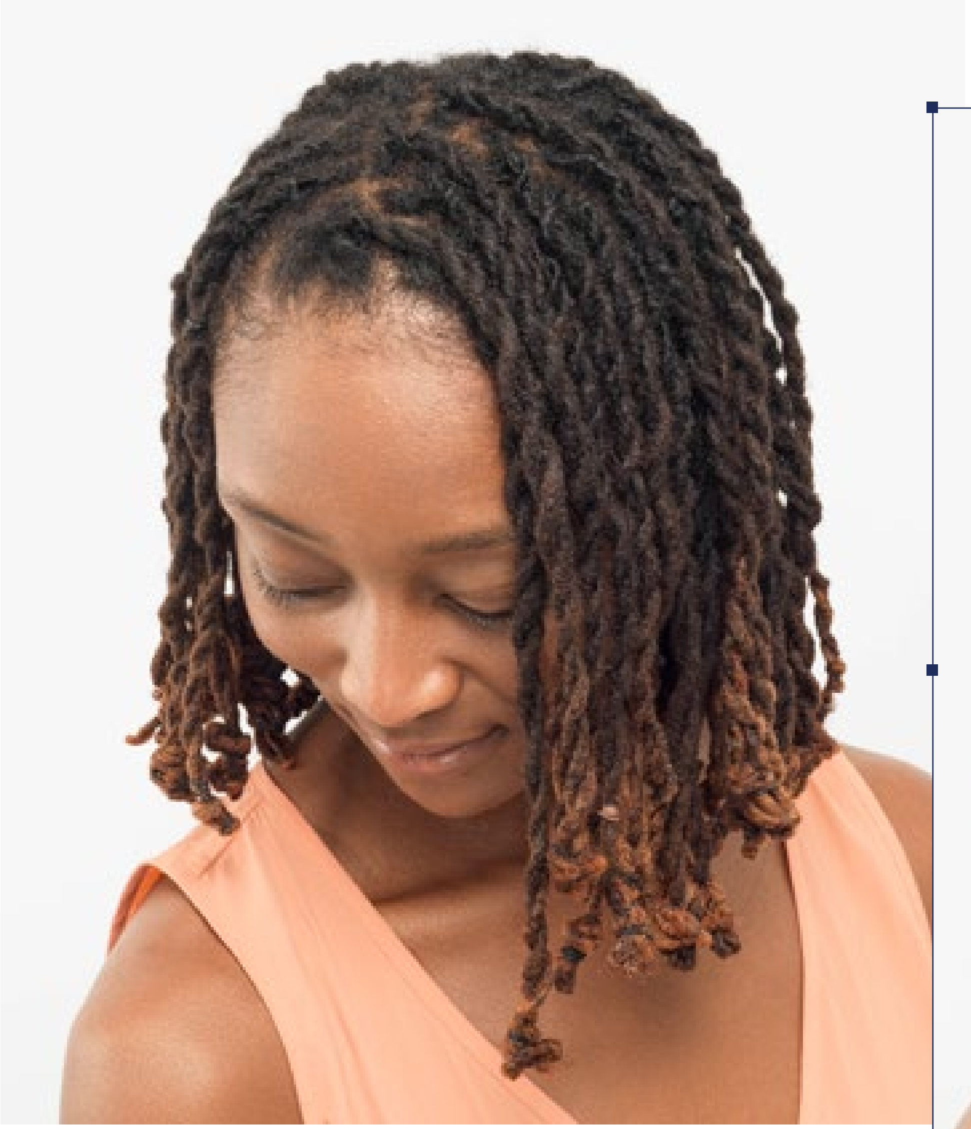


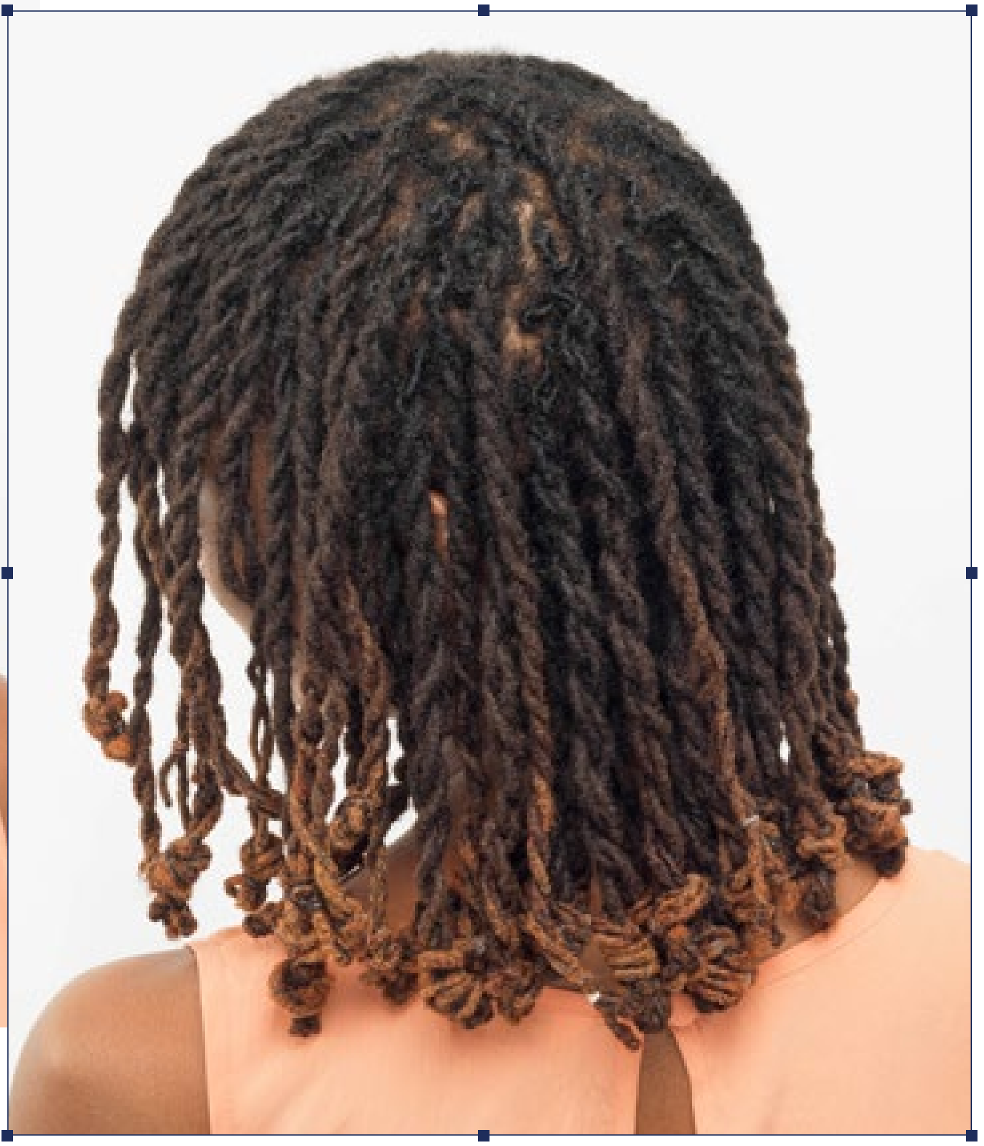
// Hairstyle

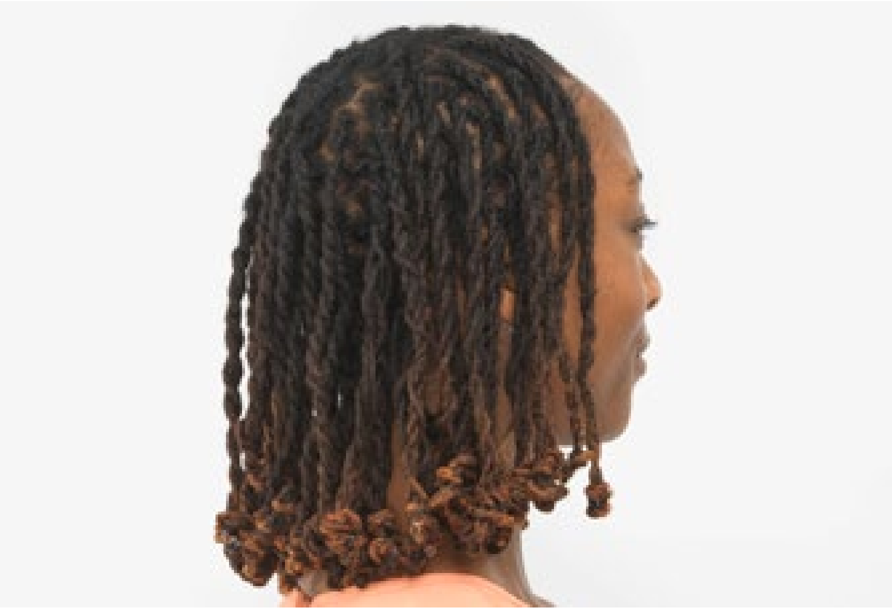
Loc’d Bob

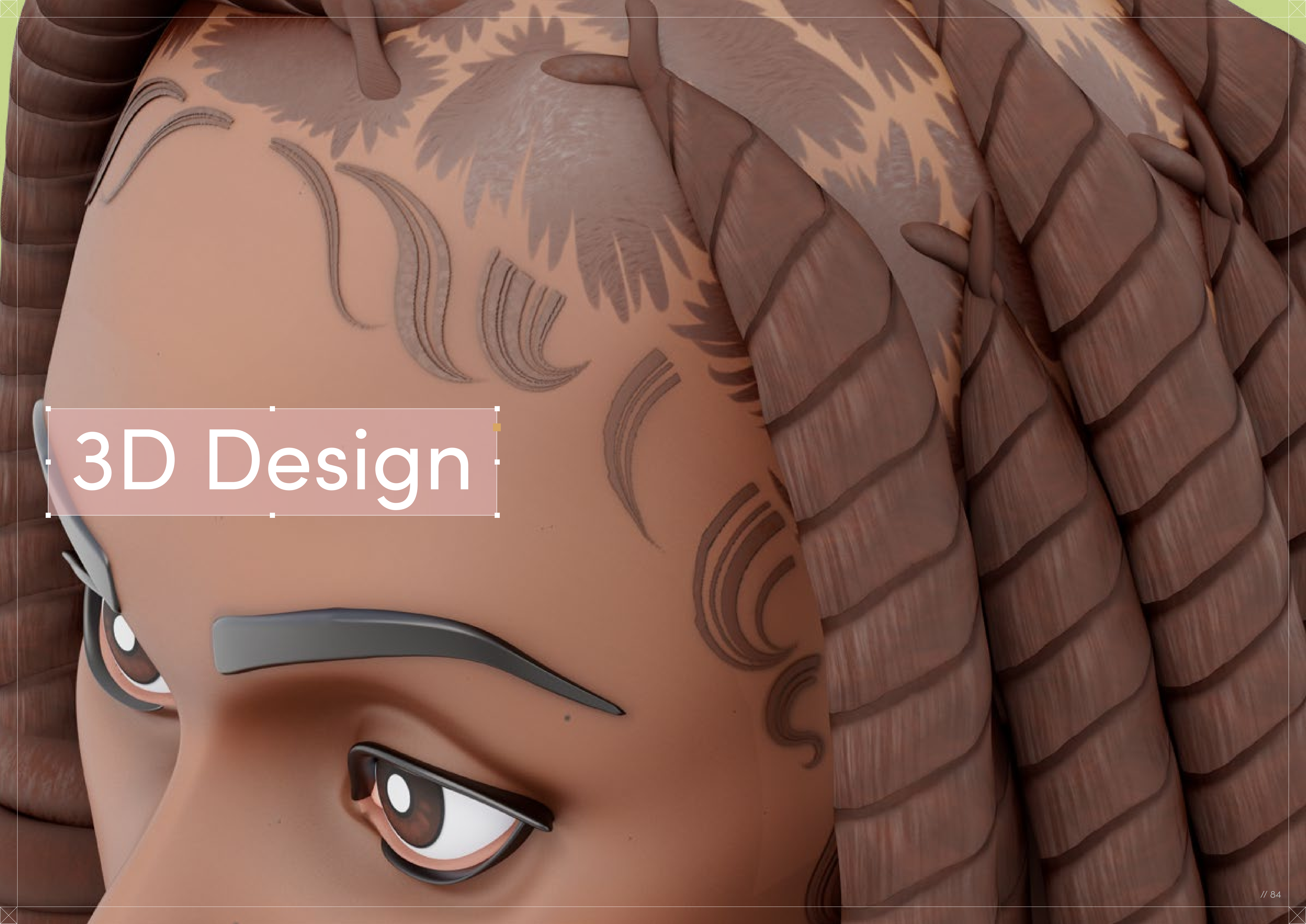
Style Background: The Loc’d Bob is a versatile style that offers a two-in-one look. It combines the elegance of two-strand Twists with a knot wrapped around the end, resulting in the appearance of a bob. This unique style brings together several styling methods with distinct and eye-catching results.

Download Images



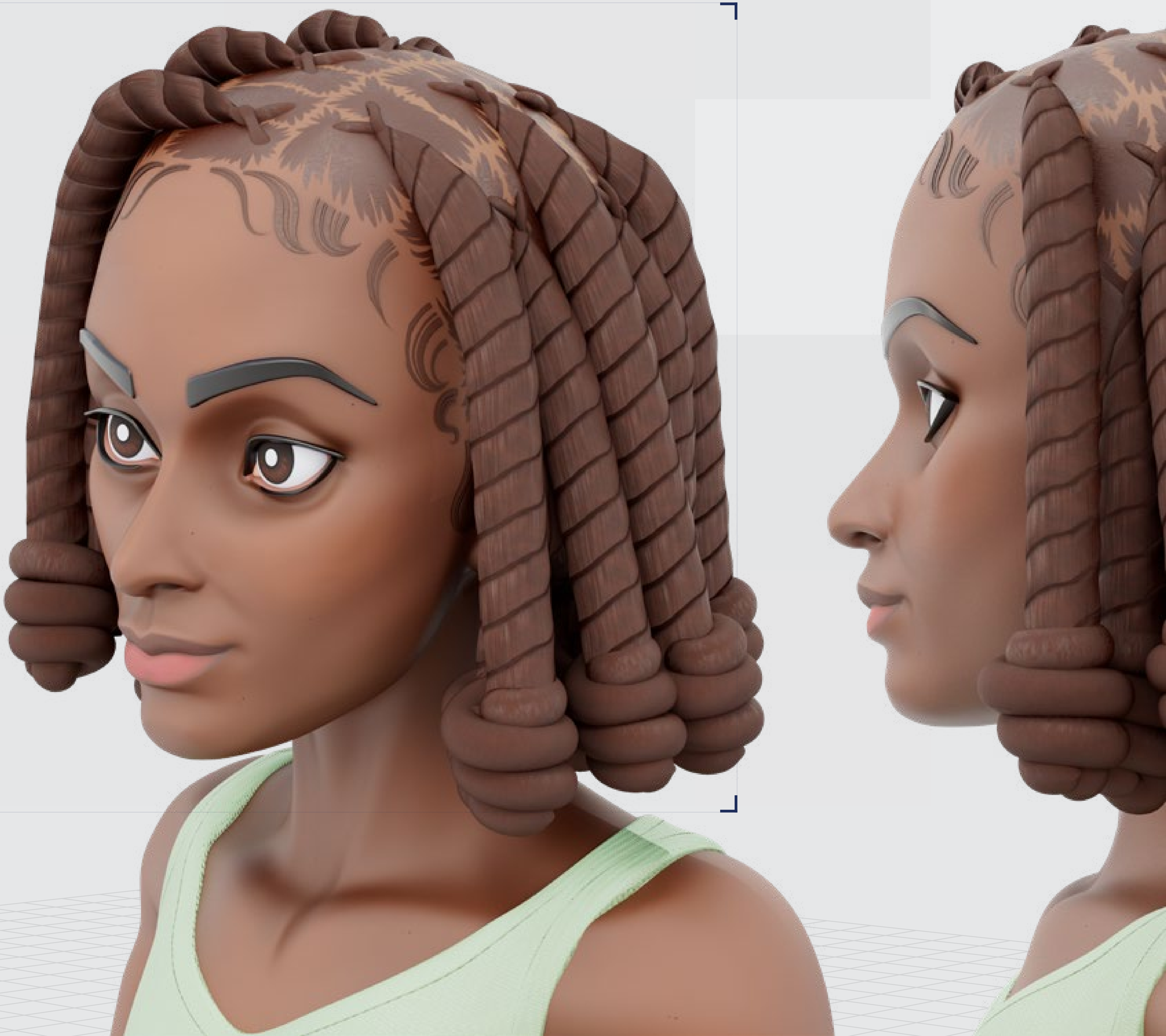






3D Design

Loc'd Bob

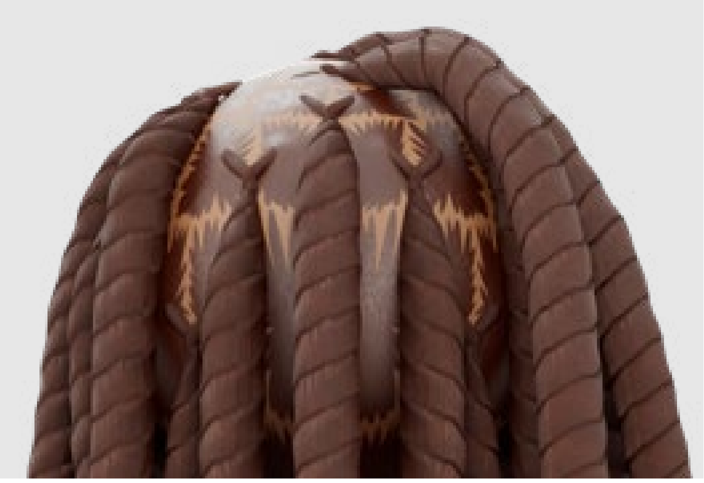


Digital Artist: **Sajhmori**

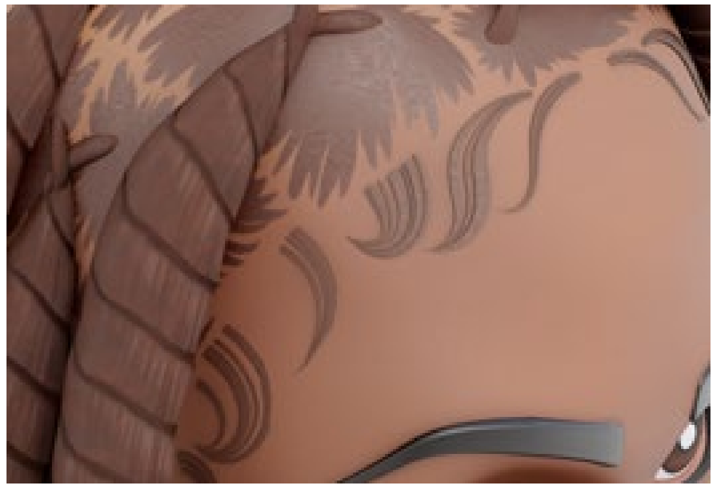
// Loc'd Bob

3D Design

Curl Type Modeled: **4B/4C**



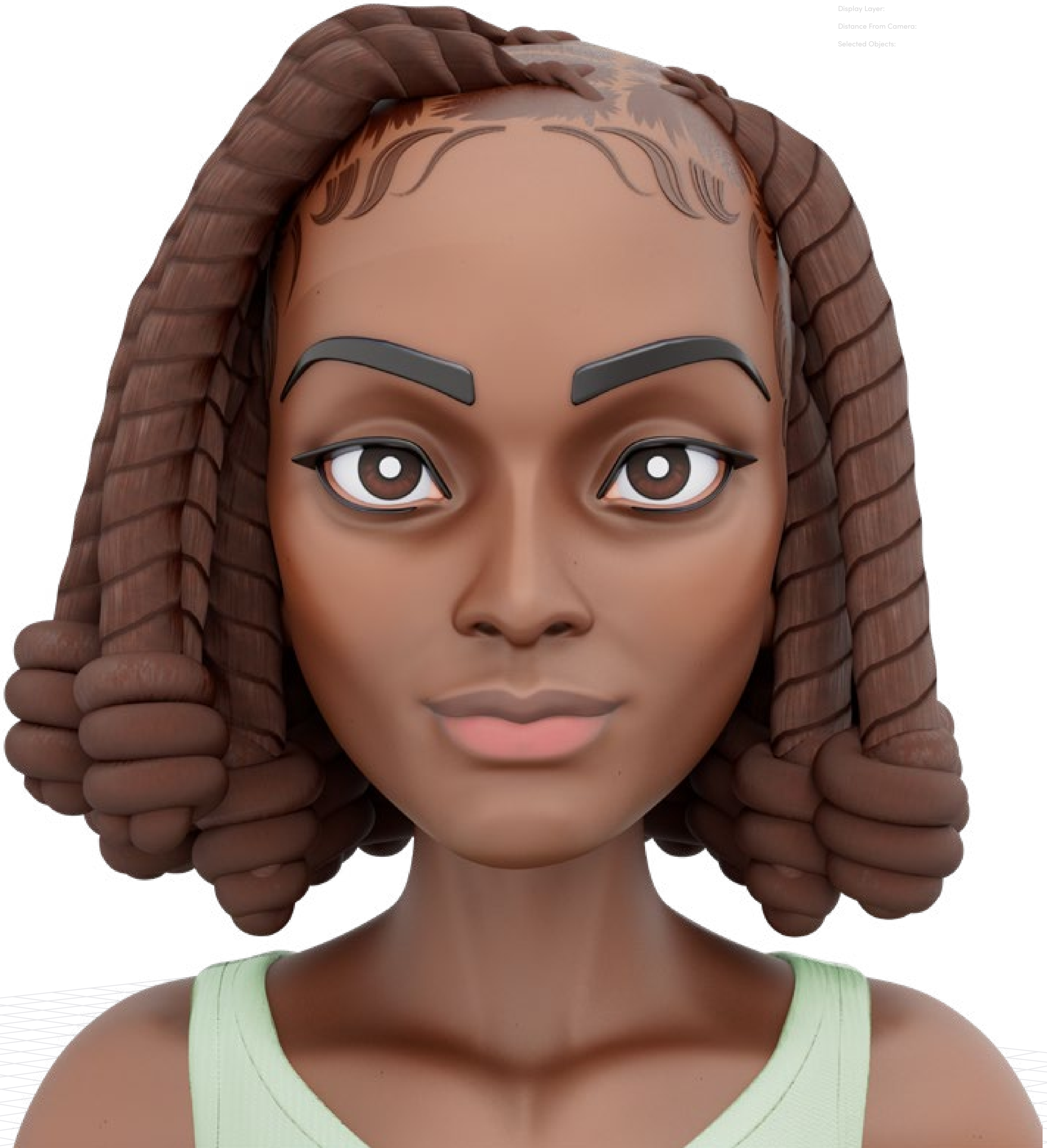
// 01



// 02

Download 3D Render

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

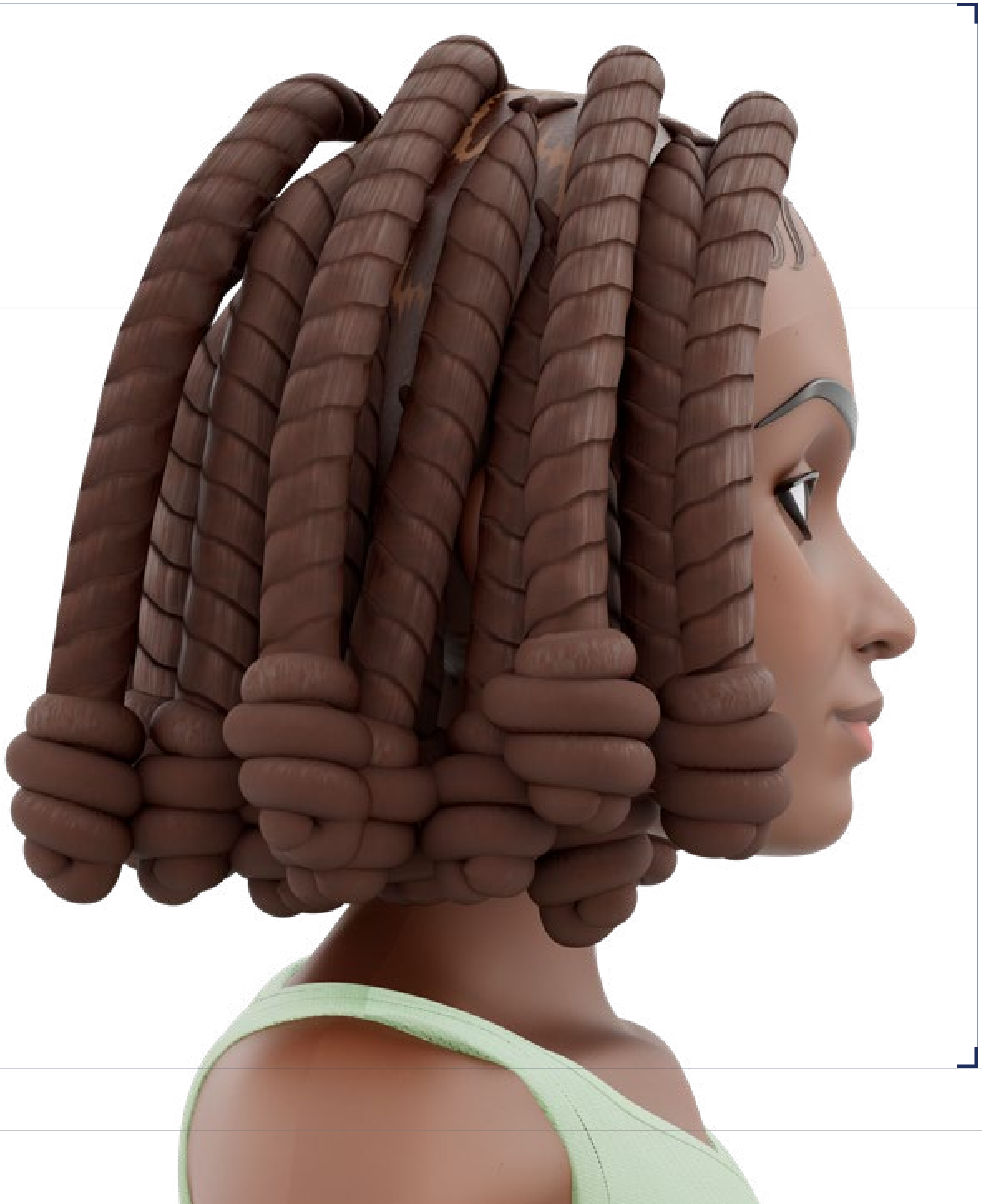
Project

// Loc'd Bob

Settings

Program Guide

Sculpting and Modeling: Blender
Texturing: Blender and Substance 3D Painter

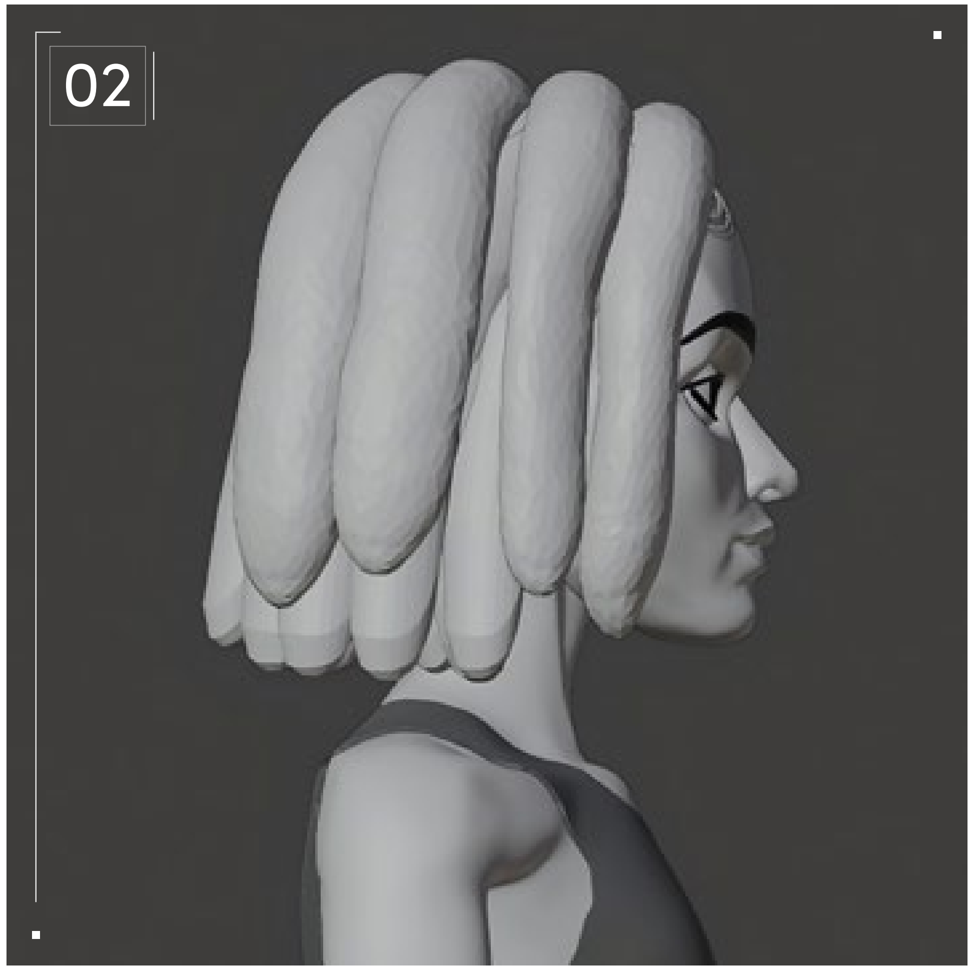


Step-by-Step Guide

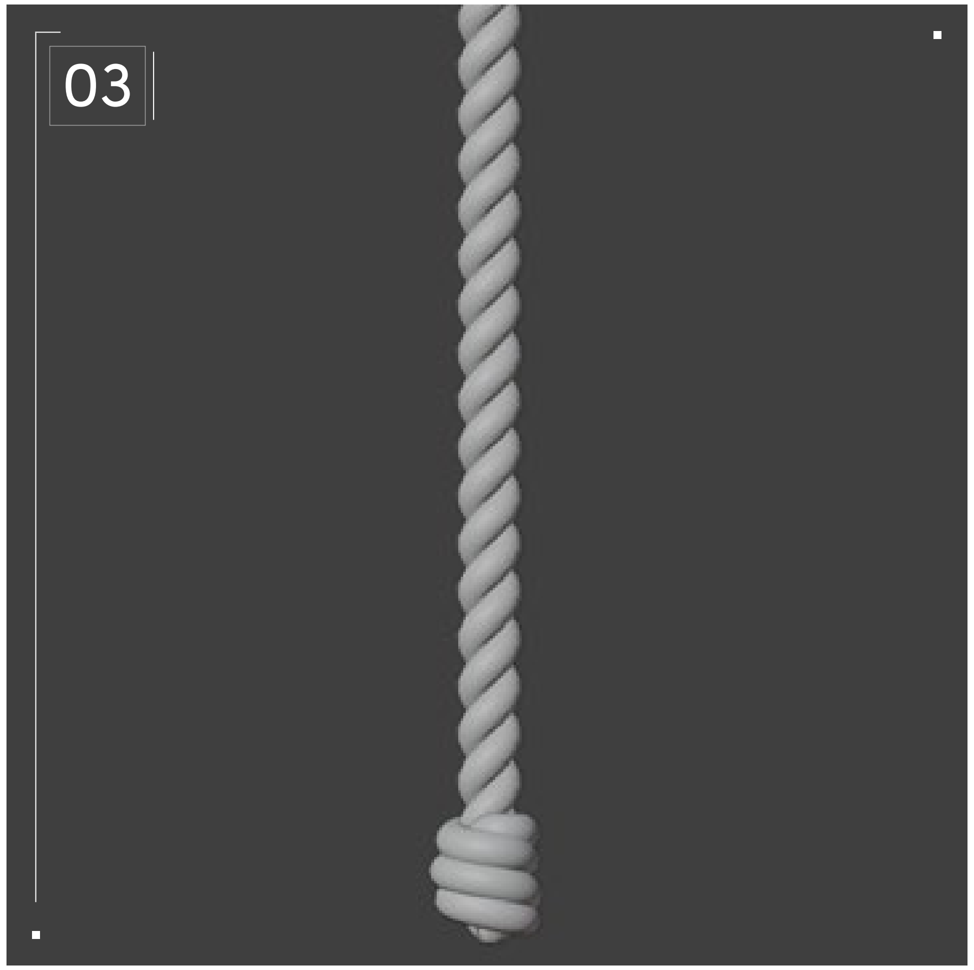
01

I always start by compiling a lot of visual references to enhance my understanding of the specific hairstyle I am creating. In addition to photos, I recommend looking at video tutorials of the hair being styled, and watching how different shapes are formed.

Make note of the key components that comprise the hairstyle. For the Loc'd Bob, the main structures to focus on are the twist itself and the spun knot structure located at the end. These are the features to exaggerate in the stylized model.

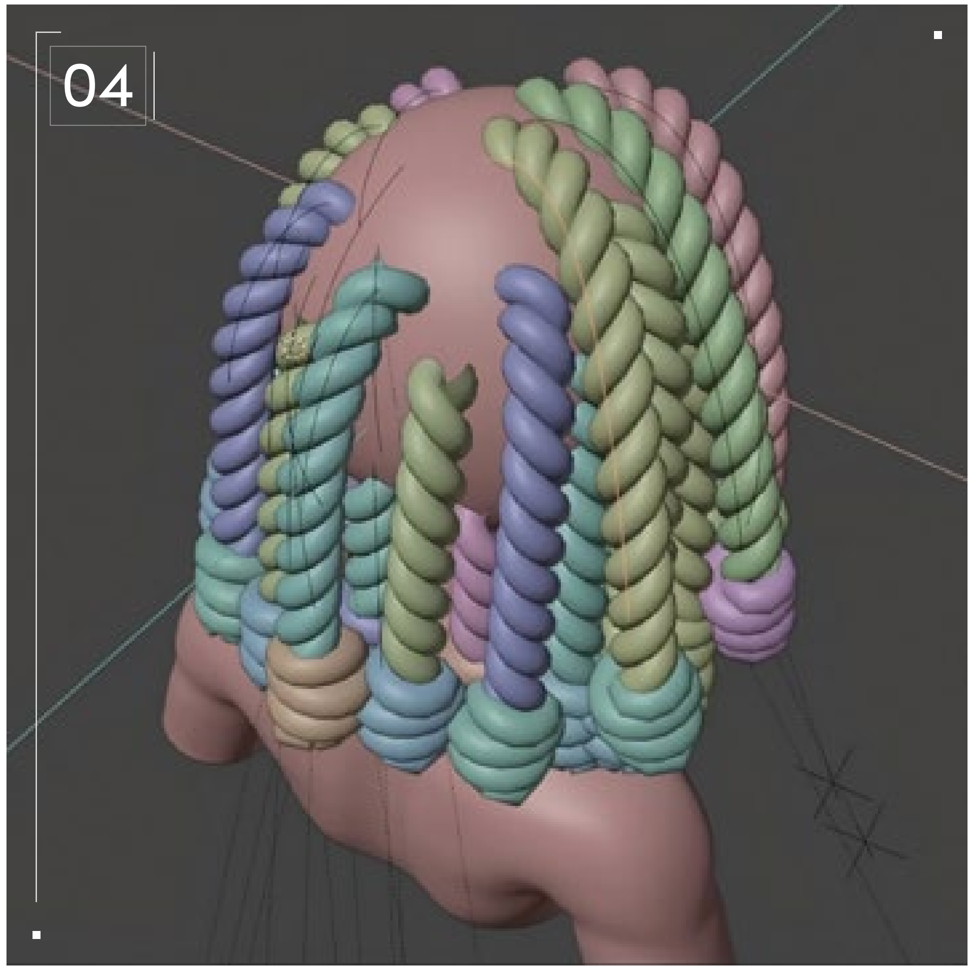


Once I can comfortably visualize the hair, I start blocking out shapes in 3D. First, I set up a base character to serve as the foundation for the hair. I then establish a silhouette for the bob using primitives spheres providing a basic shape for my hair sculpt. Using a path curve, I add some baby hair for a more detailed and polished look.

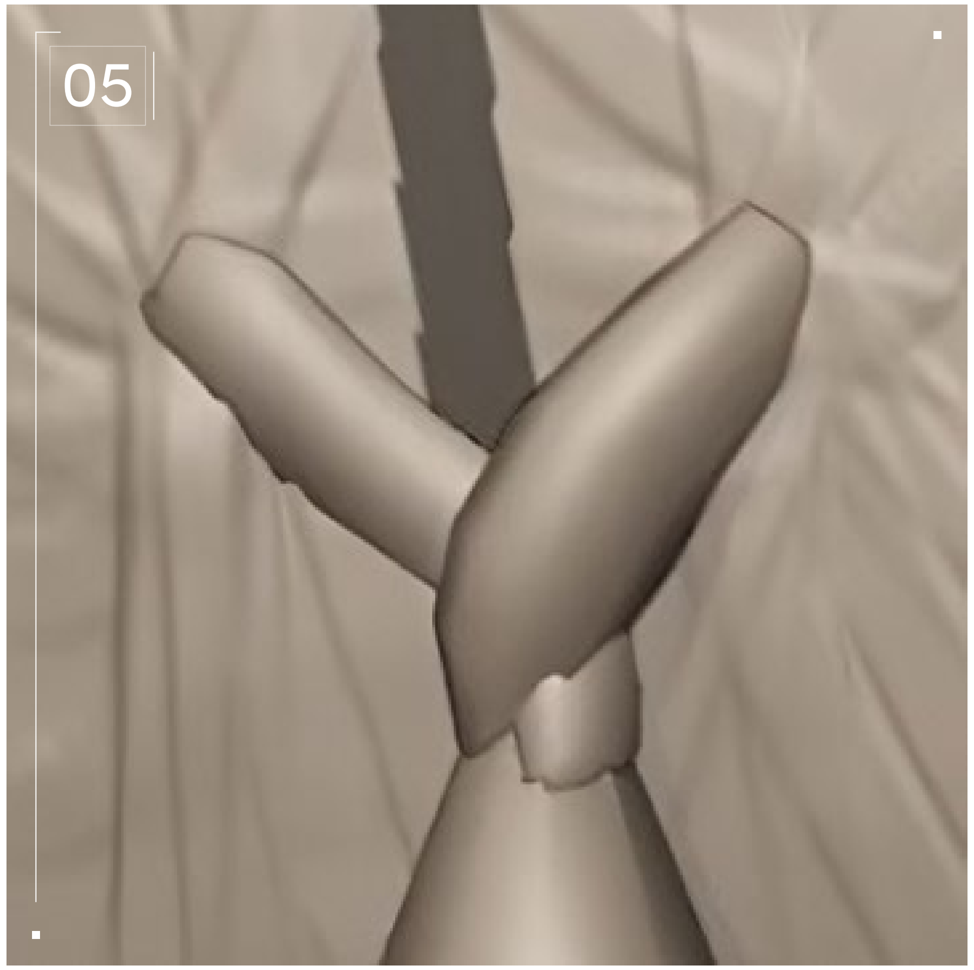


To model the two-strand twist hairstyle, I start by creating the twist itself using a mirrored circle combined with Blender's helix modifier. I form the knots of the Twists by utilizing a curve with a round profile. Next, I model a lower-poly version of the twist and create a cylinder that fits over the high-poly twist. Use your preferred baking method to transfer the details from the high-poly twist to the low-poly one.

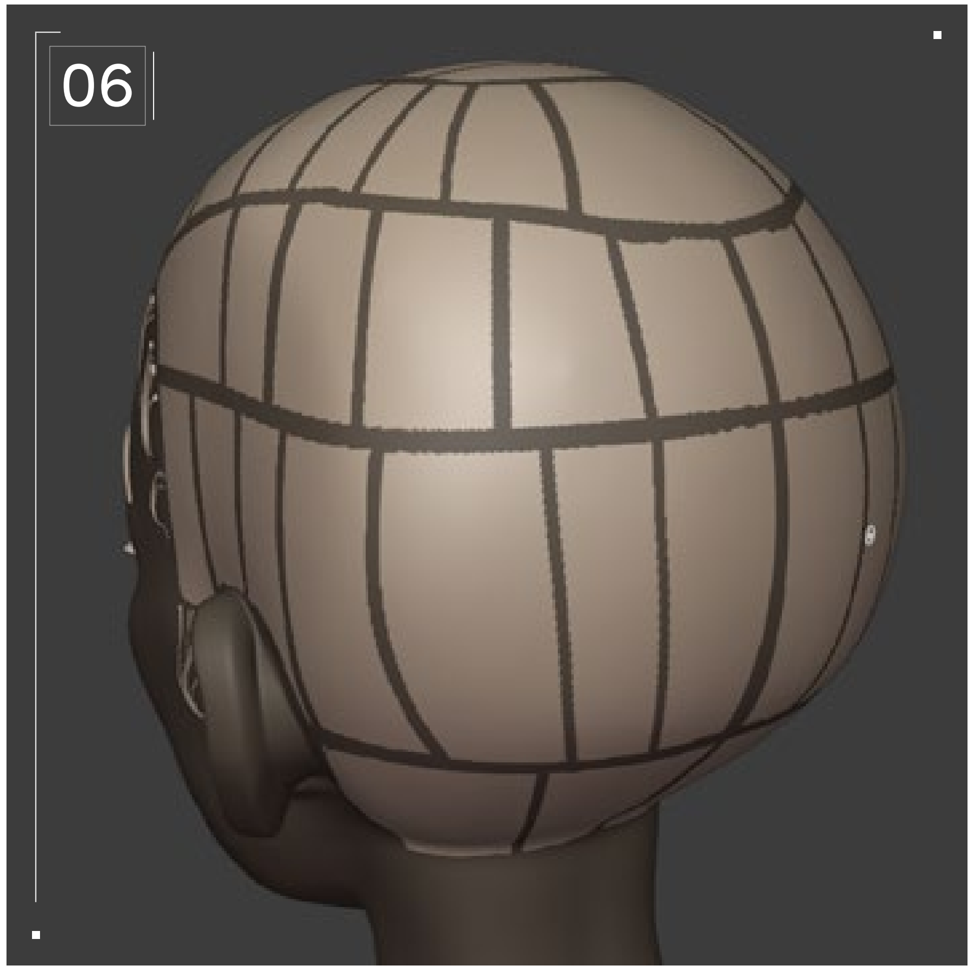
Step-by-Step Guide



Next, we'll position the low-poly strands around the head to match the previously established silhouette, ensuring a cohesive and accurate representation of the hairstyle. Be sure to leave a bit of a gap between strands for more accurate parting, especially where the style splits asymmetrically, with the result being it will look more real.

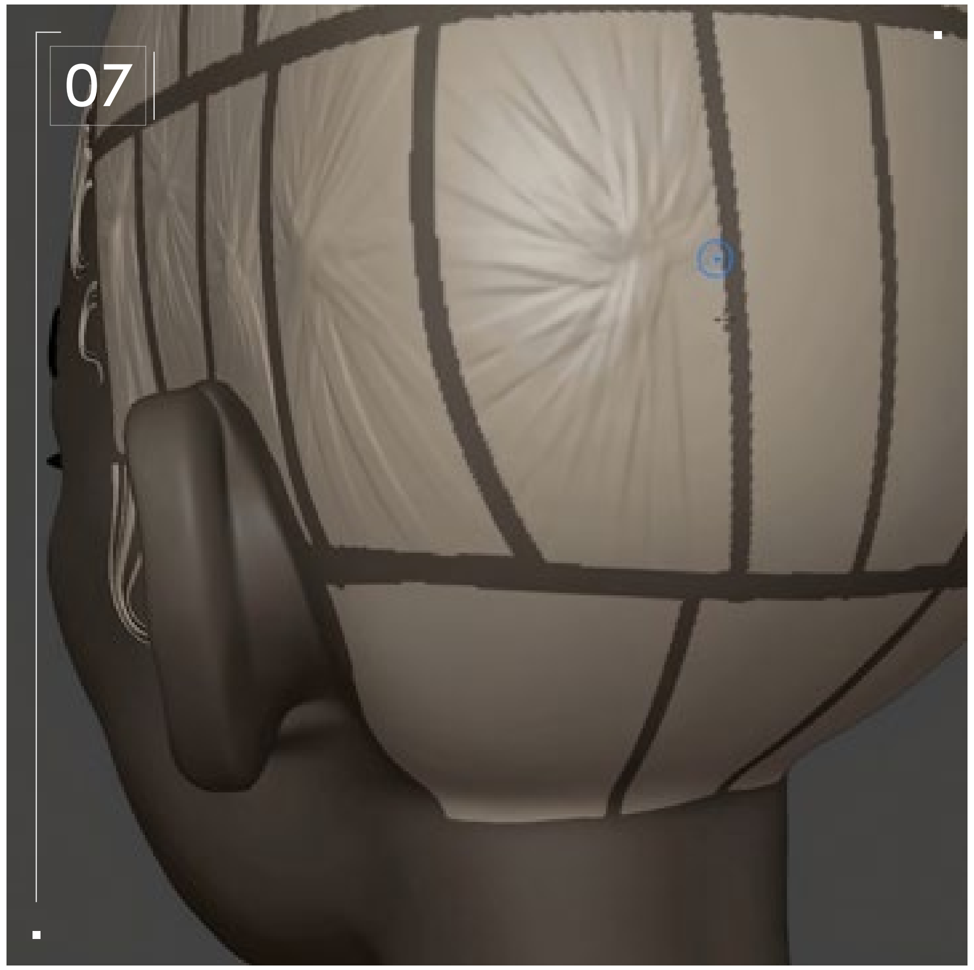


Once the low-poly Twists are placed, you can begin creating the lead-in model for the partings. Those shown were created with two sphere primitives with sculpting details. It is important to ensure they appear to twist out, mimicking the appearance of the twisted strands themselves. This can be achieved by carefully shaping and positioning the lead-in model.

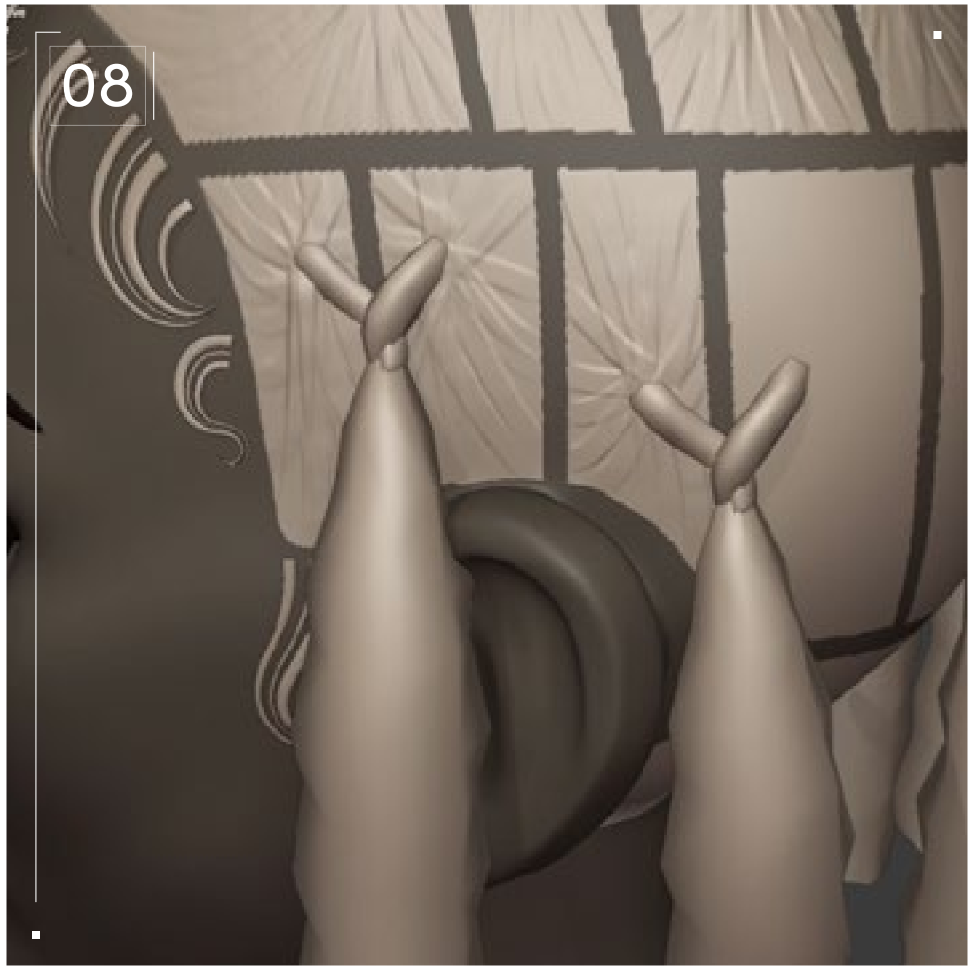


Using the mask tool, I then create a pattern on the scalp that corresponds to the location of the strands themselves. With the mask still active, I then sculpt in the parting details, ensuring precision and attentiveness to the desired design. I sculpt my strokes from the outside of the parting divisions towards the lead-in model. This technique ensures a seamless transition from the outer areas towards the central parting. Be careful not to sculpt directly on the mask, or hard edges will appear! If this happens, smooth over the edges with the smooth tool to ensure a natural transition from section to section. Along the hairline, use curved lines to simulate blended edges and the effect of moisturizing products that people often like to use on their hair in real life.

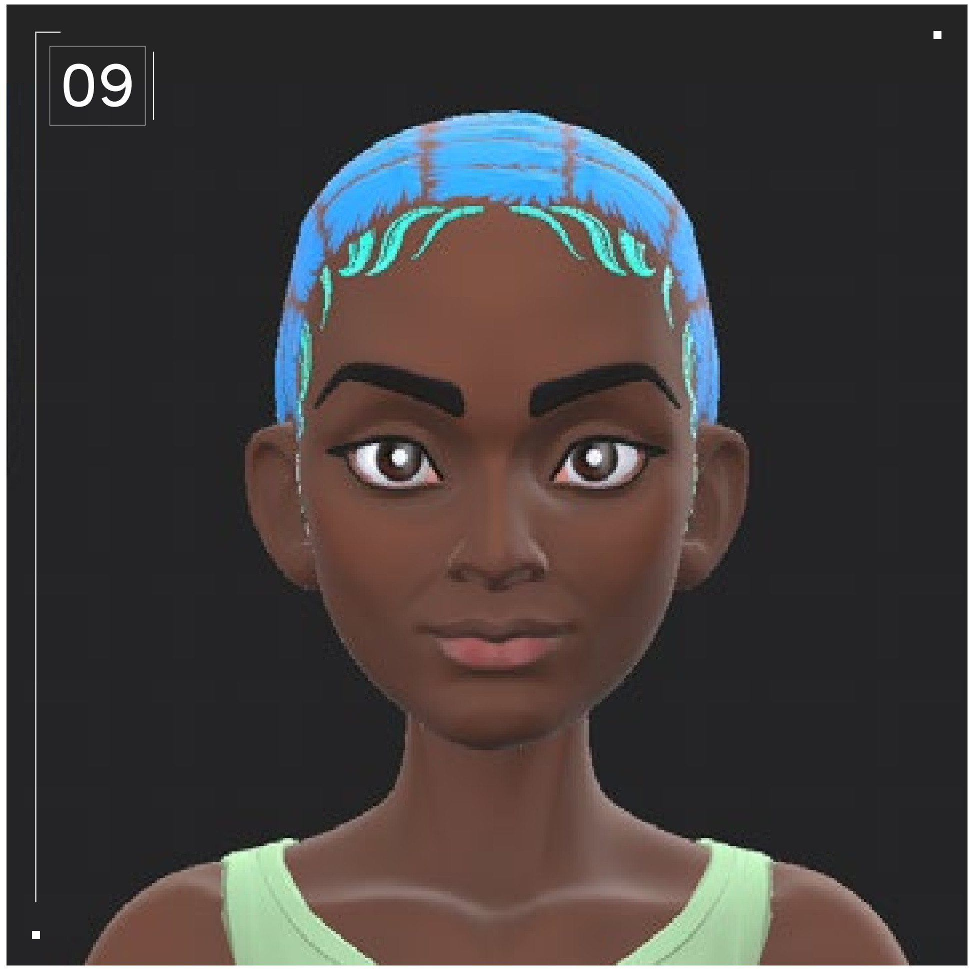
Step-by-Step Guide



When all of the sculpting is complete, I retopologize the model, reducing its geometry to ensure a clean and efficient structure before I begin texturing. Since Blender's node system and Substance Painter will be used, and I don't have any specific parameters to hit, I set a target of <50,000 tris.



I'm using Blender nodes to effectively generate the texture for this sculpt. Noise textures can generate the striated texture that Locs typically have. This noise has an impact on both the color and reflectivity of the hair, giving off a repeating pattern that resembles interlocking fibers. I use another noise texture to drive the variations of color and reflectivity.



Once the texture is complete, I bake it onto the low-poly model, ensuring that the details are accurately transferred. For the model itself, Substance Painter was used to color in both the intricate parting details and the remaining skin. From baking the high-poly model onto the retopologized low-poly model, the sculpted parting details are preserved onto a normal map. This step allows for precise and realistic coloring, adding depth and dimension to the final result. By leveraging the capabilities of both Blender and Substance Painter, the hair texture and coloring can be achieved with optimal quality and visual impact.

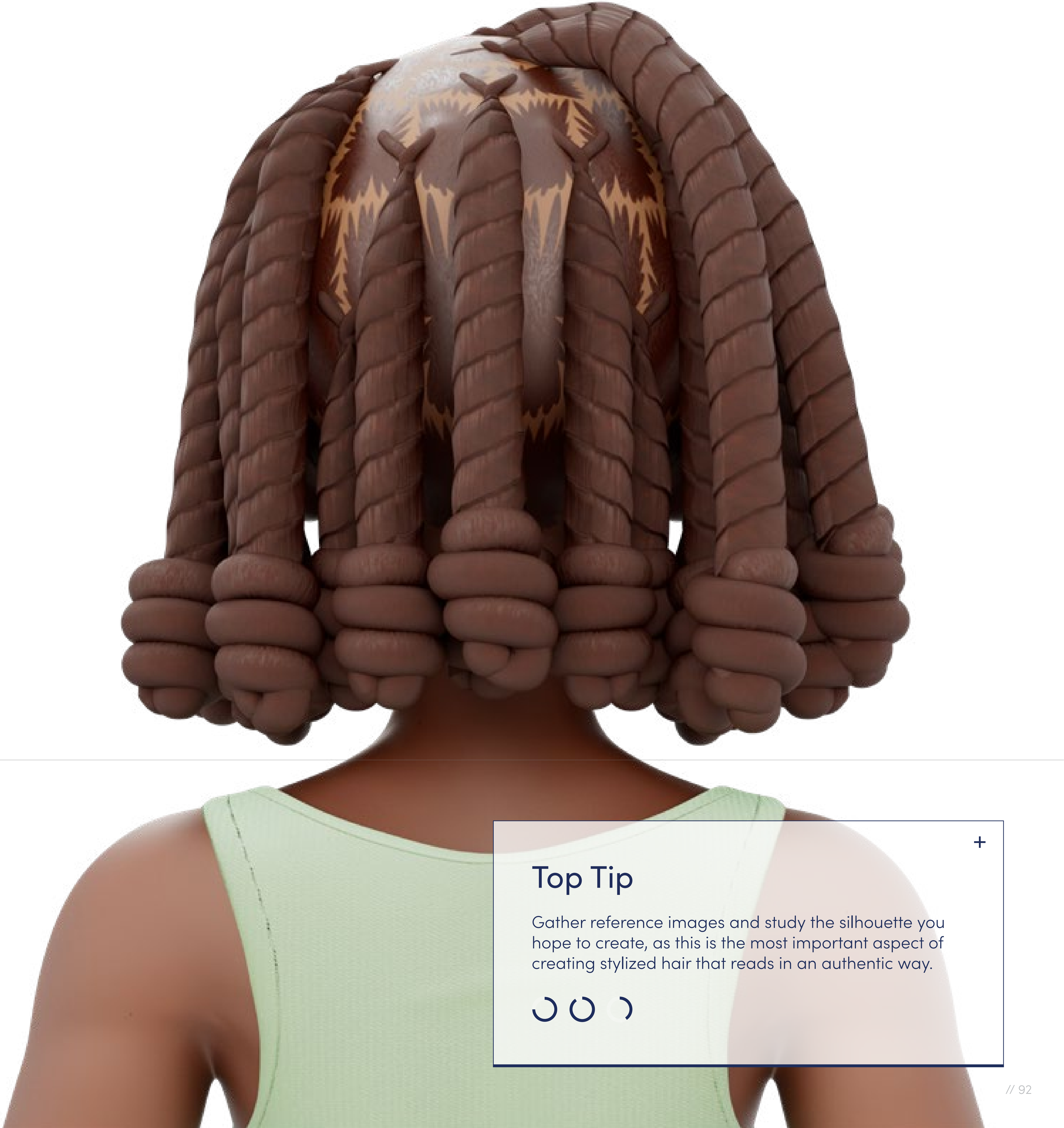
// Project Settings

Technical Watchouts

To achieve this style, you'll need to have a basic understanding of sculpting, as well as baking detail from high-poly to low-poly.

For different hair textures, adjusting the materials to either be more or less rough would be the best way to showcase hair type variability. The hair in the Loc'd Bob style was retwisted, which tends to pull and straighten out a lot of texture, so I set my materials accordingly. When creating this hairstyle in 3D, it's important to ensure that the topology of the low-poly meshes is of sufficient quality to support the baking process. This involves optimizing the mesh structure to avoid distortion and artifact issues during the baking stage.

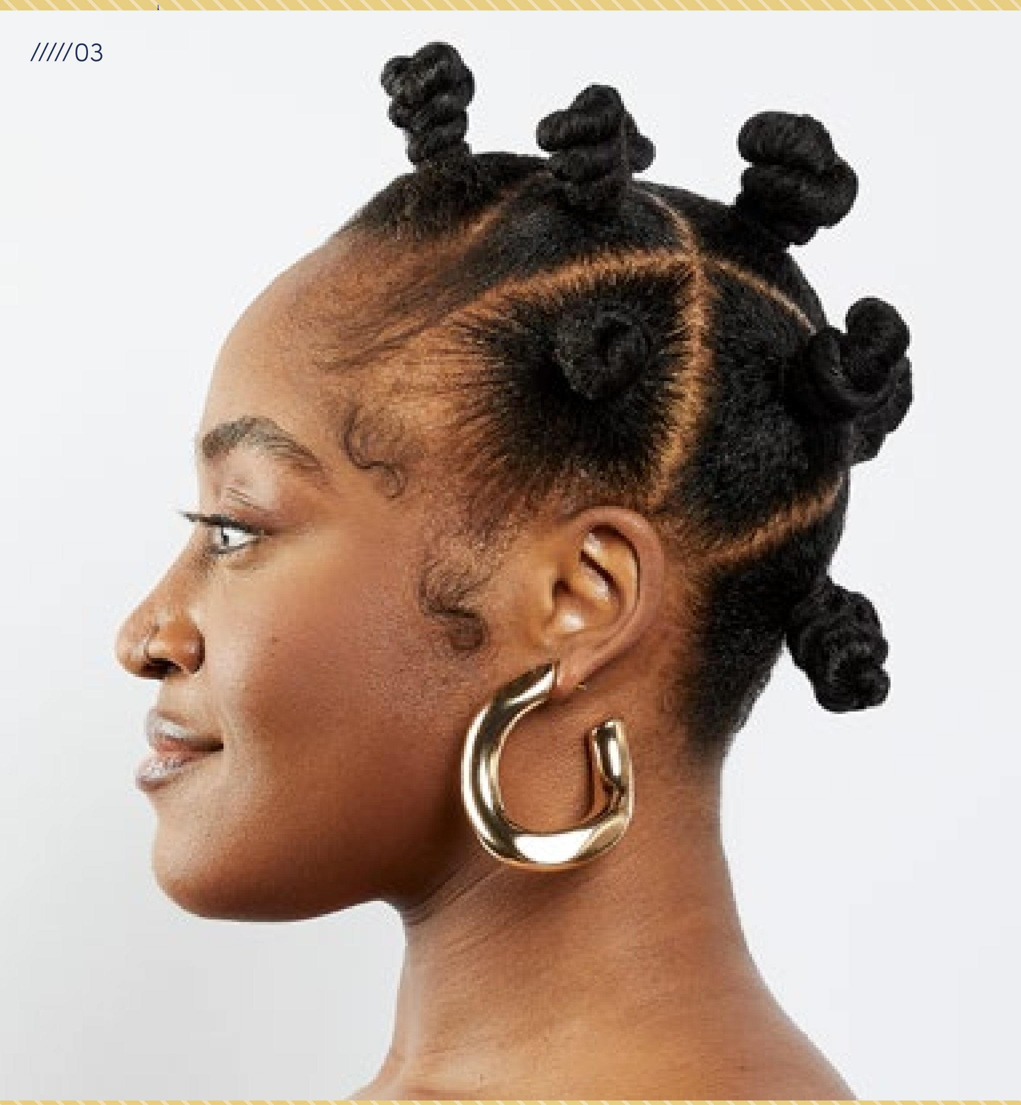
Don't be afraid to sacrifice realism to prioritize the silhouette and overall visual impact. This is the key to creating low-poly stylized hair that reflects intricate real-world styles.

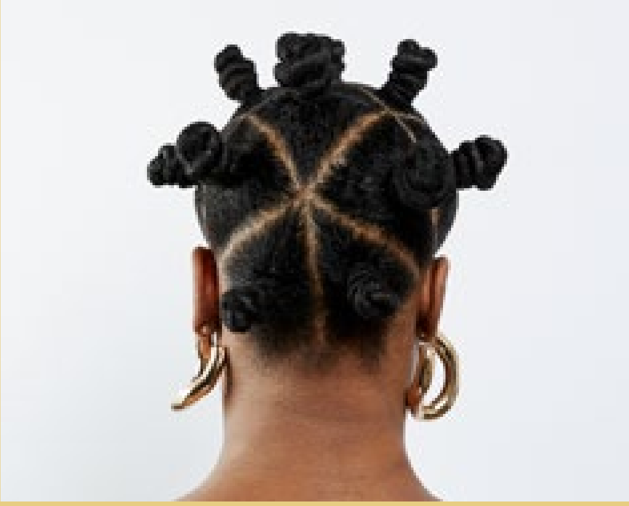


Top Tip

Gather reference images and study the silhouette you hope to create, as this is the most important aspect of creating stylized hair that reads in an authentic way.

Bantu



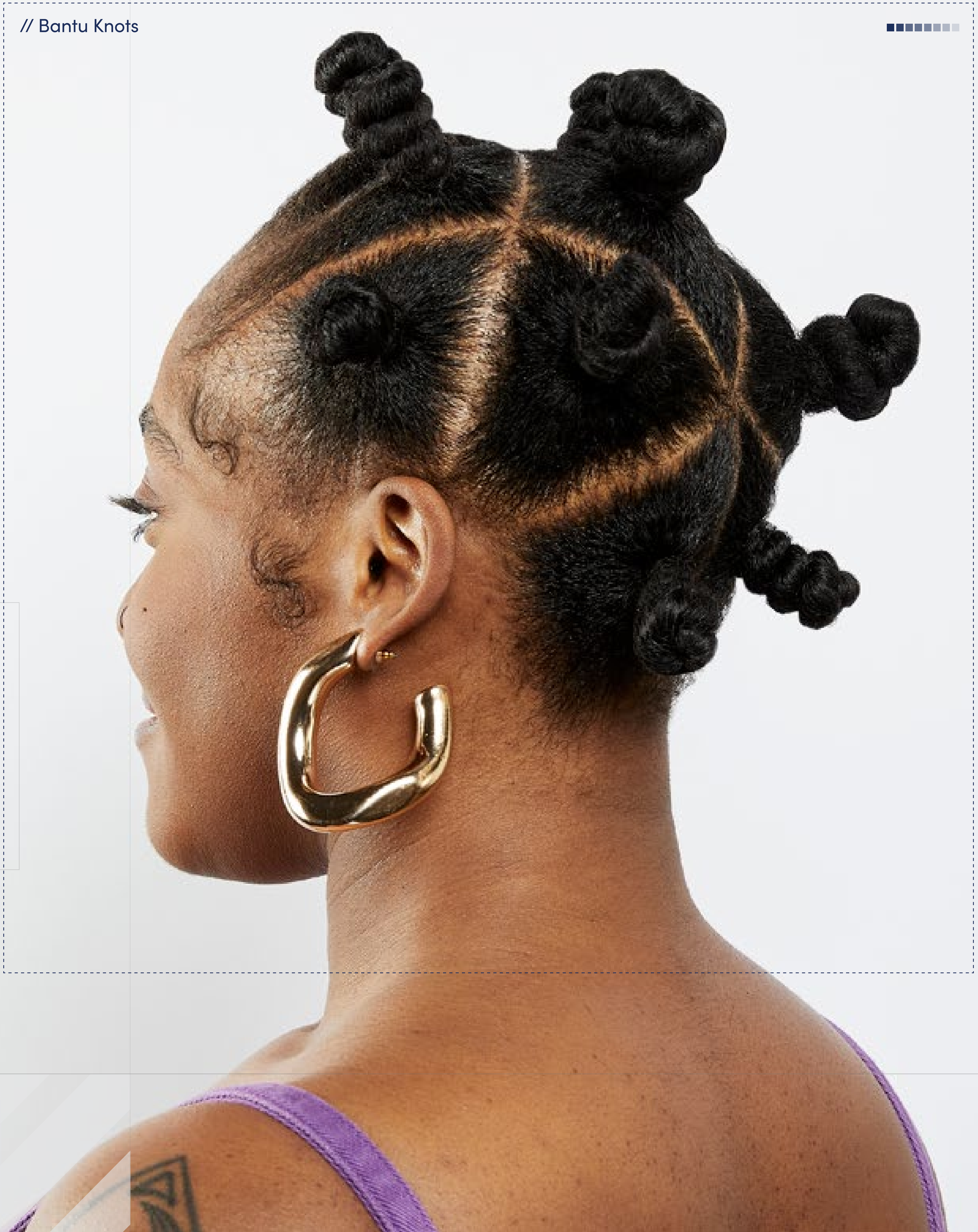


Knots

Knots
Knots

Download Images

// Bantu Knots

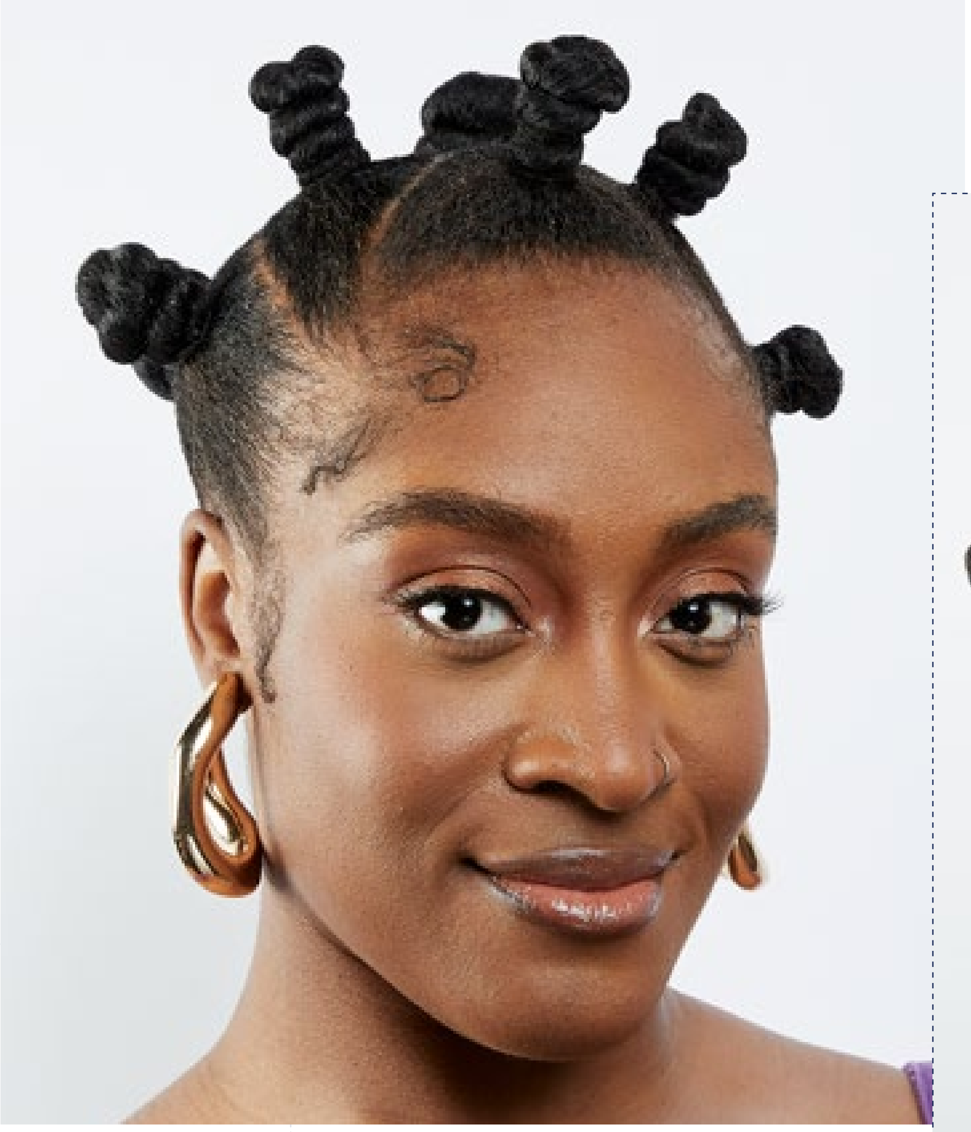


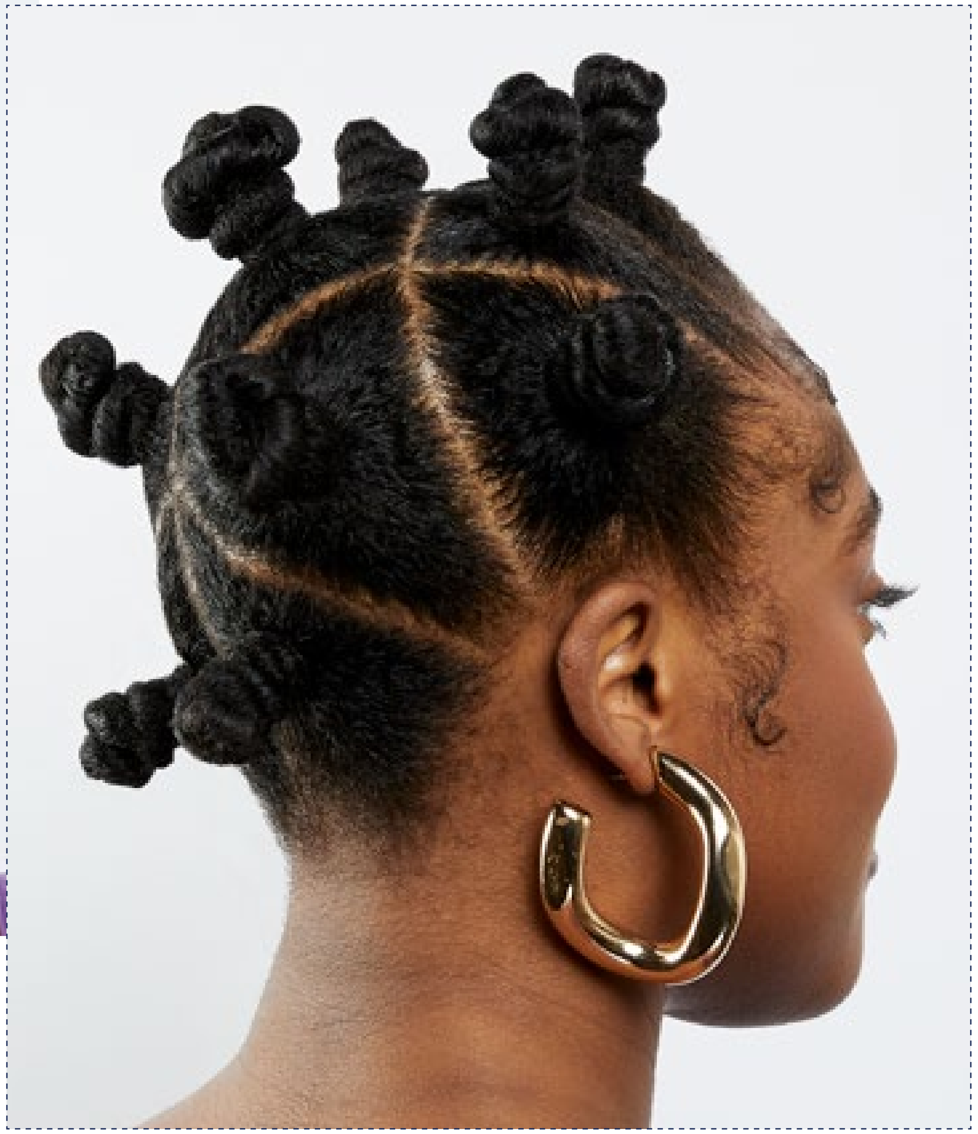
// Hairstyle

Bantu Knots

Style Background: Bantu Knots are created by sectioning your hair into several one-to two-inch sections and tightly twisting each section of hair from the roots to the end. Each twisted section is then wrapped into a small bun shape and secured with an elastic band. At one time, Bantu Knots were purely a protective style that people would wear around the house or to bed in hopes of achieving curls or waves once the knots are undone. However, Bantu Knots have transformed into a bold fashion statement in their own right. Bantu Knots are often styled with baby hairs, which are short, wispy hairs that sit around the hairline and are sculpted into curlicues or other designs to add detail to a style.

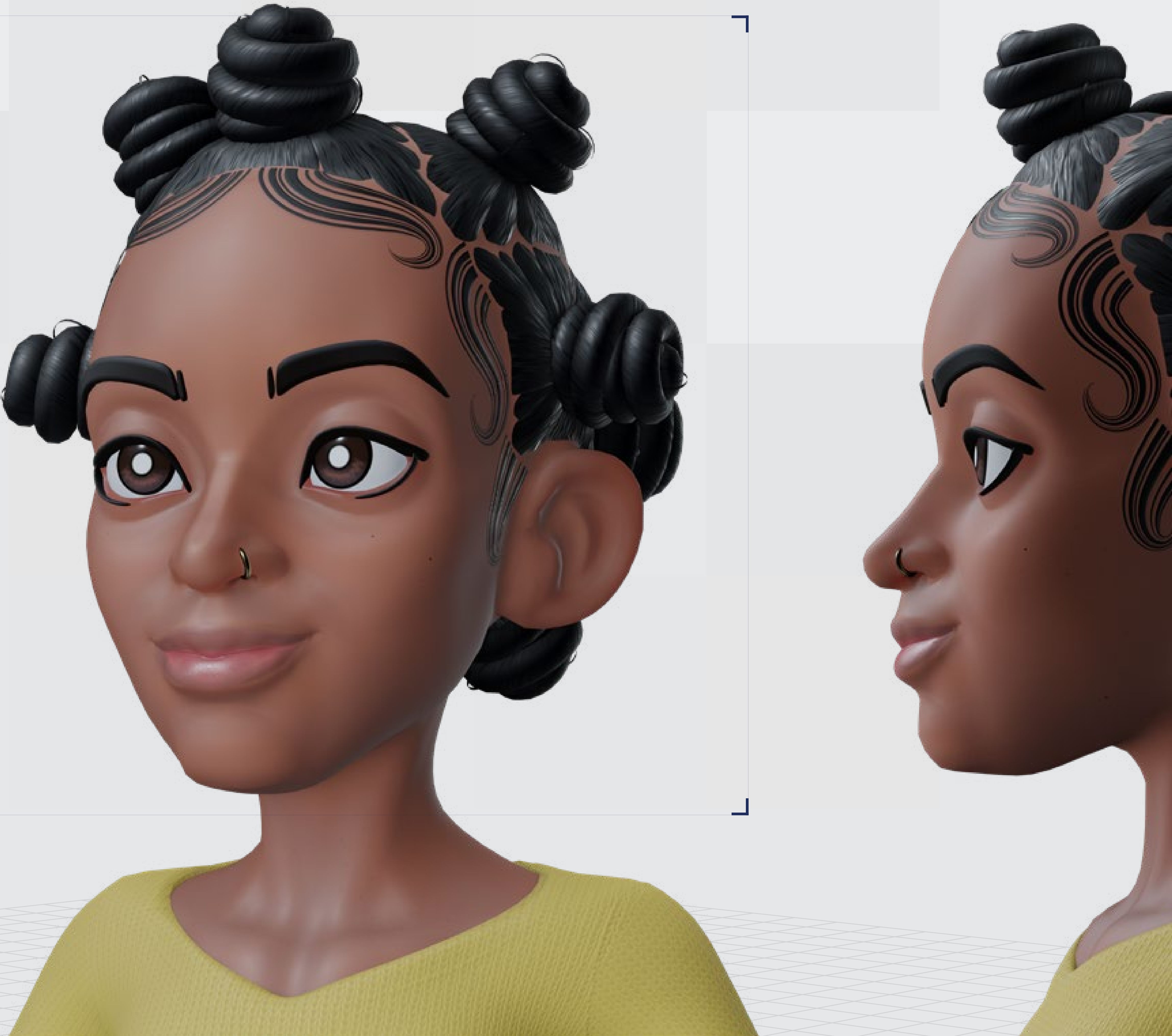
Download Images





3D Design

Bantu Knots

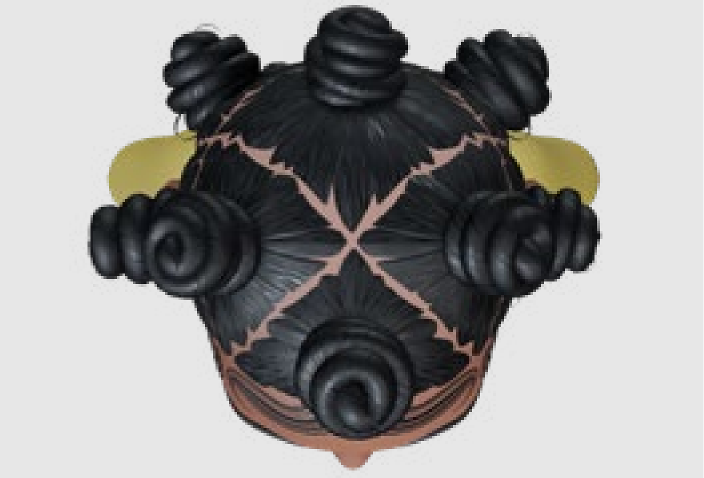


Digital Artist: **Sajhmori**

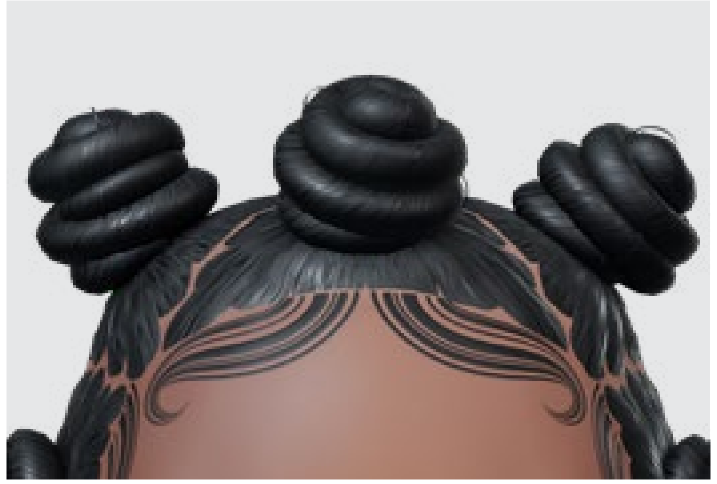
// Bantu Knots

3D Design

Curl Type Modeled: **4A/4B**



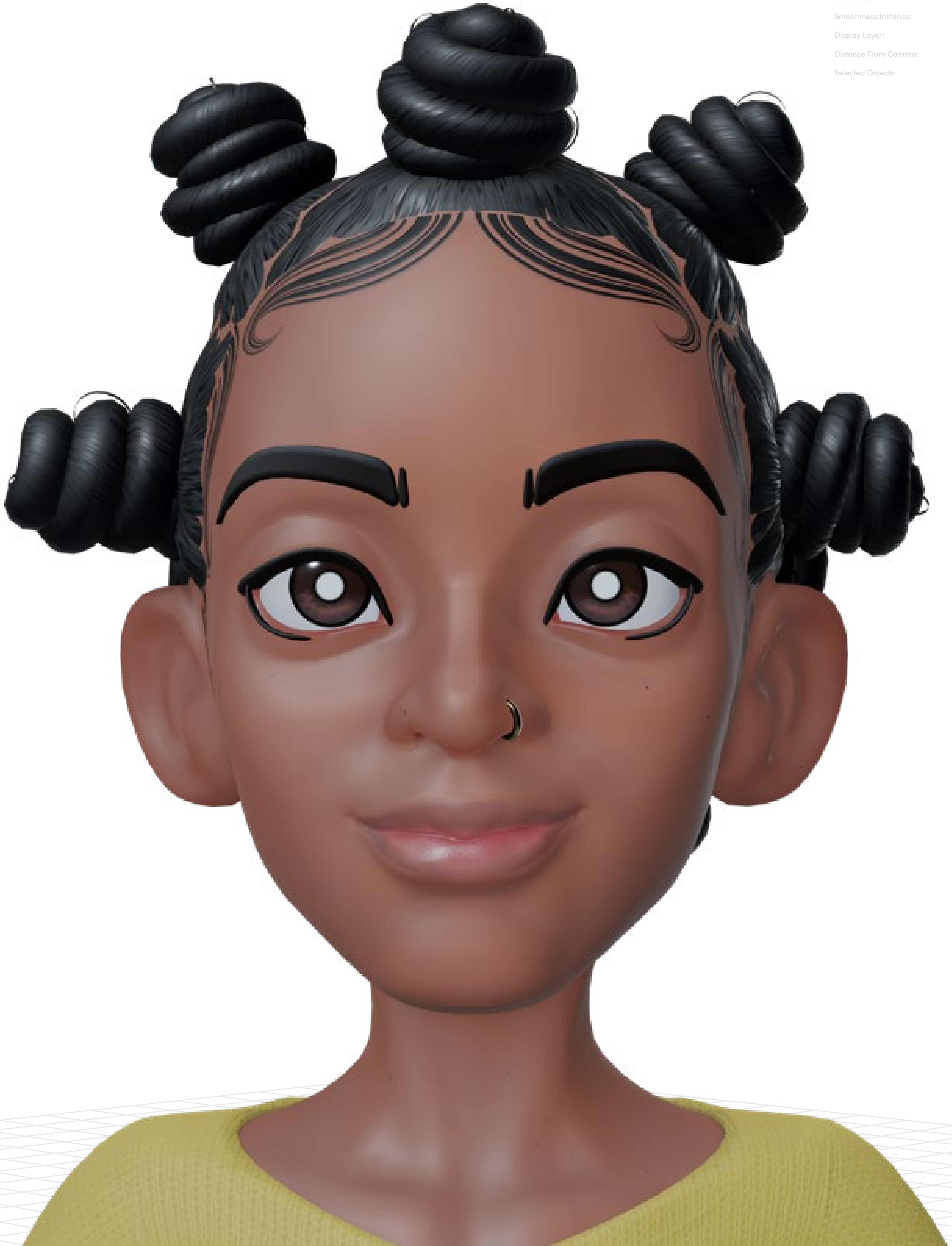
// 01



// 02

Download 3D Render

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

Project

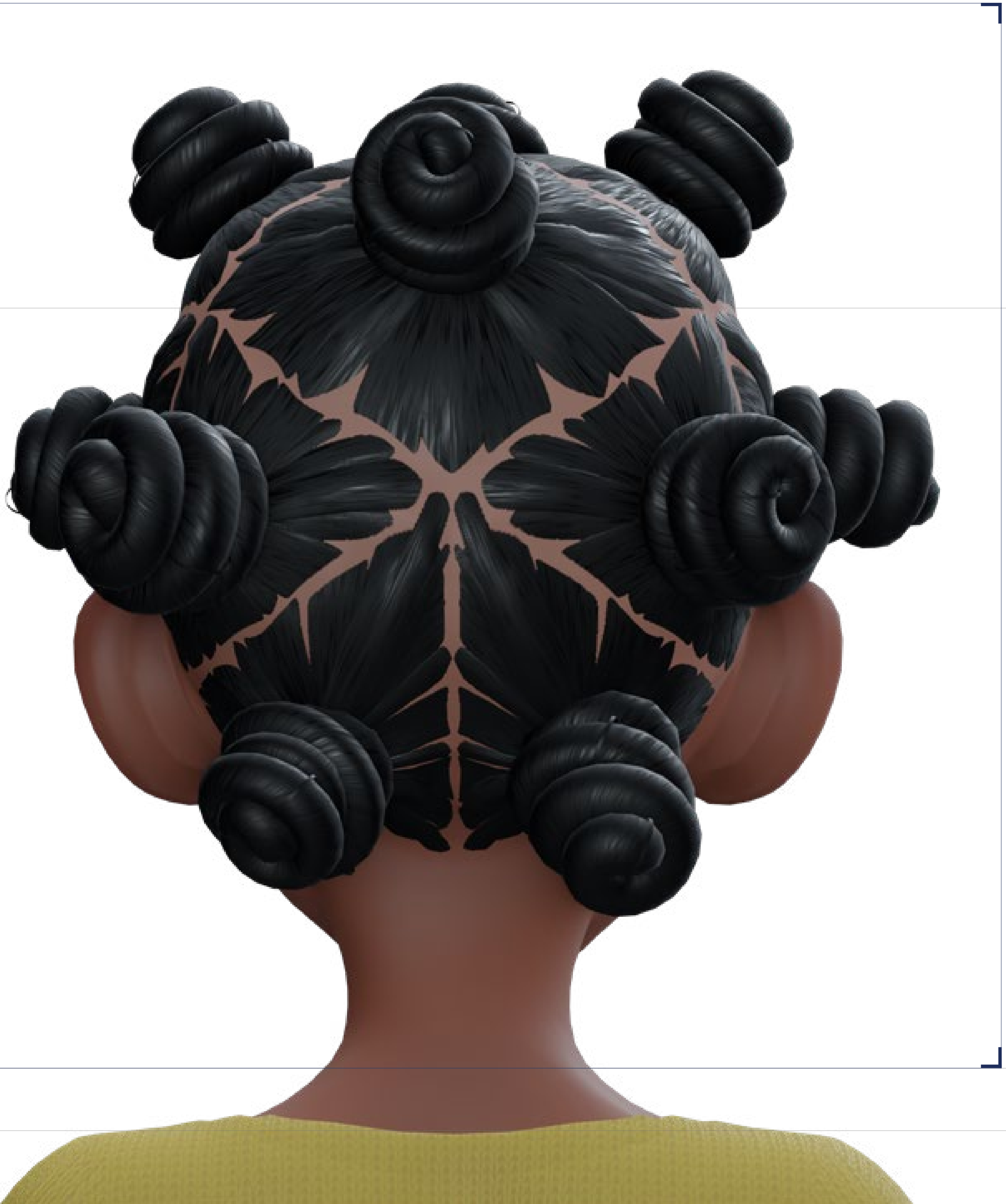
Settings

// Bantu Knots

// Project Settings

Program Guide

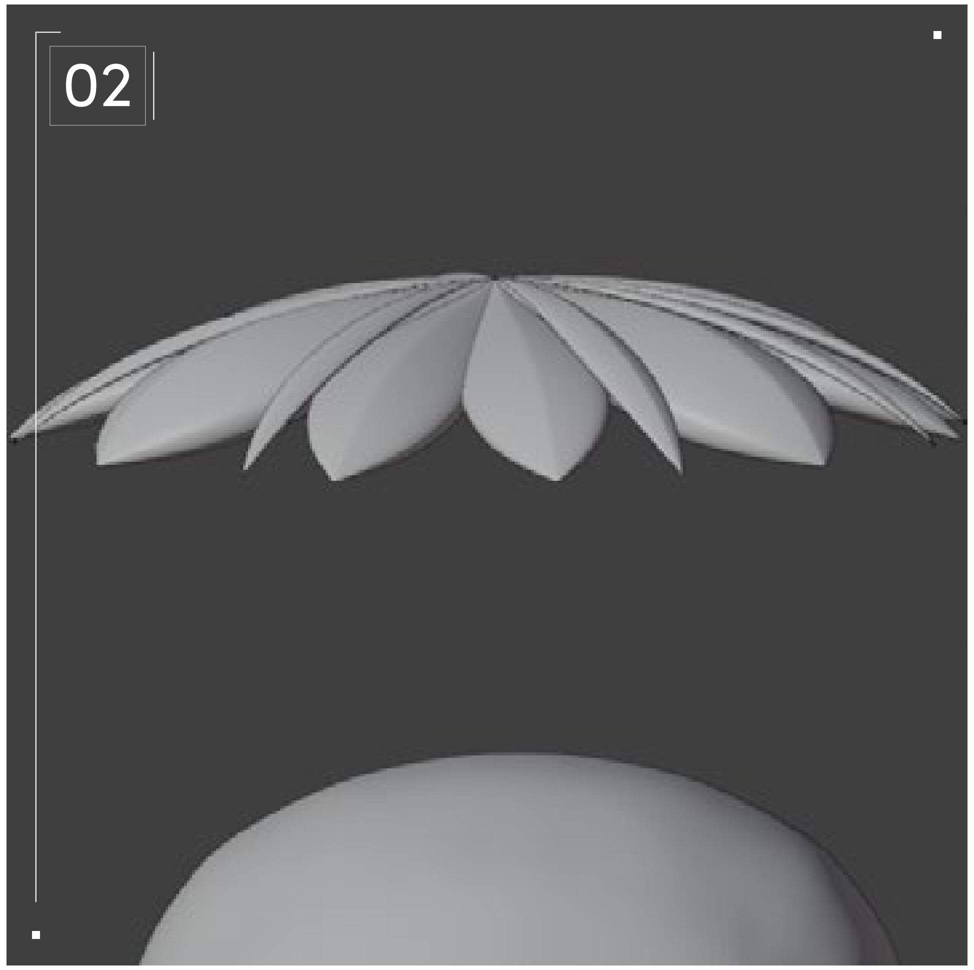
Sculpting and Modeling: Blender
Texturing: Blender and Substance 3D Painter



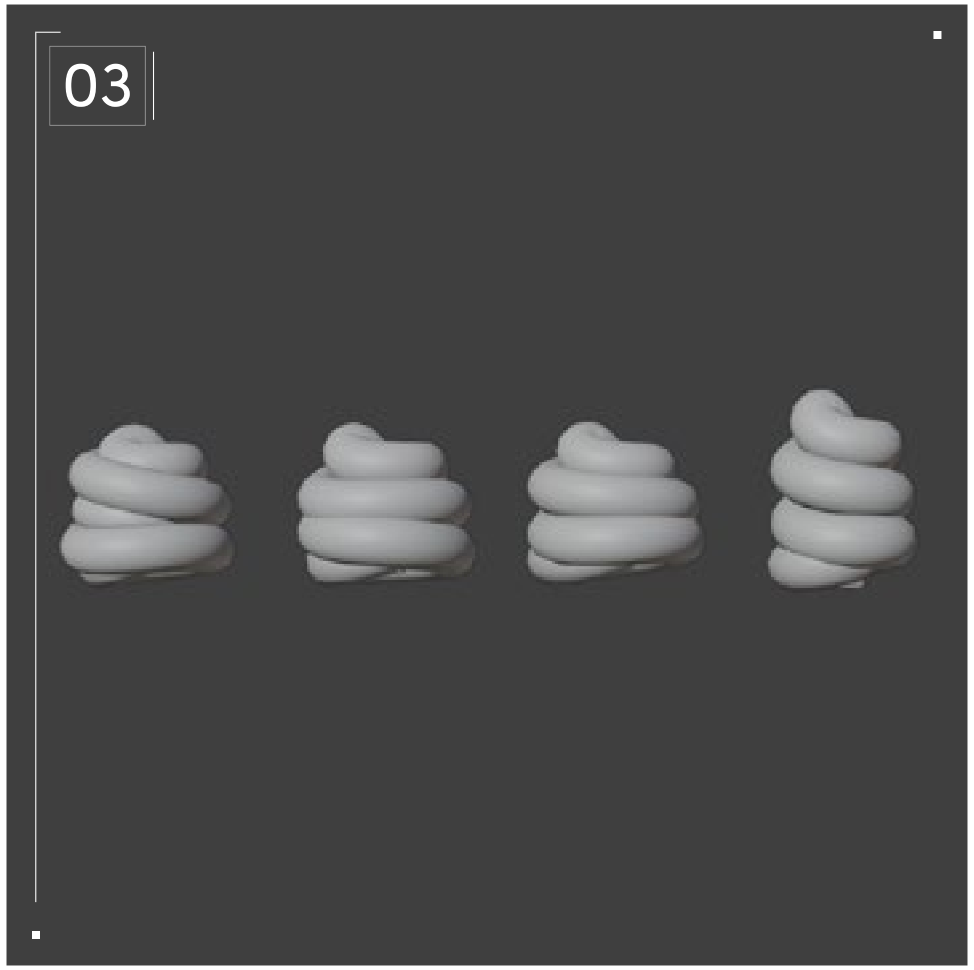
Step-by-Step Guide

01

I start off all of my sculpts by researching the style I'm trying to recreate, compiling photos and watching video tutorials to see how the style is created in real life. In addition to noticing how the key shapes are formed, I also take note of any products people use to help them to achieve the end result. This helps to understand the various details in shape, stroke, texture, or material that I'll need to set in order to create an authentic-looking style.

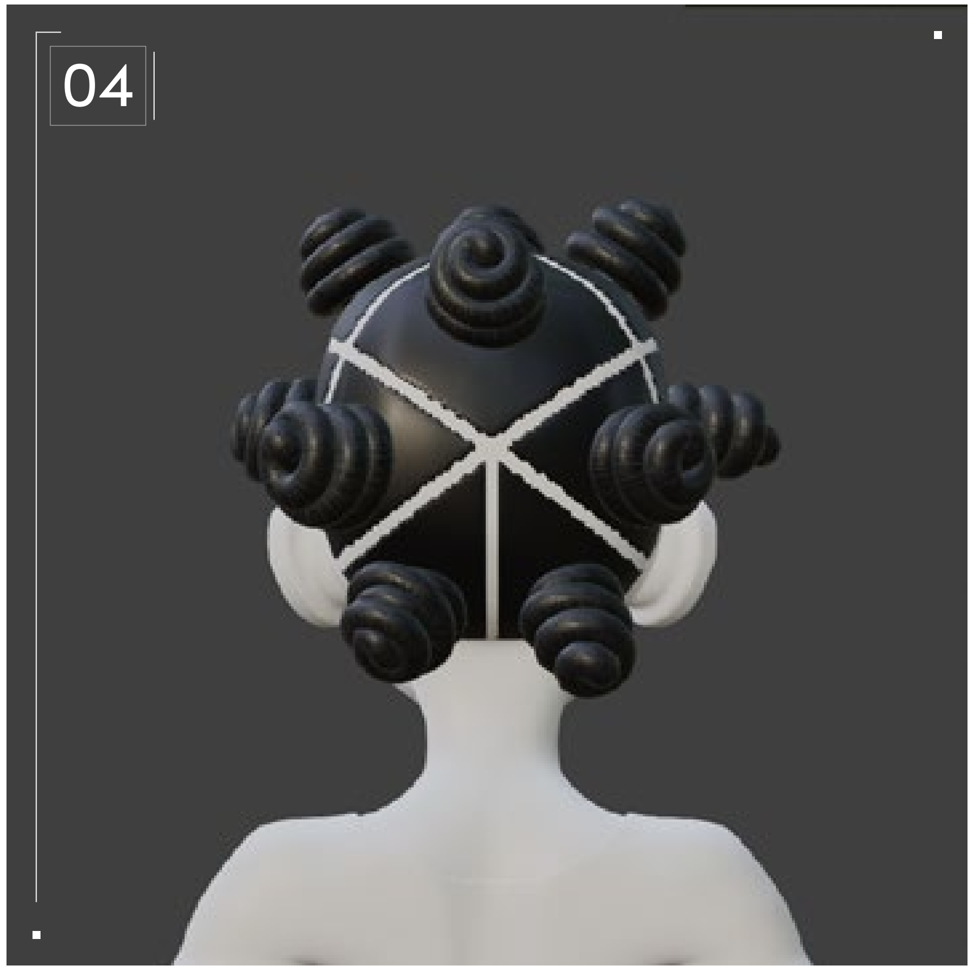


For Bantu Knots, consider drawing the knots to gain a better understanding of how they are wrapped. In addition to the knots themselves, pay close attention to the intricate details on the scalp. The parting and the way the hairline is styled are just as important as the main structures that comprise this style.

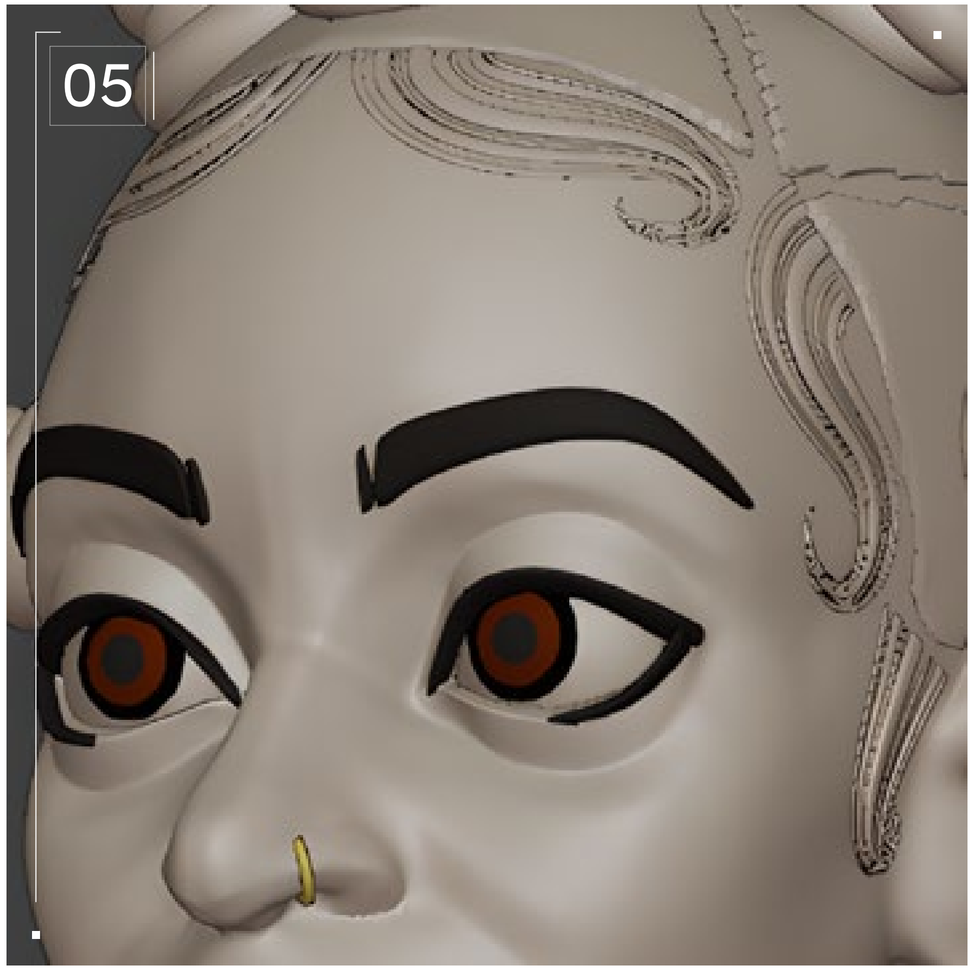


The main structures for this style include the twisted knot formations placed around the head and the pulled hair that constitutes each hair section. I enlarge these key features to make them more stylized.

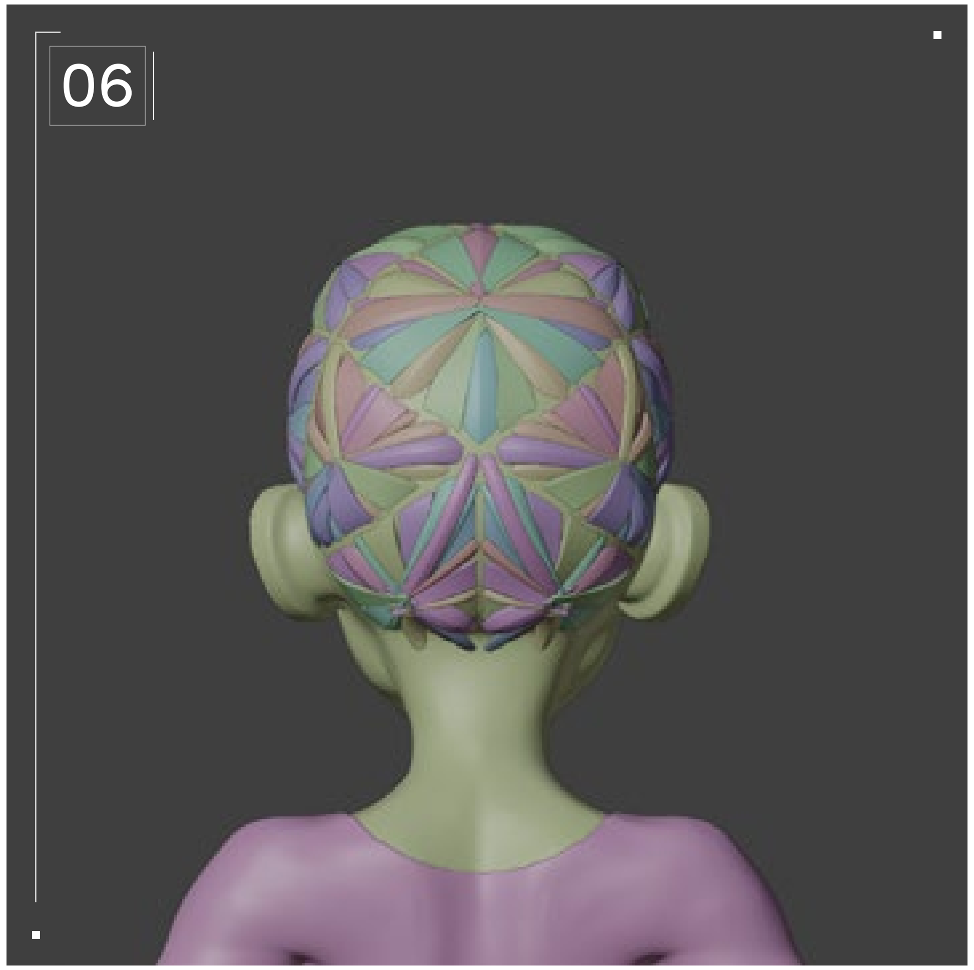
Step-by-Step Guide



Now, I set up a base character as a foundation for blocking my style. I then create the knots using a curve with a rounded profile, ensuring that the curve follows the path a real knot would take. This is where all of that studying helps! To match the stylized head, I make the knots relatively large in size, placing a knot in each hair section, using my visual references as a guide. Next, I use the mask tool to mark out the areas where the hair partings will be placed.

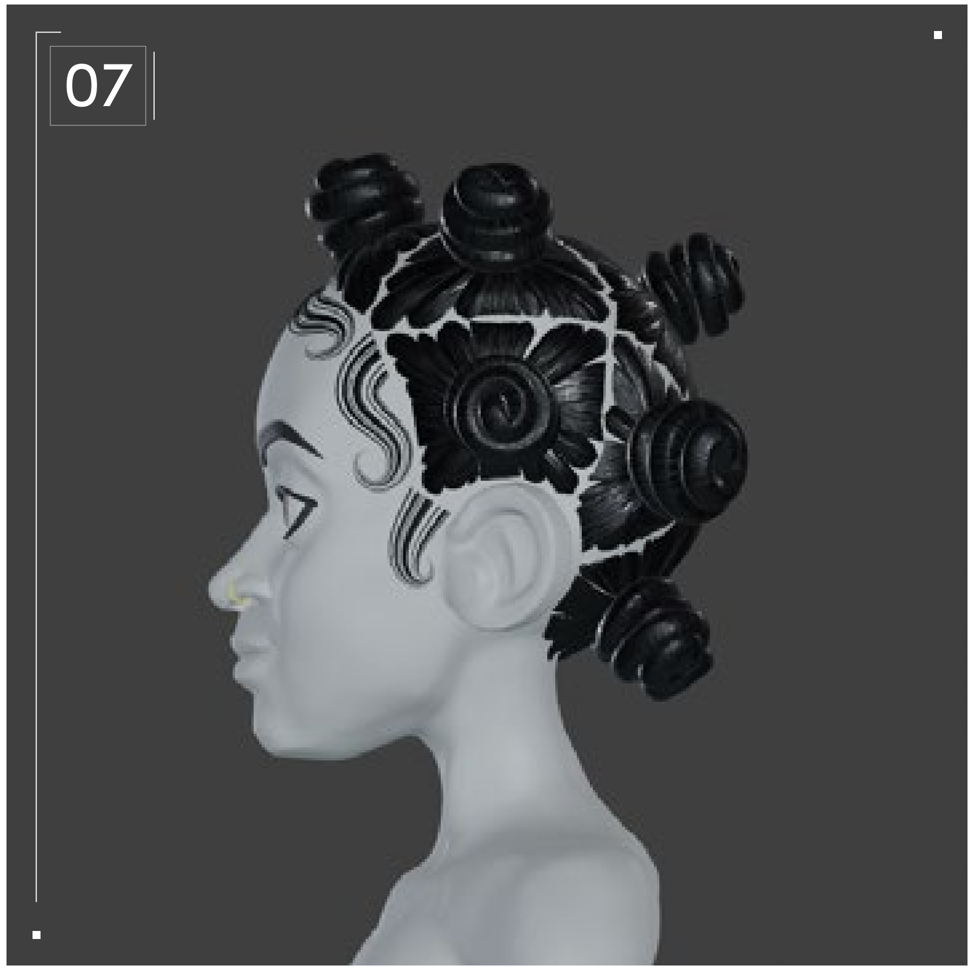


Since Bantu Knots can be a protective style, I added baby hair for a more polished look. I used a path curve with a custom profile just for baby hair, where the profile consists of separated blocks that mimic natural-looking hair strands. There are a lot of ways to style baby hair in real life, so use a variety of references and choose the shapes that best fit your style. I chose dramatic swirls to enhance the overall look.

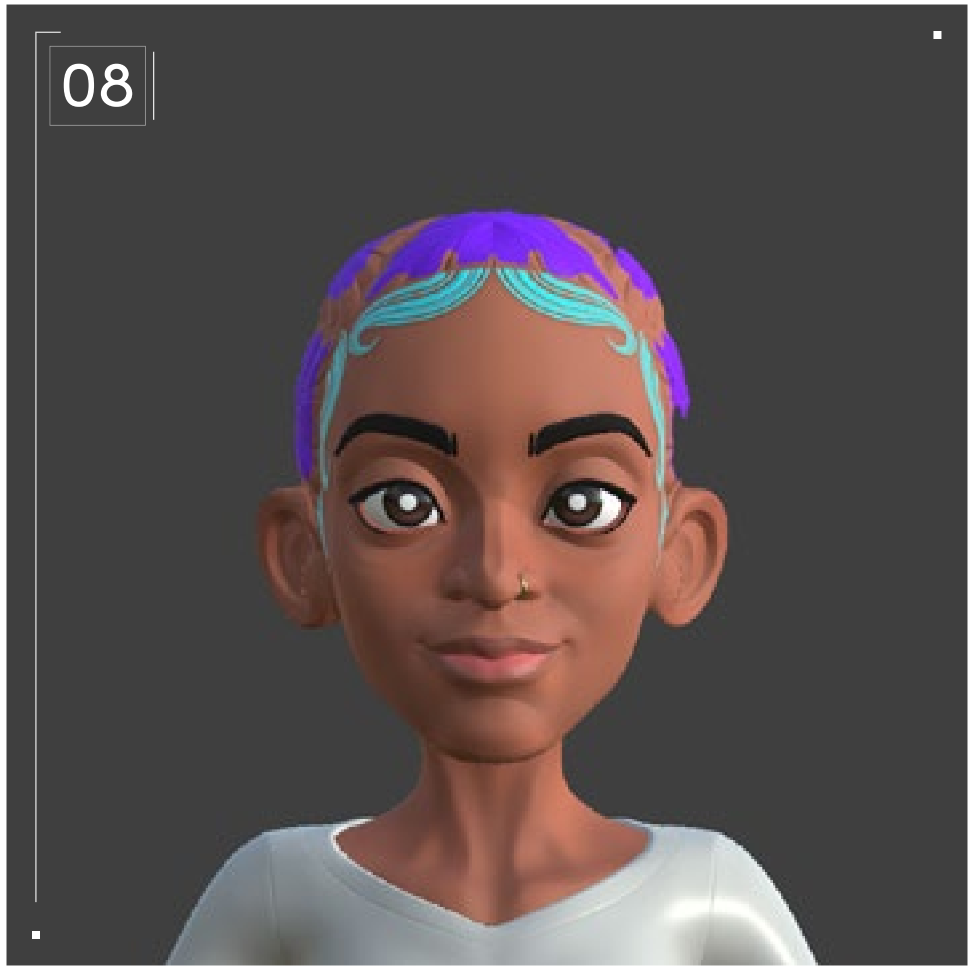


Once you're happy with your design, you can start modeling. To achieve a lower-poly count for the knots, reducing the sides of the curve profiles proved effective. This optimization technique strikes a balance between visual quality and performance. Lower-poly baby hairs were created by generating 2D planes positioned above the head, which were then baked from the high-poly curves to transfer intricate details. For the parting sections, carefully position curves that originate from the scalp and extend towards each knot, adhering closely to the pattern provided by whatever reference material you're using. I've taken care to avoid creating lengthy lines on the parting sections. By implementing these techniques, the final hair asset is visually appealing while maintaining the desired level of fidelity.

Step-by-Step Guide

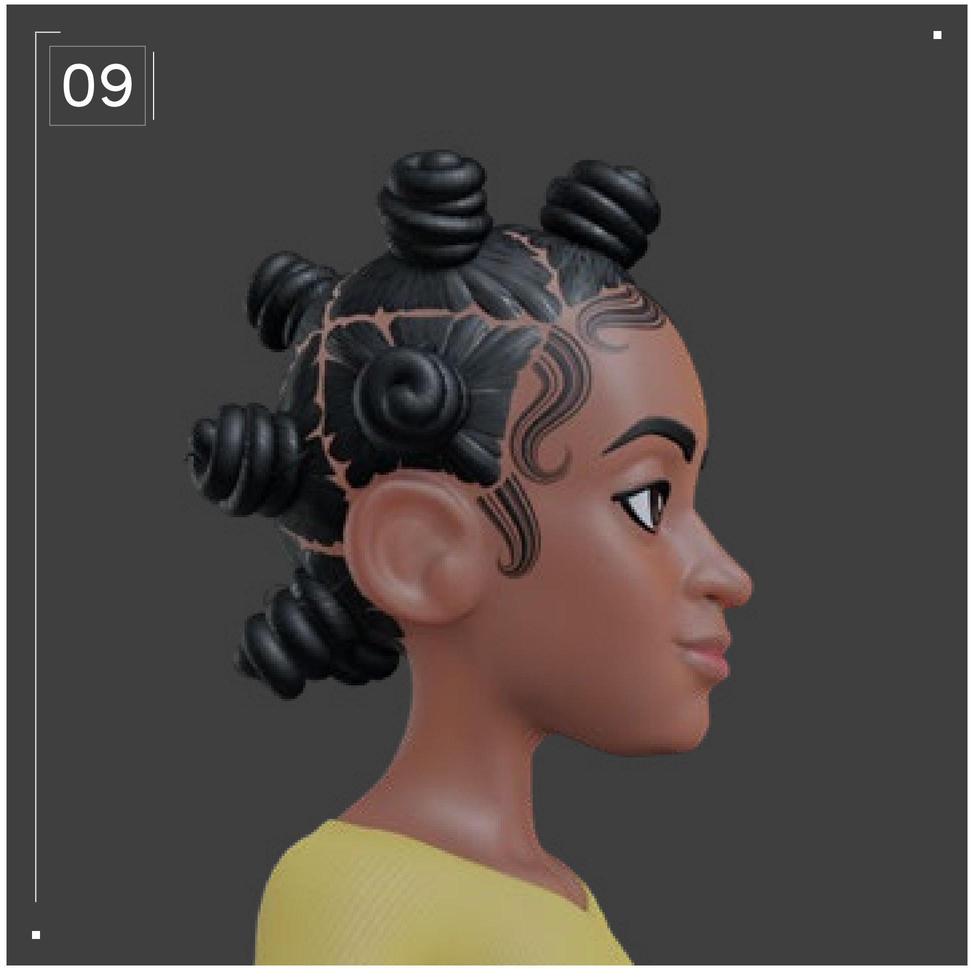


Next, join the high-poly mesh with the parting curves, merging them into a single high-poly model. This step ensures that the parting detail is incorporated seamlessly. Once completed, it is essential to retopologize the model, considering the inclusion of the parting detail in the process. This helps create a clean and optimized topology for the model while preserving the intricate parting design. While atypical at this point, Blender nodes were utilized, incorporating noise textures to achieve the desired striations and texture variation within the hair. The resulting hair texture was then baked from Blender, focusing specifically on the scalp detail to ensure accurate placement and integration. The next step involves exporting the mesh into Substance Painter.



Finally, the baked hair map can be imported into Substance Painter, where you can continue the texturing process.

In Substance Painter, pay attention to the texture of everything else besides the hair, including the skin, to achieve a cohesive appearance.

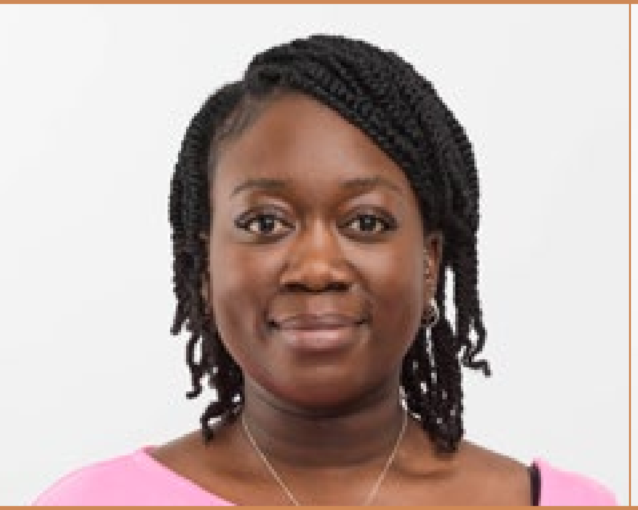


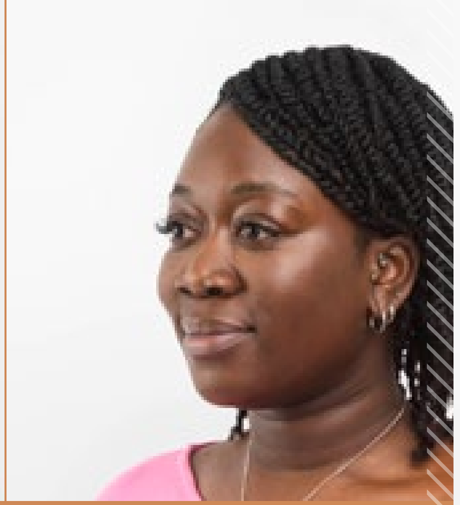
To enhance the realism further and reduce symmetry, I like to incorporate additional flyaways, strands of hair that stick out and/or up instead of lying flat against the head or within the desired style, introducing natural variations and imperfections to the hair. These flyaways contribute to a more authentic and lifelike representation of the hairstyle as seen in the reference images. Additional flyaways can also be used to convey a tighter curl pattern at the expense of a larger tri-count.

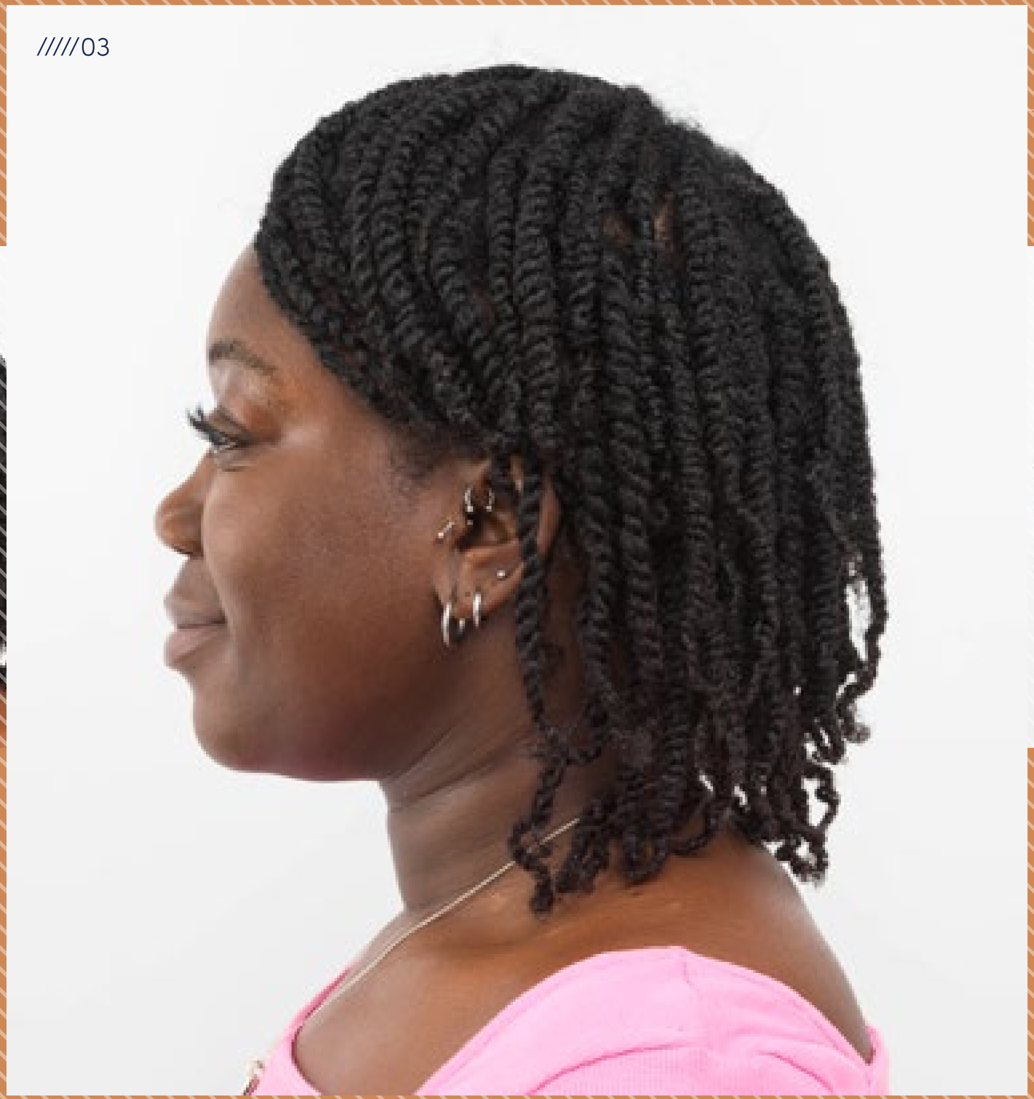
Twists
Twists

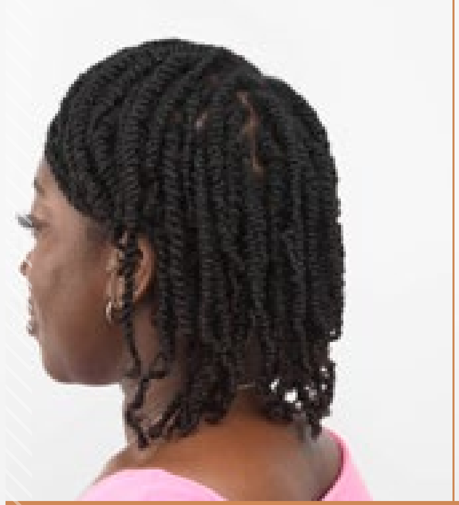
Twists

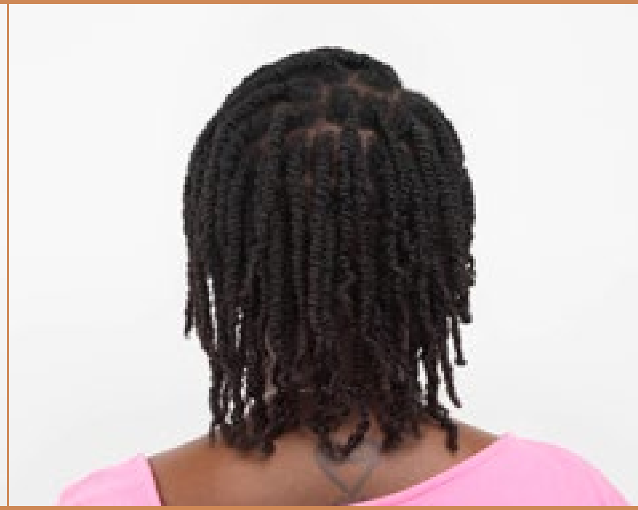
/////03







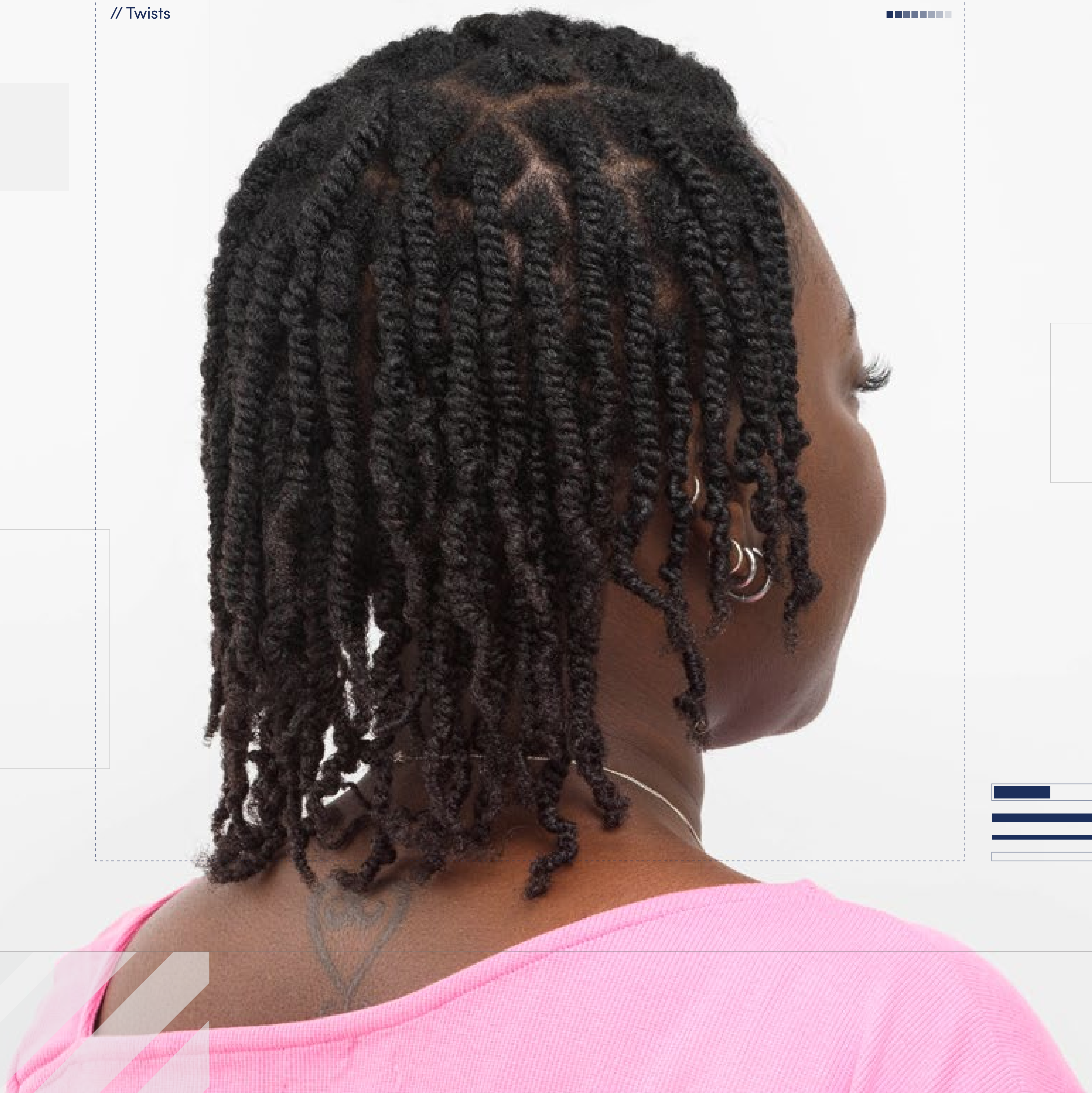




Twists

Download Images

// Twists

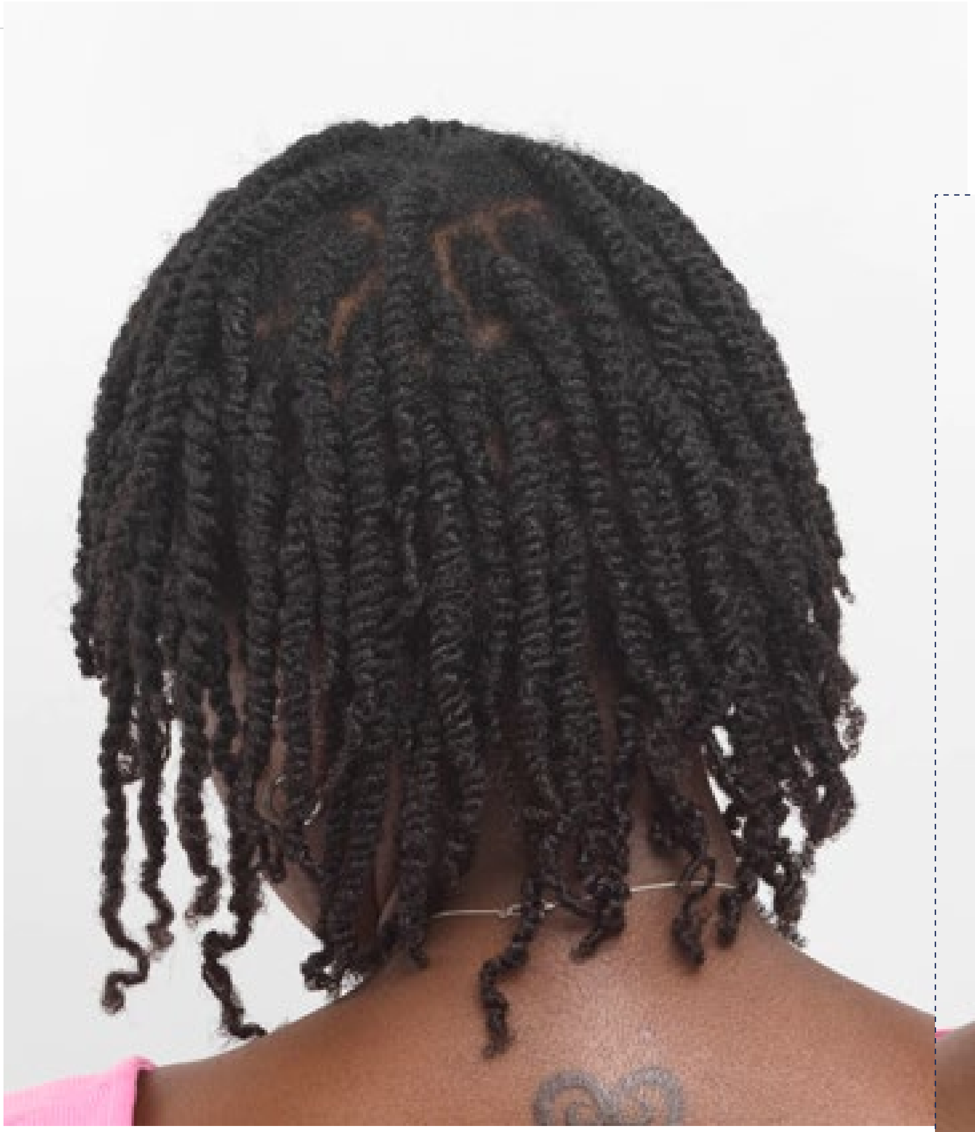


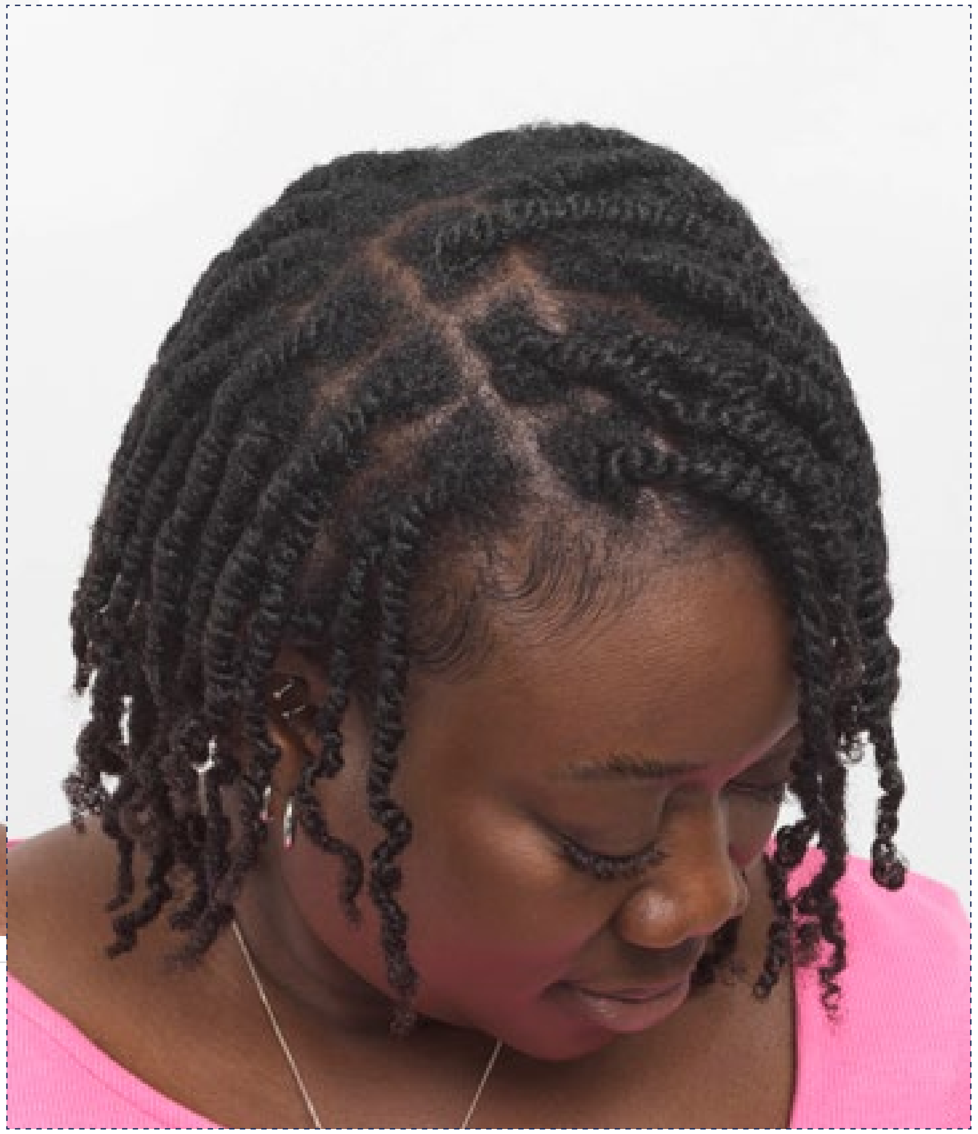
// Hairstyle

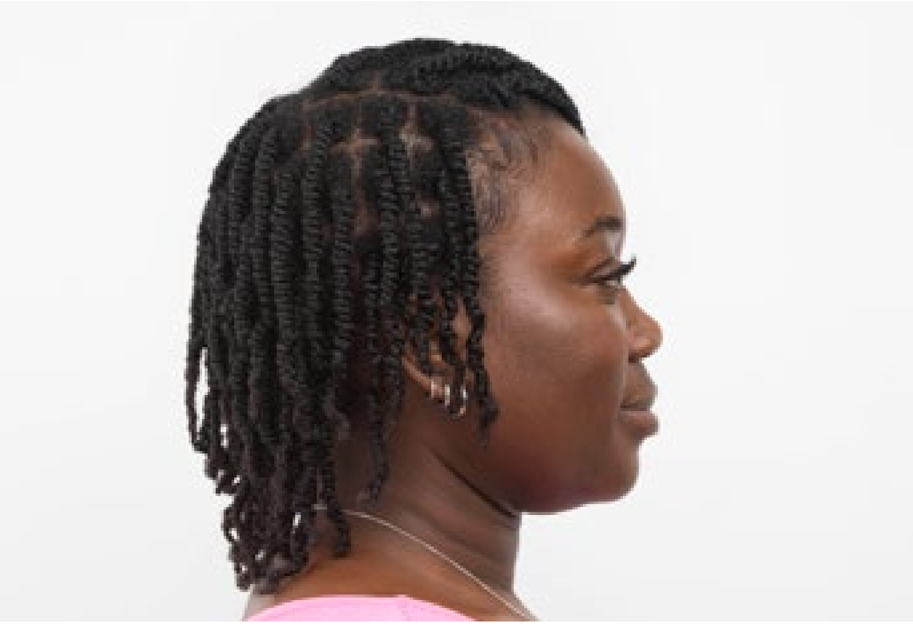
Twists

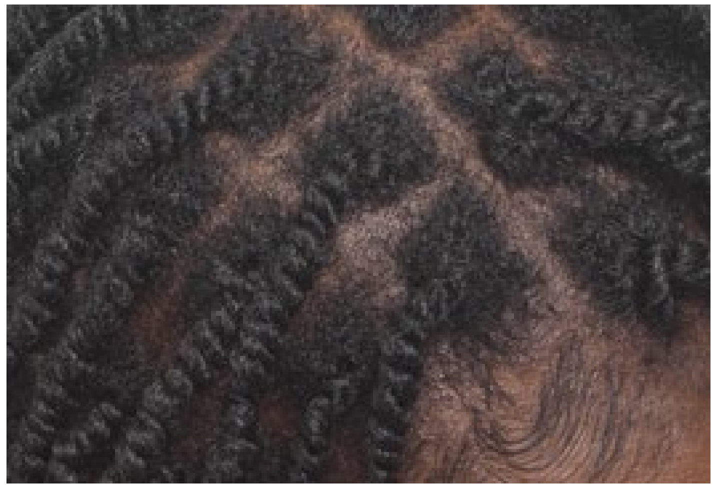
Style Background: Twists are an easy protective style that helps hair to hold on to moisture between wash days. They're created by dividing hair into sections, then splitting each section into two and twisting the strands together. Some people will wear fewer, thicker Twists while others may prefer to wear lots of thinner Twists; it's all down to personal preference. Like many protective styles, Twists can be done with natural hair alone or augmented with synthetic hair to achieve the desired length and volume.

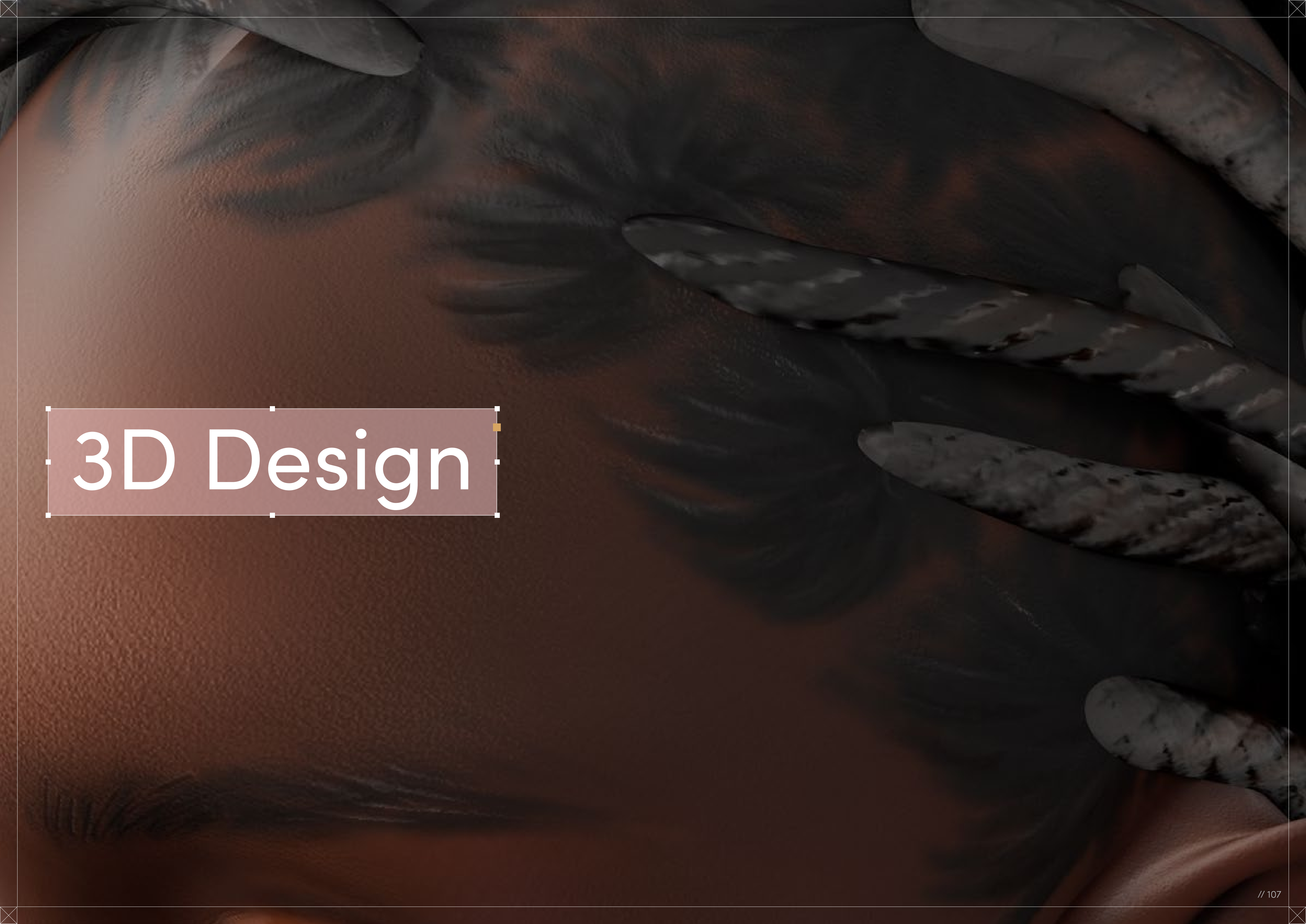
Download Images





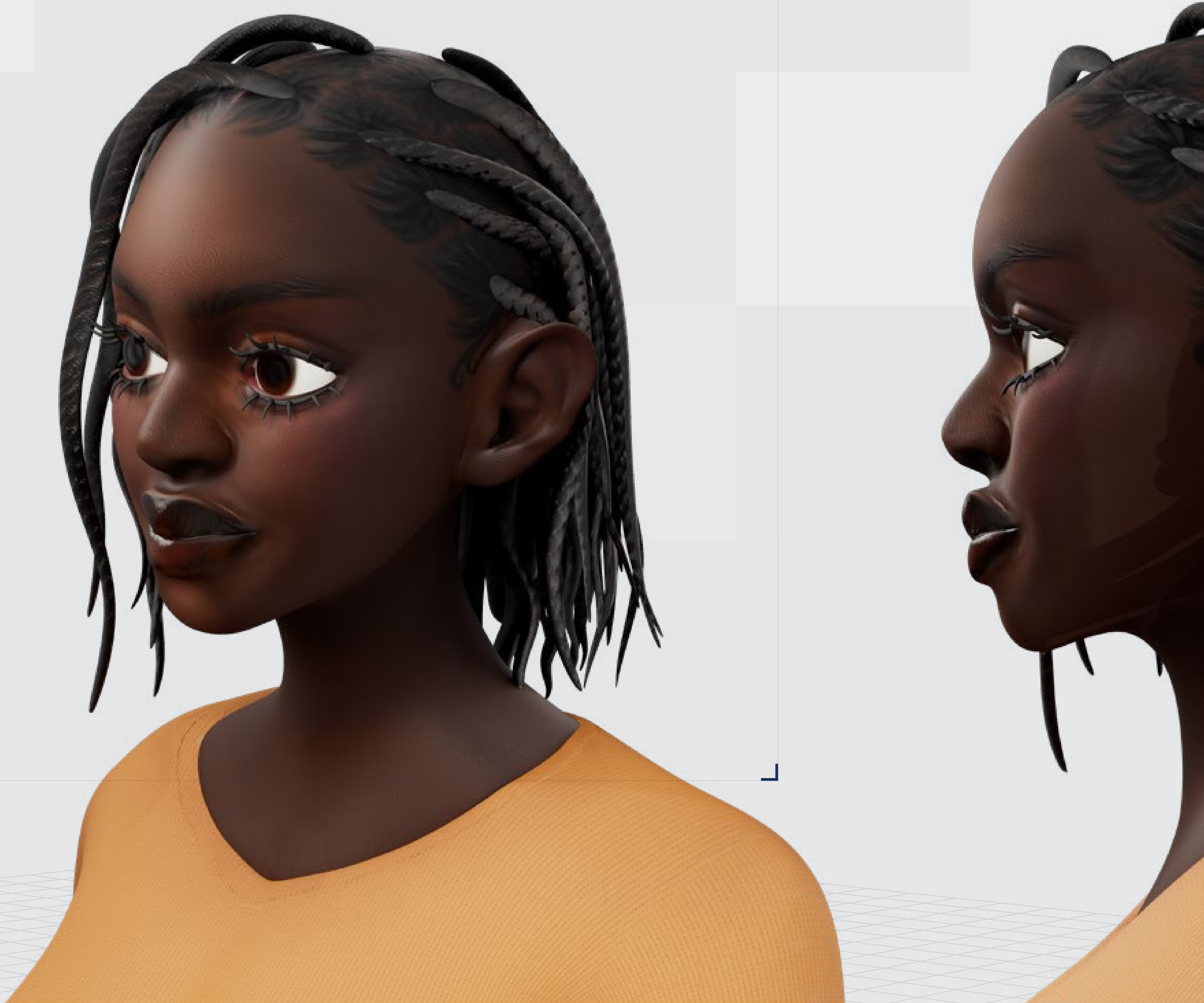






3D Design

Twists



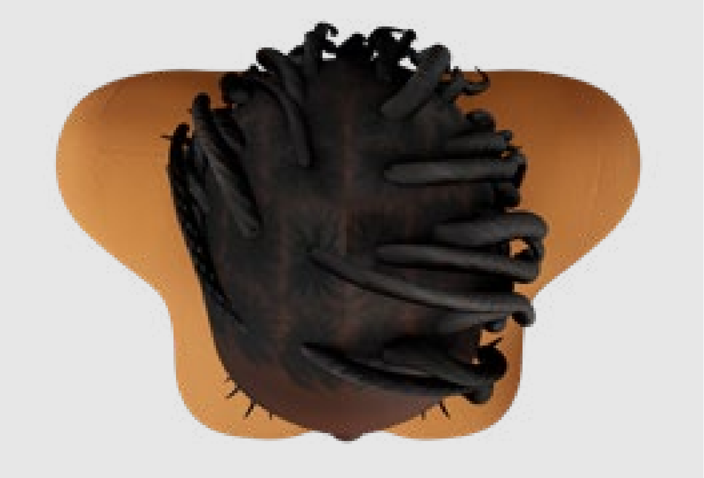
Digital Artist: **Malika**

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

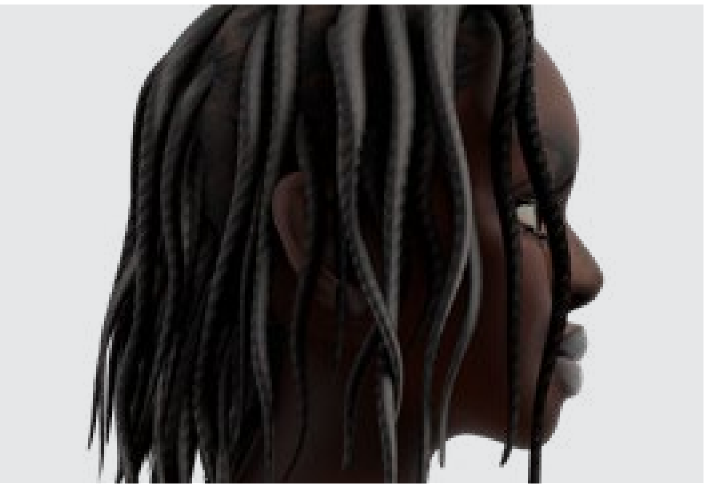
// Twists

3D Design

Curl Type Modeled: **4B**

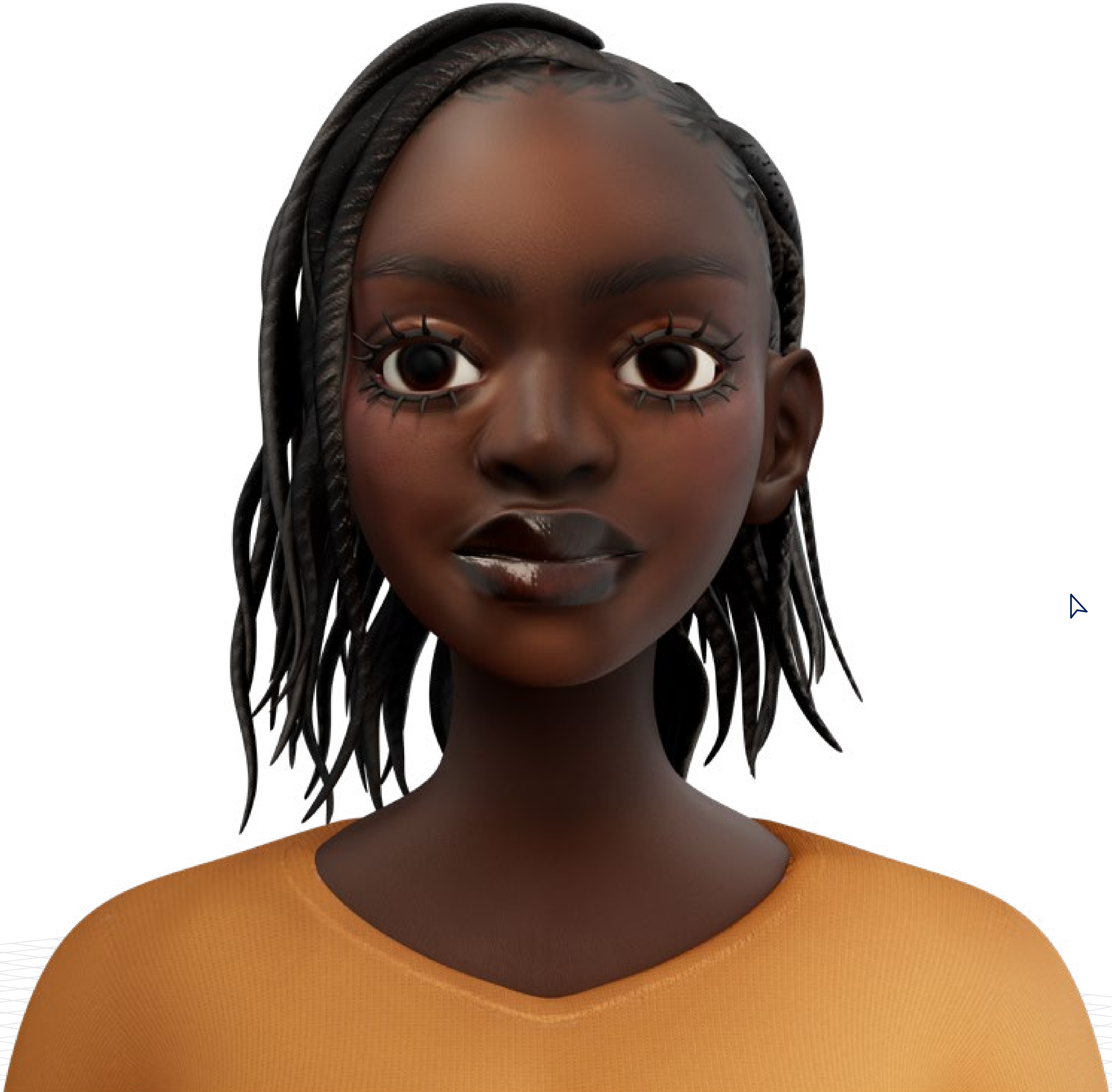


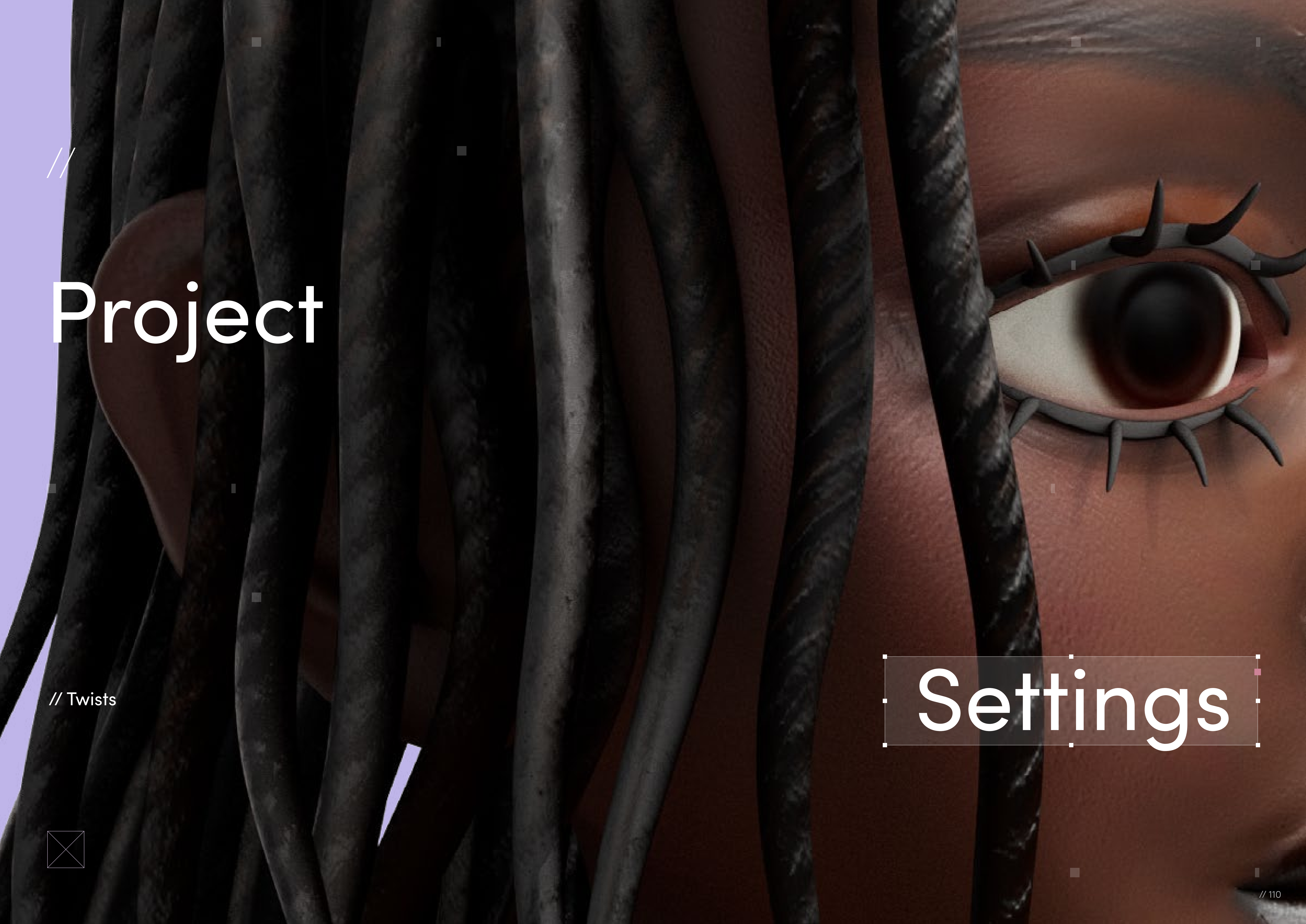
// 01



// 02

Download 3D Render





//

Project Settings

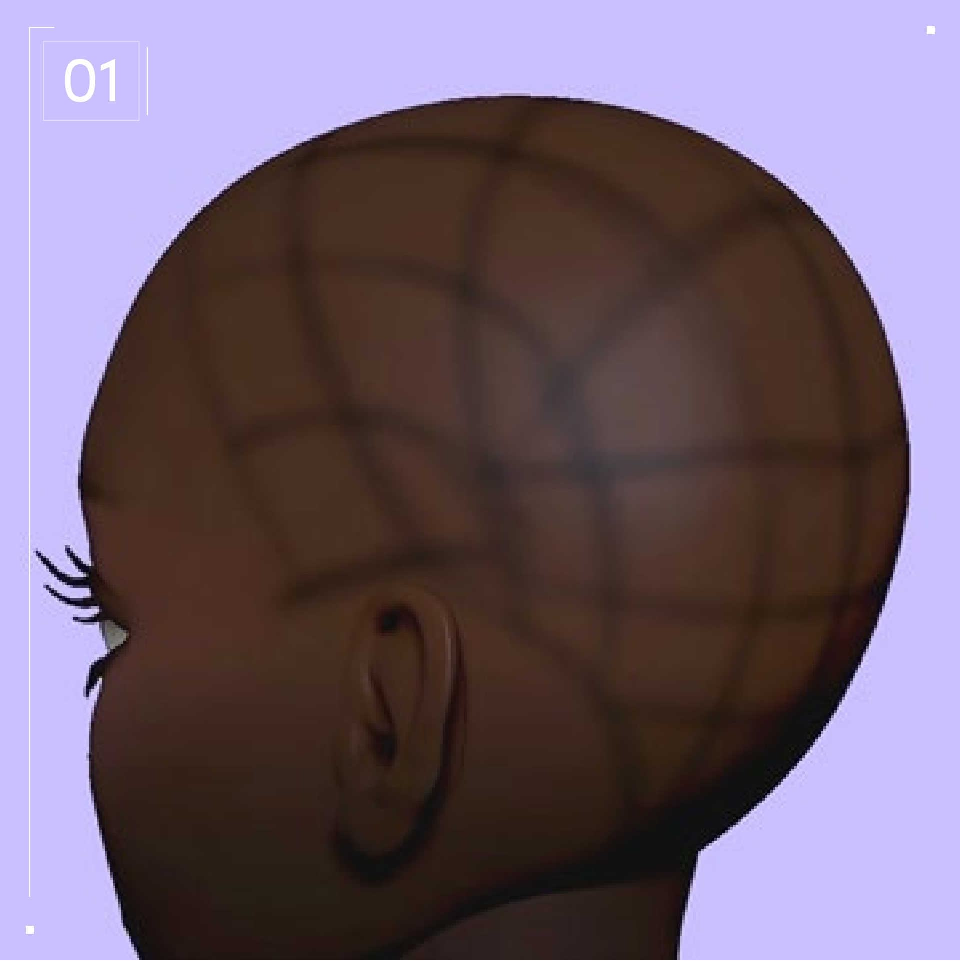
// Twists

// Project Settings

Program Guide

Sculpting and Modeling: Blender and ZBrush
Texturing: Blender and Substance 3D Painter

Step-by-Step Guide



// Blocking

Twists are a pretty classic hairstyle worn by many Black people who have mid-length to long hair and want to put their hair away until the next wash day, while keeping it looking stylish. Since I had already done Twists on myself in the past I already had an idea on how to go about it, but I still looked up some reference images as inspiration and to experiment with ways of styling them.

I believe that when it comes to natural hairstyles the best way to go about them is as if you were an actual hairstylist. I am not one, obviously, but I've seen aunts, cousins, sisters, and other girls do each other's hair my whole life. Actually, watching how the hairstyle is created in real life helped me more than just looking at pictures. That's why my first step is—just like any real hairstylist—parting the hair. To do this, I use polypainting in ZBrush (using the standard brush and only enabling the RGB button) and I part the hair by drawing on it.

People usually do Twists by themselves—unlike Box Braids, which often will be created with the help of another person. To save time, most people don't tend to make small, intricate partings like they would with Box Braids, so I make the partings as thick as possible. The best way to go about it is to split the head in four first, then create horizontal rows that will be split into vertical rows. Those last vertical rows may perfectly line up with each other or you might split the horizontal rows in some type of brick pattern instead; it's really up to you and the end look you're going for.

After that, you can either start going into detail and drawing the hair strokes or you can just leave them looking like squares for now. The hair strokes are important, but they can be done at any step. Parting is often the biggest mistake novice developers tend to make, so that's why I recommend doing the parting first and leaving the hair strokes until after. Plus, when you start actually placing the Twists you might realize some of the partings are wrong, so it could save you time leaving them as squares for the time being.

Then comes the most daunting part—placing the Twists, which is a very manual job. There are probably several ways of doing this, but personally I think this way is better for achieving something that looks more natural-looking and real. I simply create a Twist using the CurveTube brush in ZBrush. The Twists here are made of real hair so their width cannot be completely uniform, which is why using the CurveTube I go into "Stroke," then "Curve modifier," and then "Size" and tweak the graph so that my tube will be big at the base and smaller at the end. I also make sure to give a little wavy shape towards the ends, which is something that happens when braiding or twisting real curly hair.

After that I duplicate the Twists using the gizmo and holding CTRL while moving it, placing it using the parting I created previously. I still need to tweak each of them a bit using the Move Topology brush to break the uniformity, but I can do that later. Just like a real hairstylist would, it's easier to start in the back and move your way up and around when placing the hair.

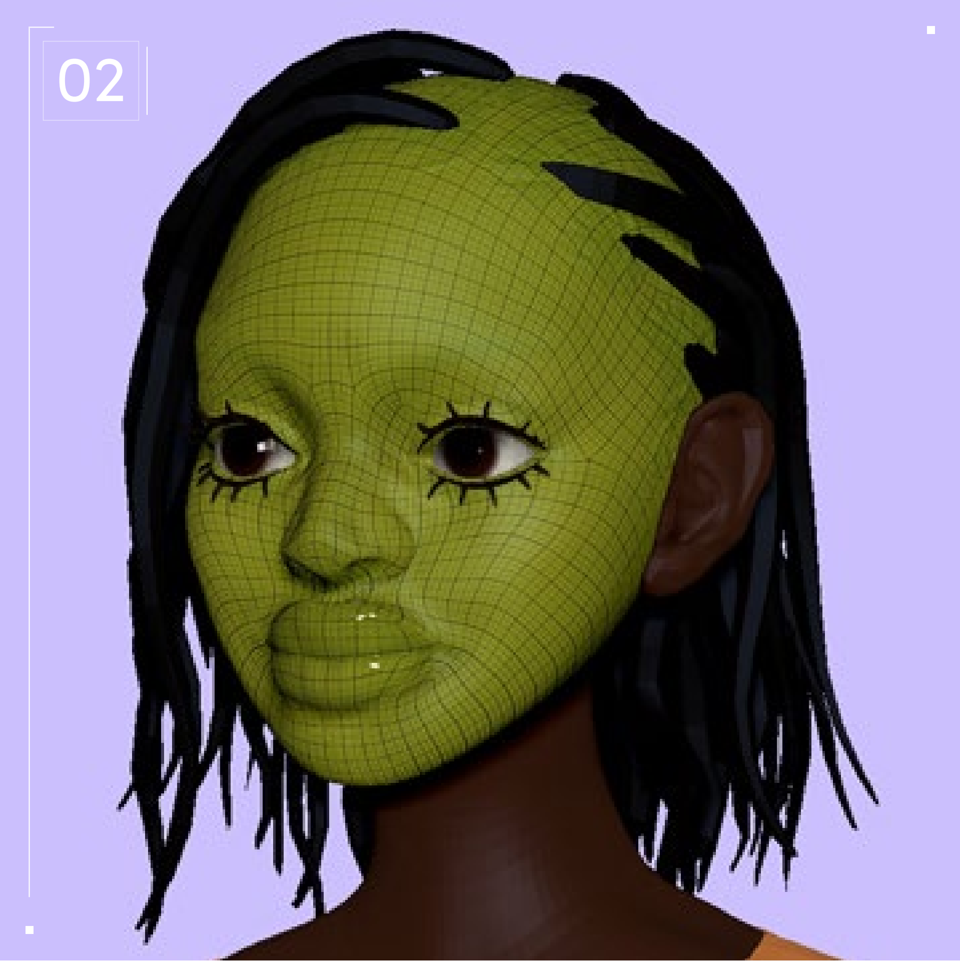
If your partings are squares that line up perfectly, once you've done a single row, you can usually duplicate a whole row instead of one Twist at a time, but in this specific example I could only do that for one or two rows, since my character has a side part with one side tucked behind her ear.

Once the back is done it's a matter of breaking symmetry and a lot of repetitive work on placing Twists in a way that fits your planned hairstyle but also follows the parting that you've already done. Sometimes some partings don't really work, so it's perfectly fine to combine some into one or split some if they're too big.

When everything is done, you could start doing the hair strokes. I personally decided to show dimension with the parting through the head's normal map instead of making them their own geometry, since this could very quickly make the whole bust way too polygon dense. It's also easier to have control over the hair flow this way, because the partings don't just go in any direction; the hair strands reunite at the center where the Twist starts, a bit like a ponytail would, but on a smaller scale. Then, I use the ClayBuildup brush to add dimension to each part.

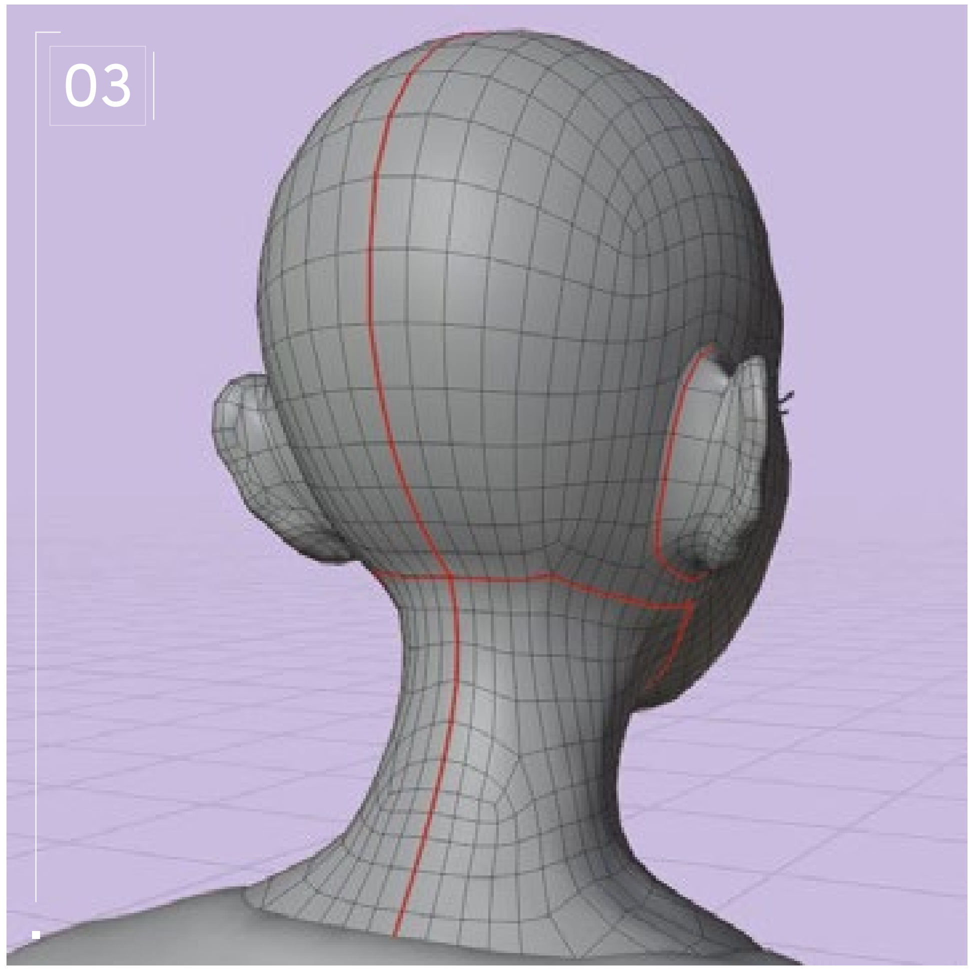
// Project Settings

Step-by-Step Guide



// Modeling

To save time, I used a base model that had already been retopologized, but I don't really like using a still face and moving it around. So, I took the time to model a face from scratch while keeping it to the same size as the base model, then I moved it around with the Move Topology brush and used the Project geometry tool to make it match my original sculpt as much as possible. Since the Twists are just simple cylinders, I remeshed them as low as possible.

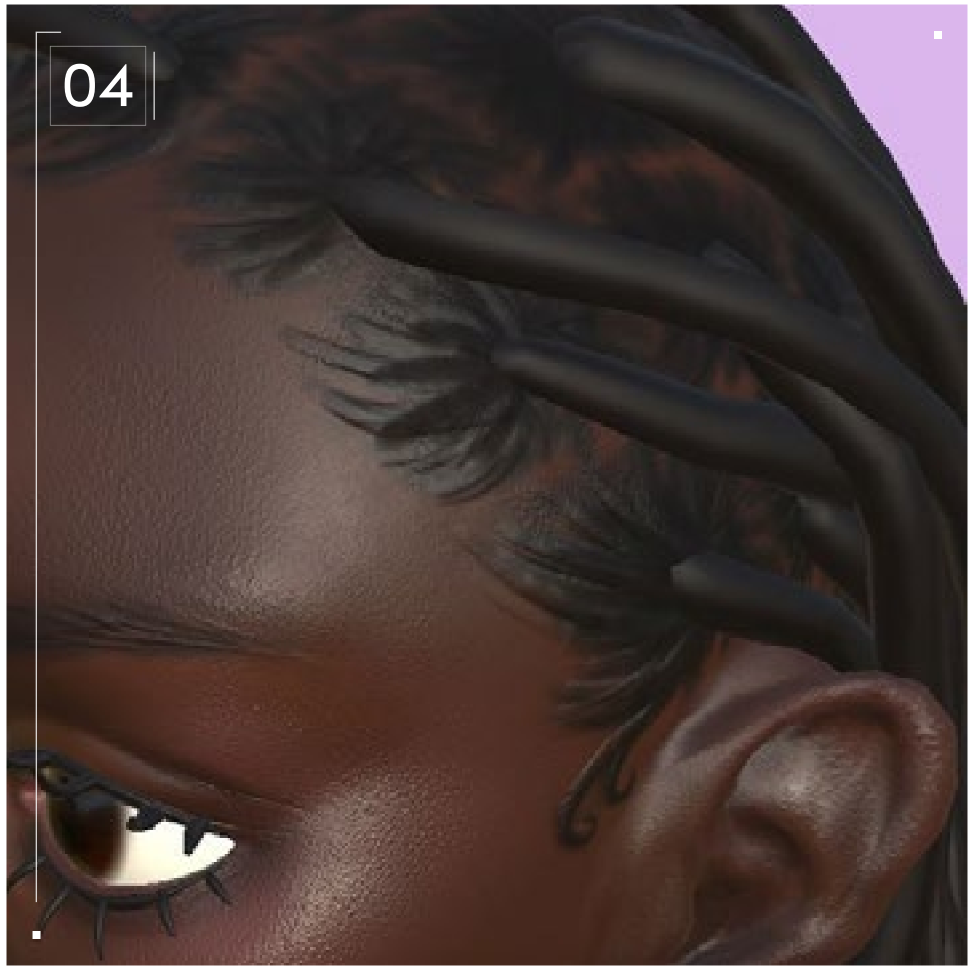


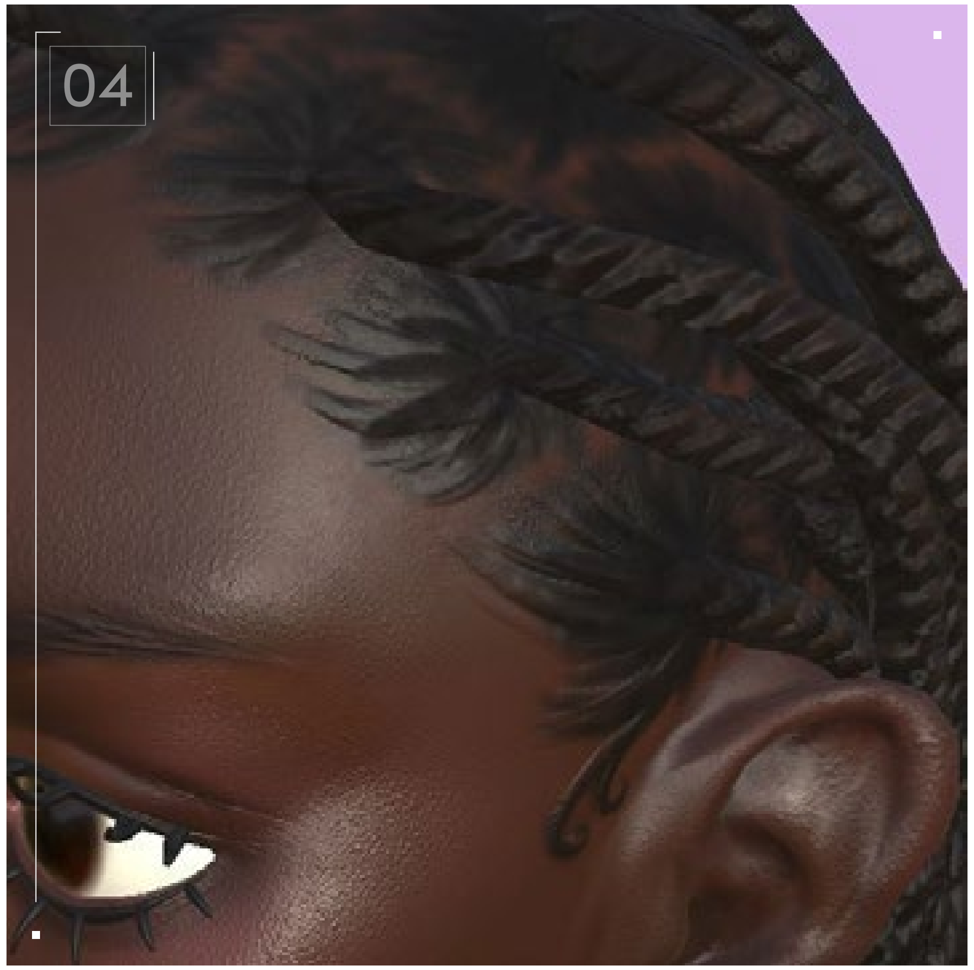
// UVing

In this instance, I chose a base model that was already UVed. But generally speaking, UVing is pretty similar for all faces, so usually I would just cut a seam at the base and back of the neck, around the ears, and across the cranium, stopping a bit before the hairline.

As for the Twists, since I was going to either draw the Twist pattern myself or use a texture, I figured it would be easier to work with if the cylinders are as straight as possible. To do that, in Blender, I select a single rectangle, click "U" to unwrap it, and "Move" the vert around to have a rectangle as straight as possible. Then I click "CTRL" and "L" to have the whole Twist selected, then click "U" and pick "Follow Active Quads" and click "OK." However, this does scale up the Twists so it's necessary to scale them all back down. This is a pretty repetitive process because this has to be done for every single Twist and they all have to be scaled to a size that's as uniform as possible.

Step-by-Step Guide





// Texturing

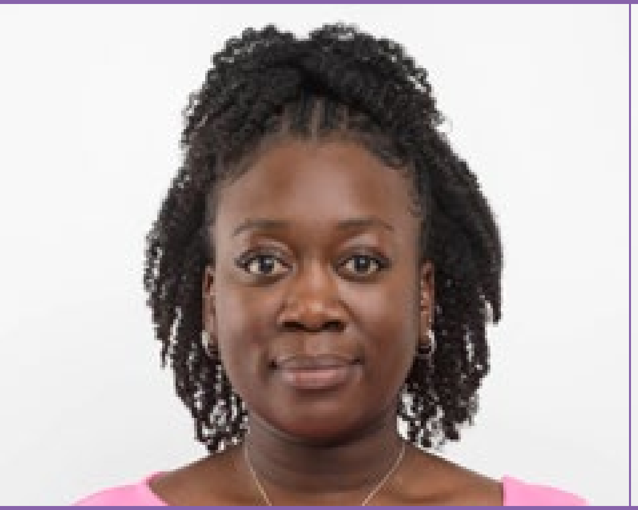
I do most of the texturing work using polypainting as I believe it helps me figure out the face and hair better, so I simply export a high-poly FBX of my sculpt from ZBrush. It's important to rename the objects that will be baked to match each other on the low and high poly. Specify them with "_low/_high" (in all lowercase) at the end of the name to make sure you don't confuse the software. To bake both the polypaint and the Normal (mostly for the head to get the parting details), I go into the Texture Setting tab and click on "Bake Maps." I personally like to set the size to 4,000 to get as much detail as possible when I size down. I then pick my high-poly, precise that the mapping needs to be by name. Also it's important to be precise for the ID that the source color is the vertex color or your ID will just simply be grey.

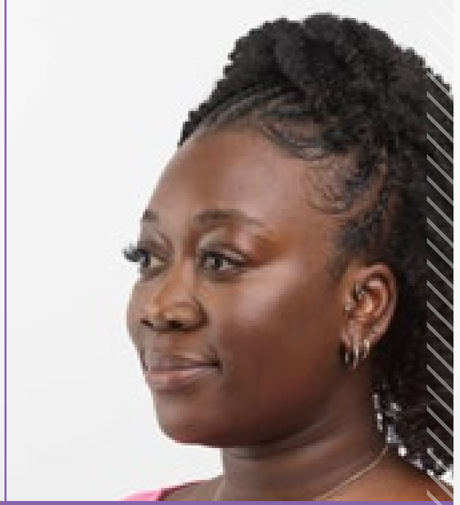
When the baking is done, simply create a fill layer and drag your baked ID and Normal to their respective channels. I then use additional fill layers – or just a paint layer – to fix anything that is off on the ID. In this case, the ears and neck were not connected to the high-poly model so the Normal was badly baked there. But that was fine since these were details that I didn't really need. For the more important parts I simply used a white mask on the fill layer that has the baked Normal and I erased the parts I didn't need.

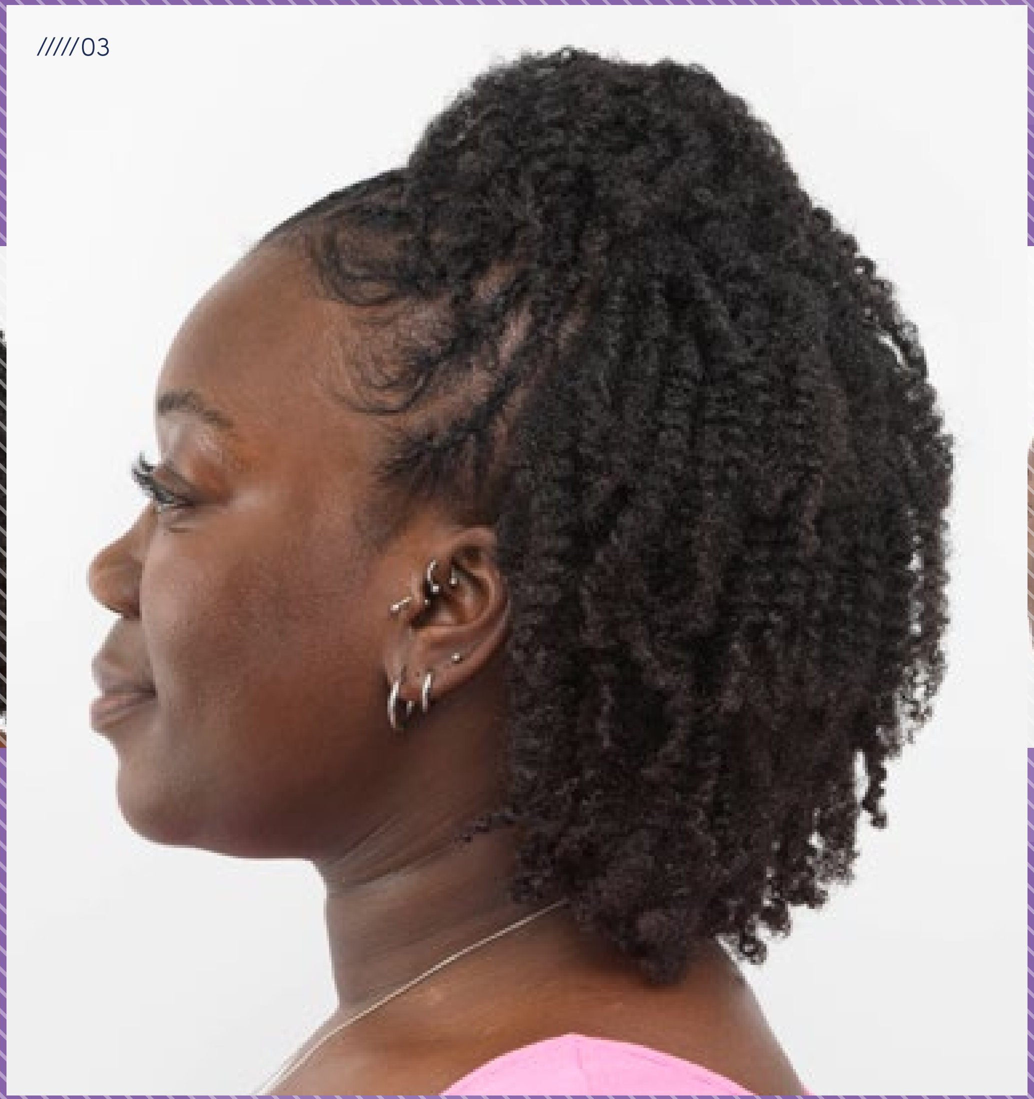
I usually go for a very stylized look and I like my designs to look a bit toy-like, so usually I would stop here. But this time I decided it would be nice to have more roughness variation as well as skin texture, so I added that using the skin materials already available in the Substance library, varying them for each part of the face – for example, so that the skin on the nose doesn't look the same as the skin on the cheek. To make the hair partings as visible as possible, I made the back of the head more matte than the overall head and I created an even more matte layer to add some very dark hair strokes.

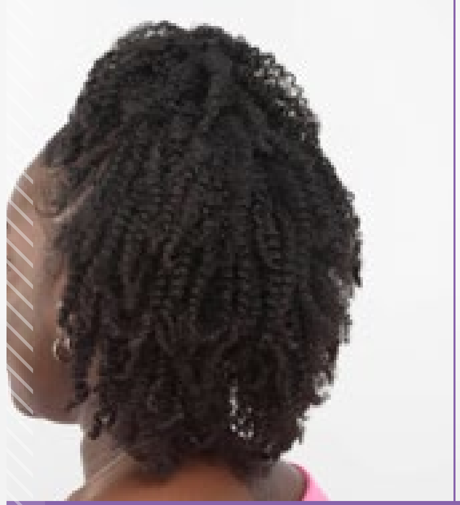
For the Twists, I used a gaussian spots texture to give the illusion of tiny curls within the Twists. For the Twists texture itself I used a fiber texture, although it ended up a bit off at certain parts because some of the Twists were not UVed fully. I manually added some hair stroke details and made some of the Twist indents a bit deeper with a height fill layer at certain parts, as well as some brown hair strokes to give some color variation.

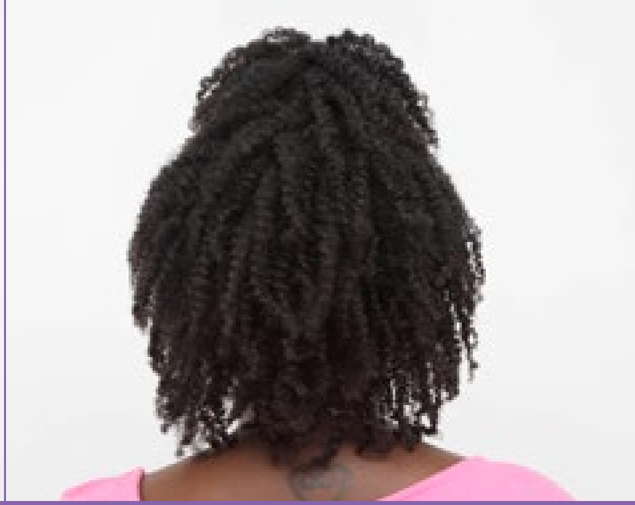
Half Up







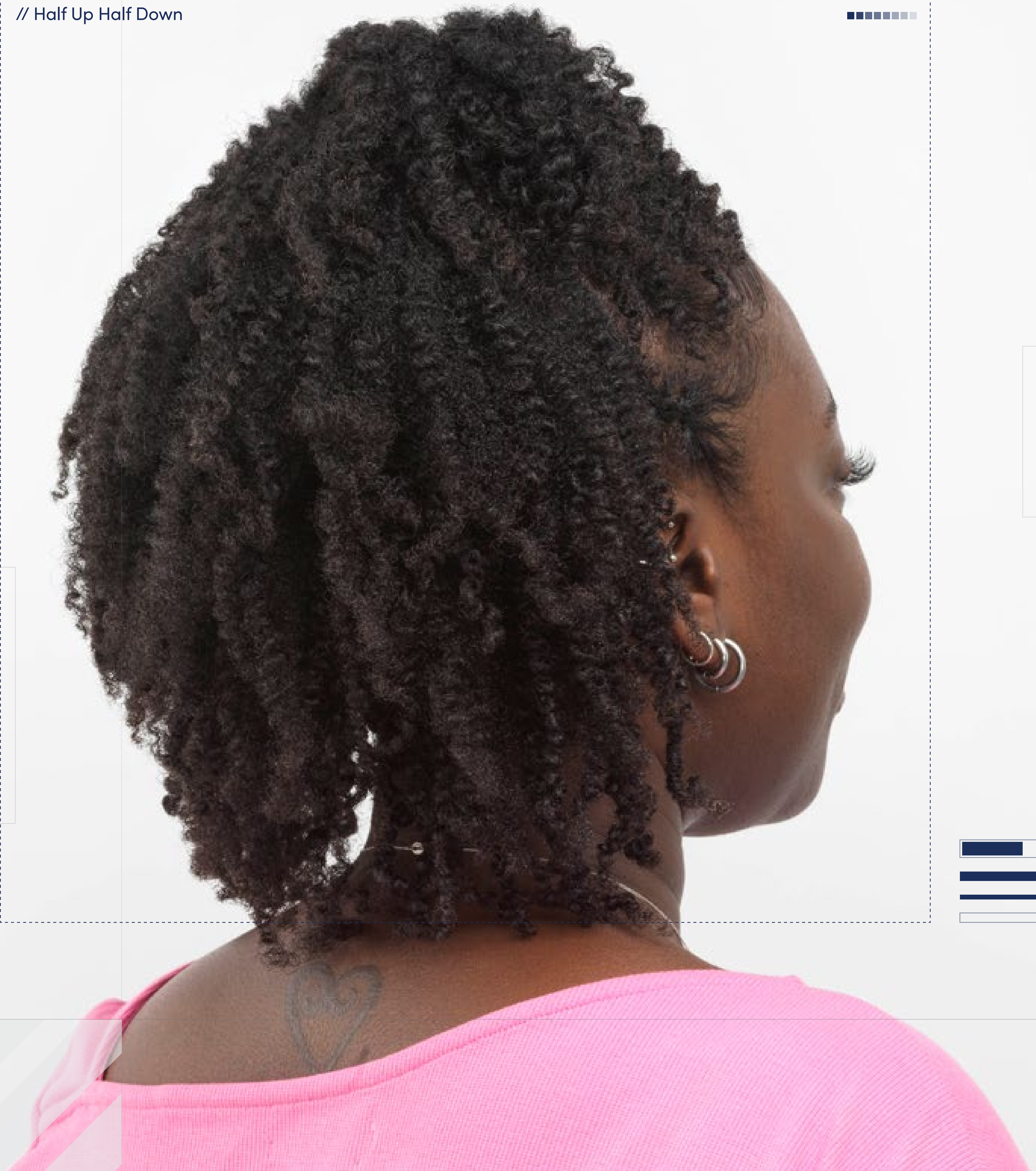




Half Down

Download Images

// Half Up Half Down

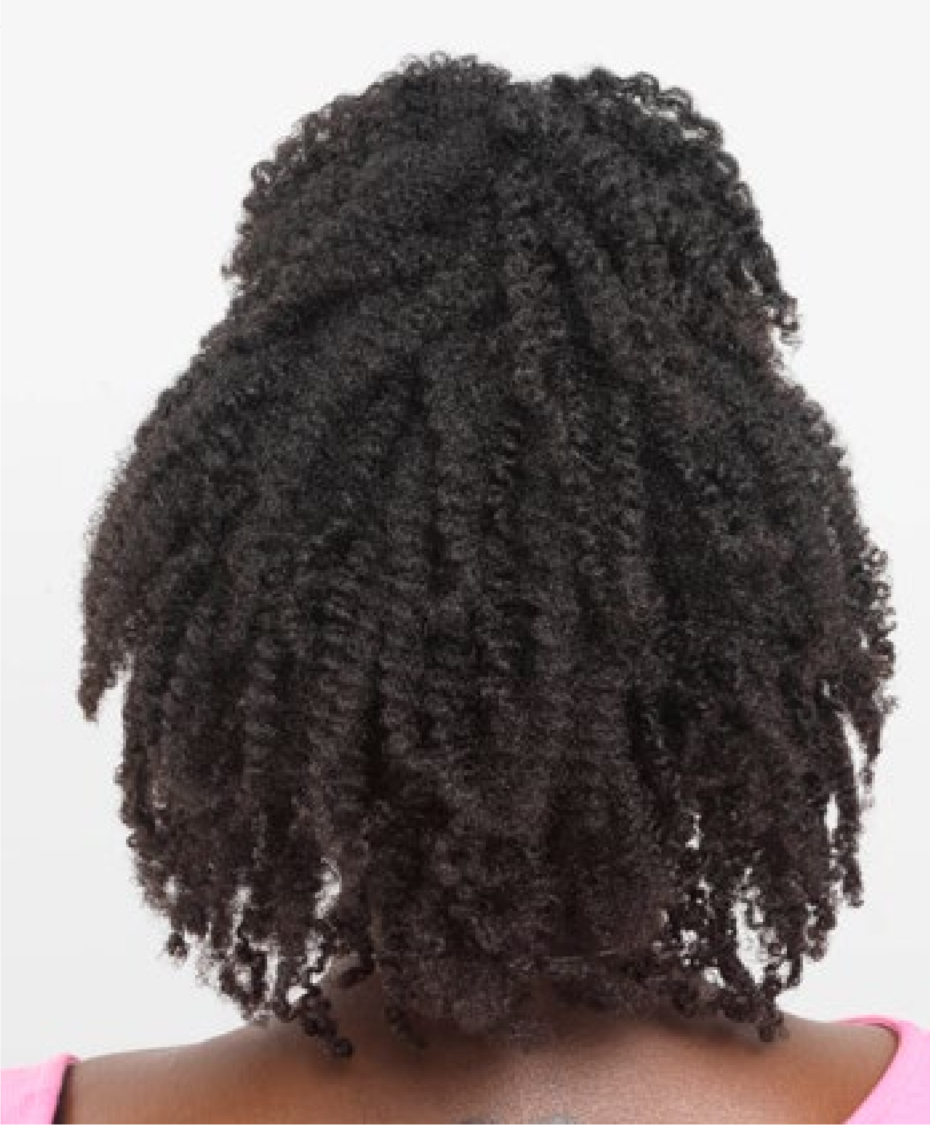


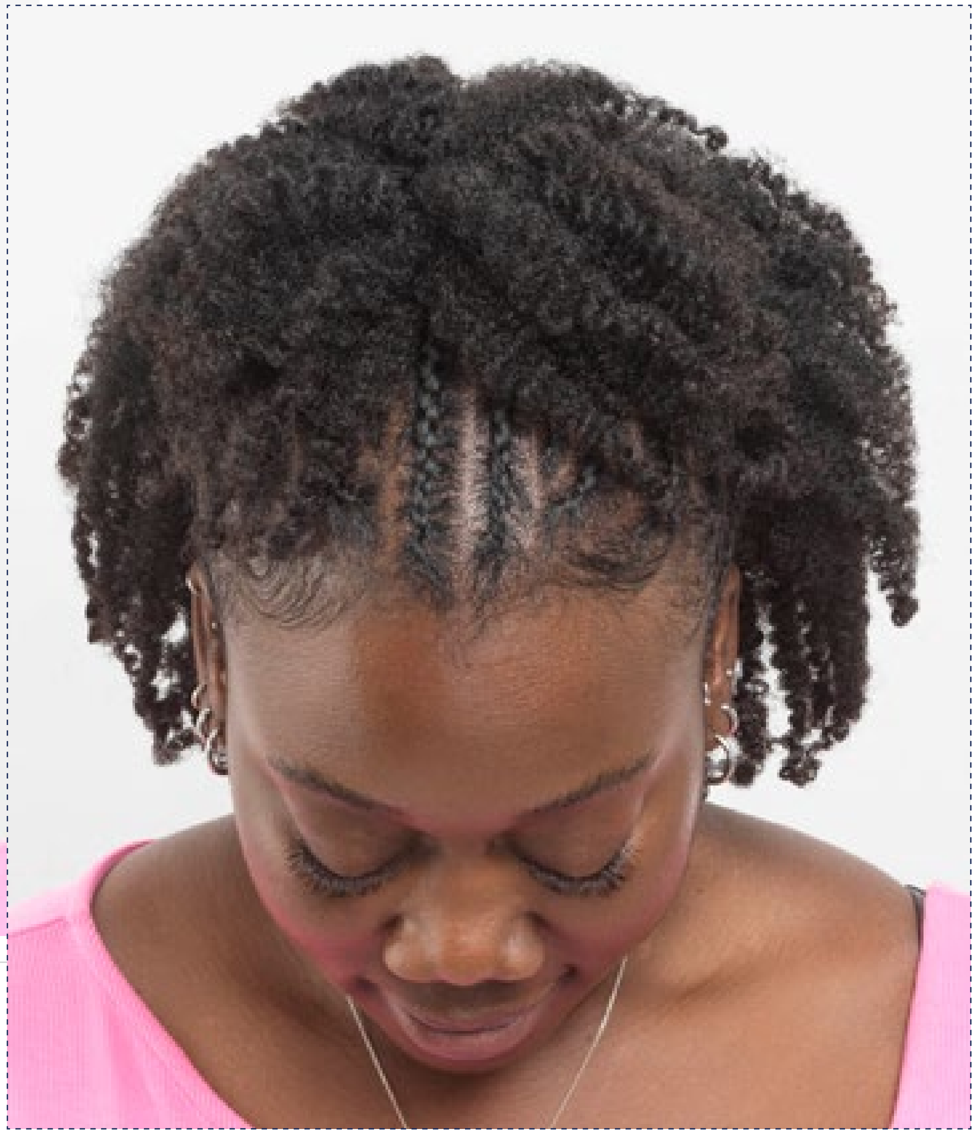
// Hairstyle

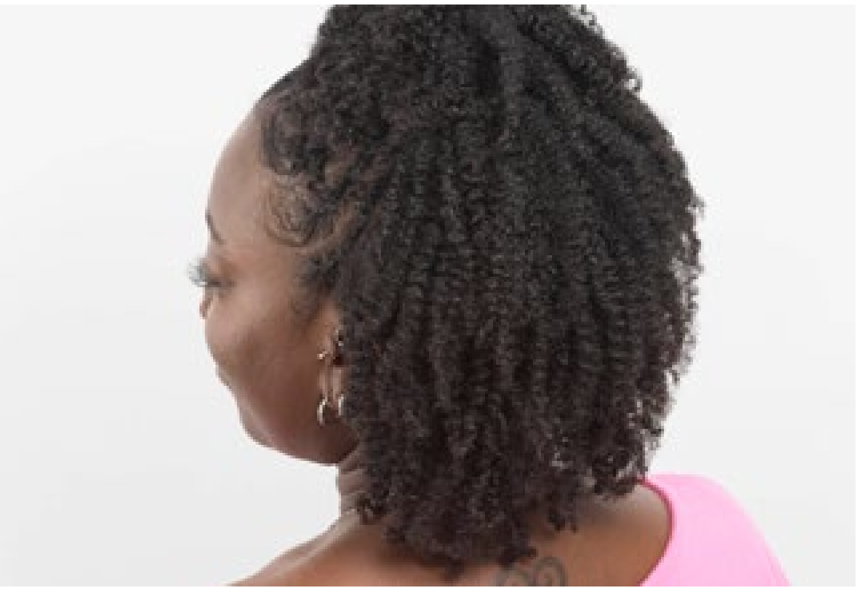
Half Up Half Down

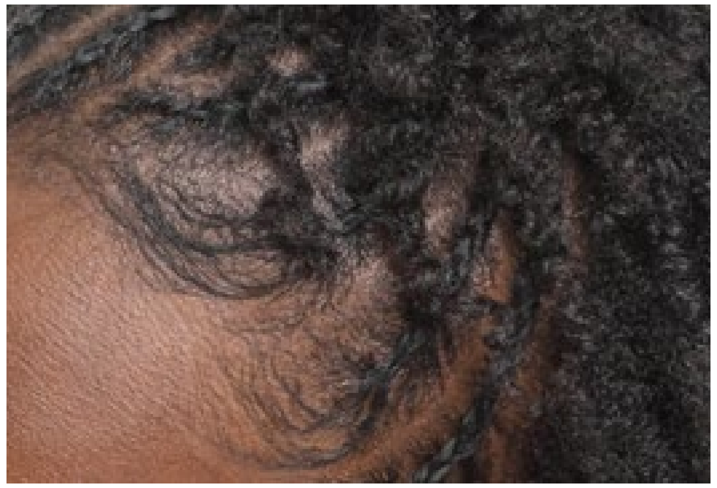
Style Background: A half up, half down hairstyle, is when a section of hair is pulled away from the face and tied up on top of the head (half up), while the rest of the hair hangs freely in the back (half down). This style has the front section of the hair in Cornrow Braids (hair braided flat to the scalp) and gathered at the crown of the head to form a ponytail. Braids are a versatile protective style that can be worn in many different ways. Strands can be loose and natural or in individual braids to achieve a half up half down look. The style is practical and suitable for any occasion.

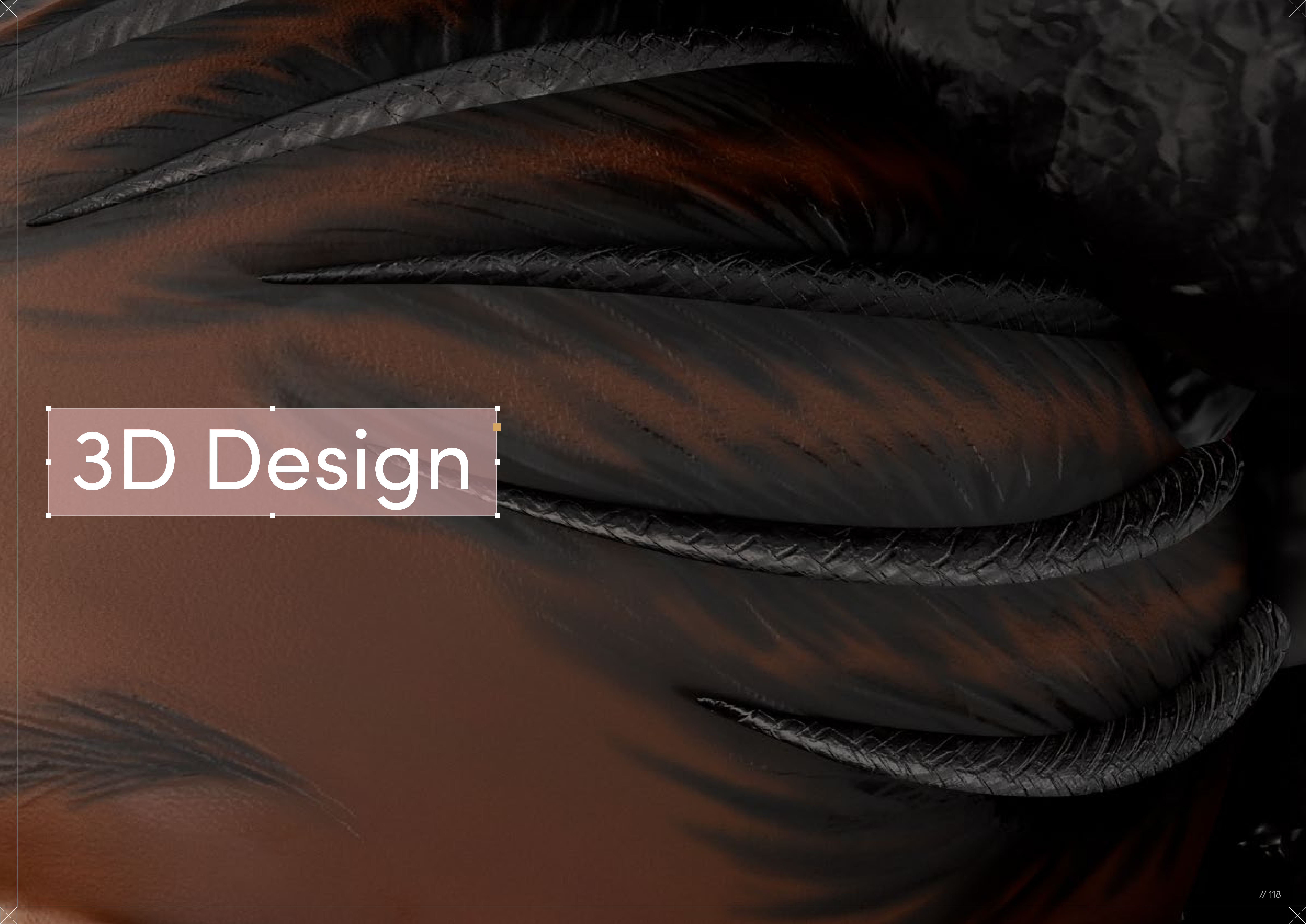
Download Images





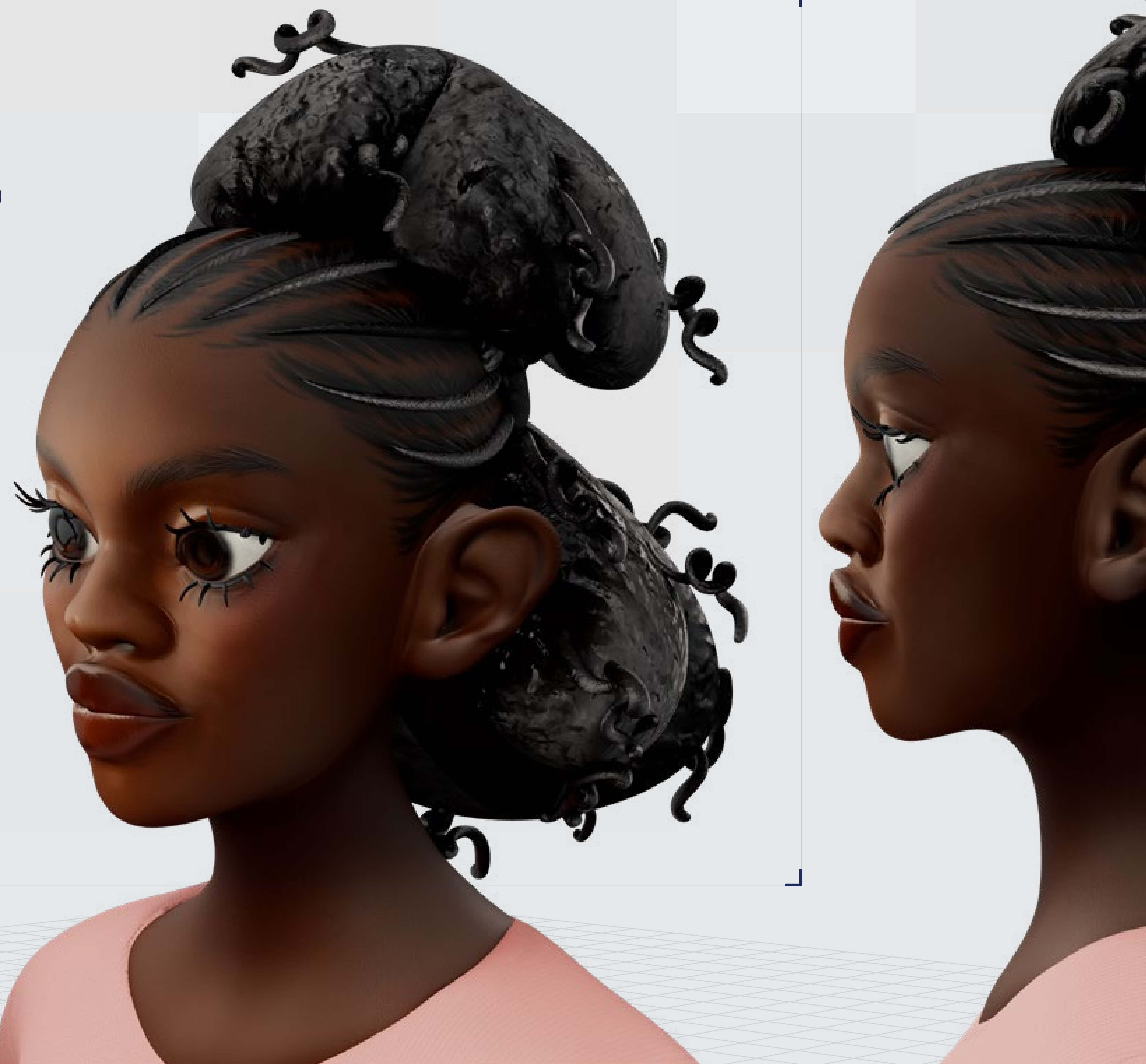






3D Design

Half Up Half Down



Digital Artist: **Malika**

// Half Up Half Down

3D Design

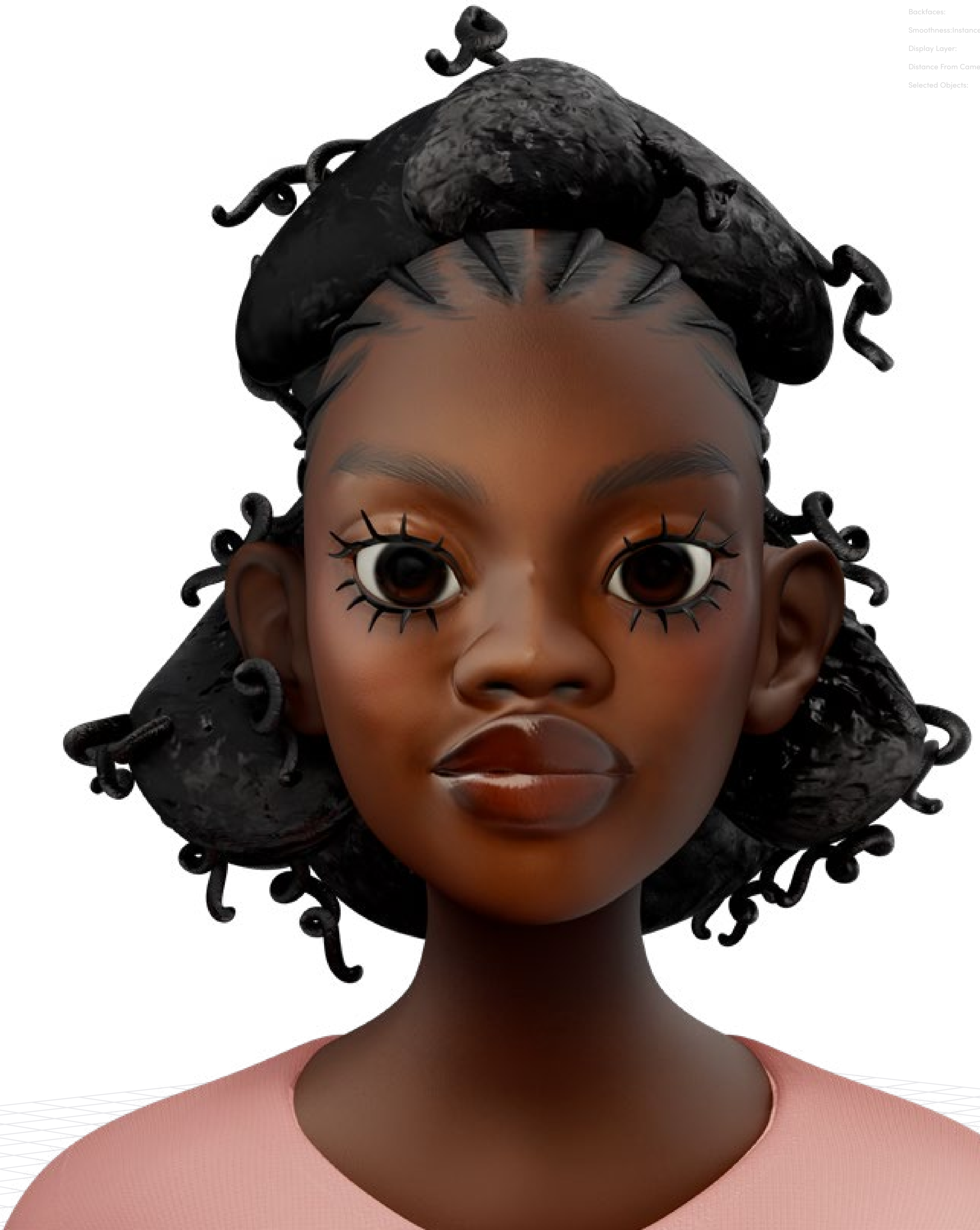
Curl Type Modeled: **4B**

// 01

// 02

Download 3D Render

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

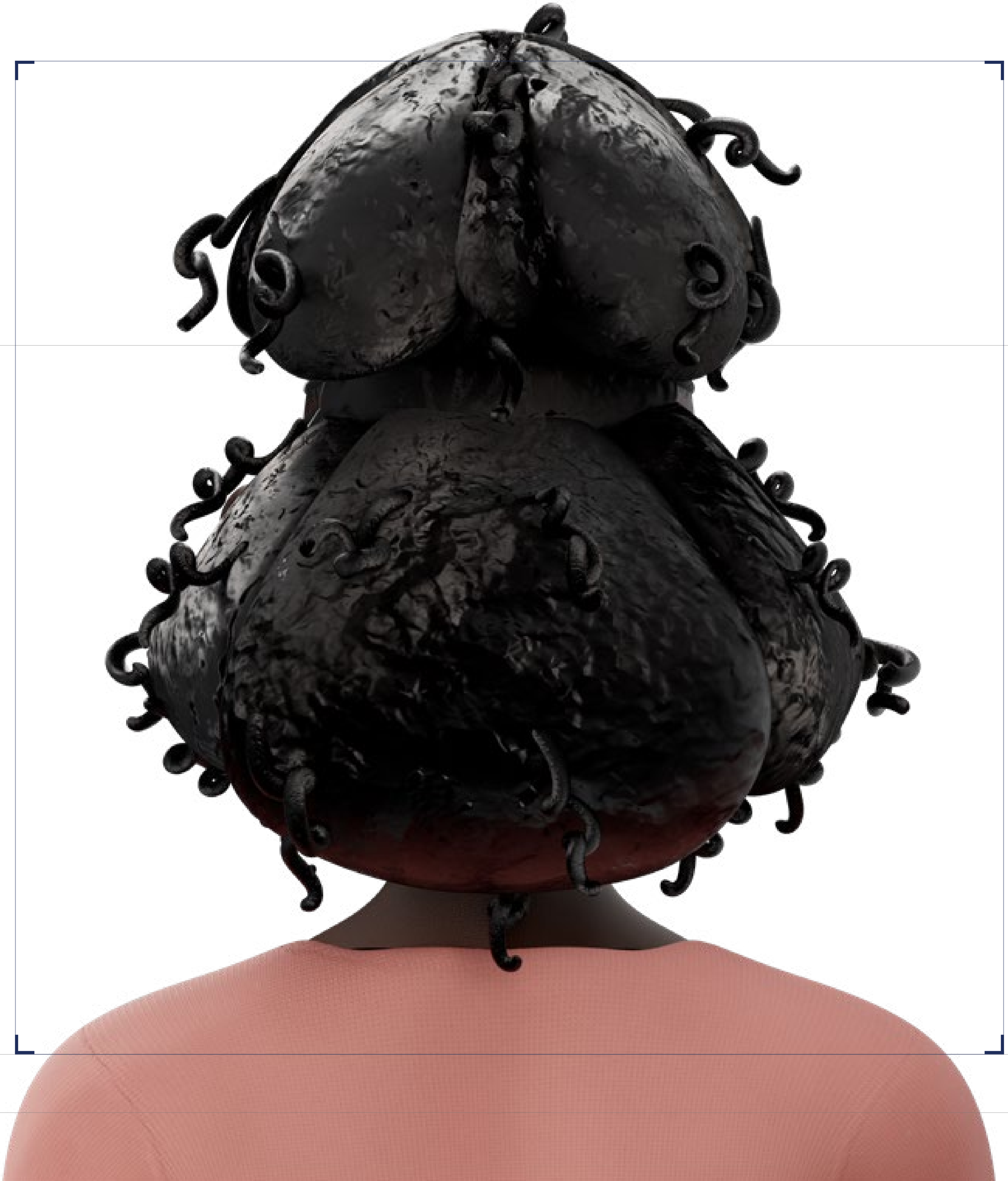
Project Settings

// Half Up Half Down

// Project Settings

Program Guide

Sculpting and Modeling: Blender and ZBrush
Texturing: Blender and Substance 3D Painter



Step-by-Step Guide

01

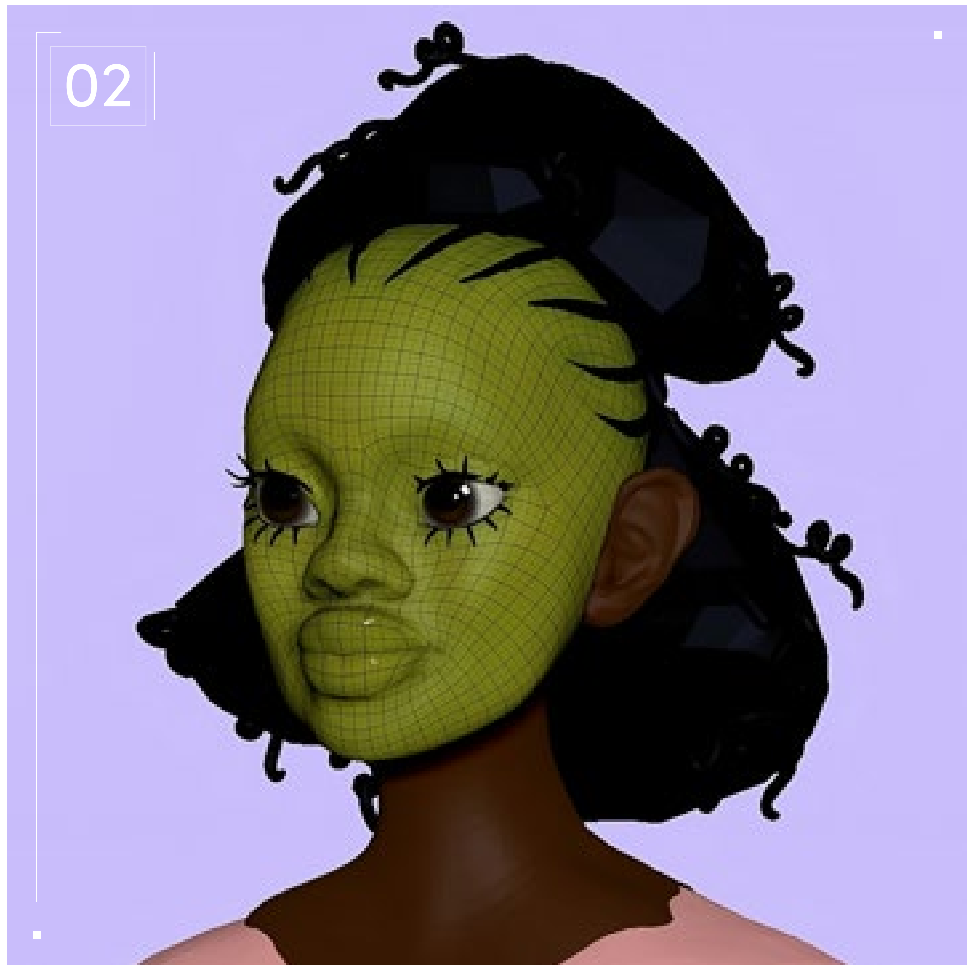
I also vaguely planned out the placement by polypainting how many cornrows would feel right on the head. I then painted the hair strokes and added dimension to them using the ClayBuildup brush.

// Blocking

The half up, half down hairstyle is pretty straightforward. I simply placed spheres using the IMM Primitives brush and the Move Topology brush to have an idea of what I wanted the general silhouette of the hair to look like.

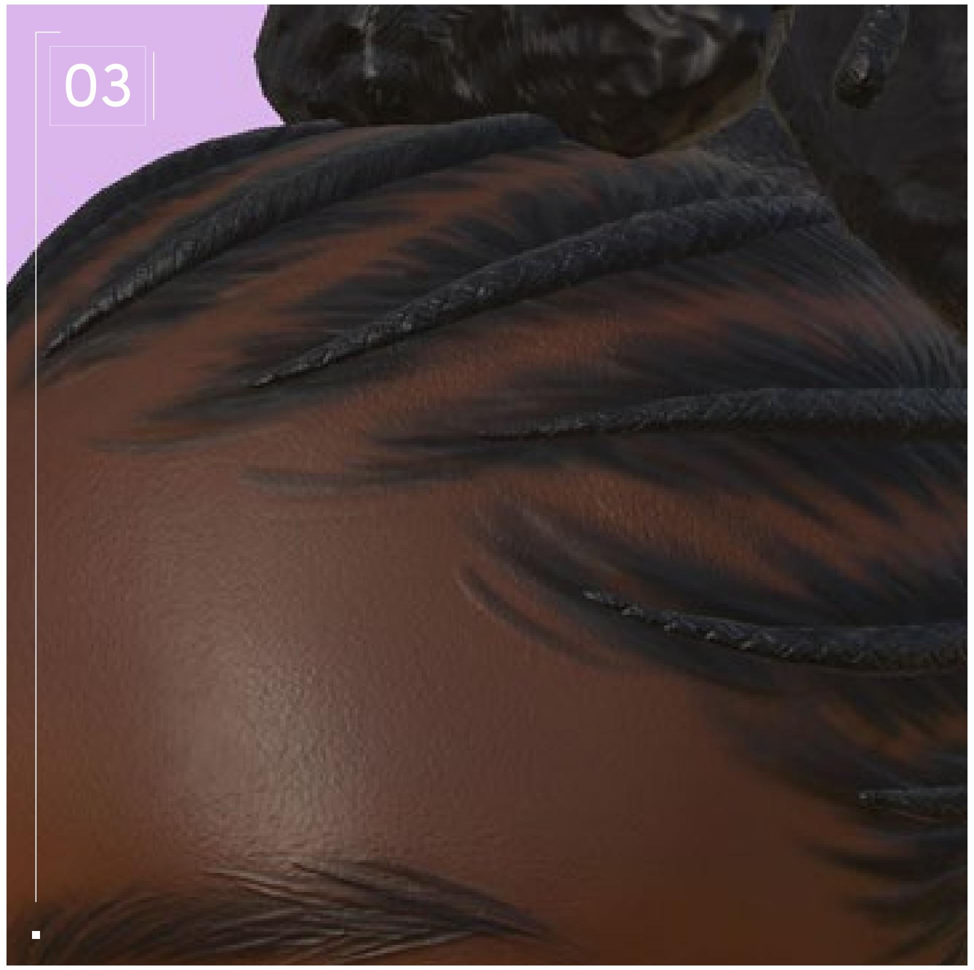
For the cornrowed part, I used curve tubes with an inverted graph so that it starts small and ends big. I also changed the Picker depth to Cont Z so that when I drag my brush to make the cornrows it will follow the shape of the head without needing too much tweaking.

Step-by-Step Guide



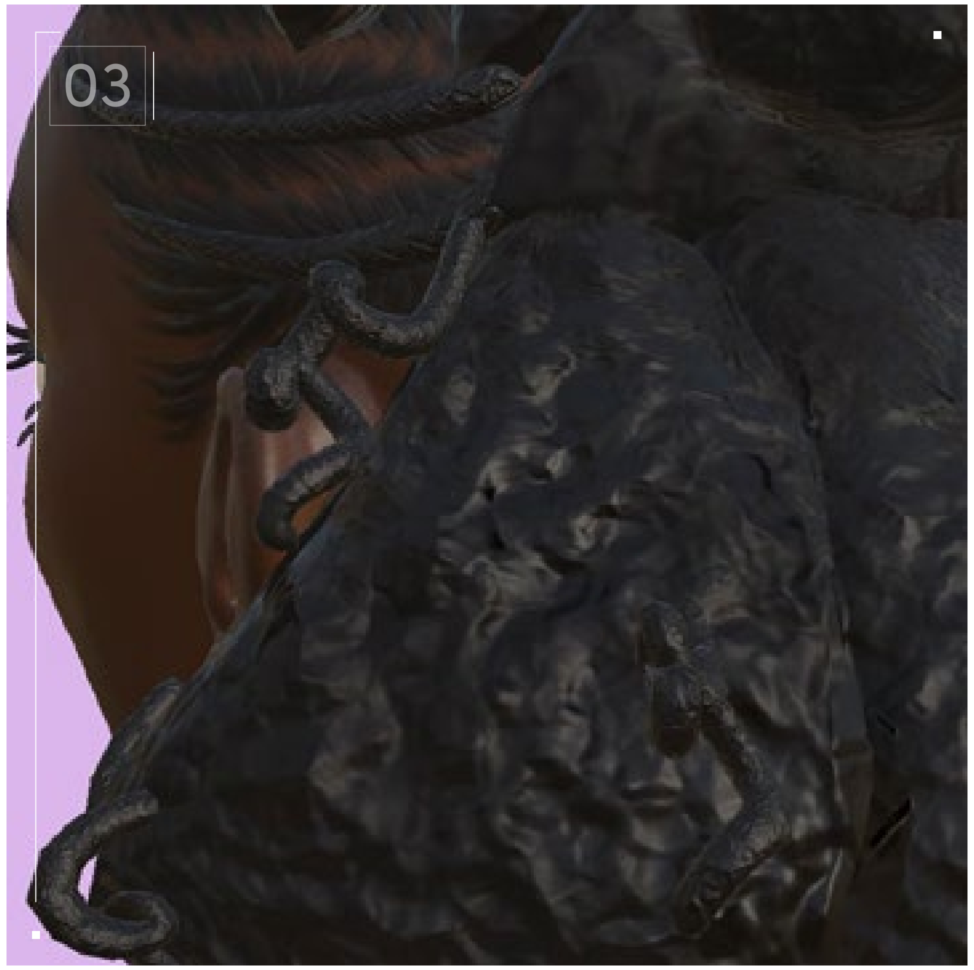
// Modeling

To save time, I used a base model that had already been retopologized, but I don't really like using a still face and moving it around. So, I modeled the face from scratch while keeping it to the same size as the base model, then I moved it around with the Move Topology brush and used the Project geometry tool to make it match my original sculpt as much as possible. To give the illusion of curls I did some random squiggles on the main hair using the DamStandard brush. I then broke the silhouette with some random curls using the CurveTube.



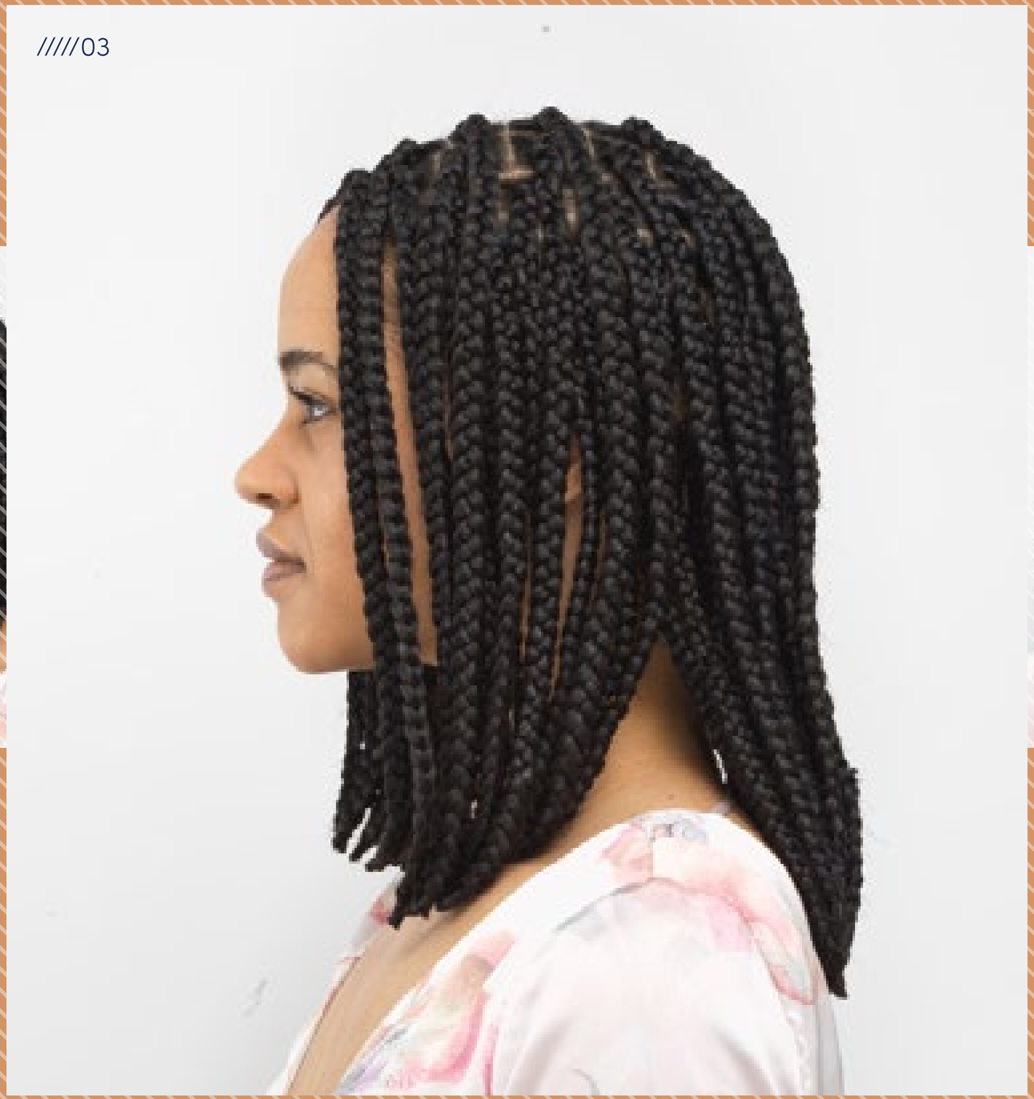
// Texturing

I do most of the texturing work using polypainting as I believe it helps me figure out the face and hair better, so I simply export a high-poly FBX off my ZBrush. It's important to rename the objects that will be baked to match each other on the low and high poly. Specify them with "_low/_high" (in all lowercase) at the end of the name to make sure you don't confuse the software.



When the baking is done simply create a fill layer and drag your baked ID and Normal to their respective channels. I then use additional fill layers — or just a paint layer — to fix anything that is off on the ID. I like to go for a very stylized look, with designs that look a bit toy-like, so usually I would stop here. But this time, I decided to add a noise texture to create a bit more dimension on top of the already existing one I had made in ZBrush. I also added some variation in roughness throughout the hair to give the impression of products being on the hair. I also added a braid texture to avoid having to draw the braid by end, but I still added some dents with a height map to add some randomness to the cornrows.

Braided

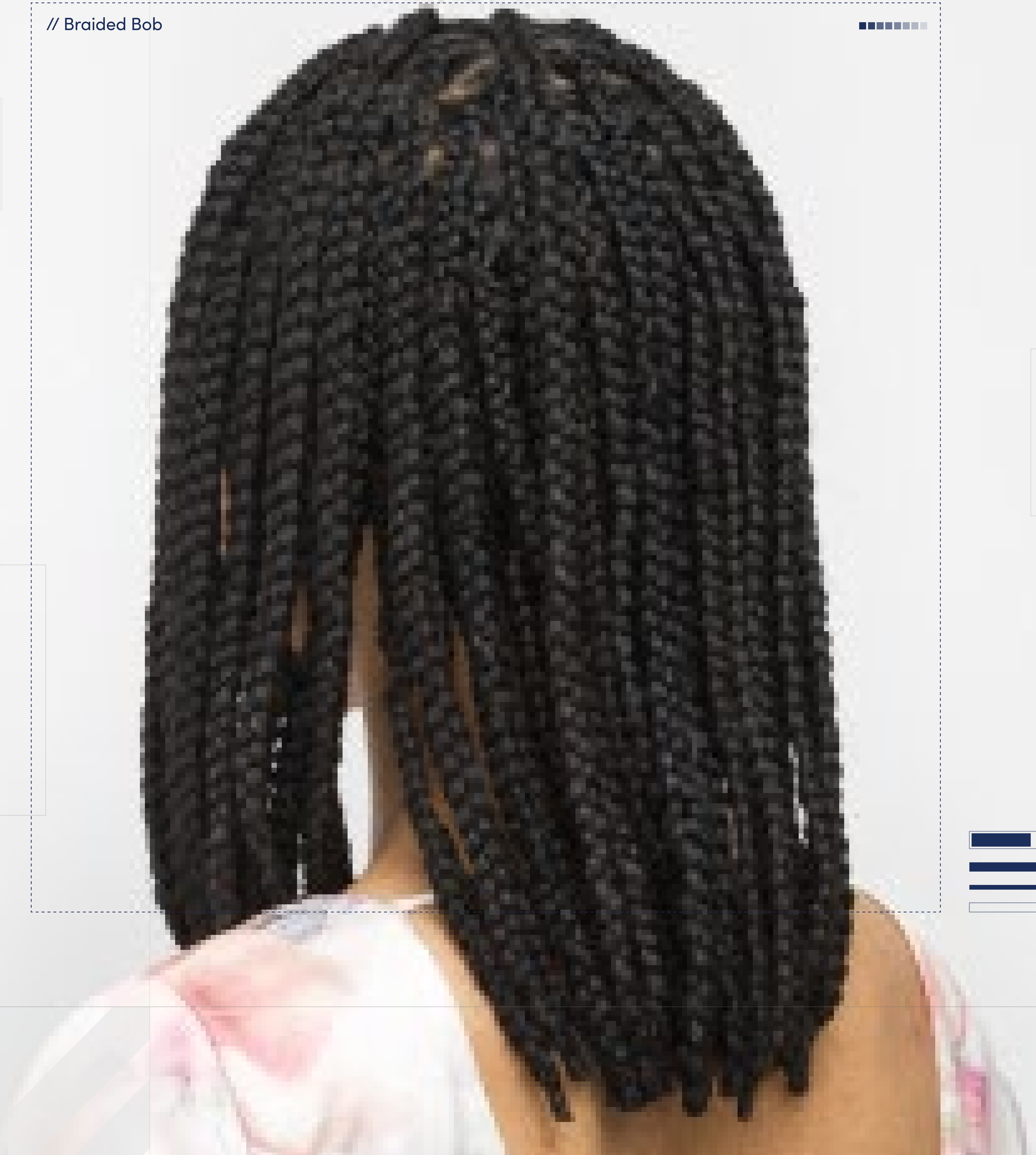


Bob

Bob
Bob

Download Images

// Braided Bob



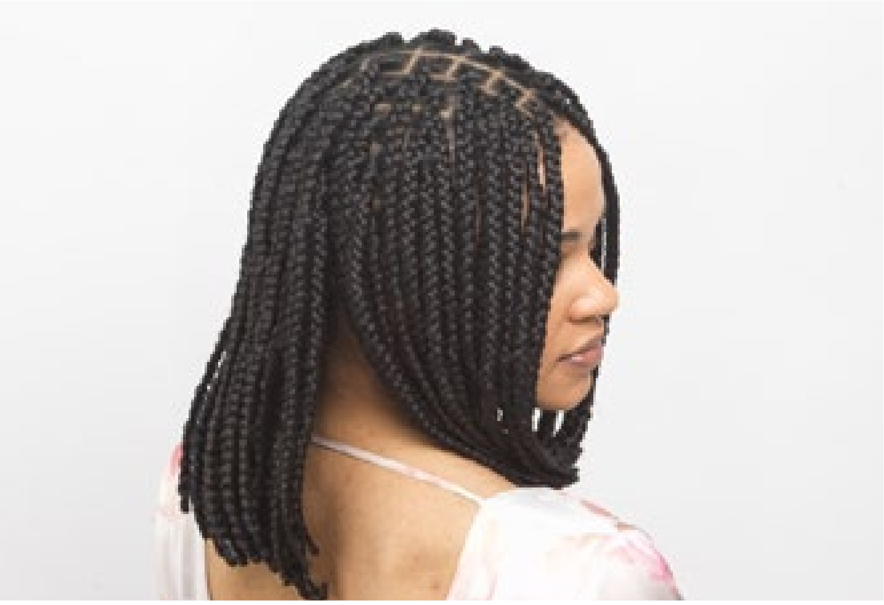
// Hairstyle

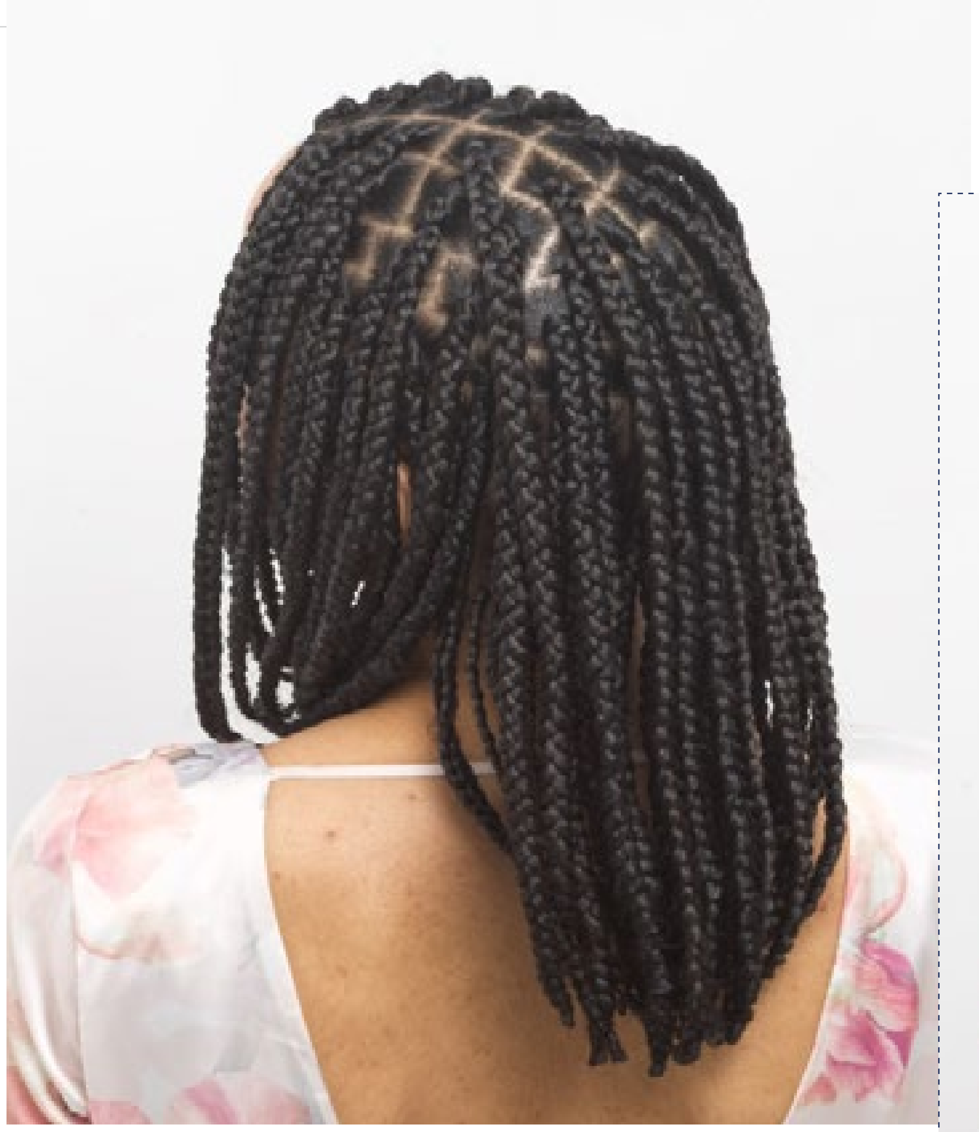
Braided Bob

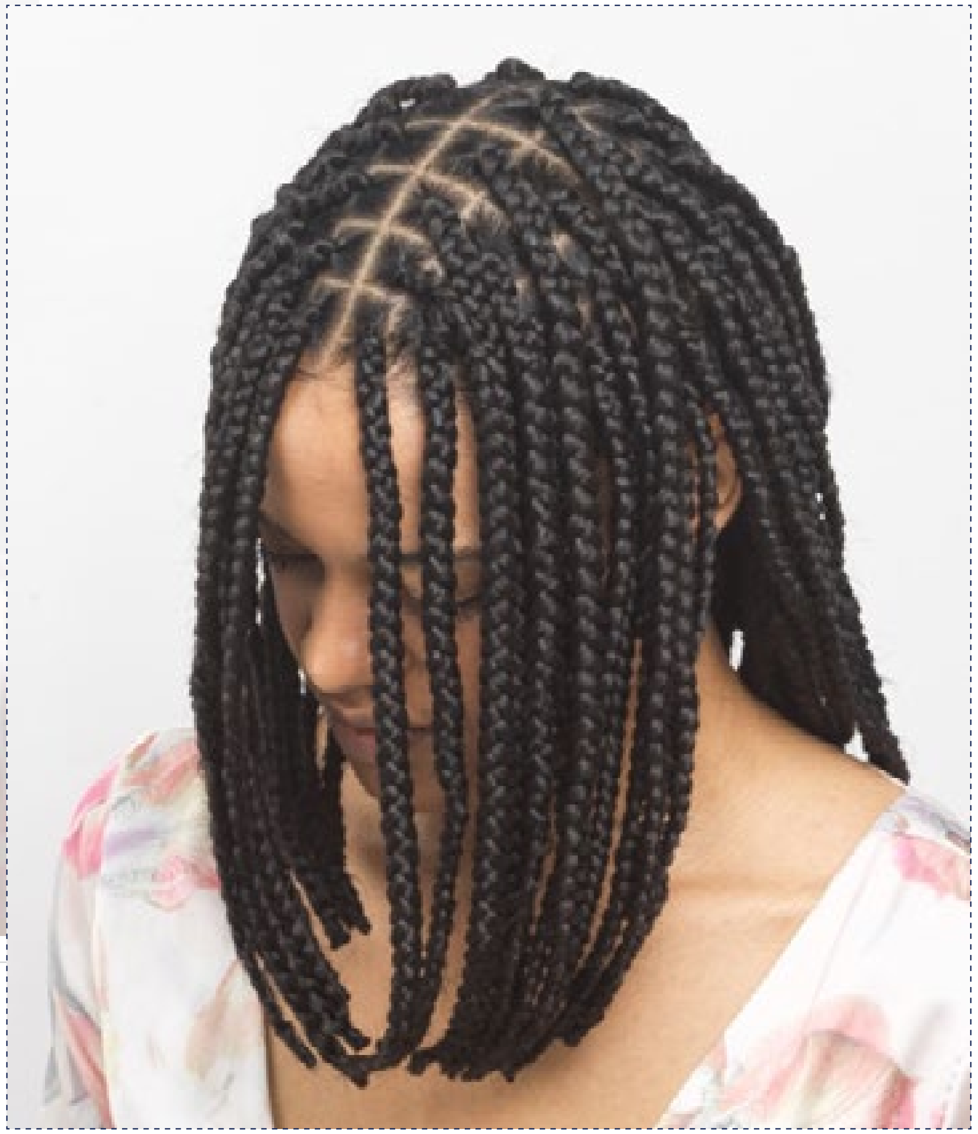
Style Background: Braided Bobs can be achieved with Box Braids or knotless braids that are short enough to resemble a bob haircut; they can be as short as chin-length and as long as collarbone-length.

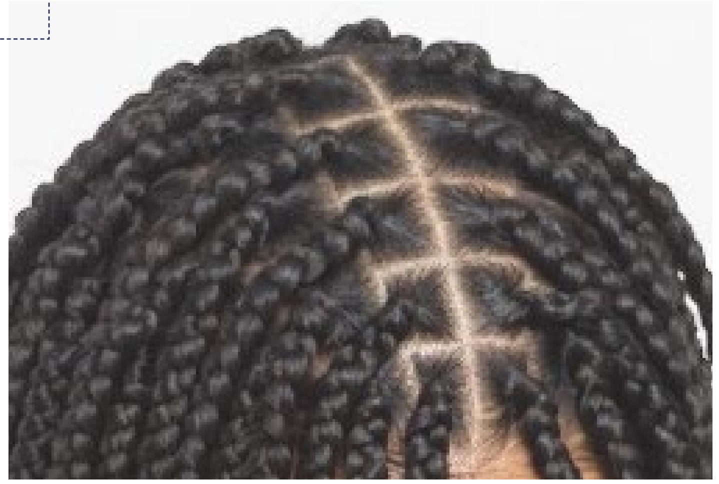
Box Braids are achieved by parting hair into sections, then parting those into smaller sections and braiding. People usually weave synthetic or human hair extensions into each braid to add length. Recently, knotless braids have become hugely popular. Traditionally, Box Braids would have a knot at the base of the braid, close to the scalp, and this is how the extension hair is added. However, many people now prefer to start off each braid with their own hair and feed in the extension hair as they go along, allowing for a sleeker, knotless look. The style featured is a Braided Bob with knotless braids.

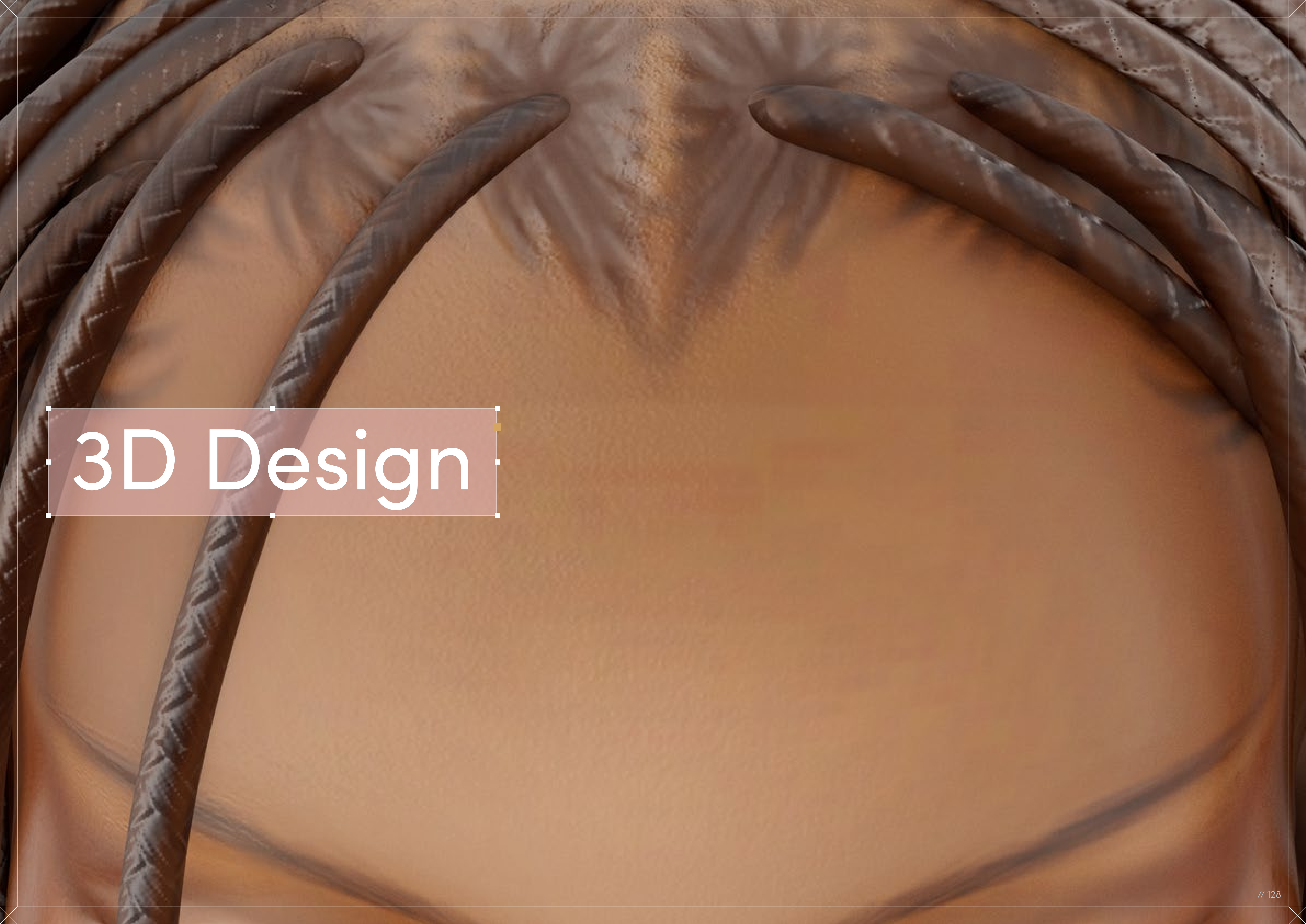
Download Images





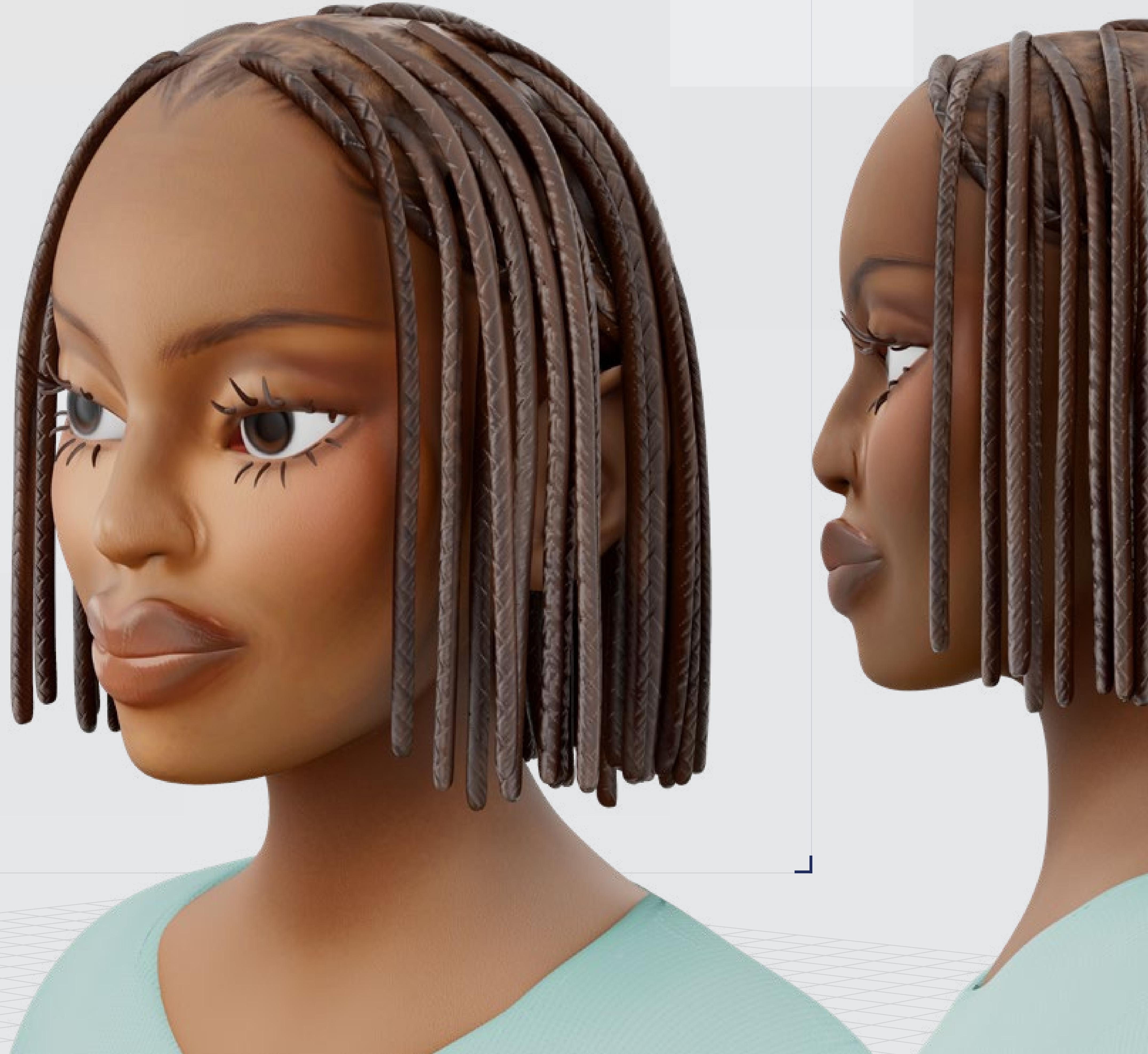






3D Design

Braided Bob

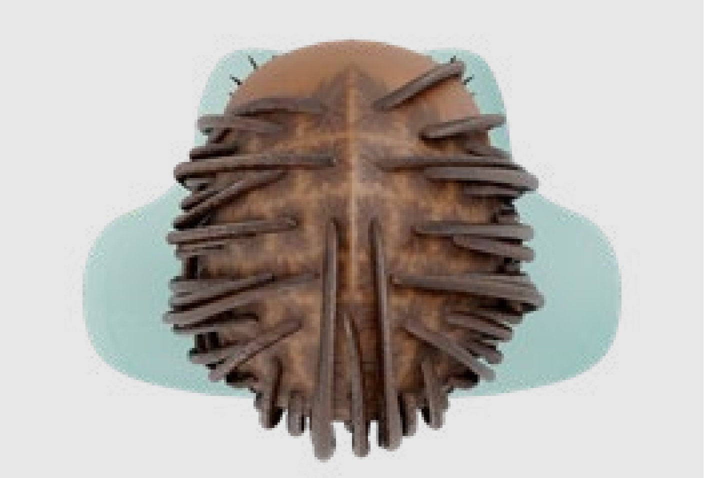


Digital Artist: **Malika**

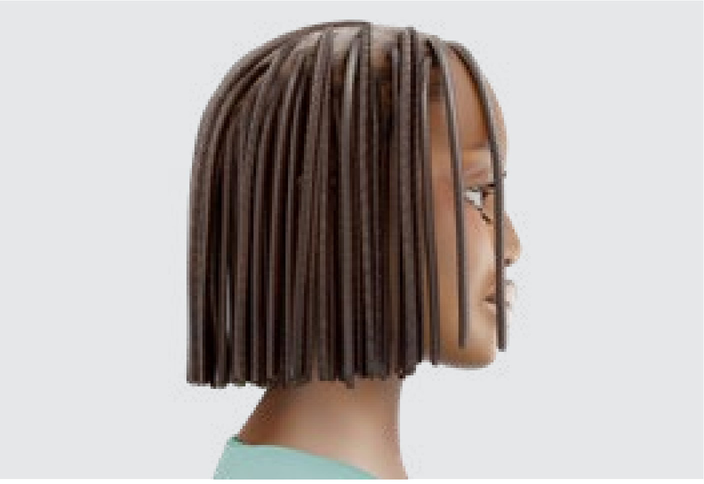
// Braided Bob

3D Design

Curl Type Modeled: **3B**



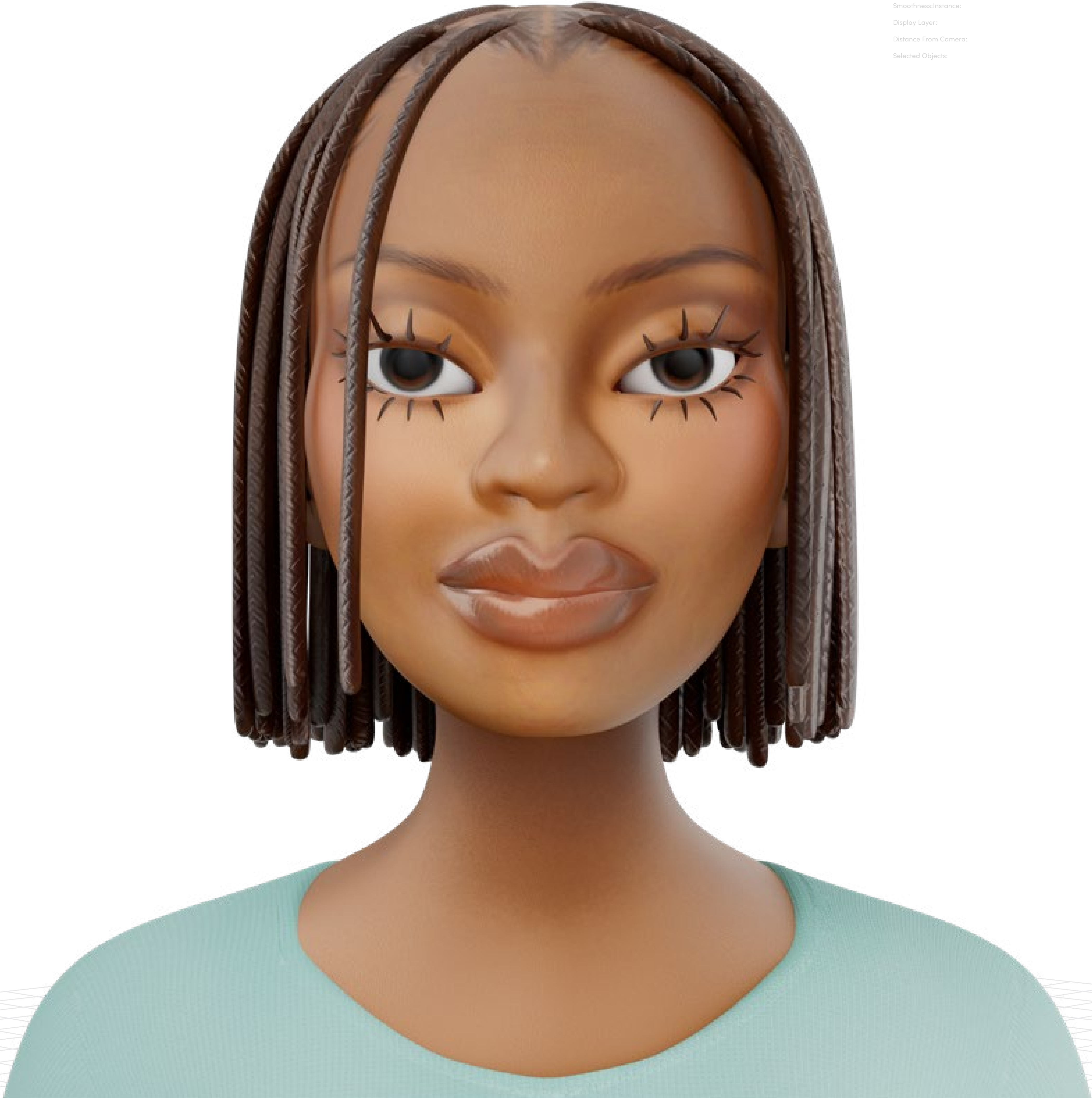
// 01



// 02

Download 3D Render

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994



//

Project

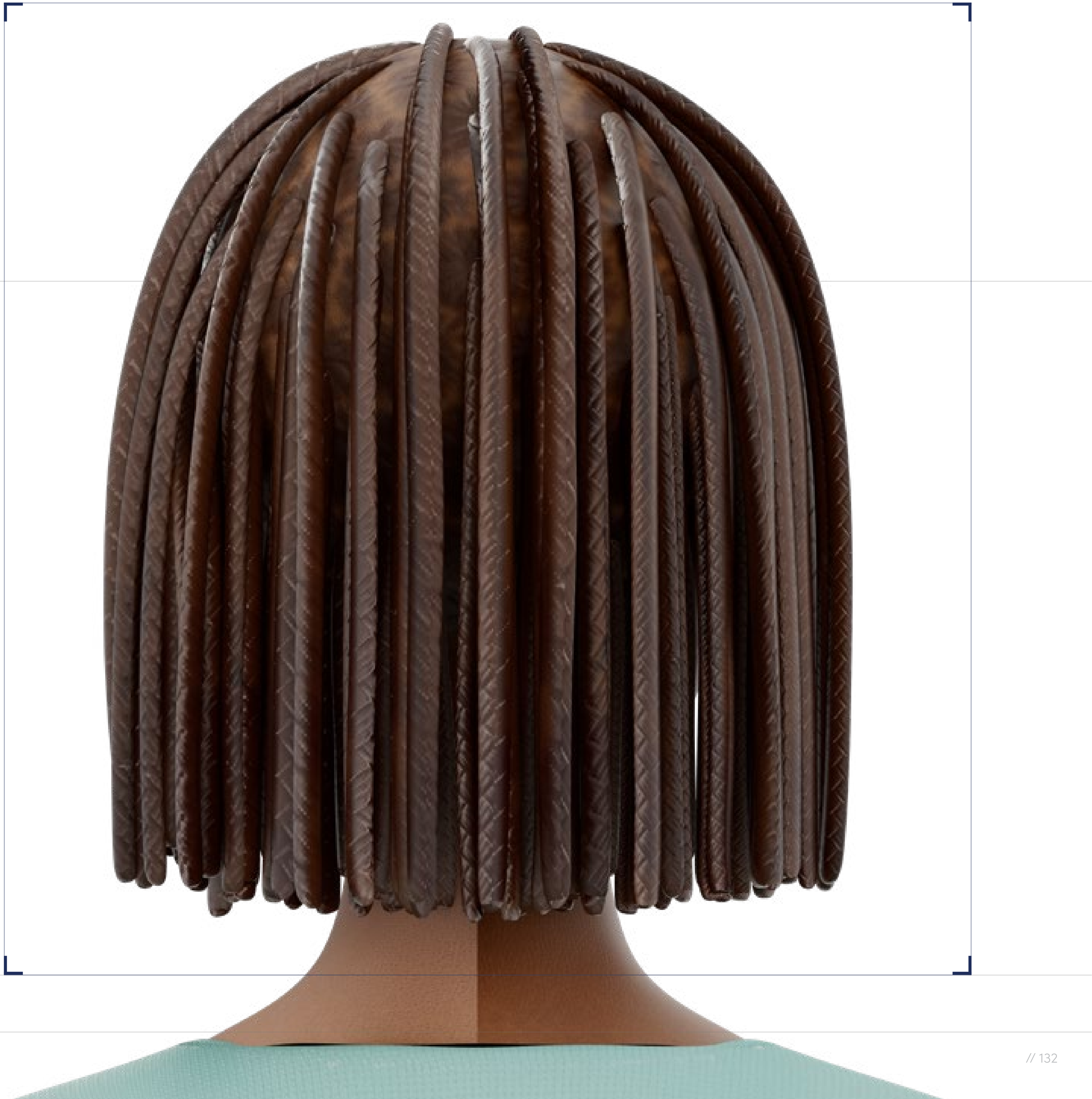
// Braided Bob

Settings

// Project Settings

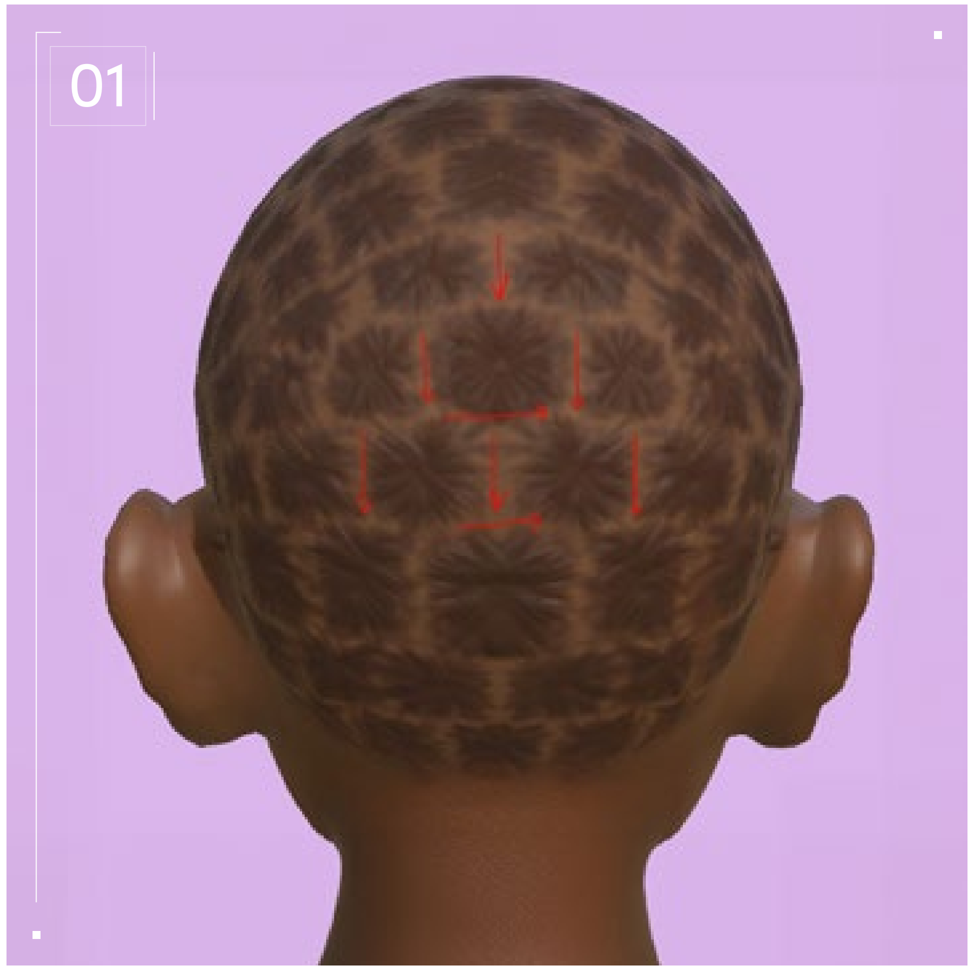
Program Guide

Sculpting and Modeling: Blender and ZBrush
Texturing: Blender and Substance 3D Painter



// Project Settings

Step-by-Step Guide



// Blocking

The Braided Bob is a pretty complicated hairstyle to create because it can very quickly become dense in polygon count. That's one reason why I wanted to save on my polycount by having the scalp be texture instead of geometry, same as for the braiding pattern. I also personally feel like Braided Bobs in real life can aesthetically be a bit of a hit or miss, so it was important to get something that would still look stylish. That's why I went for a very sharp, straight bob.

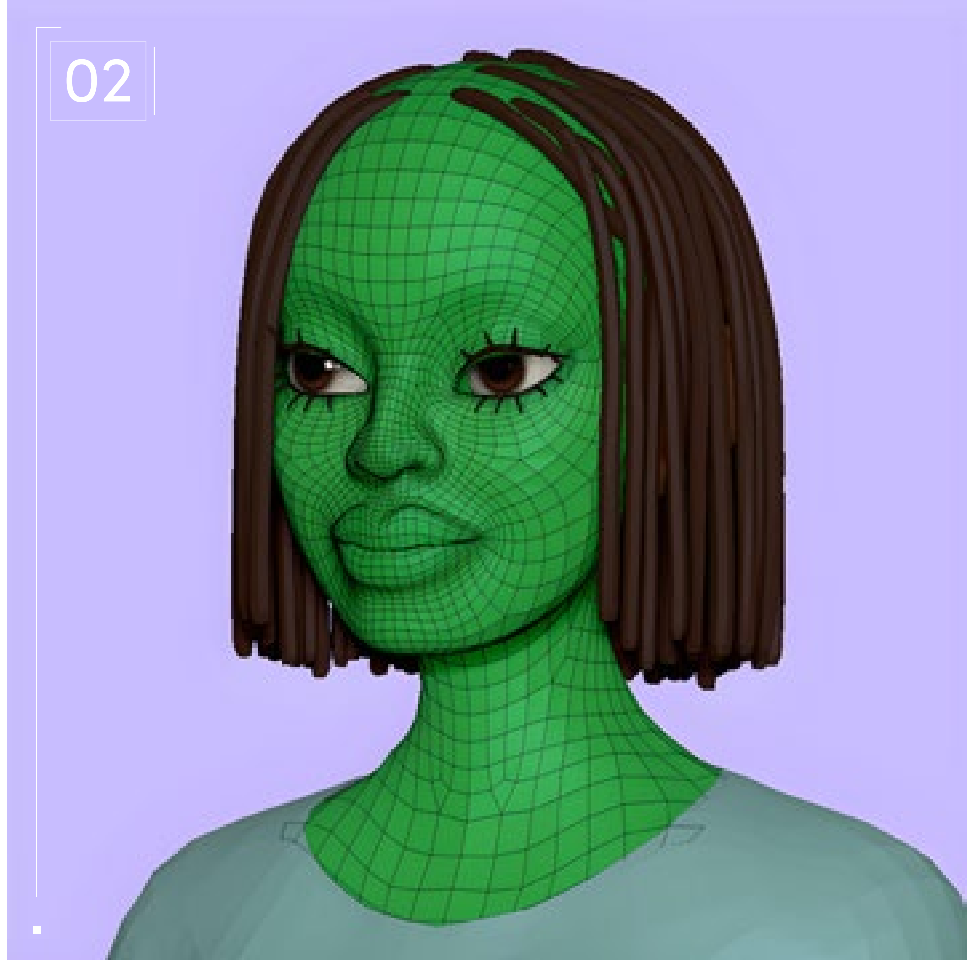
I start by planning the parts — I decided not to go with aligned parting but more of a brick-like parting so that it would lie better at the back.

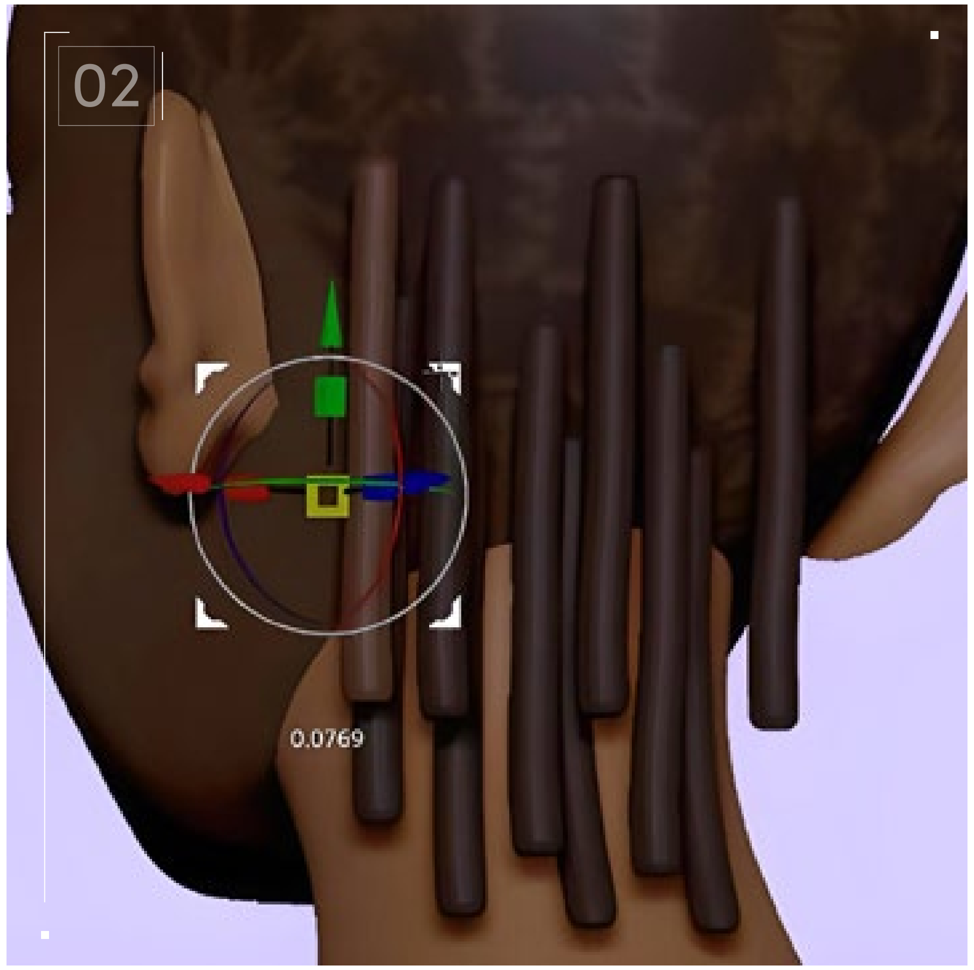
I then created the tube with the CurveTube brush and disabled the size in the curve modifier to make sure I would have an even size throughout. To save time, I created a single tube and duplicated it using the gizmo and holding CTRL. I tweaked them as little as possible because it's important for Box Braids to be as identical as possible. After that it was pretty much the repetitive job of placing each individual braid while following my partings as a guideline. If your partings are squares that line up perfectly, once you've done a single row, you can usually duplicate a whole row instead of one braid at a time. I went for a brick pattern but you can still work quickly here by duplicating every other row. Once the back is done, it's just a matter of breaking symmetry and placing braids in a way that fits your planned hairstyle while also following the parting that you've already made. Sometimes some partings might not line up exactly, so it's perfectly fine to combine some into one or split some if they're too big.

Next, start doing the hair strokes using Polypainting. I decided to show dimension with the parting through the head's Normal map instead of making them their own geometry, since this could very quickly make the whole bust way too polygon dense. It's also just easier to have control over the hair flow, because yes, the parts don't just go any ways — the hairs actually reunite at the center where the twist starts. A bit like a ponytail but on a smaller scale.

I then use the ClayBuildup brush to add dimension on each part to give the effect of real hair on the scalp.

Step-by-Step





// Modeling

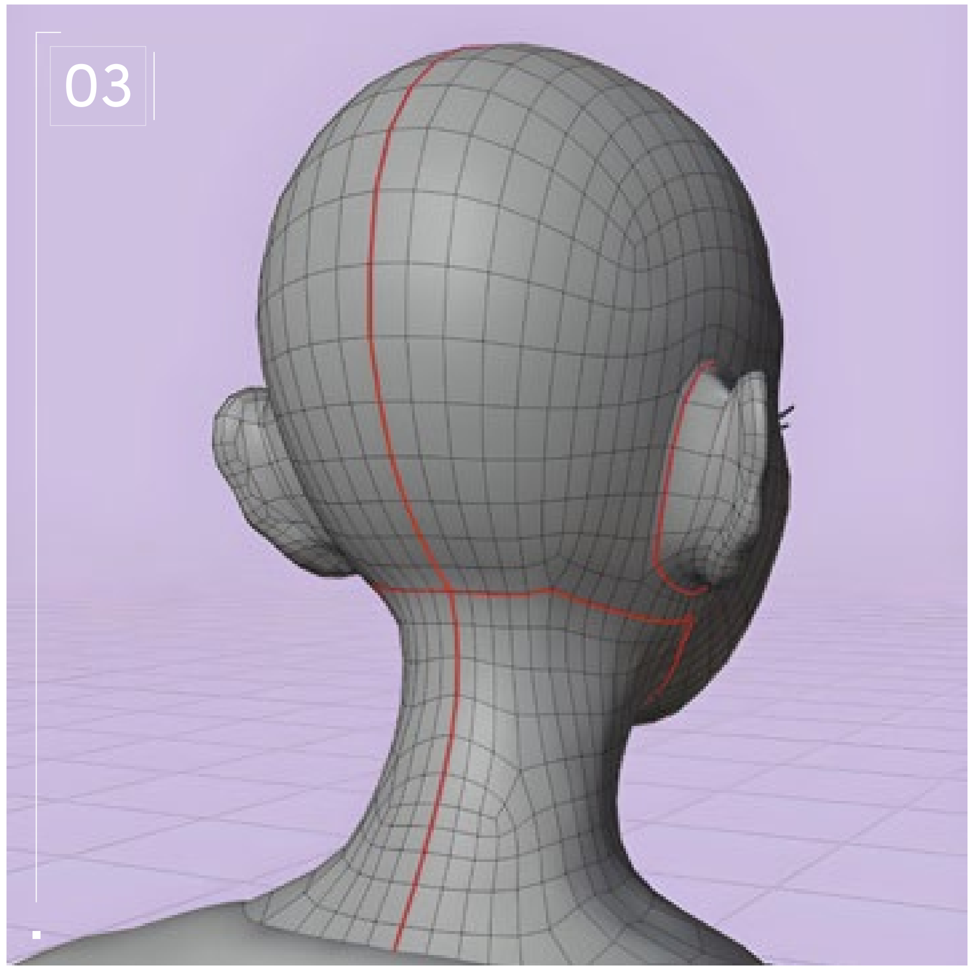
To save time, I used a base model that had already been retopologized. I used both the Move brush and the Move Topological brush to move around the base model so that it could match the bust.

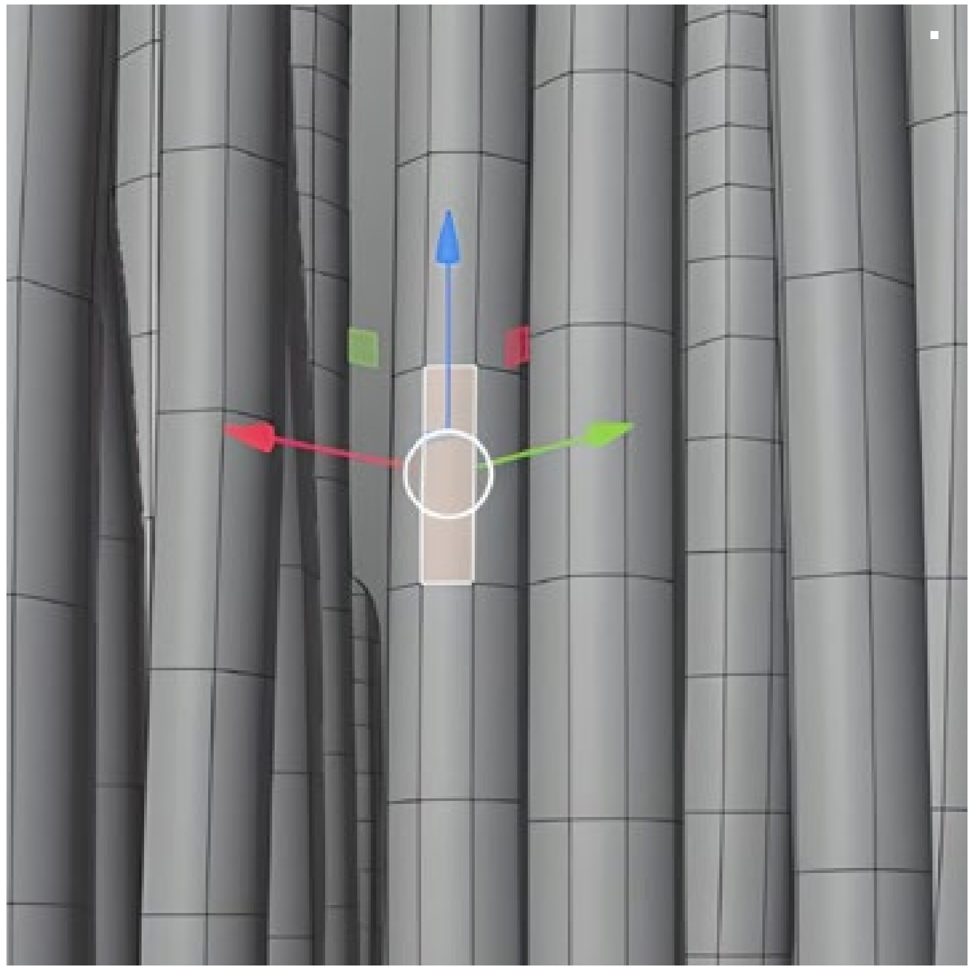
Once the base model was as close as possible to the original sculpt, I then used the Project tool in the Subtool tab and clicked "Project All." To make sure that all the Polypainting and small geometry details of the original bust would be copied on the retopologized base model, I divided it (hotkey: D), hit "Project All," and did this several times until I was satisfied with the results.

Quick note: To use the projection tool, all subtools must be hidden except for the subtool whose details you want projected and the subtool you want the details projected on. You also need to be on the subtool that will get the details projected on.

Since the braids are just simple cylinders, I remeshed them as low as possible using the ZRemesher tool that can be found in the Geometry tab, setting the Target Polycount as low as possible.

Step-by-Step





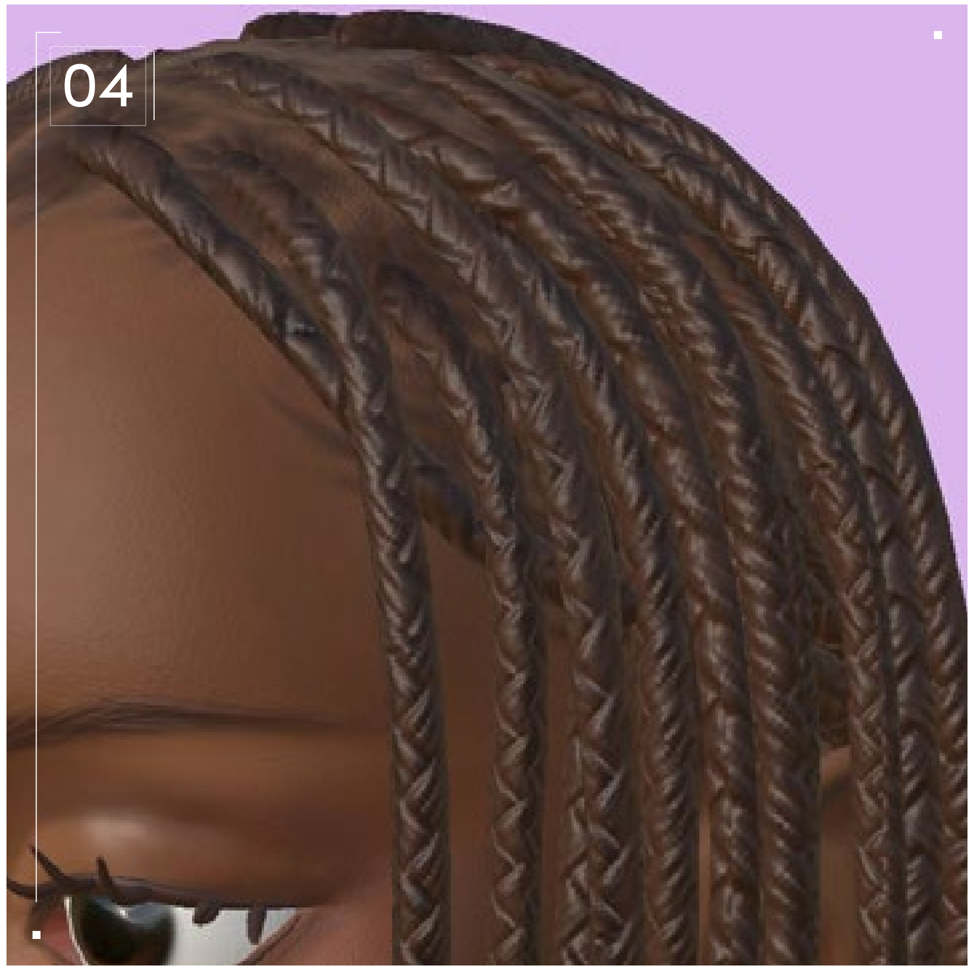
// UVing

With UVing, I tend to go off vibes, to be honest. My base model was already UVed so I didn't have to figure it out, though face UVing is pretty similar for all faces. I personally just cut a seam at the base and back of the neck, around the ears, and across the cranium, stopping a bit before the hairline.

As for the braids, since I was going to either draw the braid pattern myself or use a texture, I figured it would be easier to work with if the cylinders are as straight as possible.

To do that, in Blender, I select a single rectangle, click "U" to unwrap it, move the vert around to have a rectangle as straight as possible, click "CTRL" and "L" to have the whole twist selected, click "U," and pick "Follow Active Quads" and click "OK." This unwraps the braids, making them very scaled up, so it's necessary to scale it back. Sadly, this has to be done for every single braid and they all have to be scaled to a size that is as uniform as possible.

Step-by-Step



// Texturing

I do most of the texturing work using Polypainting in ZBrush as I believe it helps me figure out the face and hair better. So next I simply export a high-poly FBX off my ZBrush using the FBX ExportImport tool, in the Zplugin tab.

It's important to rename the objects that will be baked to match each other on the low and high poly and specify them with "_low/_high" (in all lowercase) at the end of your name to make sure there is no confusion by the software.

To bake both the Polypaint and the Normal (mostly for the head to get the parts detail), I go into the Texture Setting tab and click on "Bake maps." I personally like to set the size to 4,000 to get as much detail as possible when I size down. I then pick my high-poly, precise that the mapping needs to be by name. Also, it's important to be precise for the ID that the source color is the vertex color or your ID will just simply be grey.

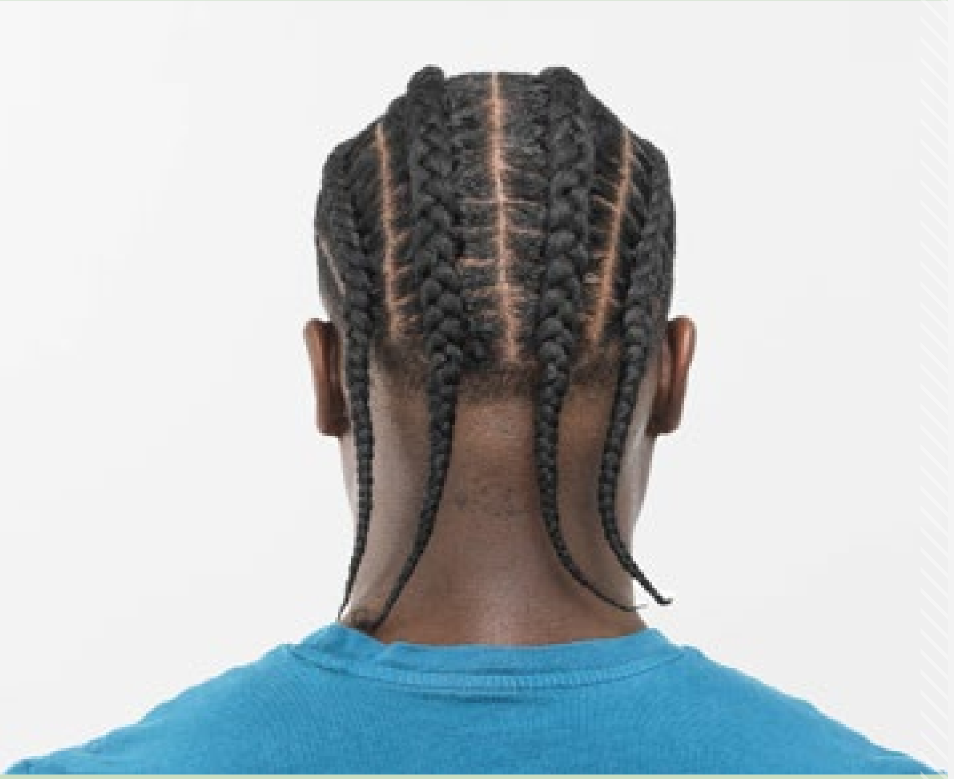
When the baking is done, simply create a fill layer and drag your baked ID and Normal to their respective channels.

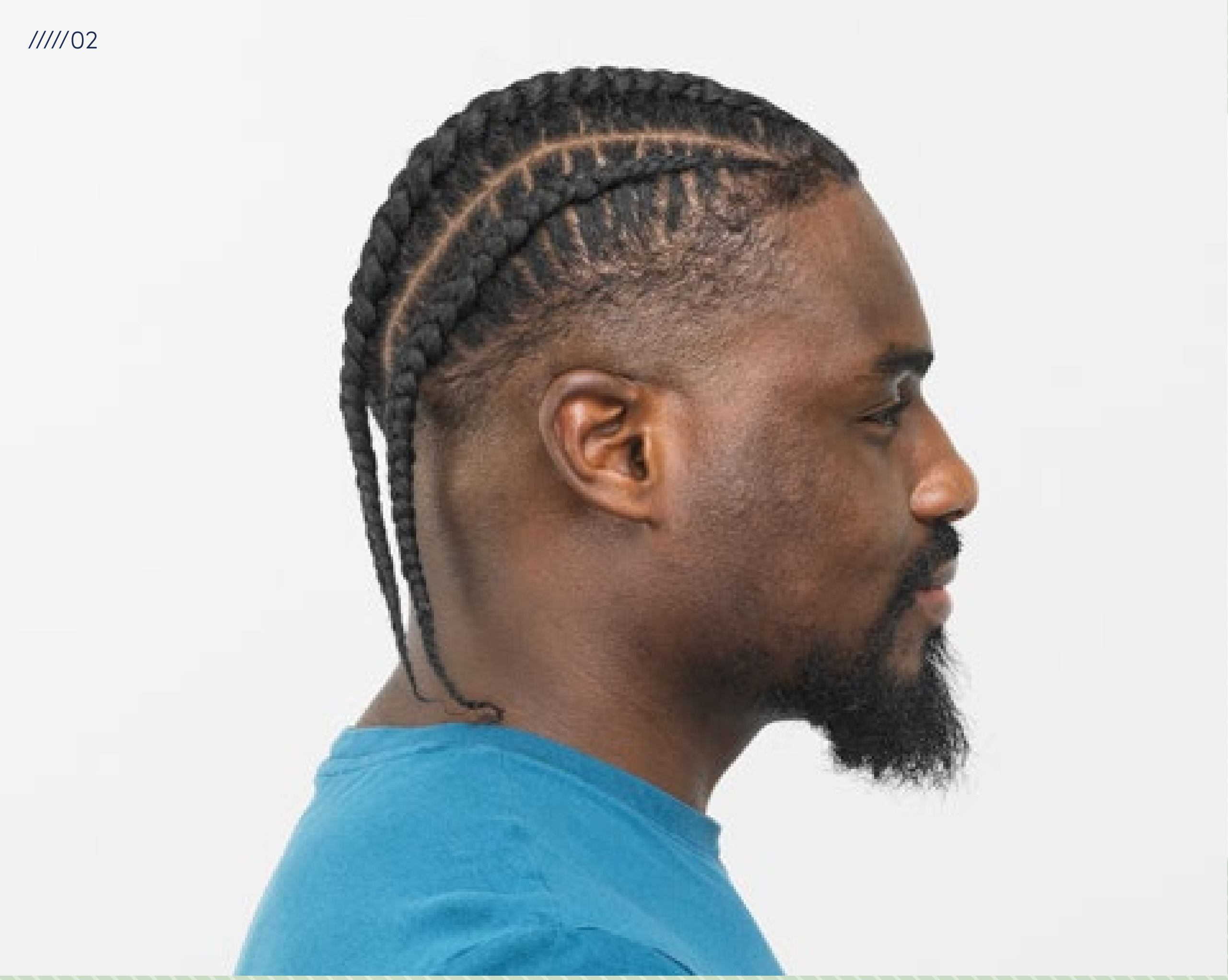
I then use an additional fill layer or just a paint layer to fix anything that is off on the ID. In my case the ears and neck were not connected to the high-poly model, so the Normal was badly baked there but it was fine since they were details that I didn't really need. The hair parts were the most important, so I simply used a white mask on my fill layer that has my baked Normal and erased the parts I didn't need.

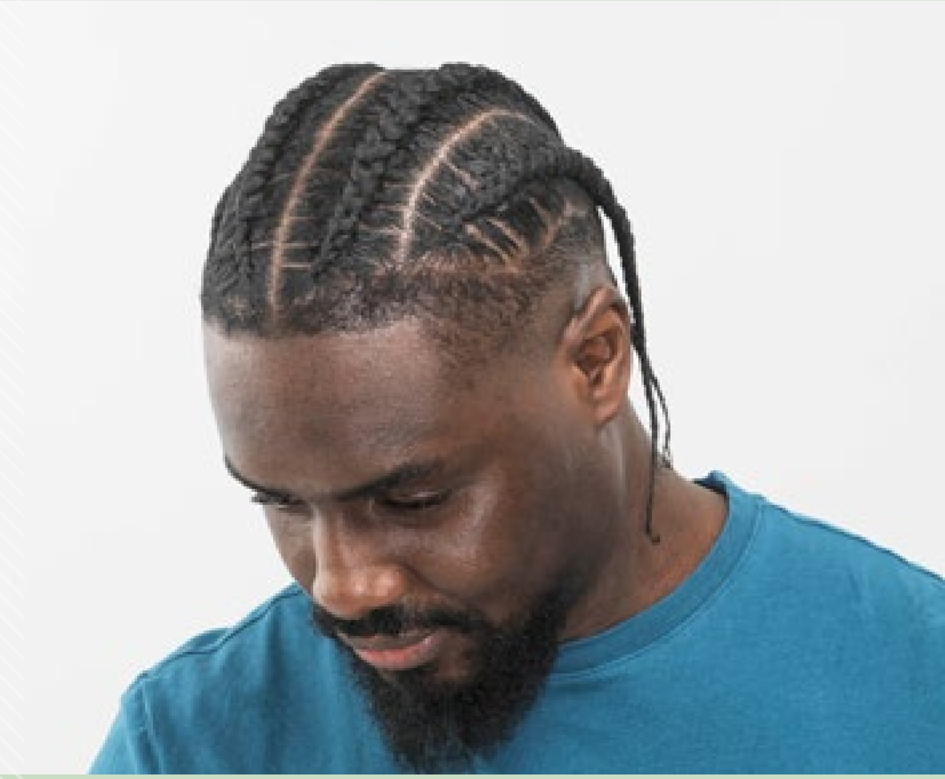
I usually go for a very stylized choice and I like things to look a bit toy-ish so I usually stop there. But this time I decided it would be nice to have more roughness variation as well as skin texture, so I added that using the skin materials already available in the Substance library. Varying them for each part of the face (the skin on the nose doesn't look like the skin on the cheek, for example, and materials exist for a lot of body parts in this way.)

To make the hair parts as visible as possible, I made the back of the head more matte than the overall head and created an even more matte layer to add some very dark hair strokes. For the braids, I used a braid texture to give the illusion of real braided extensions. I still manually added some hair stroke details and made some of the braids' dents a bit deeper with a height fill layer at certain parts, as well as some brown hair strokes to give some color variation.

Stitch Cornrows





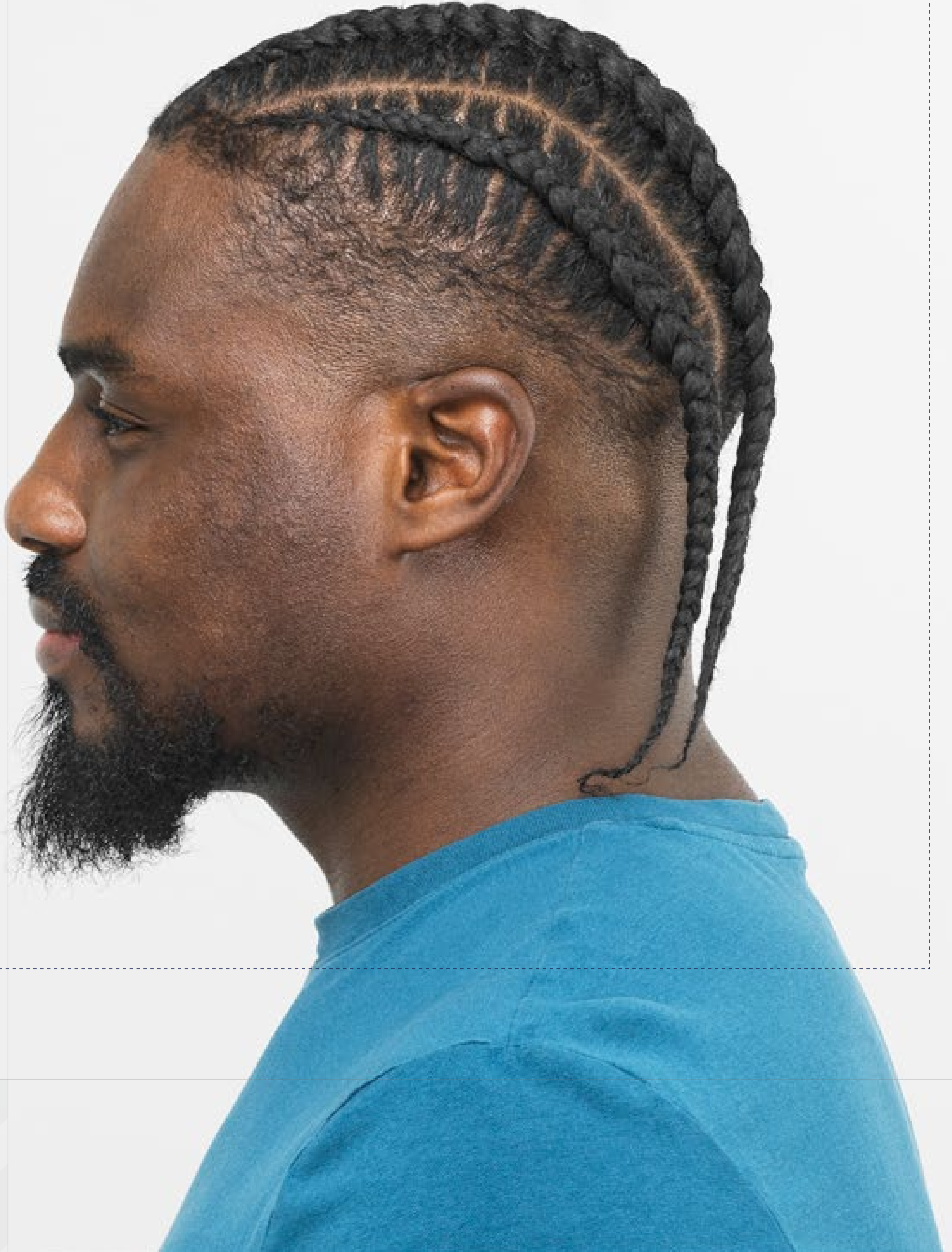


++++

with Fade

Download Images

// Stitch Cornrows w/ Fade

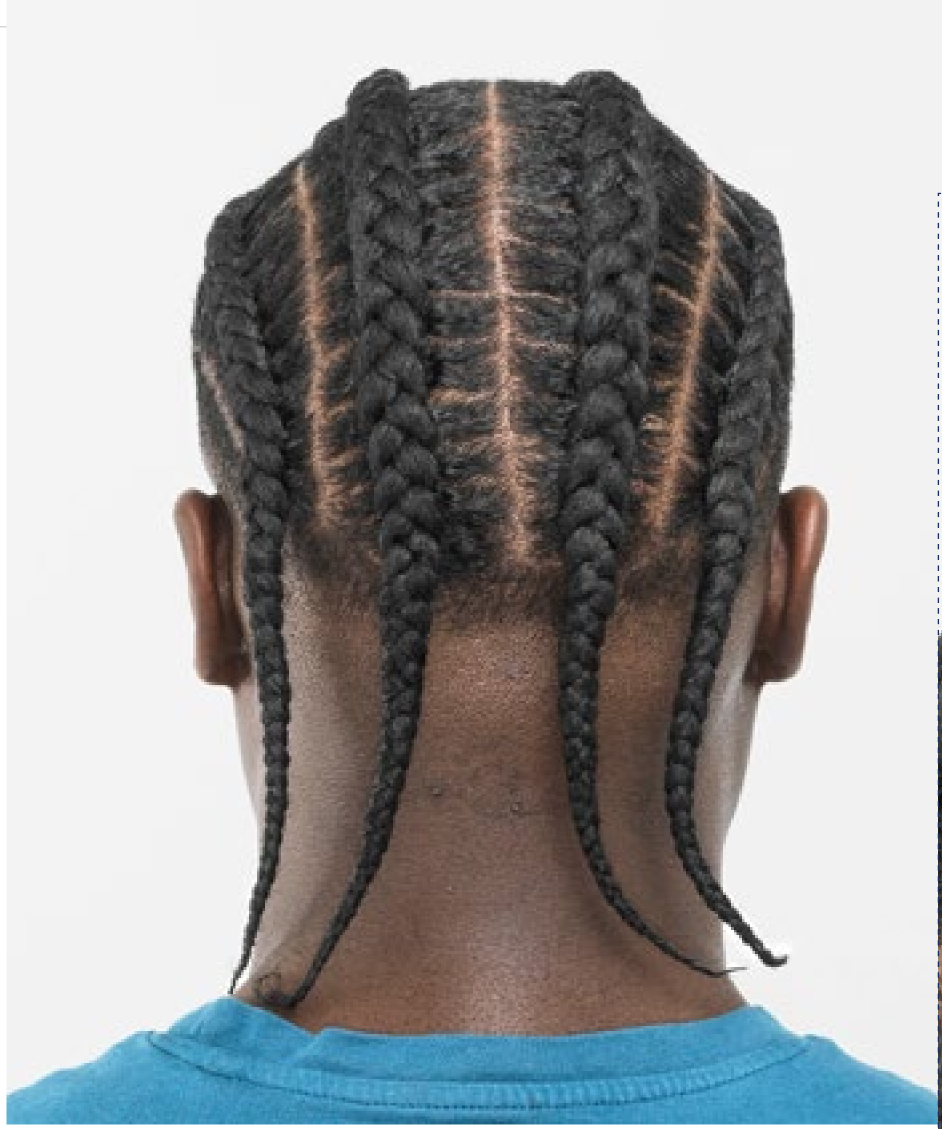


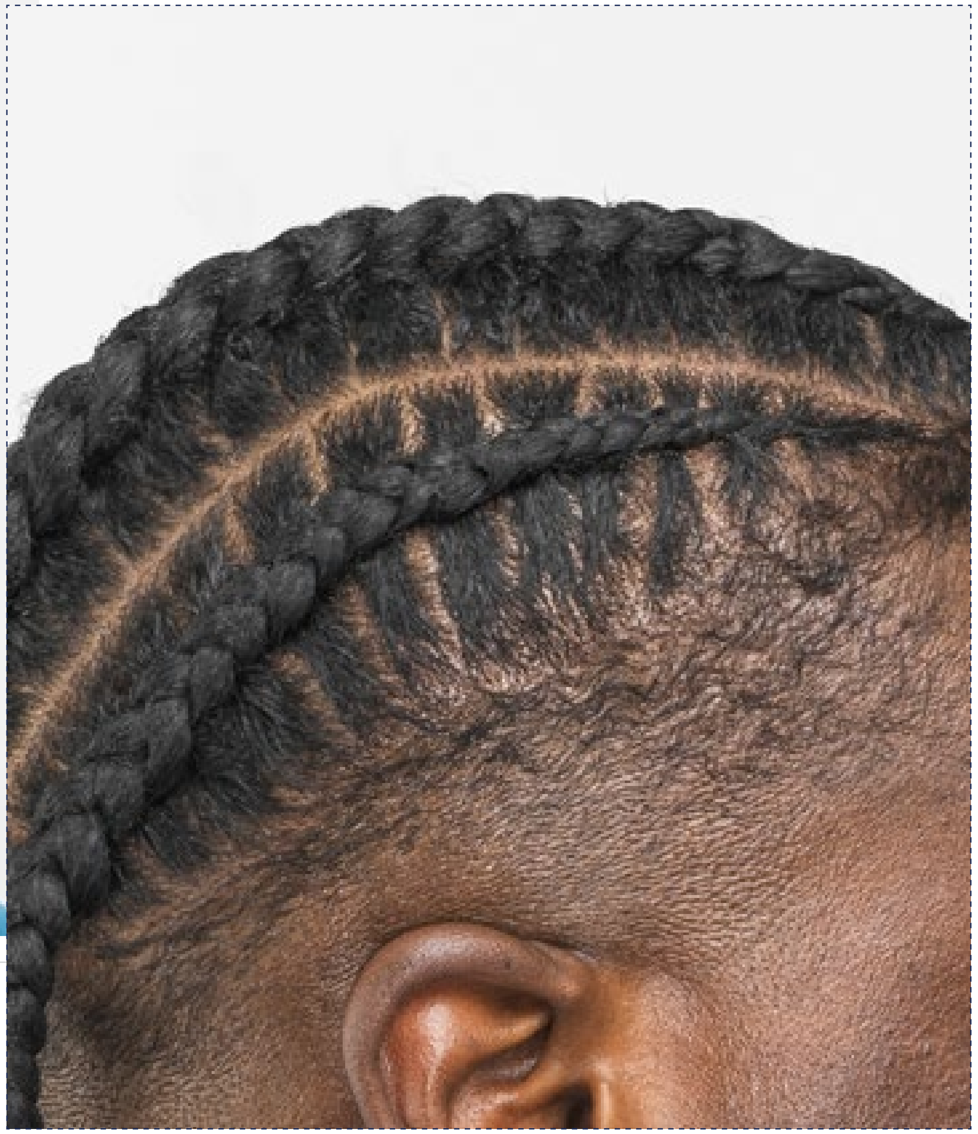
// Hairstyle

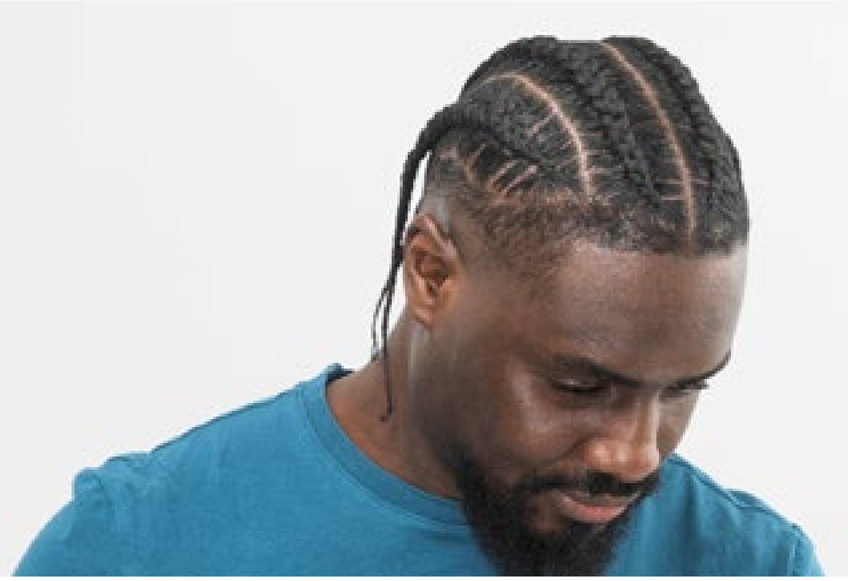
Stitch Cornrows with Fade

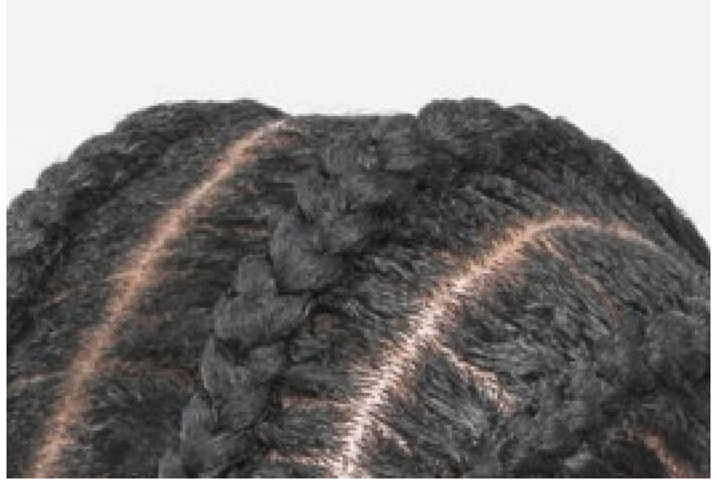
Style Background: Hairstyles often come and go, but Cornrows have remained firmly in fashion — in one form or another — for centuries. Alone, Cornrows are a sleek, low-maintenance protective style that can last weeks but they can also be a base for creating other styles, like adding sew-in extensions. Stitch Cornrows have slices based on the technique used to gather the hair into the braid, creating defined parts in between the braids, in between each little section. Cornrows or Stitch Cornrows are often paired with other styles — such as a Fade, which is a cutting technique that cuts hair in a gradient so it eventually "fades" into the skin/scalp, or an Afro — to create different looks. Cornrows are an incredibly versatile hairstyle, worn by both men and women.

Download Images





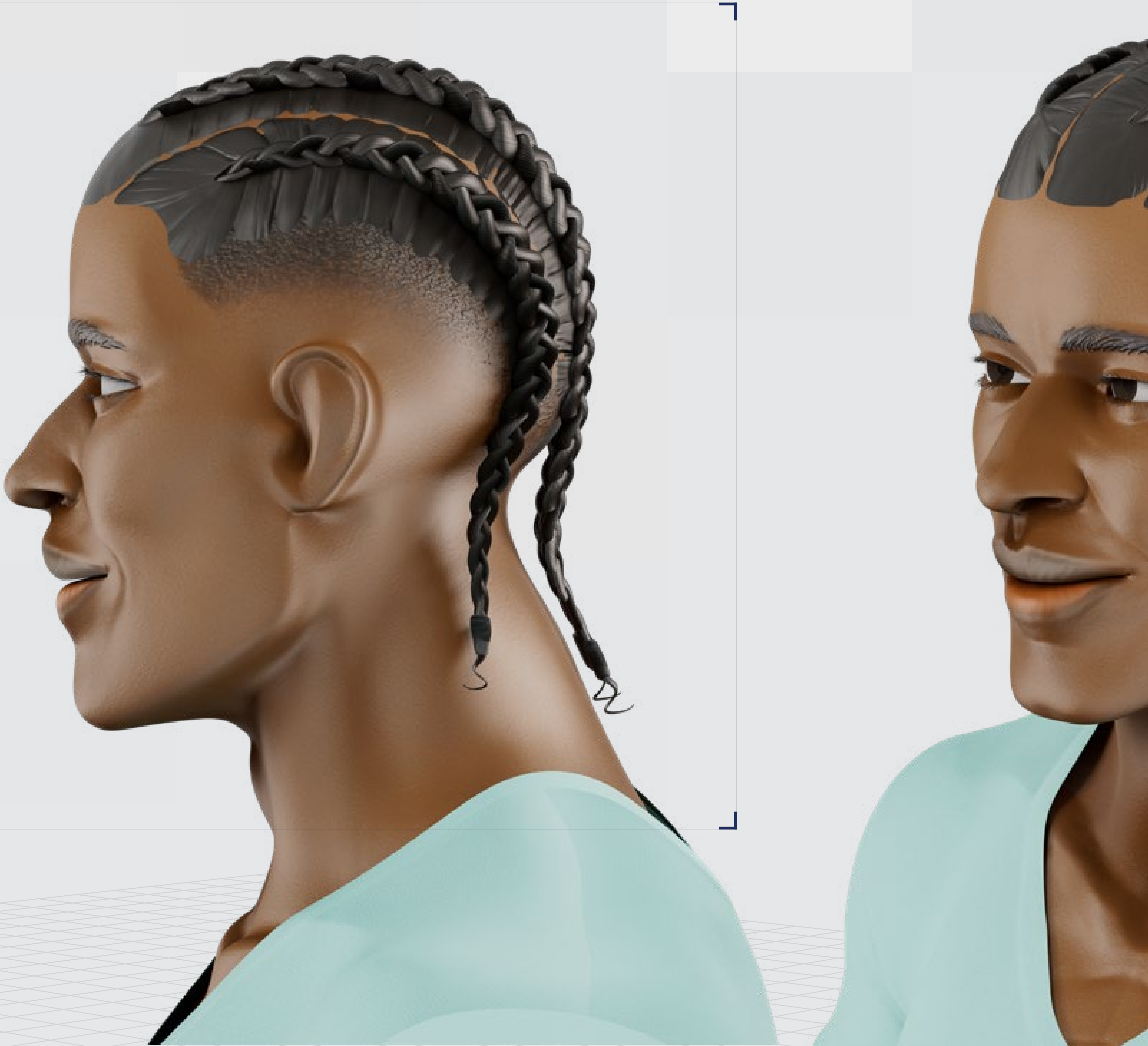




3D Design

Stitch Cornrows

with Fade



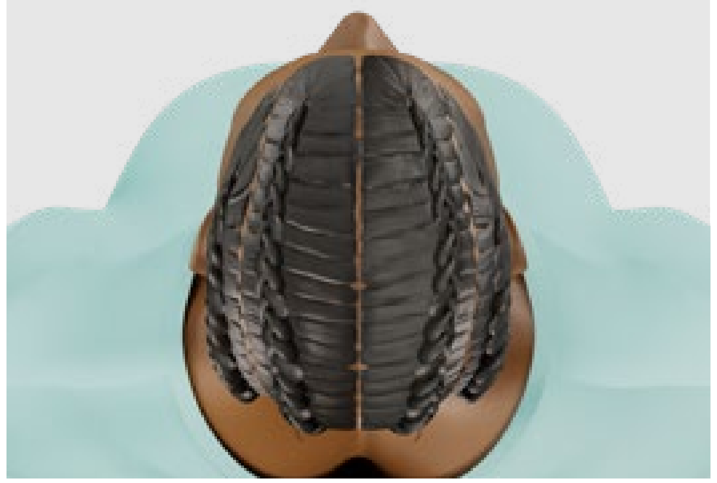
Digital Artist: **H.D.**

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

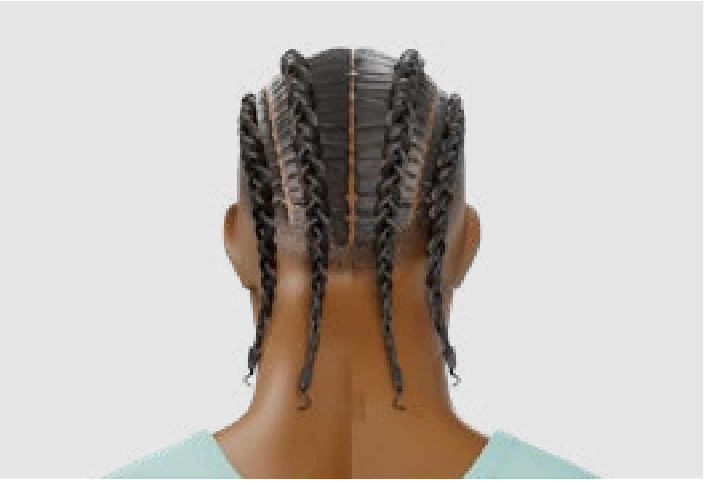
// Stitch Cornrows with Fade

3D Design

Curl Type Modeled: **4A/4B**

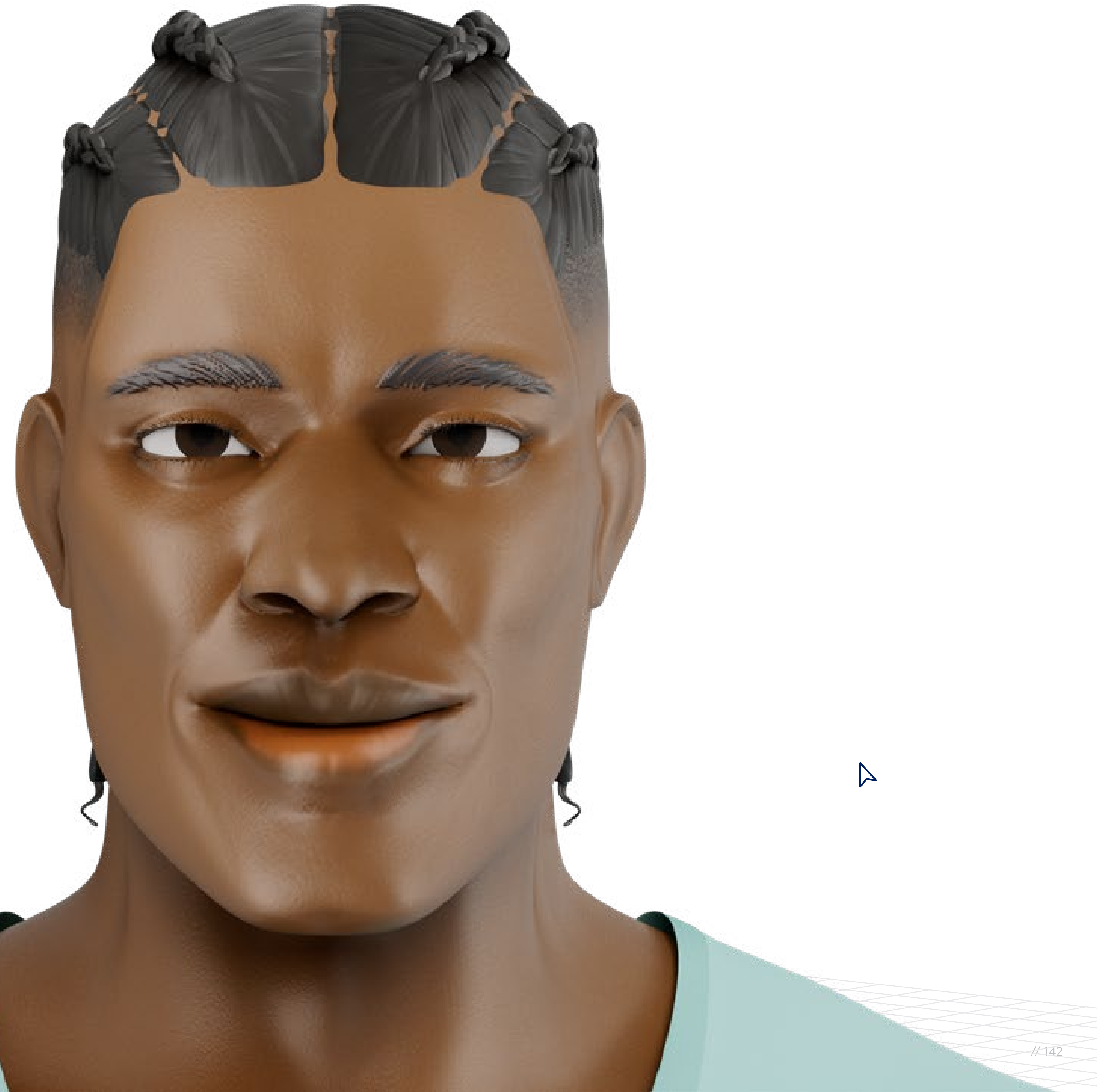


// 01



// 02

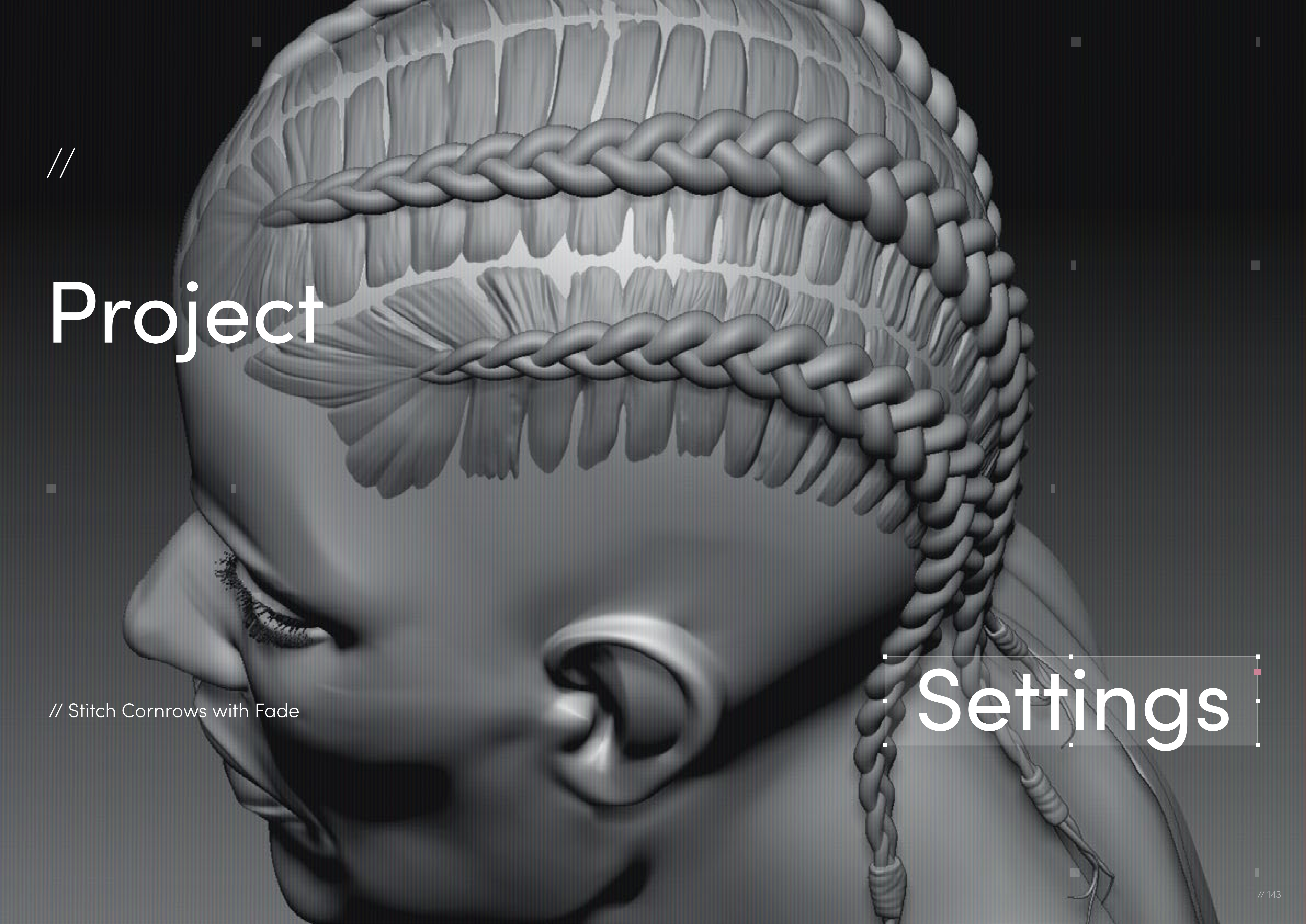
Download 3D Render



//

Project Settings

// Stitch Cornrows with Fade



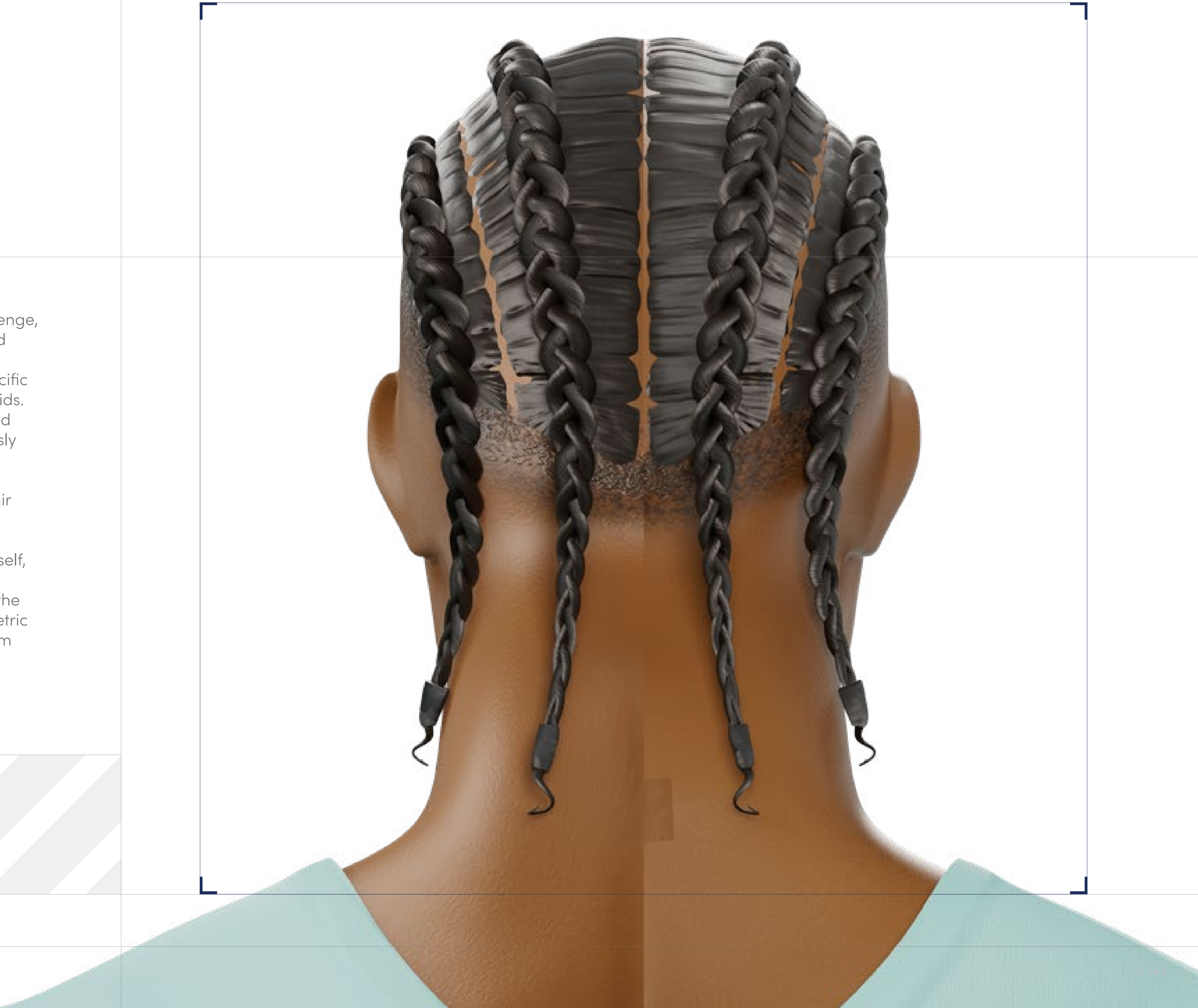
// Project Settings

Program Guide

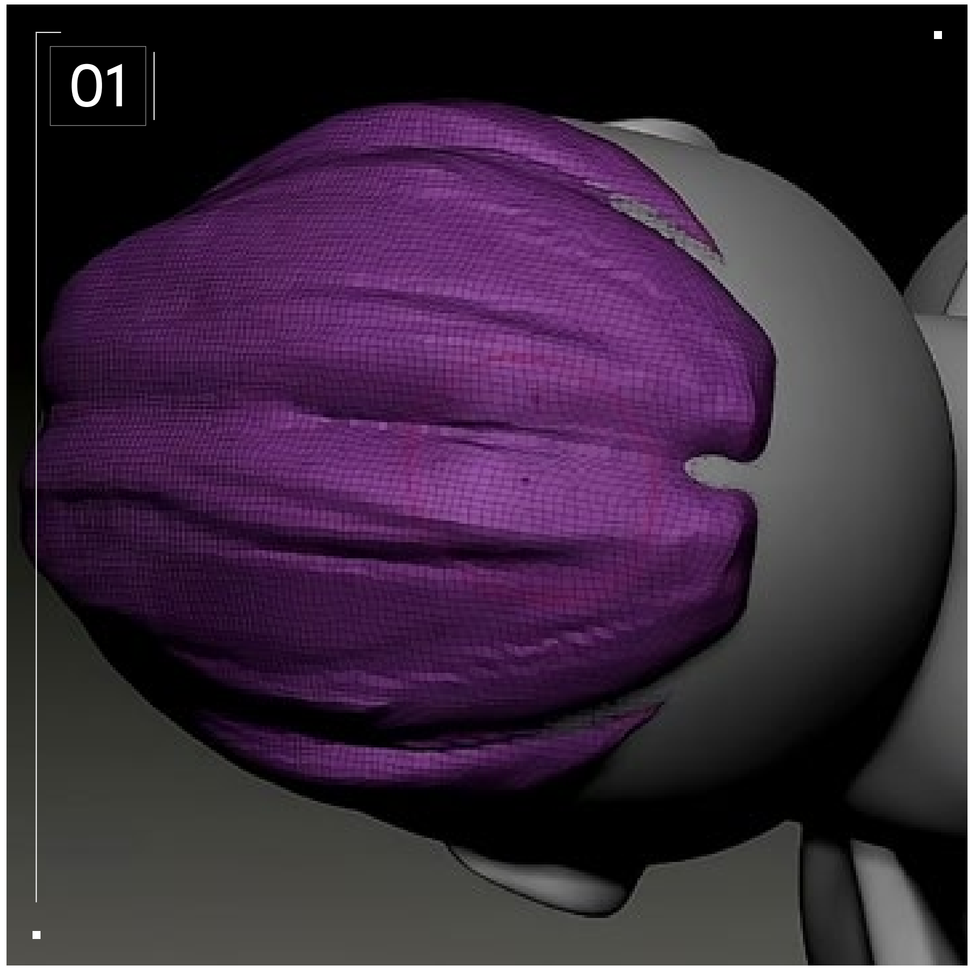
Sculpting and Modeling: ZBrush
Texturing: Substance 3D Painter
Render/UVs: Blender

This hairstyle, when I first saw it, presented as a challenge, because I knew from the references that I would need to be concerned with not just the braid placement (which was straightforward enough) but also the specific nature of the Fade and the hair that supports the braids. This was a particular hairstyle that could be supported technically by a textured scalp but that would obviously be better served by creating a supporting mesh to embody the nature of the stitches and create a more legible delineation between the Fade hair and the hair being fed into the braids.

I was also struck by the inventive nature of the style itself, which is just one of a long line of creative expressions using hair that has defined Black people throughout the course of history. It was a bold synthesis of the geometric language of braids with the pure aesthetic minimalism of a Fade — it was immediately intriguing!



Step-by-Step Guide



// Blocking

I began by laying the base color on the area of the model in ZBrush that would outline the area of the scalp. This was to ground me when laying the braids out and give me an impression of how it would eventually look.

Next, I selected a braid curve brush that I have in my library. There are many that can be bought; however, in the interest of learning, customization, or even to take the opportunity to sell your own, it can be beneficial to make your own.

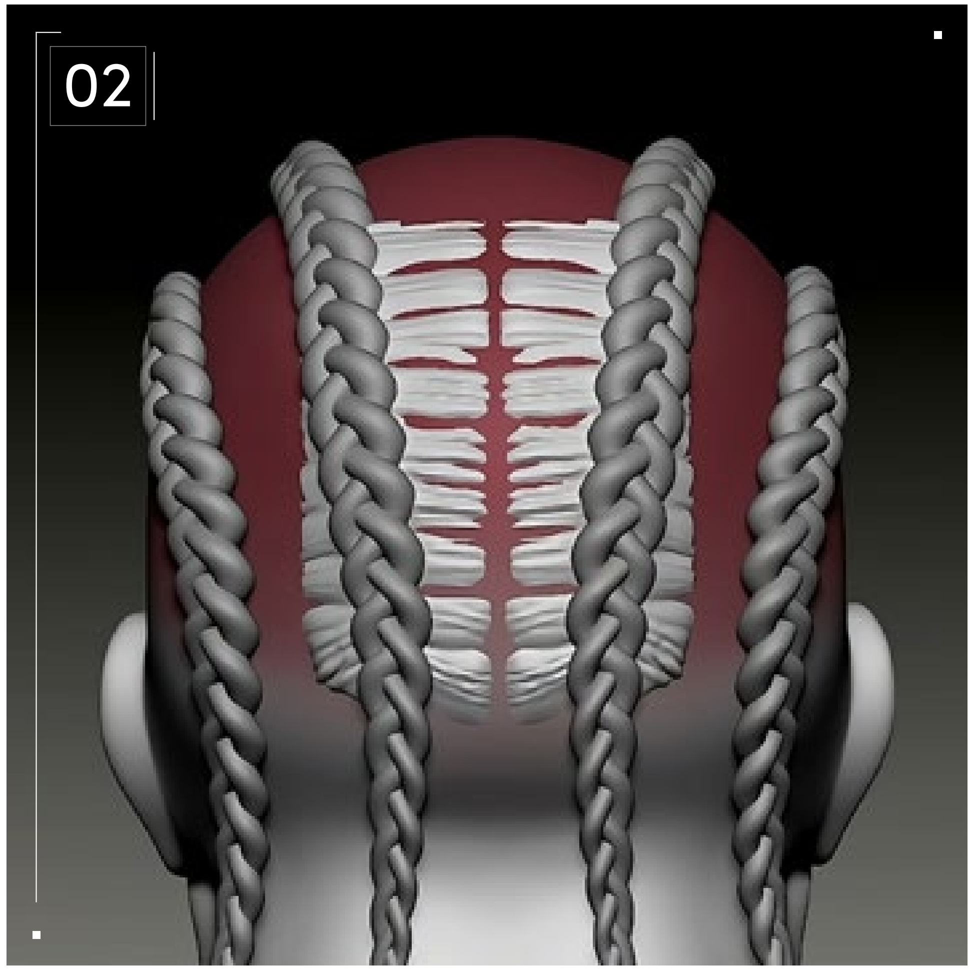
I knew that with four braids they should be symmetrical, and since they will "detach" from the scalp high up in the back, they shouldn't follow the skull down to the nape but hang freely. Knowing this, I drew the braid out beyond the head and manipulated the curve until it ran along the scalp appropriately and hung in a way that I liked.

When that was done, I switched back to the base mesh, and, using the color I'd laid down as a guide, drew out a mask along where the supporting hair would be, and then extracted it as a new mesh. I drew the basic lines where the hair would be separated to feed into each braid and examined it from a number of angles, trying to think about where the Fade would be.

After the supporting hairs for the braids had been blocked in, I then erased the color for the scalp map until I'd recreated where I would want a Fade. With the pieces in place, I moved on to the modeling step where I would create the details.

///

Step-by-Step Guide



// Modeling

Using Dynamesh, I sculpted the supporting hair into shape, working to build up volume while also trying to be mindful of the stitched nature of the style in question — it would be easy to muddy the lines and end up with surfaces that only suggested the stitching.

My goal was to create stitching that looked like it actually fed into the braids, so I tried to use my strokes to build that flow, building up with clay brushes and then carving with Dam Standard, until eventually it had finally taken shape. I inflated the braids so that they lost the thin and loose appearance and became more full, which added to the believability of the hairstyle from a distance. At some point I also realized I could add forms to the ends of the braids (to represent the rubber bands that would be used in real life) by adding a cylinder to one braid end and sculpting it with "Sculptris Pro" on.

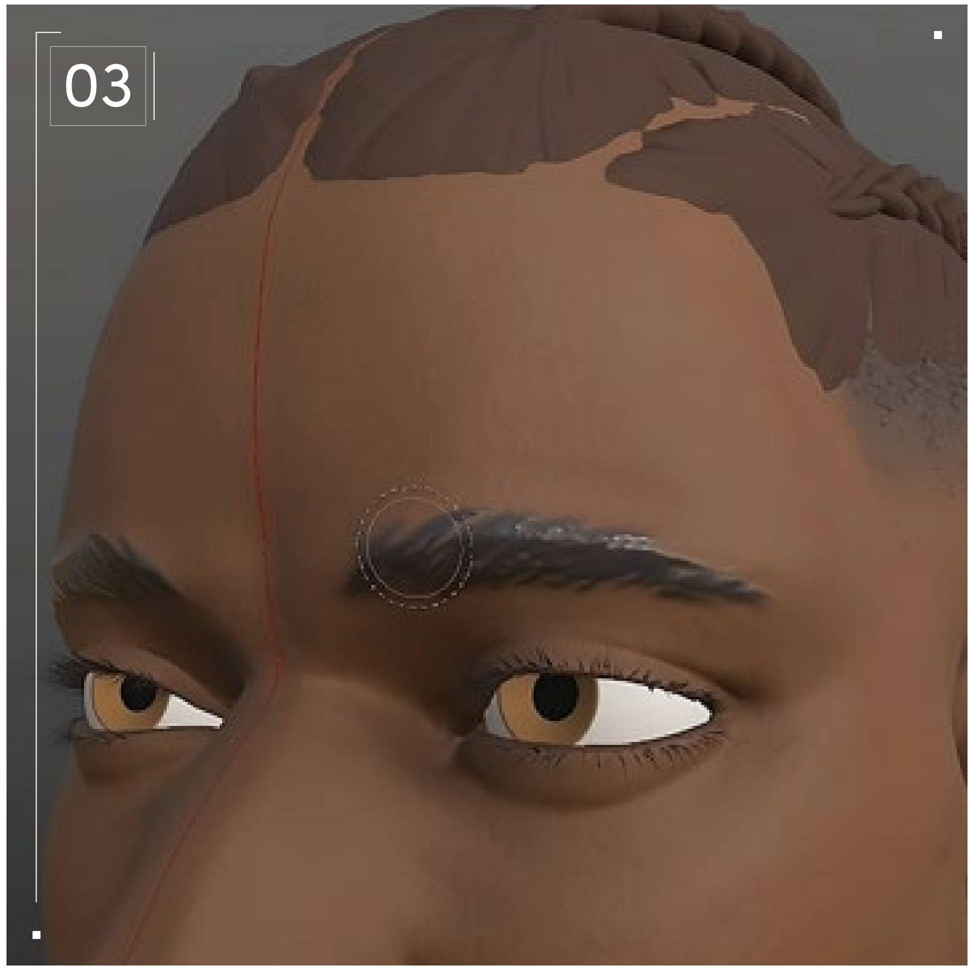
After it was satisfactory, I duplicated it and moved them to each braid, and once added, they further enhanced the effect of the style's silhouette. With the braid, supporting hair, and braids at a reasonable level, I went into the Zremesher feature in the Geometry palette. First duplicating the supporting mesh and the hair ties, I then remeshed the original models, bringing the polycount down and getting good topology, and then began subdividing them, projecting detail from the high- poly duplicates until I had captured everything necessary.

The braids already had good topology so all that was left to do was to do the UVs for the meshes and export for texturing. This is when UVMaster — a plug-in that comes standard with ZBrush — shines. Without leaving the software you can achieve a reasonable result with UVs and export them as FBX files into Blender, Substance Painter, or 3DCOAT.

Using the poly groups, I split up the meshes until the pieces and seams were as neat as I could get them while keeping the seams out of sight. After the UVs were good to go, I exported the models using FBX ExportImport, first into Blender to set them up for final rendering, and then to Substance Painter.

///

Step-by-Step Guide



// Texturing

In Substance Painter, I knew that I needed to incorporate the Fade element into the texturing process for the skin of the model. The challenge this presented was that making a Fade work without it being a separate mesh meant that I needed to create the Fade and make it blend seamlessly into the skin as would be the case in real life.

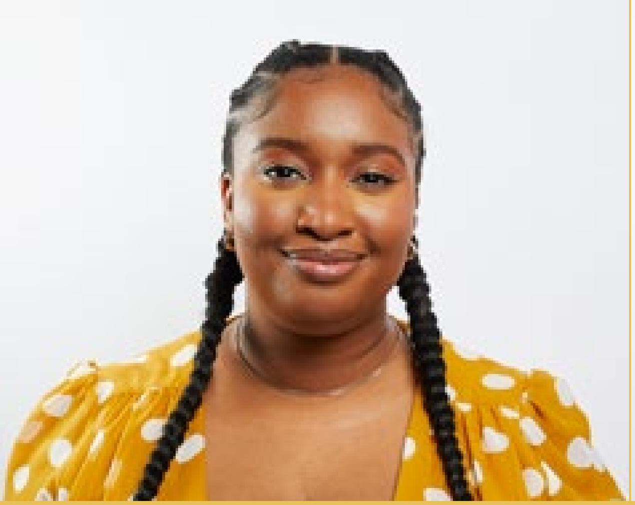
Thankfully, Substance Painter's folder and layer systems were very accommodating to this approach. After I'd managed to do the proper tones for the skin, I had to add in the Fade. Because the Fade had been colored in ZBrush, it had been baked into the exported FBX and could be read in the ID channel (by selecting "vertex" during the baking process); therefore, I could simply use it as a mask.

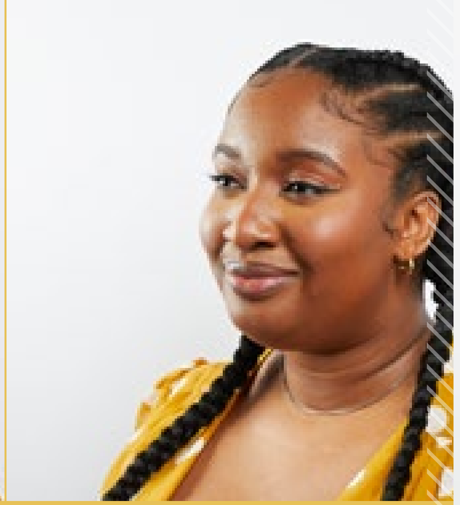
Then I could work within a folder on a masked layer and build it up using noise brushes in Substance Painter, then carve it into a layer of color that would serve as the basis for the Fade. When that layer was finished it was a flat color layer without any height information. But for it to read as slightly longer, I needed to have areas where there was detail that would give the impression of raised hair that can sell the illusion of strands of hair.

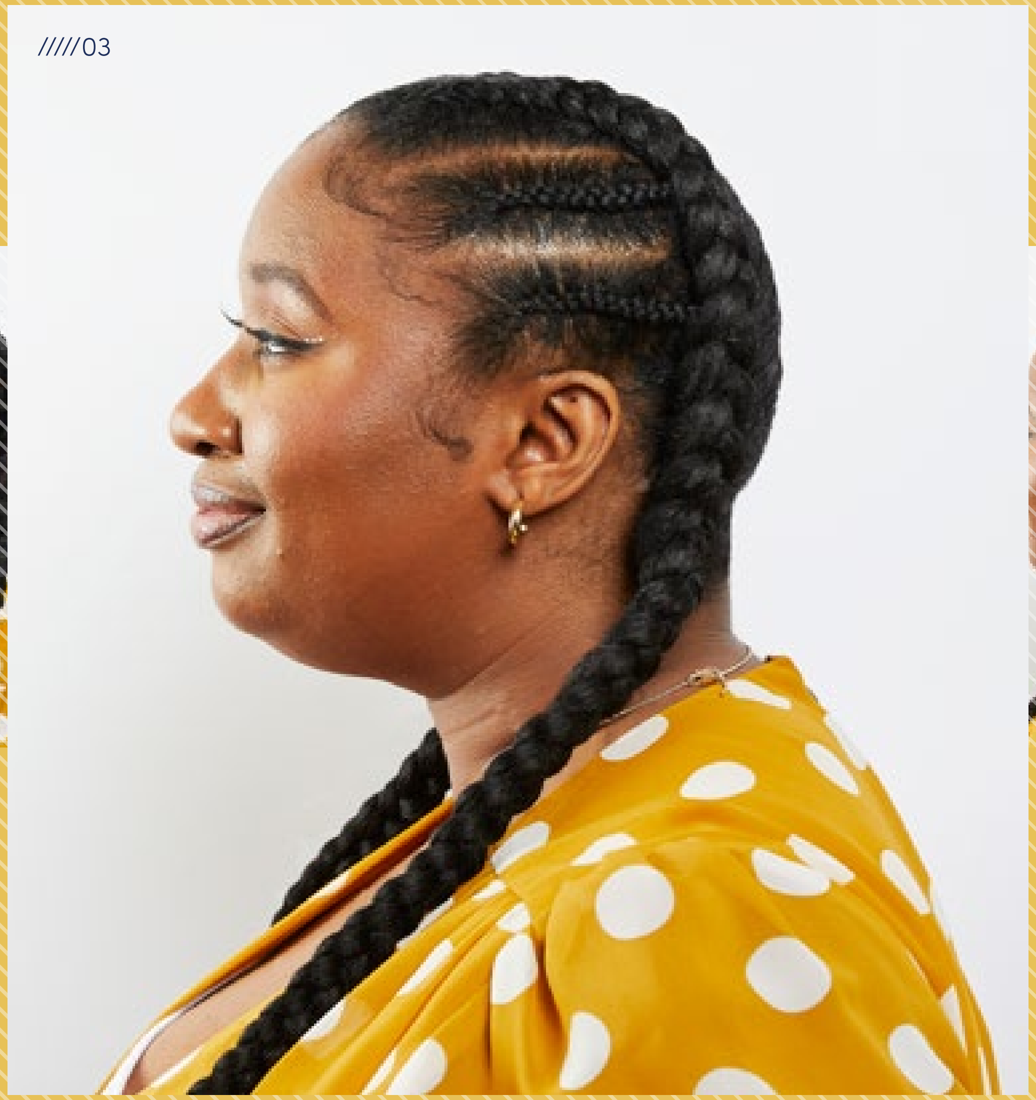
Making a new layer in that folder I put a black mask on it, and then used the Kyle's Concepts – Dragonscales brush to paint in detail. I used the basic white color on this layer so it would stand out during the process, and then adjusted the height of the layer to create raised detail. Then I matched the color to the Fade layer below it and pushed the color slightly lighter so it would contrast with it.

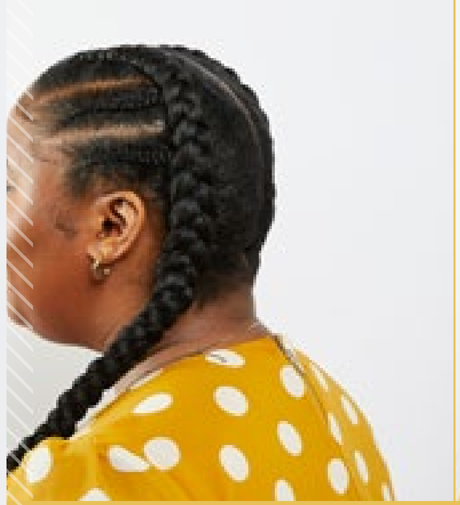
The Braids and supporting hair was a straightforward process, as I only needed to worry about colors and not trying to reproduce the texture of a real hairstyle. I used an anisotropic generator on the braids to add the illusion of hair strands to the braids themselves as it would bring just a bit of that visual weight to them. After I checked that all of the tones worked together in a pleasing and believable whole, I was able to export the textures into Blender and render the model, where my hard work sprang to life with the colors and textures added!

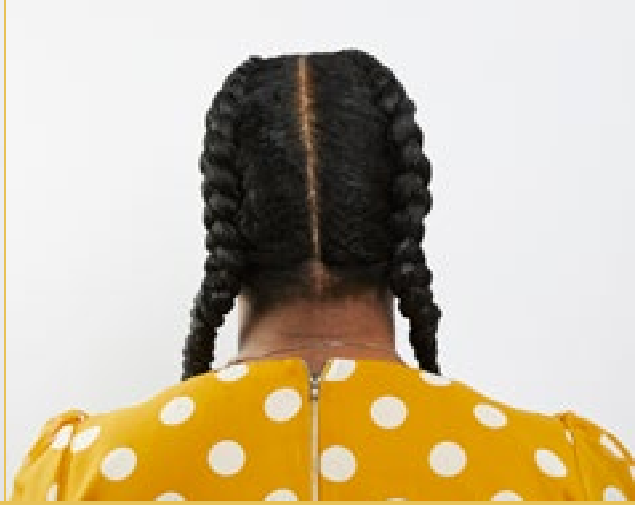
Cornrows with Accent Braids





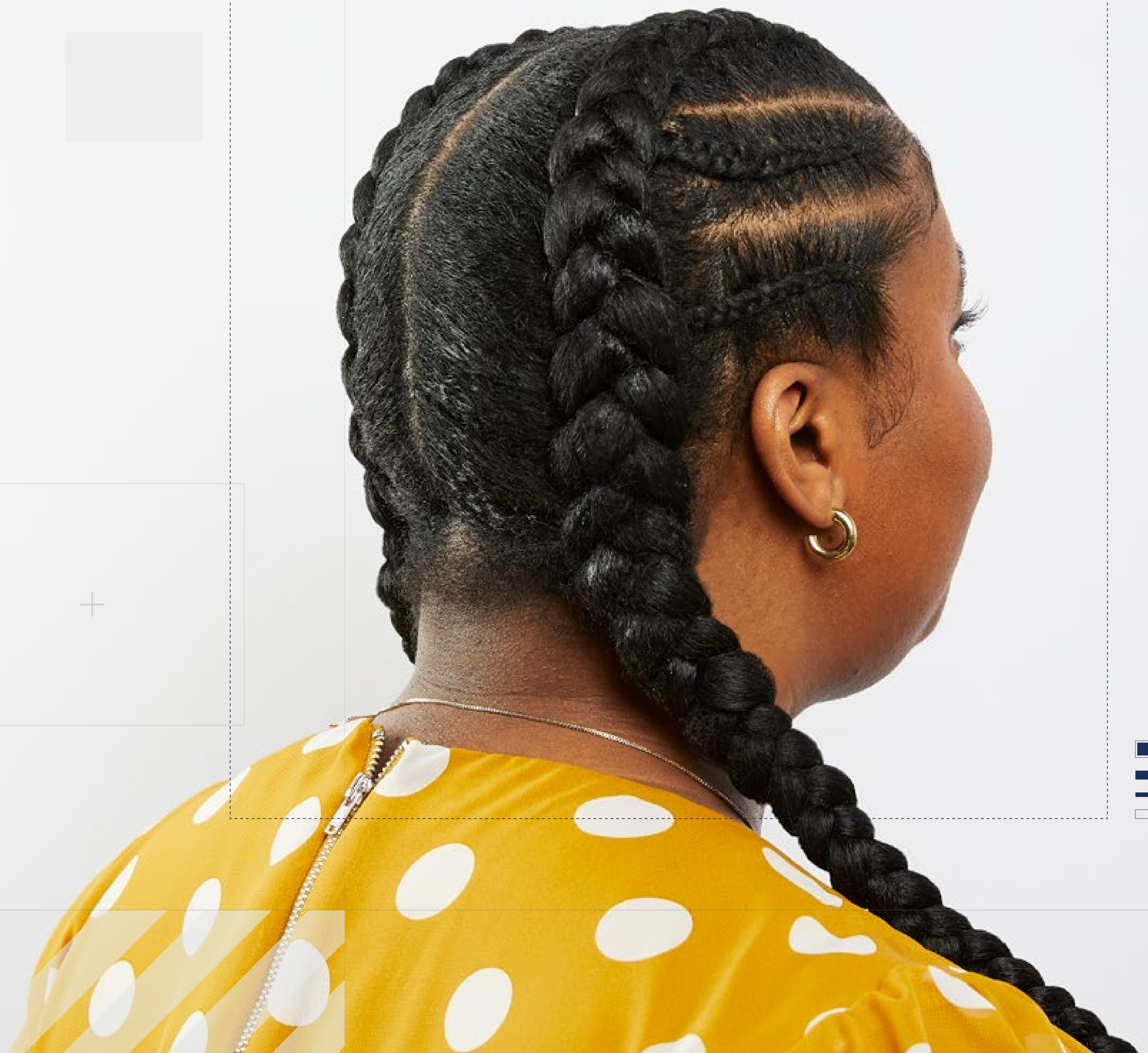






Download Images

// Cornrows w/ Accent Braids

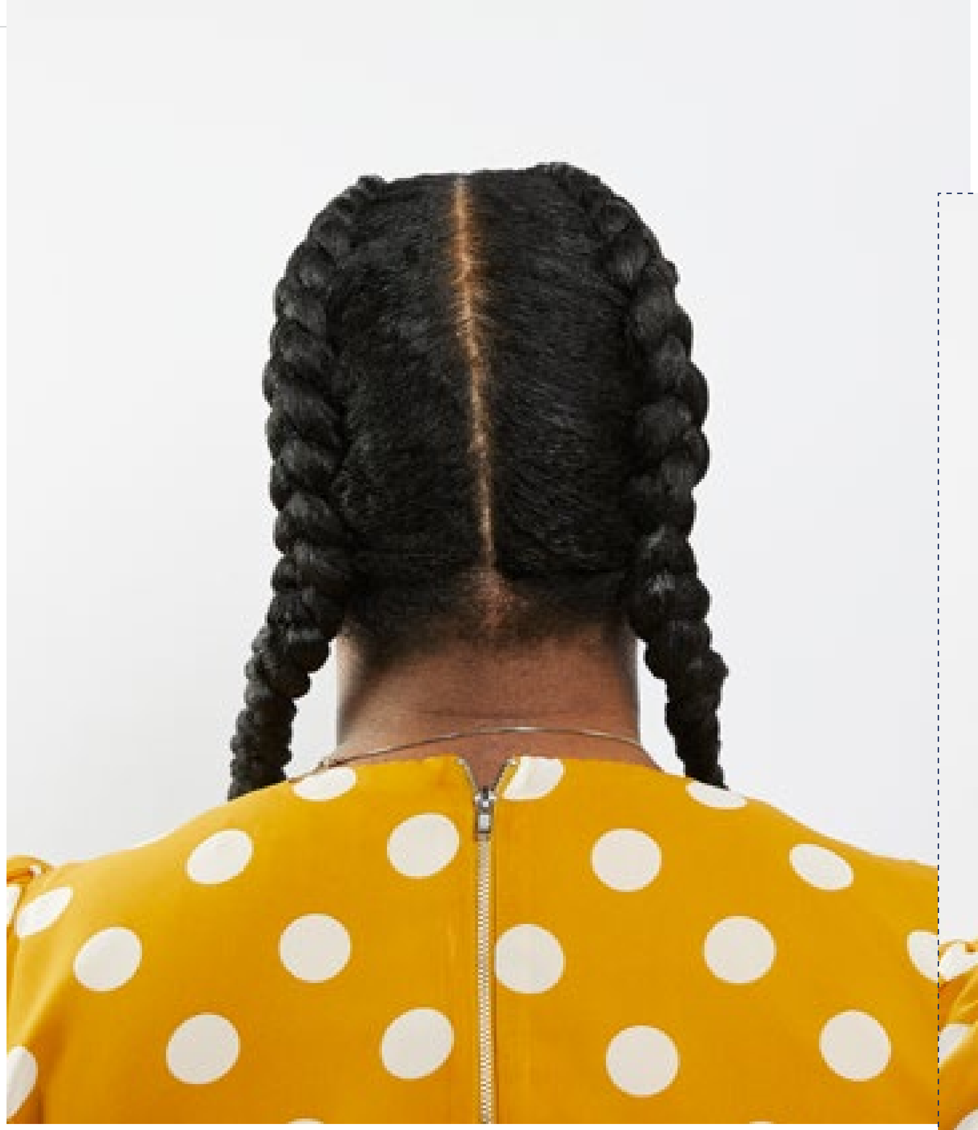


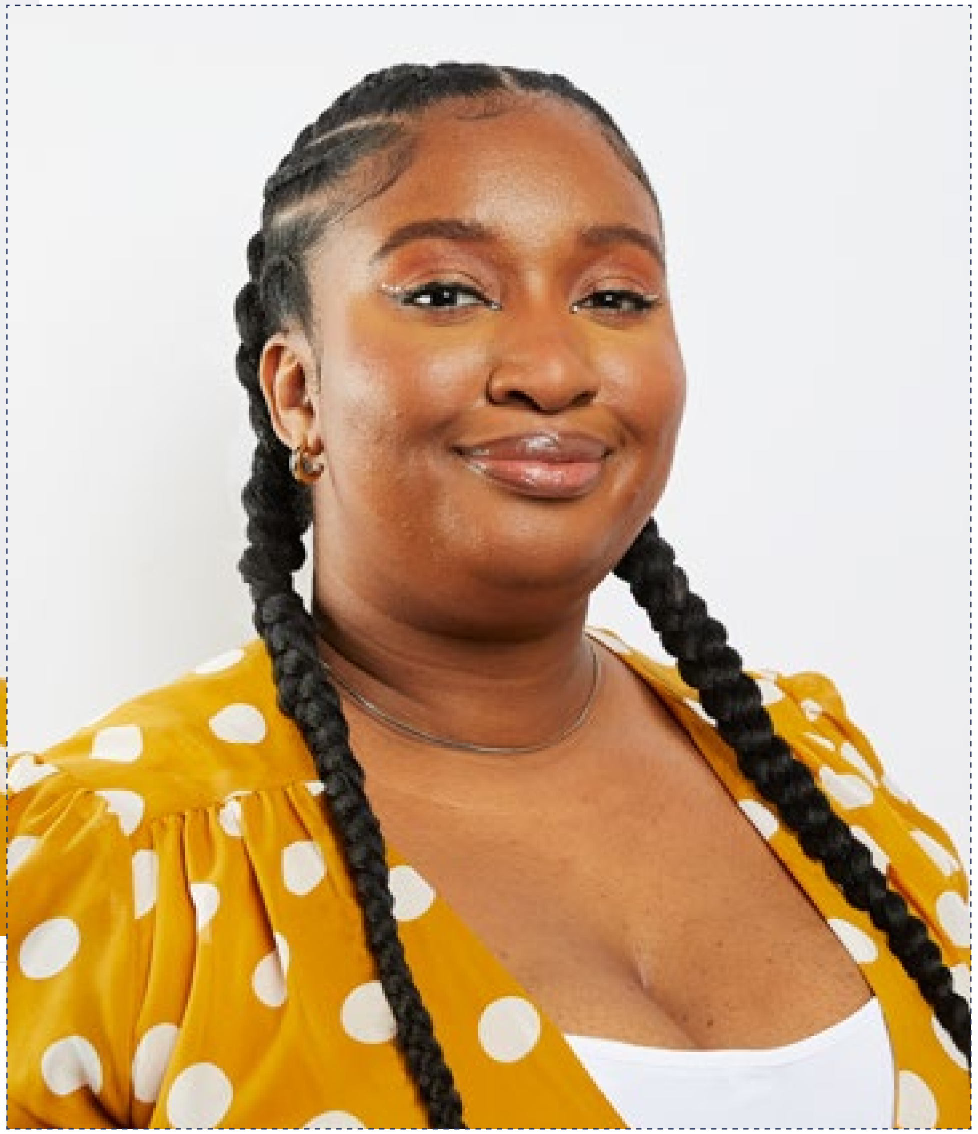
// Hairstyle

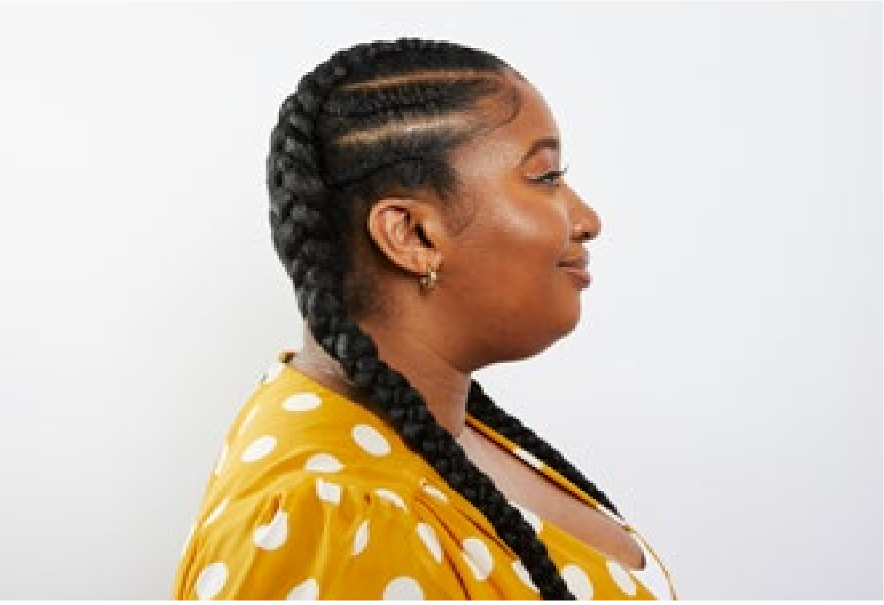
Cornrows with Accent Braids

Style Background: Jumbo Cornrows, or French Braids, are a common style among all hair types. For curly-coily hair, Cornrows also serve as a simple protective style. The hair is carefully parted into sections, then each section is parted into three and braided together with the hair held taut so that the braid is very close to the scalp. Typically, Cornrows run the length of the scalp, from the face to the nape of the neck, and each braid is secured at the end using elastic bands or clips. Creative parting and the addition of accent braids or jewels can give this traditional style a unique twist.

Download Images

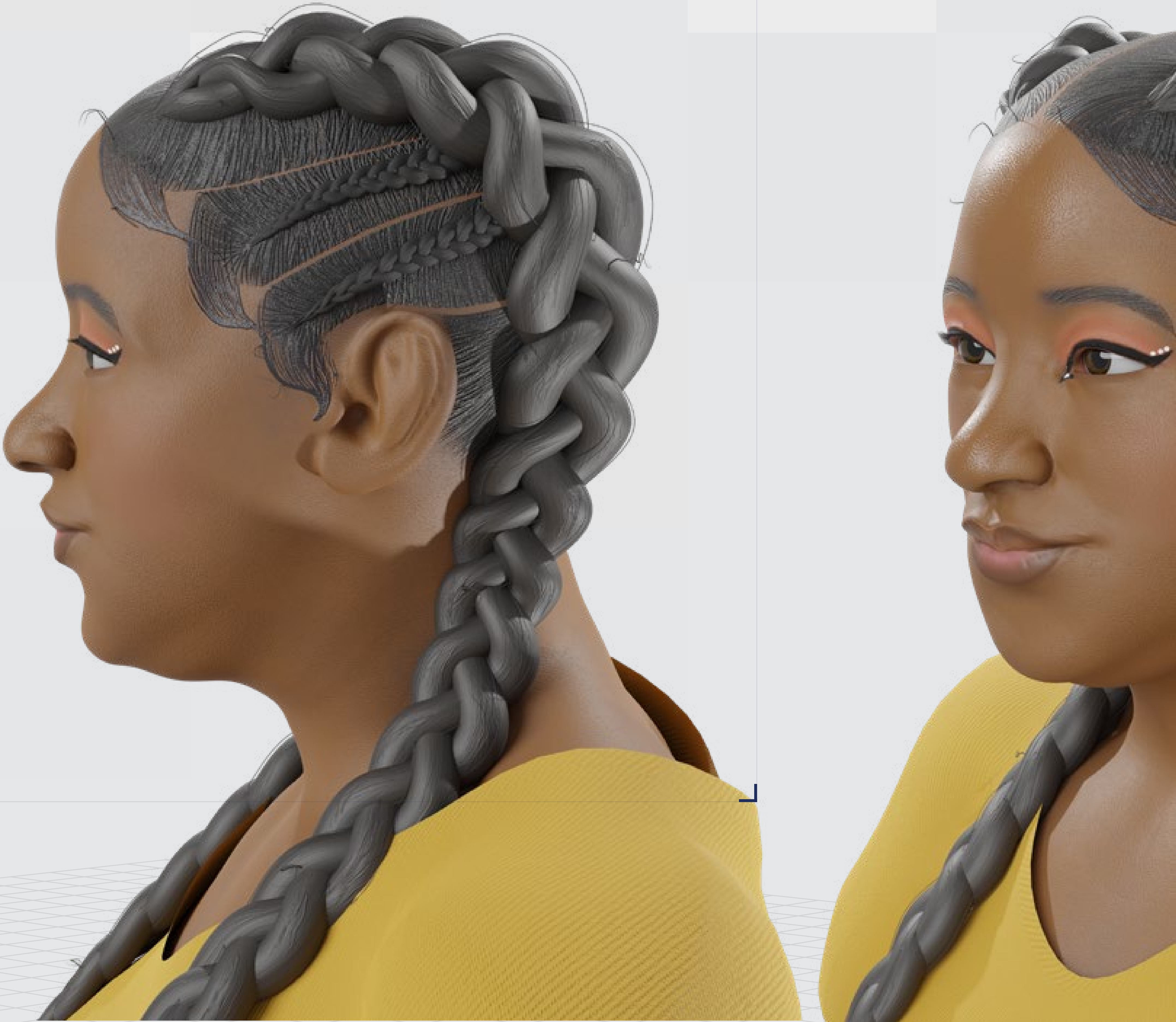






3D Design

Cornrows with Accent Braids

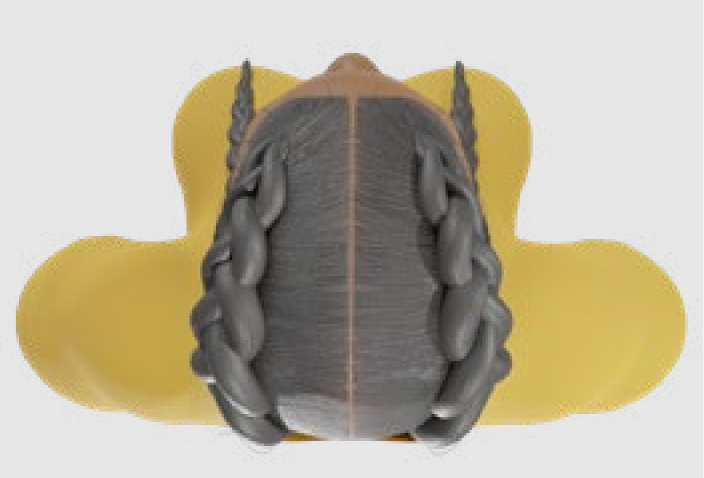


Digital Artist: **H.D.**

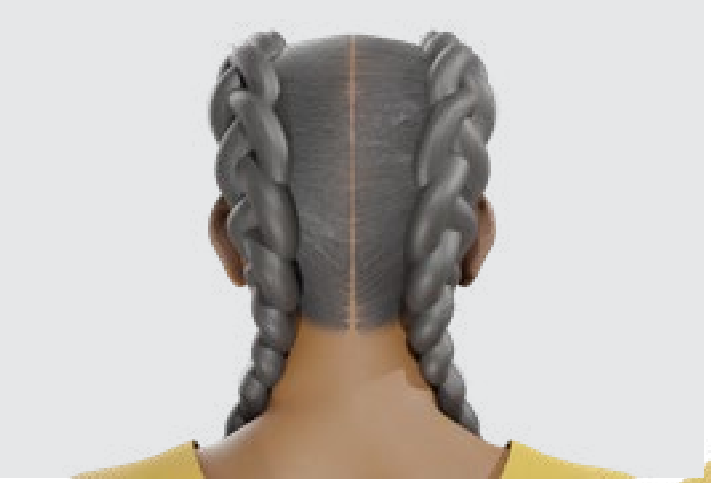
// Cornrows with Accent Braids

3D Design

Curl Type Modeled: **4A/4B**

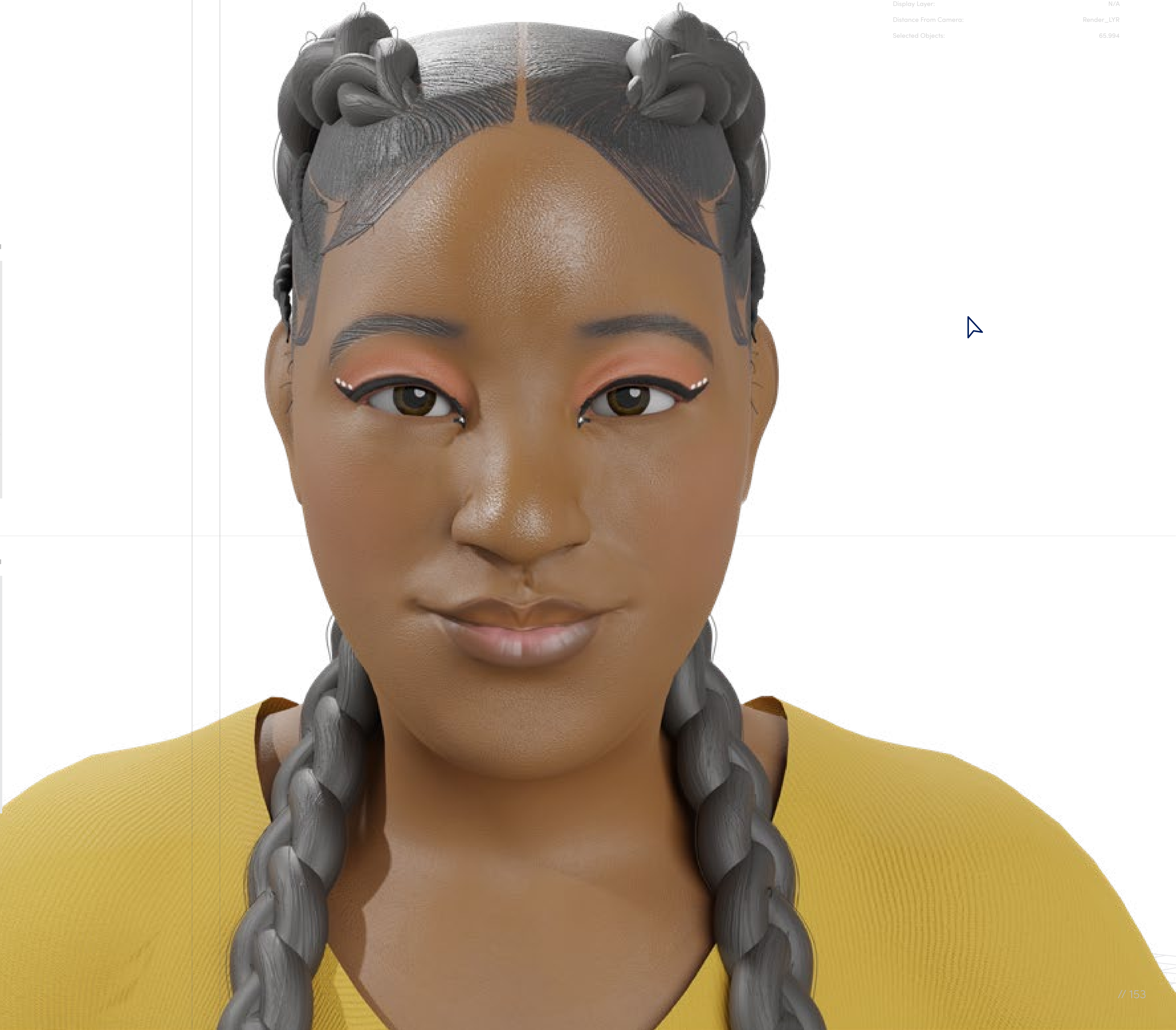


// 01



// 02

Download 3D Render



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

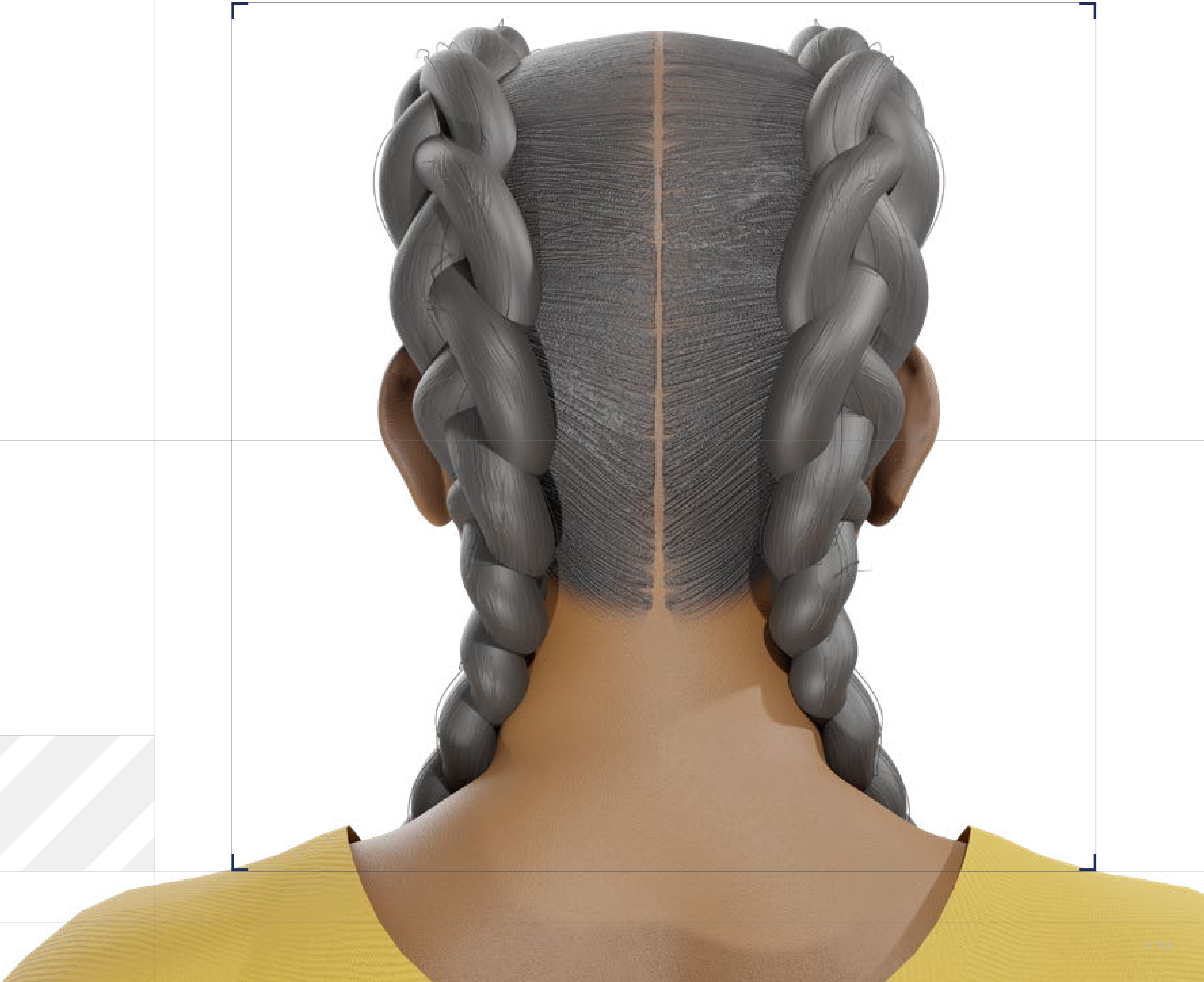
Project Settings

// Cornrows with Accent Braids

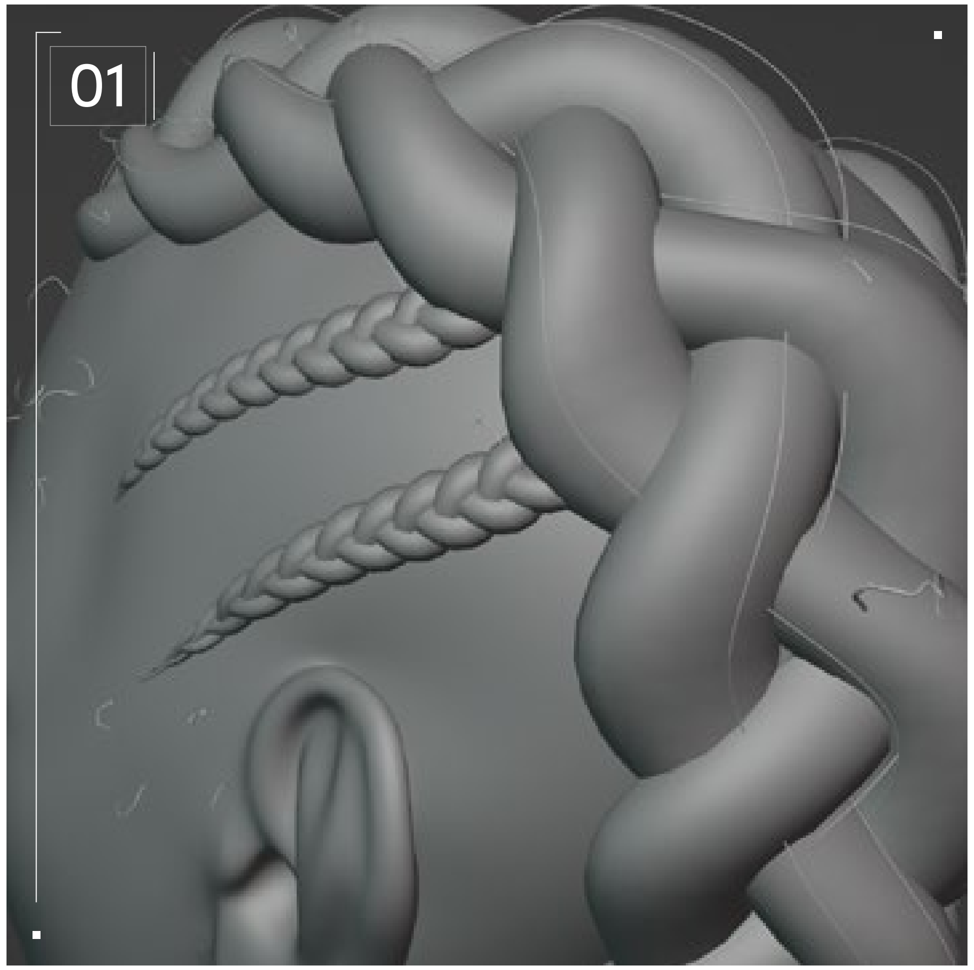
// Project Settings

Program Guide

Sculpting and Modeling: ZBrush
Texturing: Substance 3D Painter
Render/UVs: Blender



Step-by-Step Guide



// Blocking

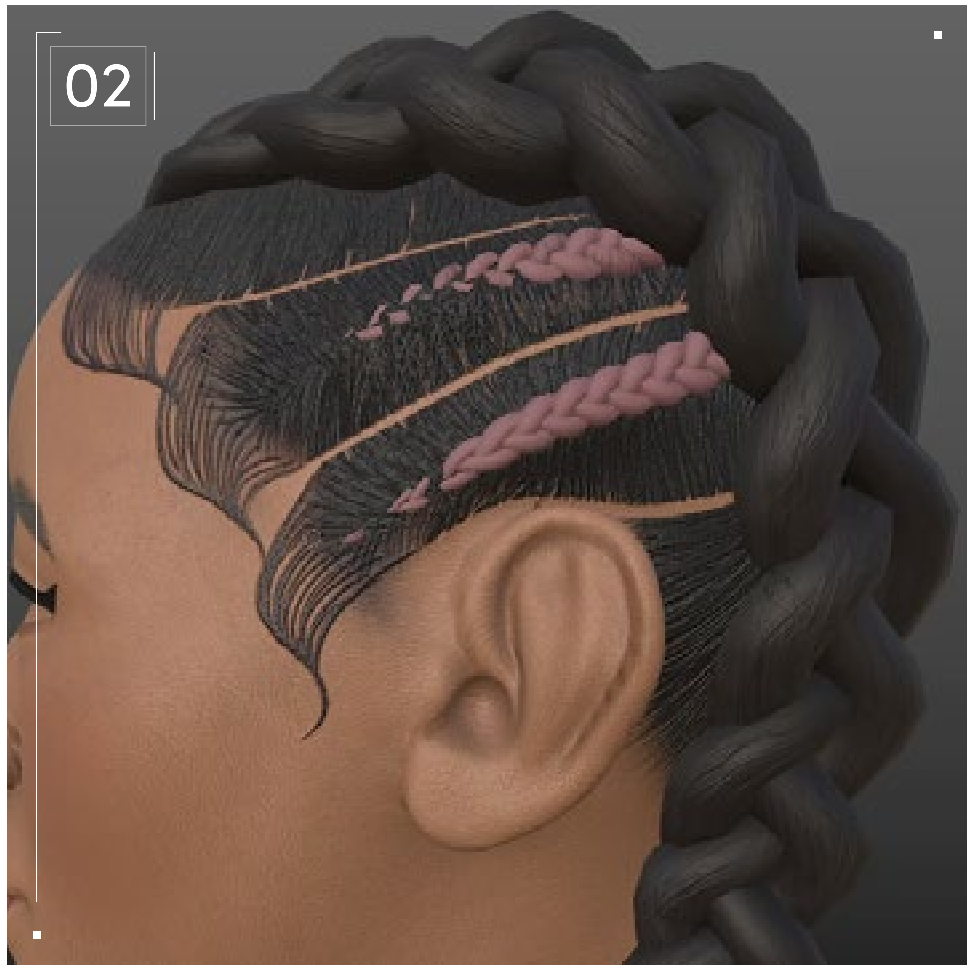
It's important to have the major forms ready first to get the most natural curve to the braid. So, once you have your base mesh [as we use elsewhere] ready, you can use a braid curve brush — if you are using ZBrush — that you made or one that you bought if you want to have one that's premade.

If you would rather sculpt it manually, you can use a basic curve brush to draw out the path of the braid, and then model it. The benefit of doing it that way is saving on polygon count and a simplified UV situation in the end. The sacrifice is that you lose out on the fidelity of having the meshes actually form a braid. The decision of which to use will depend on the actual needs of your project, in any case. With regards to the accent braids, make sure that the far ends tuck under the main braids; it wouldn't look right to have visible ends.

If you're creating a supporting mesh, think about not just the actual form, but also how it should feed into the braids. Sculpt with the intention of creating the illusion that the supporting mesh is actually joined to the braids.

Once you get the primary forms in the supporting mesh, the braids are arranged as you like, and are properly tapered at both the beginning and end to match how such a braid would exist in real life — you can move on to the next stage.

Step-by-Step Guide



// Modeling

Now that you have the basic forms together, you need to decide how much detail will be sculpted in the braids and any support meshes you've created, and how much will be textured, creating the illusion of physical detail.

In ZBrush, you can use the "auto groups" option under the Polygroups palette to give each strand of the braid its own group. That means you can hide and sculpt on each one individually if you decide to sculpt in the detail, rather than rely on textures.

Obviously, if you're sculpting the braid from a curve, you will have to build the forms up manually. But you can create poly groups in the pattern of the target braid by using masking to match what would be a strand's path through the braid and choosing "group mask clear mask" under the Polygroups palette. This will enable you to sculpt the braid's form without too much headache.

If you choose to sculpt in details on the braids, make sure that the strands you create follow the path of the braid so that it doesn't jar the viewer out of the experience. You want to avoid causing the textured surface of the braid to look like patterns in cut wood.

If you have decided upon a support mesh, now is the time to think about parting patterns as well as secondary details that aid the idea of hair being pulled into the Braid. Because this will be baked down into a low-res mesh, the detail level here isn't very restricted, which is a nice side effect of using this method.

When you've reached the desired level of detail, you can duplicate the supporting mesh and run your original mesh through Zremesher under the Geometry palette, lowering the polycount of the supporting mesh and getting better topology flow.

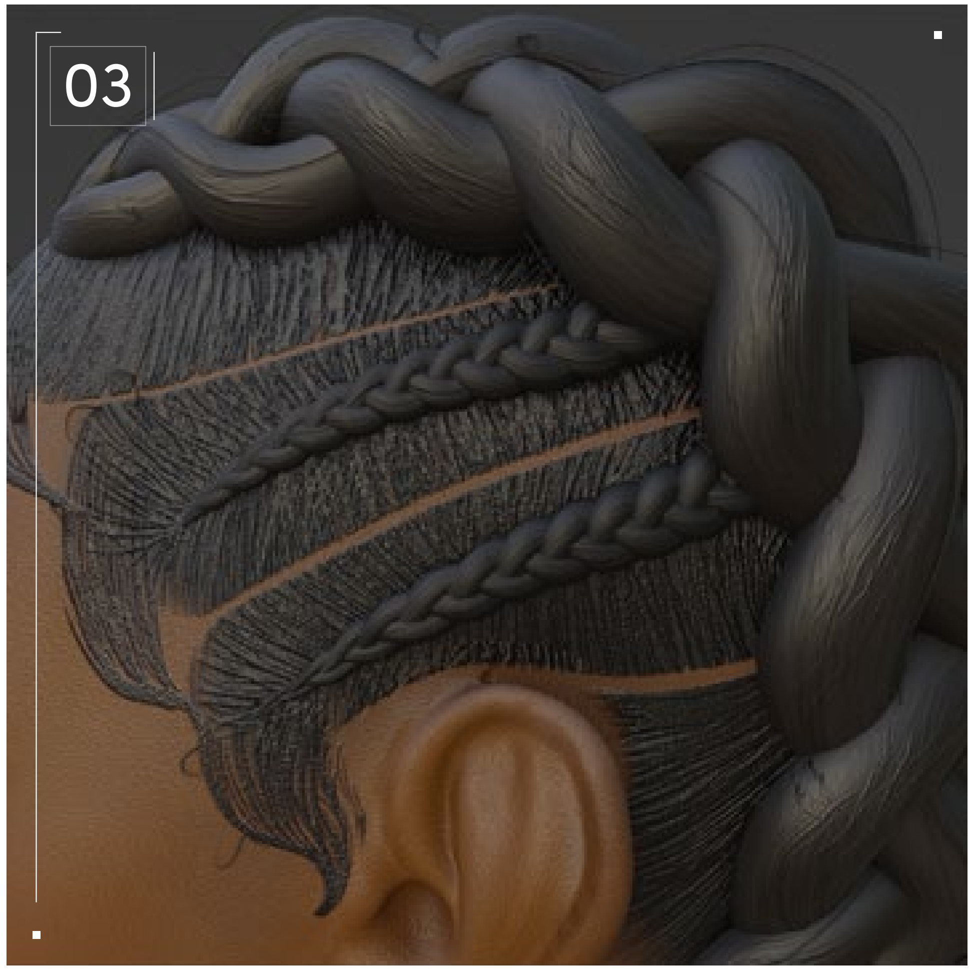
Once you have a good level of detail that you feel captures the form well enough, you can switch to the duplicated mesh. Using the "Project" menu under the Subtools palette you can project details onto other visible meshes with "Projectall" after subdividing the original retopologized mesh. Because it works on all visible meshes, the smartest way to use this feature is to hide the other meshes so that only your original mesh and the duplicate are visible in the Subtools palette.

For best results, repeating the step of reprojection after each new subdivision level will capture all details and adjust the form of the base mesh to match the final form better. At the end of that step you would have the necessary details preserved on a mesh that can be pushed up and down in subdivision levels. If you used a braid curve brush, this step will be unnecessary for the braids themselves, as any details sculpted into them would have been preserved during the subdivision process.

After all necessary meshes have been treated in this way, you can now work on UVs. You can either do this in ZBrush itself via its automated UV Master plug-in, which comes shipped with ZBrush, or by exporting the low-poly versions of the meshes as an .fbx file using the FBX ExportImport plug-in (which also comes standard with ZBrush), then importing into Blender or some other similar DCC software for more precise control. When the UVs have been split appropriately, the high-poly versions of all other meshes can be exported using the same FBX plug-in.

///

Step-by-Step Guide



// Texturing

The texturing process will be heavily influenced by how many, if any, details were created during the modeling phase, and whether support meshes for the braids were created. However, the thought processes aren't that different either way — the aim is to hit the levels of realism or stylization necessary while also matching the nature of the hair. If you didn't create a supporting mesh, now is the point at which you will create the scalp hairs, feeding into the braid.

To get started, I found it helpful to first make a sort of scalp-map that showed where the hairline would exist normally. In Substance Painter it's easy to put a fill layer on the base mesh head and use a black mask to hide it.

After that you can paint the area in, unmasking the layer until it fills only the area where you want the scalp to be. You can copy the mask from that layer into a layer folder, and then every layer within that folder can only display within that masked area. If you take the time to draw the hair part area in the folder mask, then it will save you time going forward.

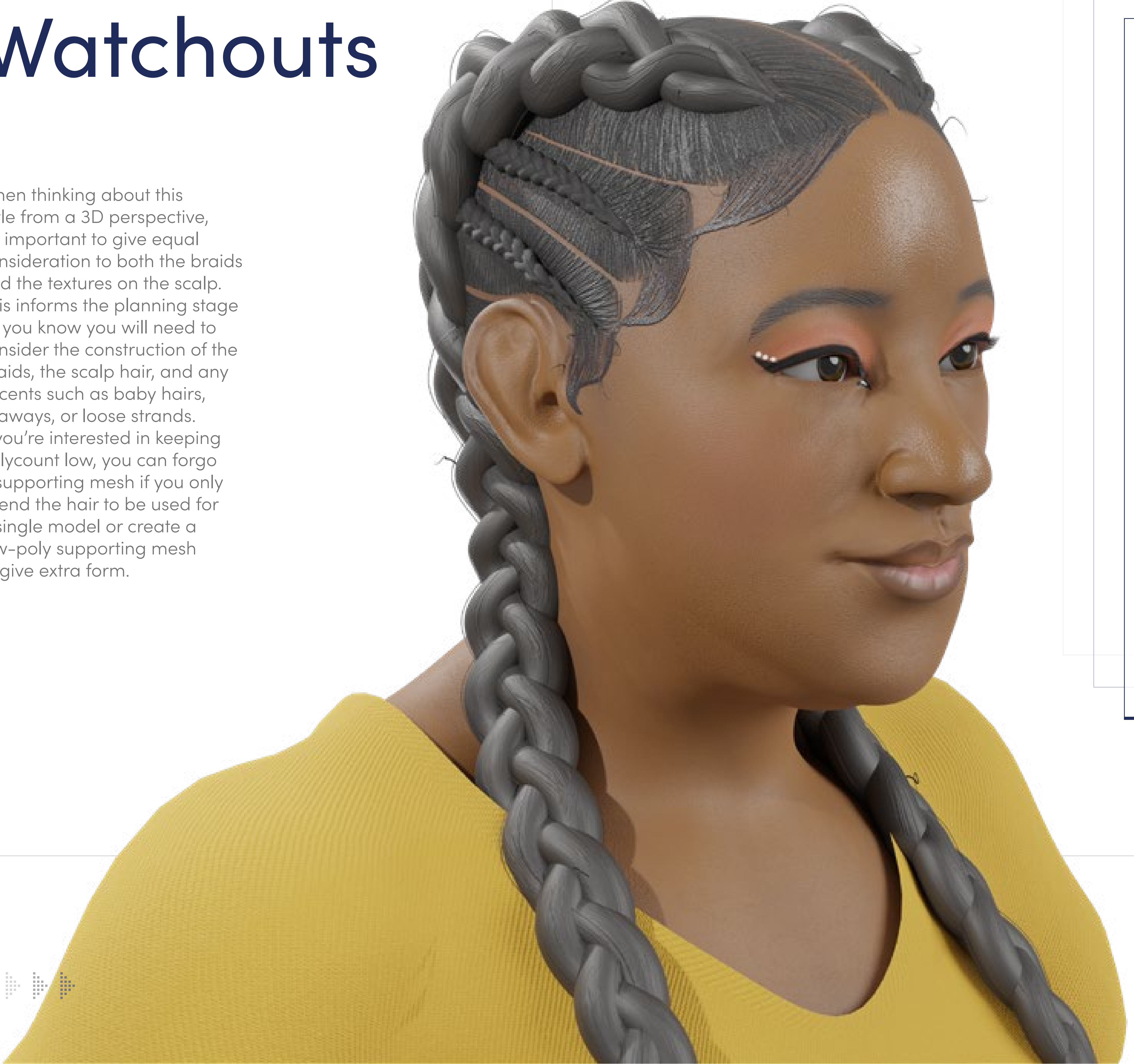
Then, with the final scalp appearance in mind, you will want to make another fill layer. In the layer properties panel, increase the height slider slightly — just enough to cause visible detail to pop from the smooth surface. On this layer, use a black mask, and then draw out individual hair strands to guide you, going from the hair part or the edge of the hairline backwards towards the braids.

Once all of those guidelines are in place, you can do secondary and tertiary strands to fill out the areas between them, giving them fuller forms. However, you're not intending to completely cover the area with these strands, merely to draw details in.

+ Beneath this layer create another layer with a color close to the one that will be the final hairstyle color, but darker. This will give some contrast between the two layers and add to the sense of depth.

Technical Watchouts

When thinking about this style from a 3D perspective, it's important to give equal consideration to both the braids and the textures on the scalp. This informs the planning stage as you know you will need to consider the construction of the braids, the scalp hair, and any accents such as baby hairs, flyaways, or loose strands. If you're interested in keeping polycount low, you can forgo a supporting mesh if you only intend the hair to be used for a single model or create a low-poly supporting mesh to give extra form.

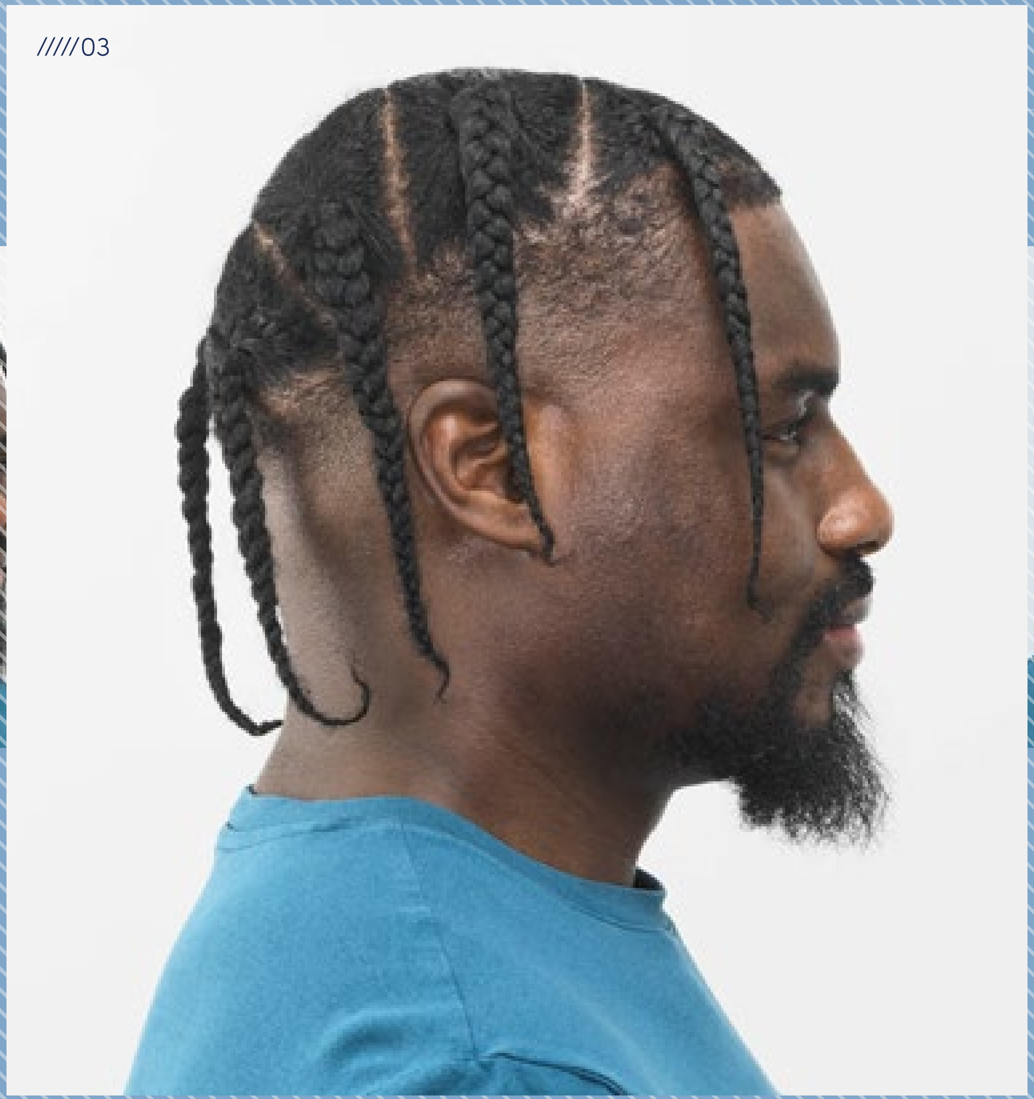


Top Tips

For the braids, you can add a new fill layer, use a black mask on it, and then add anisotropic noise with a generator to the height parameter in the layer panel. This will cause the braid to take on a strand-like texture when the mask is erased. Using the anisotropic generator on the layer rather than the mask allows you to manipulate the tiling as well as rotation of the noise. An alternative to this method, which will create a very uniform appearance, is to carefully create the detail on the braid using the brushes in Substance — I found "Kyle's rakes - sparse rough" was a good one for this method!

The goal with this is not to recreate what a photo-real braid would look like, but instead to suggest the texture in a way that blends with the aesthetics of the hair without drawing too much attention. By using these methods and carefully adding detail to the braids and scalp with an eye not for photorealism but just for capturing the feel of the references at hand, by the end of the process you will be able to achieve a beautiful result!

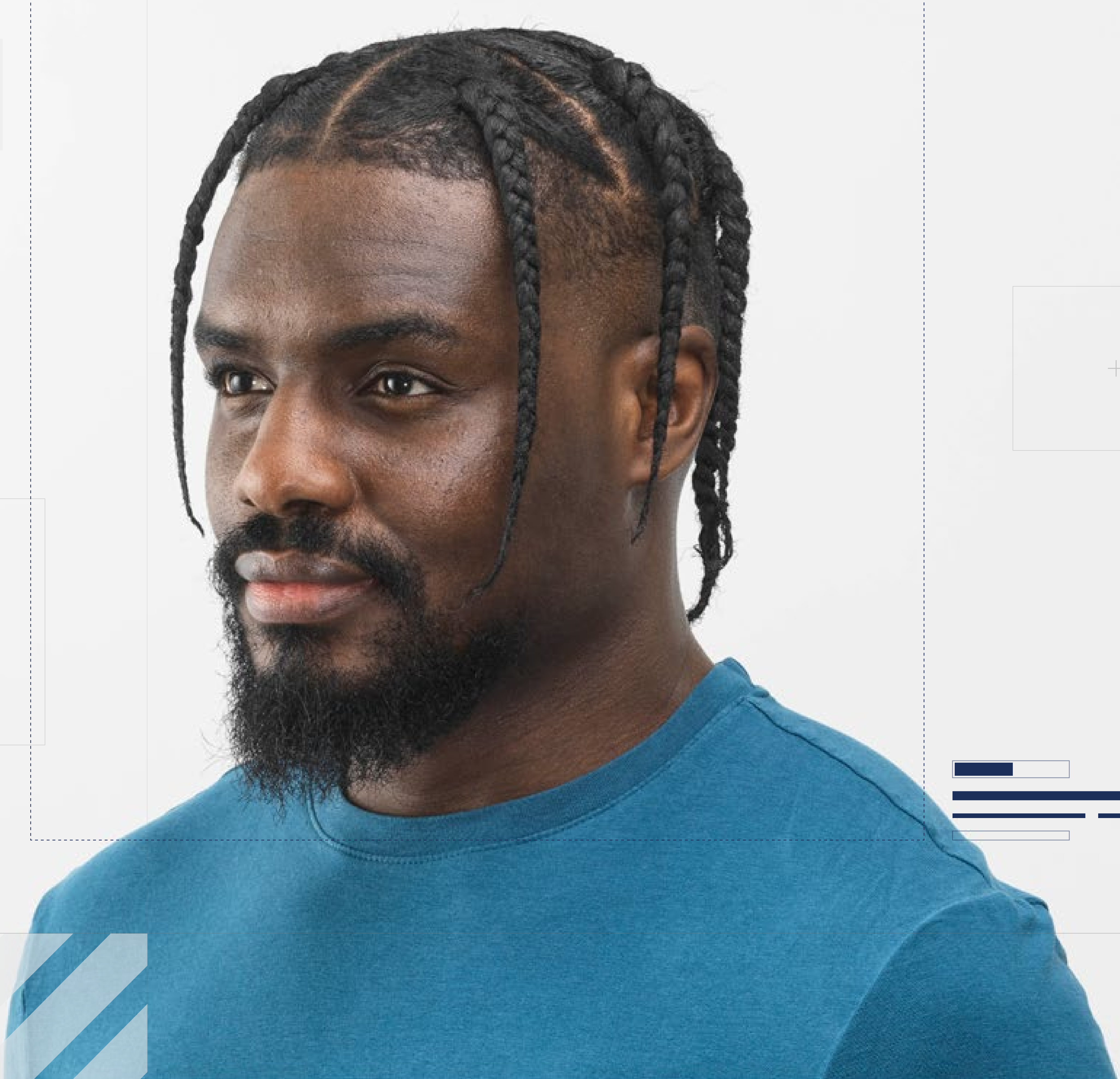
Braids with Fade



/////03

Download Images

// Braids w/ Fade

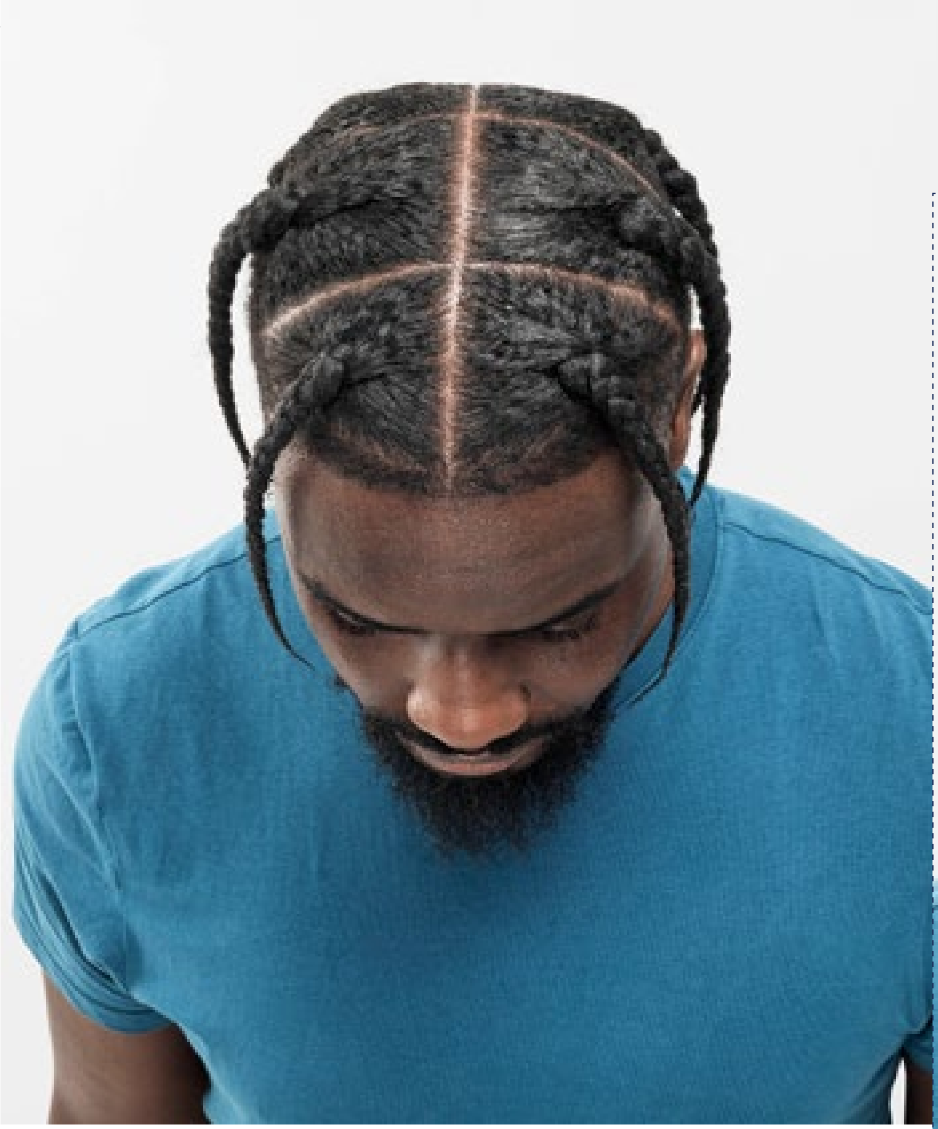


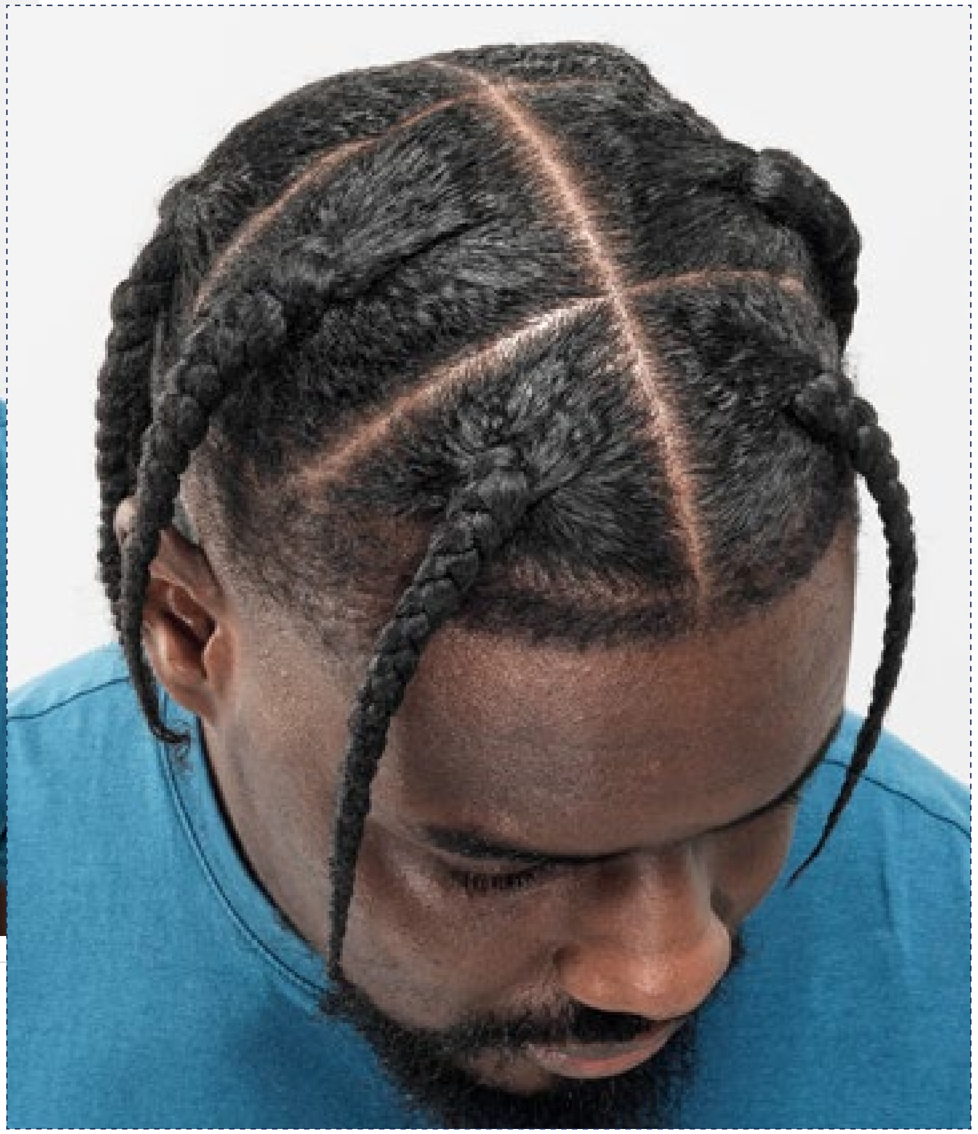
// Hairstyle

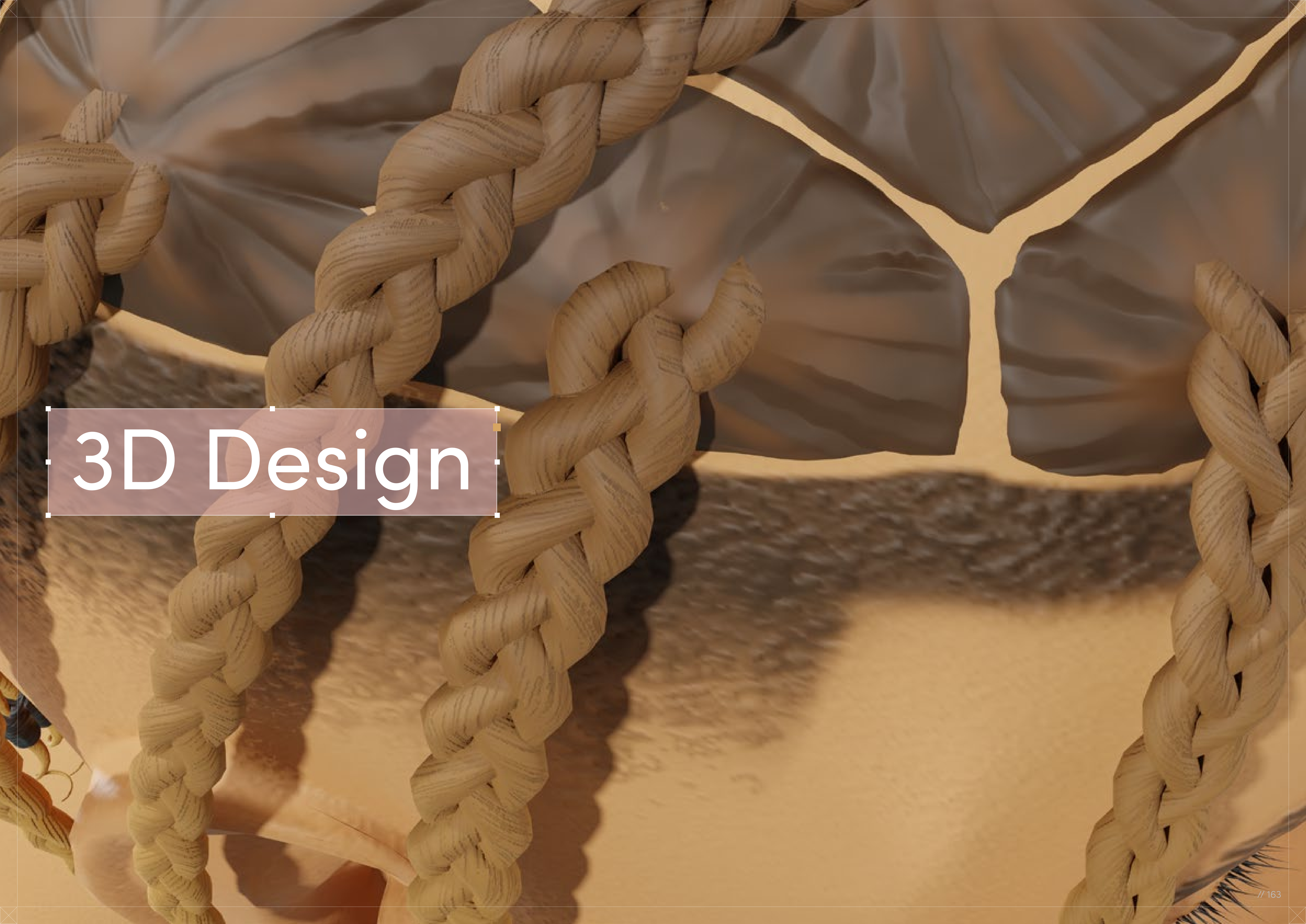
Braids with Fade

Style Background: This hairstyle is really a combination of two iconic styles: the Fade and braids. Hair is parted and styled into either traditional Box Braids that are tight to the scalp while the hairline is shaved in such a way that creates a gradient by buzzing the hair strands to different lengths. Fades have always been a part of the culture of Black hairstyles, and in recent years a new generation of stars have restyled this classic look, pairing the Braids with Fade to create dynamic variations.

Download Images

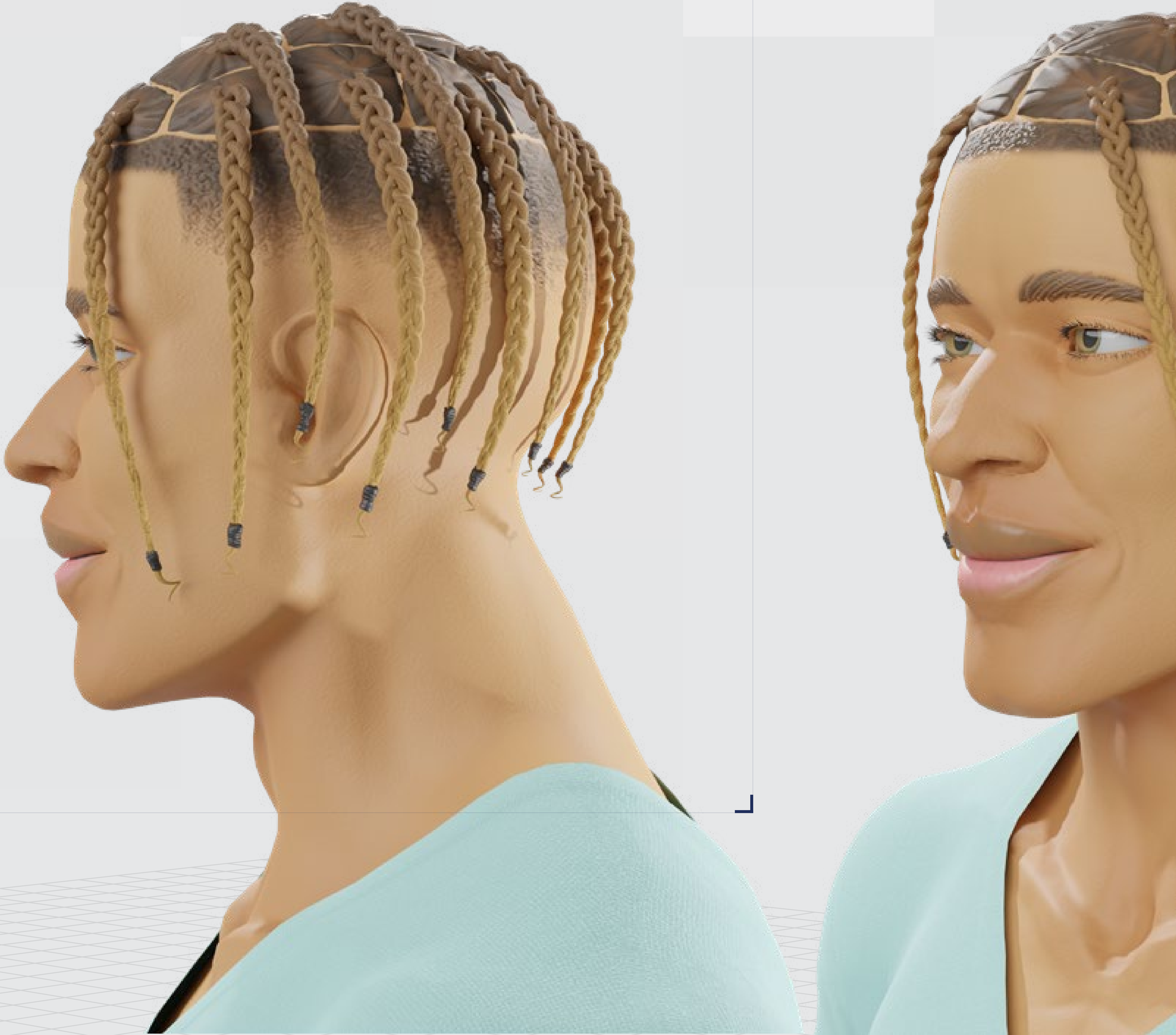






3D Design

Braids with Fade

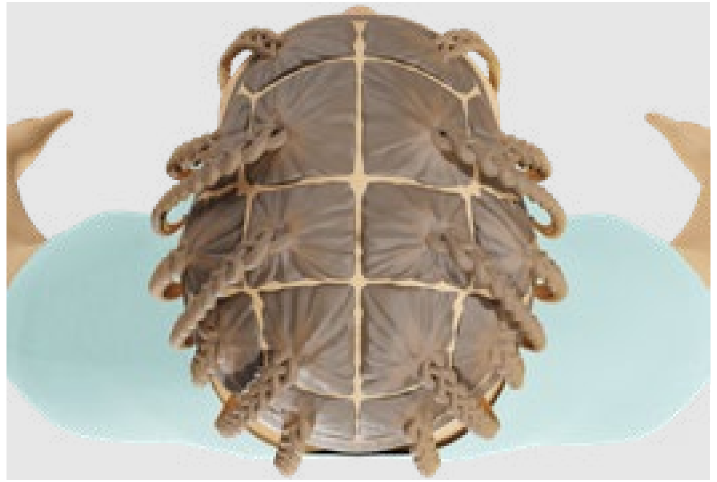


Digital Artist: **H.D.**

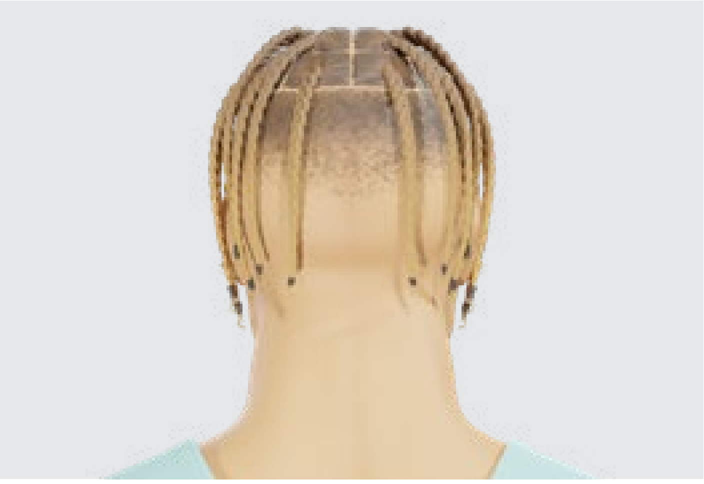
// Braids with Fade

3D Design

Curl Type Modeled: **4A/4B**

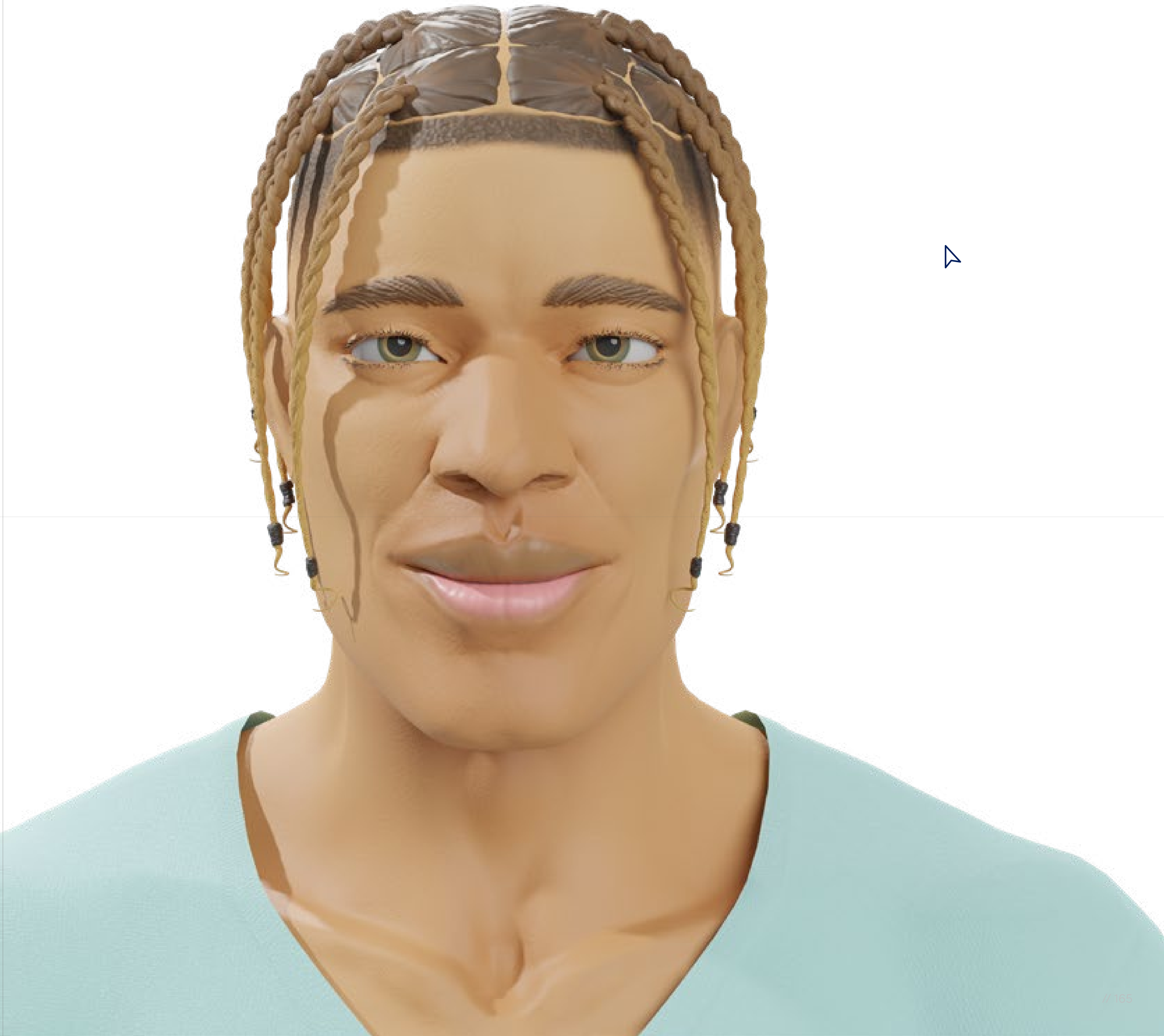


// 01



// 02

Download 3D Render



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

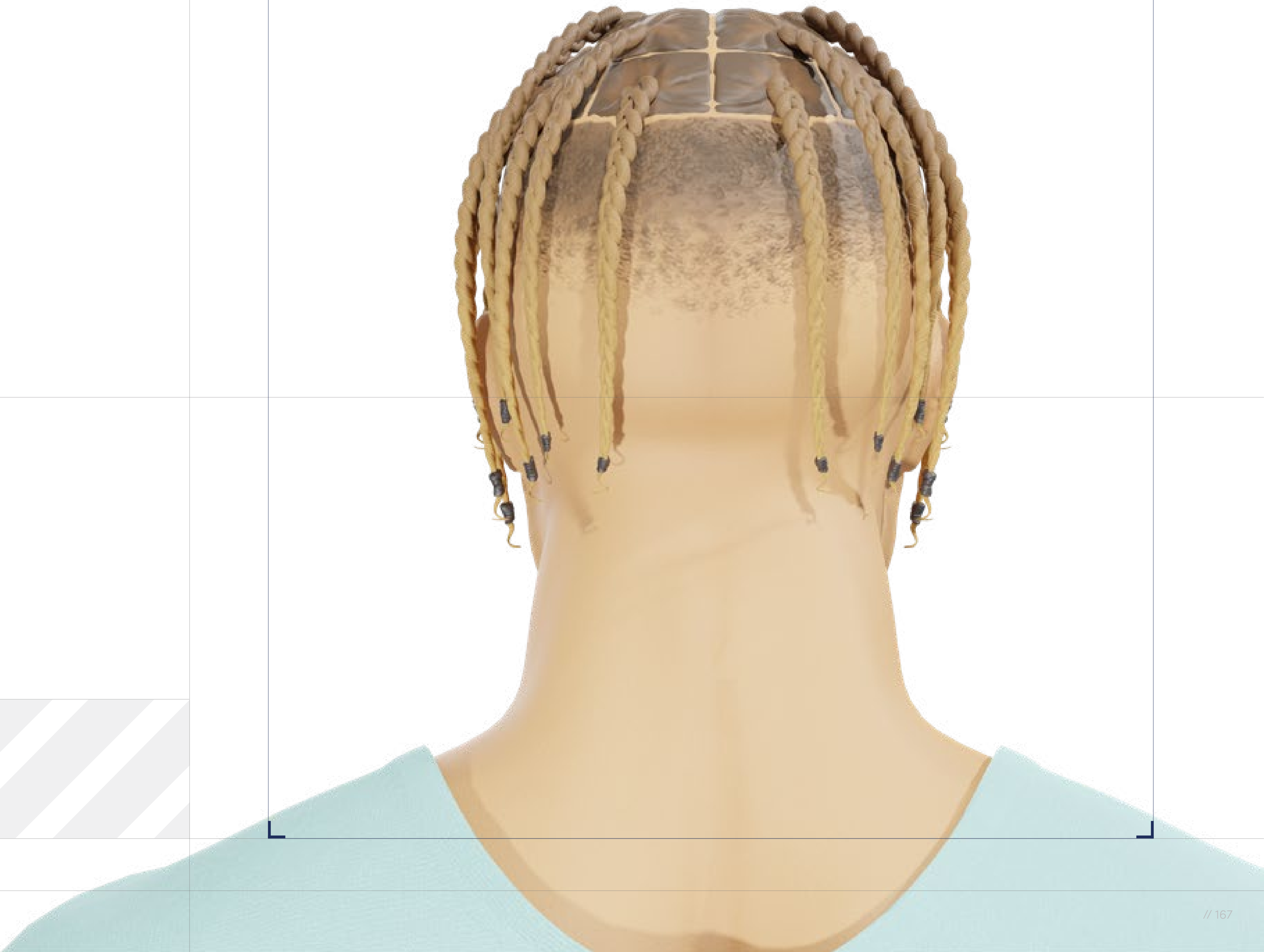
Project Settings

// Braids with Fade

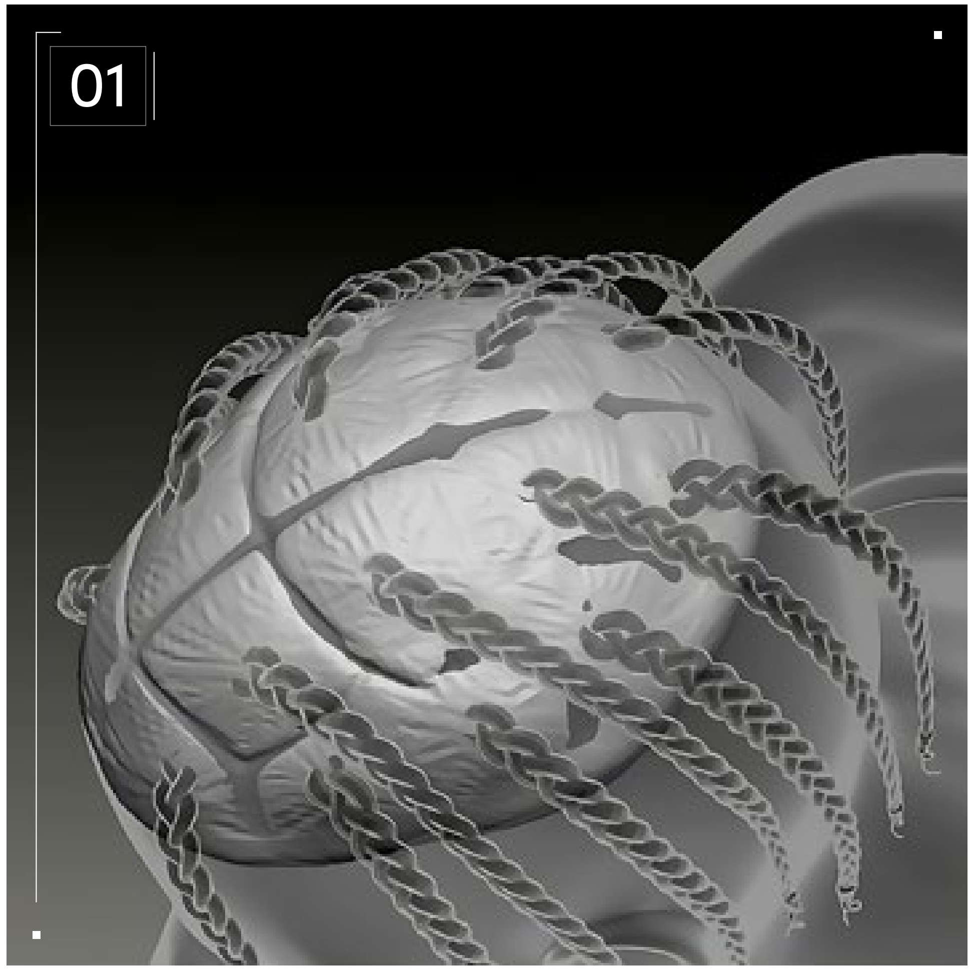
// Project Settings

Program Guide

Sculpting and Modeling: ZBrush
Texturing: Substance 3D Painter
Render/UVs: Blender



Step-by-Step Guide



// Blocking

This style makes me think of a plant, which leads me into viewing the hair as akin to a garden that we carefully maintain and attempt to marshal. And like a garden, our hair has its own will sometimes, and defies our efforts to control it. Braiding seems like the first line in attempting control so this style gives the impression of a compromise, where the braids are allowed to sway free.

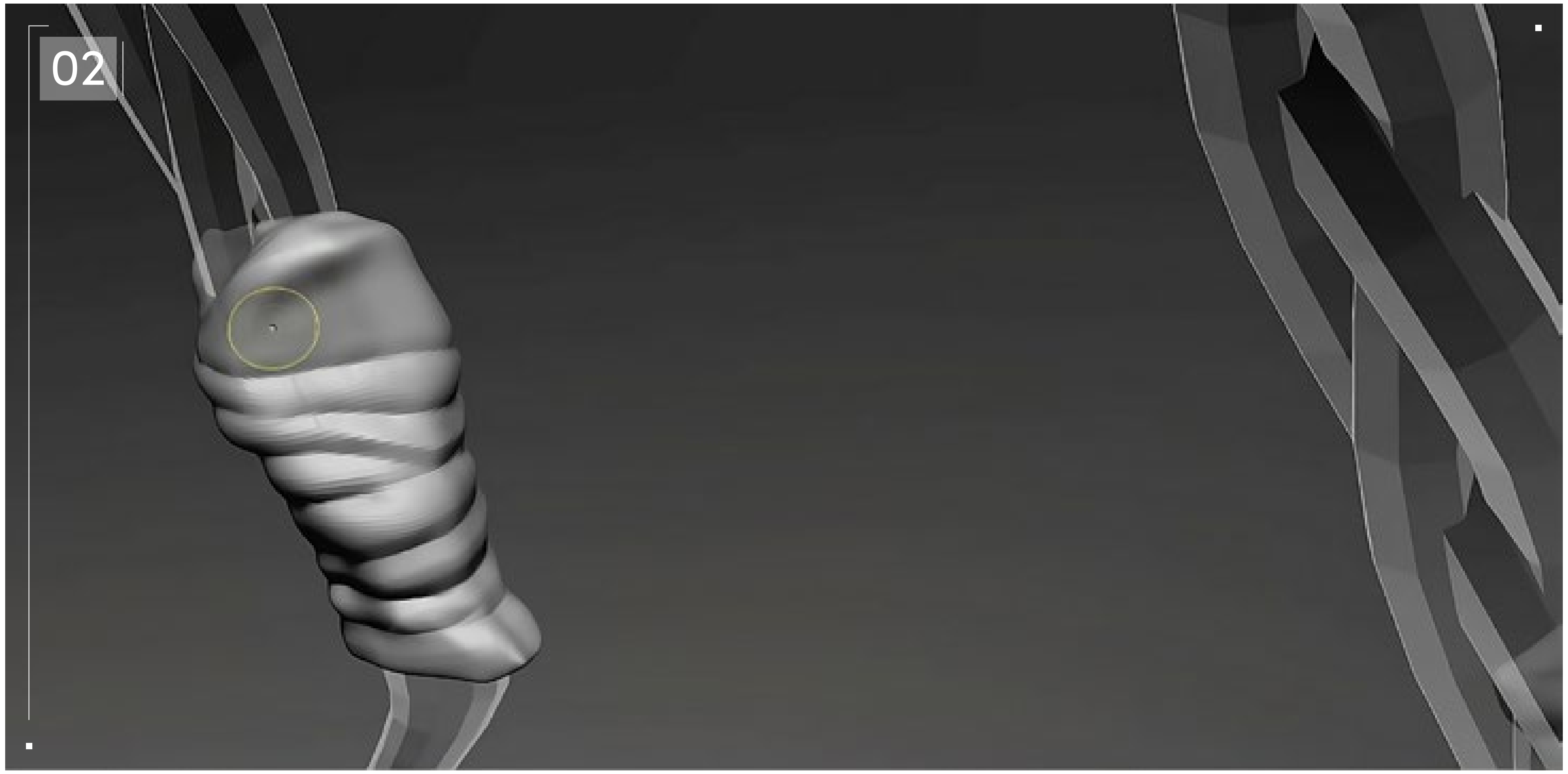
Thinking of the structure for recreating it, I knew I would need to think in the three parts: the Fade, the hair at the base of each braid, and the braids themselves. It would be important to make sure that the curve of the braid falls naturally and yet pleasingly, aesthetically speaking. From previous experience I knew that it would be better to make this Fade purely as texture, the hair at the base of the braids as separate meshes, and the braids themselves via curve brushes.

I began by painting in the area for the scalp on the base mesh model to help guide me as I built the rest of the hair. Looking at the official references and examples from around the world, I could see that there is a specific type of arrangement to the braids, especially when considering the framing of the face, so when I made the supporting hair, I masked out a sort of "cap" and then extracted it into a mesh.

Next, I drew in the braid patterns with the Dam Standard brush — not adding detail yet, just sketching in what would become the bases for each braid. Then I drew out the braids using a curve brush with the patterns I needed, and one by one arranged them, so they felt as if they were naturally falling as they grew out of the scalp. Afterwards, I masked and then extracted what would represent rubber band hair ties.

The Fade would be created via textures, not sculpting, so I was free to move on to the modeling stage at this point.

Step-by-Step Guide

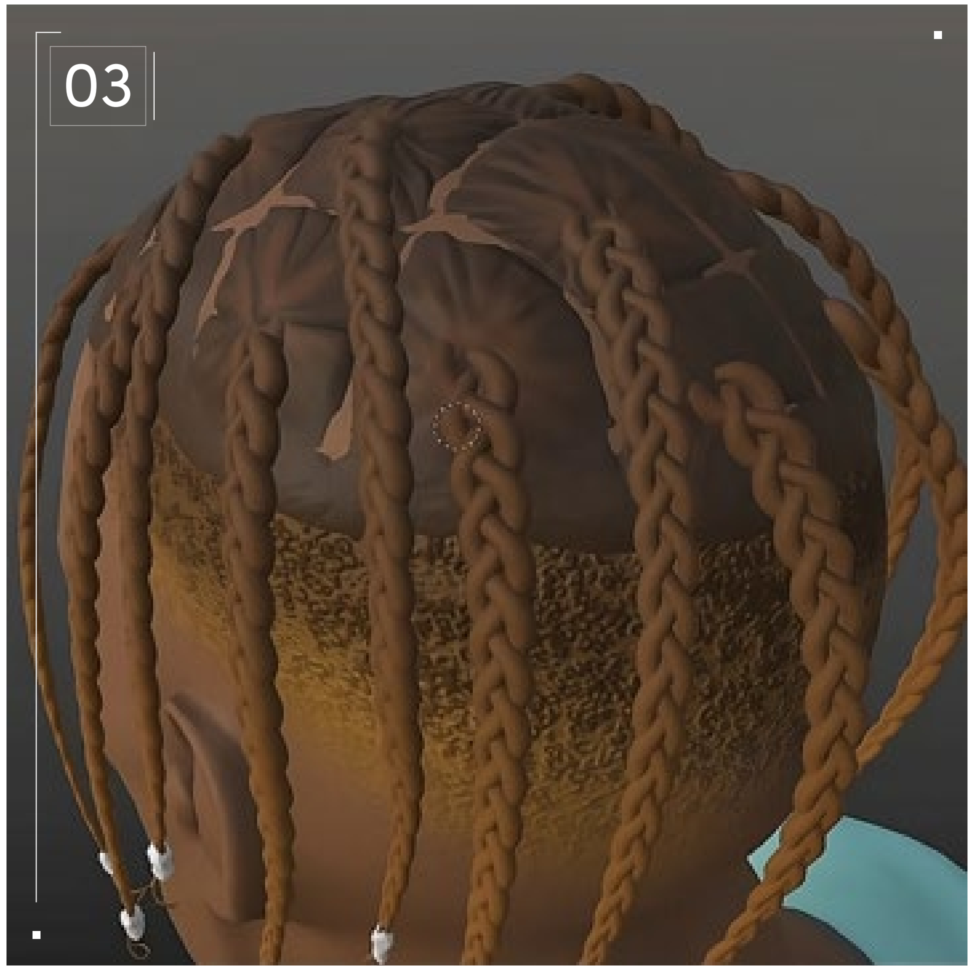


// Modeling

Now that the braids were laid, it was time to add detail to the supporting hair, which meant isolating each of the “islands” gathered to support the braids, and sculpting in details that flow from the edges of each island into the base of the braid. It was important to sell the mesh flow so that it would add to the readability of the style at a glance.

In the end, the hair ties were a small but fully necessary element that helped sell the style. With the supporting hair and hair ties done and UVed with UV Master in ZBrush — as well as the Fade and any braid details intended to be textured instead of sculpted — I was ready for the jump to FBX export to send it over to Substance Painter for baking and texturing.

Step-by-Step Guide



// Texturing

In Substance Painter, the first thing I did after baking was add a layer to the braids to serve as the base for surface detail. I put a black mask on the layer and then an anisotropic generator on the mask, and, with height information turned up and the generator set to "resolution based", I was able to simulate the presence of hair strands but with a stylistic appearance. I also put a gradient on them so that they would fade from a lighter color towards a darker one that matches the intended color for the supporting hair.

On the supporting hair mesh, I added a darker tone for a base, and then painted a lighter color in places flowing into the braid to suggest the variation in hue was happening naturally. Once the braid and scalp were colored, I needed to tackle the Fade, which, unlike the Stitch Cornrows with Fade hairstyle, would go completely around the hair, not just on the sides. In addition, it was important to create a clear delineation between the scalp hair and the Fade itself, to indicate the point at which the hair was pulled tightly into the braid.

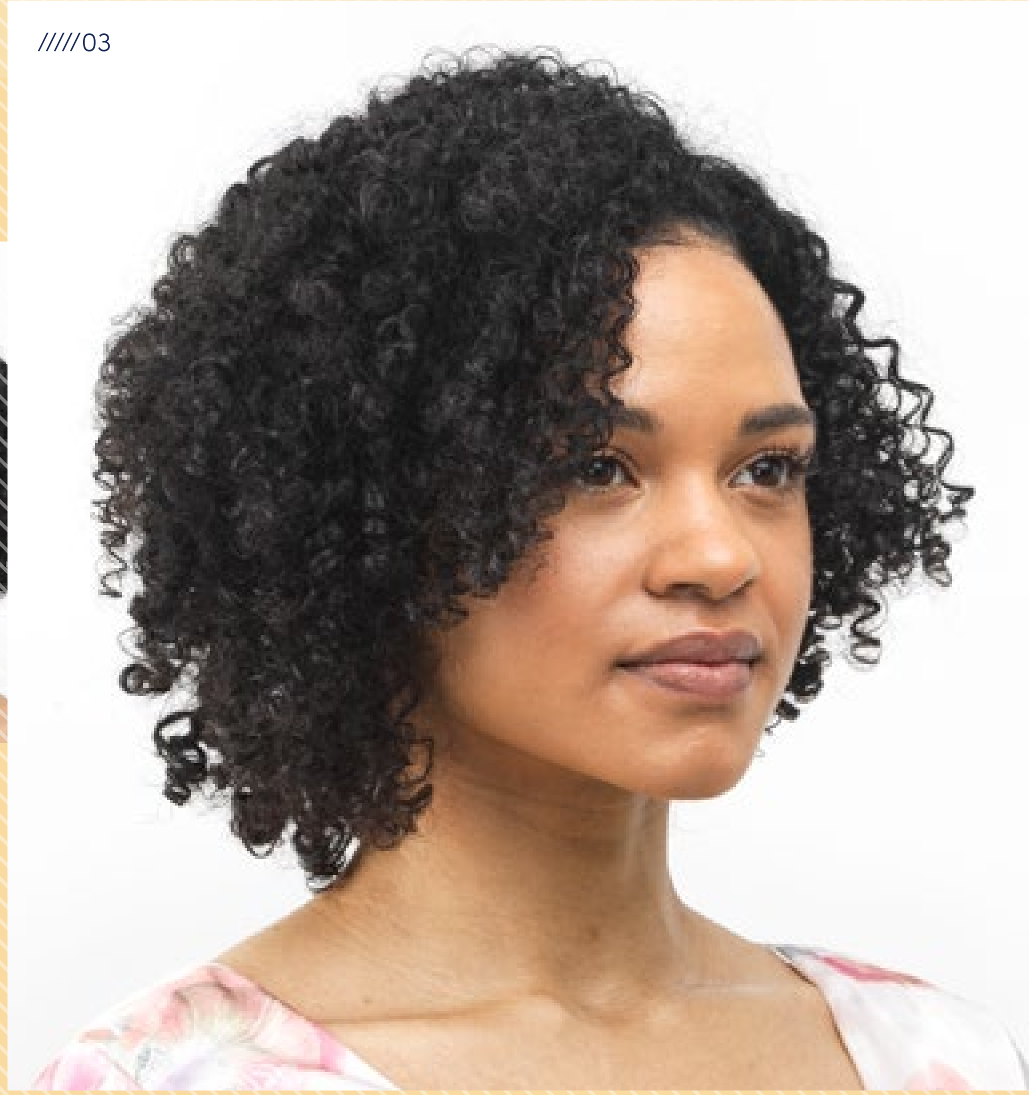
The Fade along these areas would be longer than the area furthest from the scalp and so I would have to create a second layer to build it up to differentiate between the two lengths. First, I added a new layer on the base mesh, limited to the area of the Fade I painted in ZBrush, and used noise brushes to add and erase areas until it looked like a believable, if flat, Fade. I darkened the color to more closely align with the scalp hair and braids, and then added a second layer on top of that one.

On this layer I increased the height information and then added a black mask to it, which I painted on with the "Kyle's Concepts - Dragonscales" brush, creating the illusion of short curly hair. Because it was raised, when I would erase with a soft brush, it appeared as if I was creating further gradations of length as the hair traveled down the scalp and head.

I adjusted this layer to be a lighter color than the base Fade, further adding visual depth. I deliberately did not match the hair ties to the hair because I liked the contrast they added to the lighter ends of each braid. Once the entire style was pulled together and the colors adjusted to make for a cohesive whole, I sent the textures over to Blender to finalize a render that I felt truly came to life!

Curly

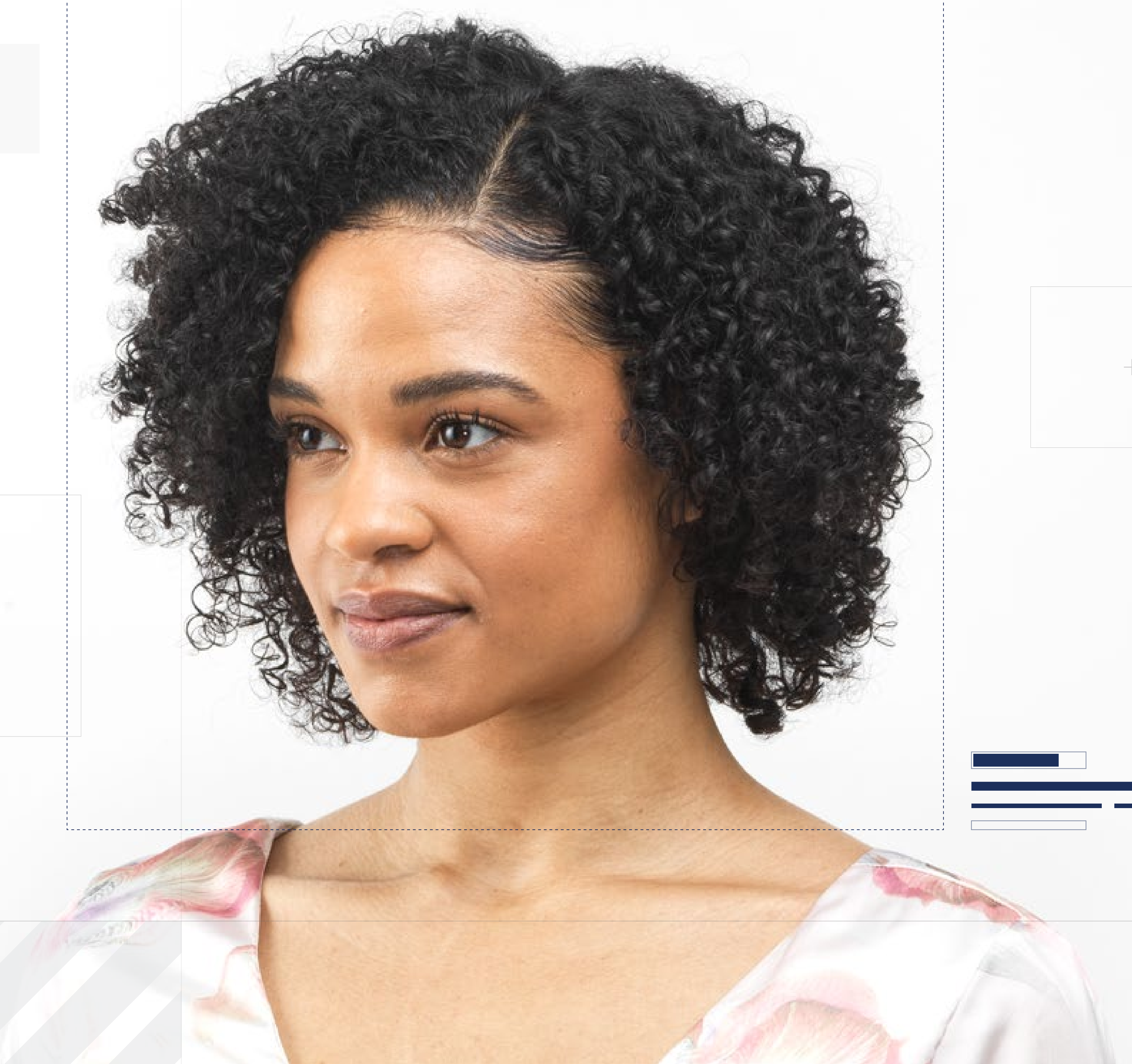
/////03



Bob
Bob
Bob

Download Images

// Curly Bob

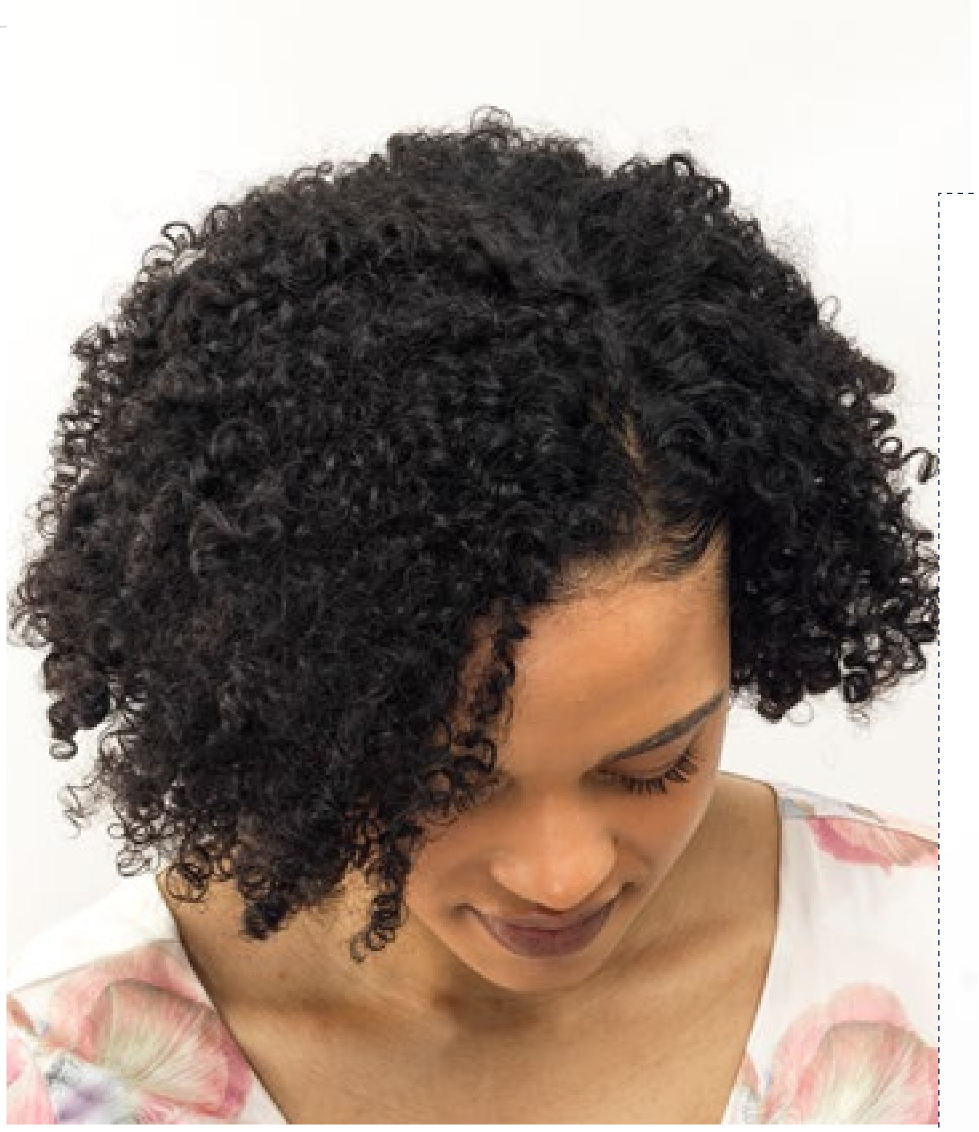


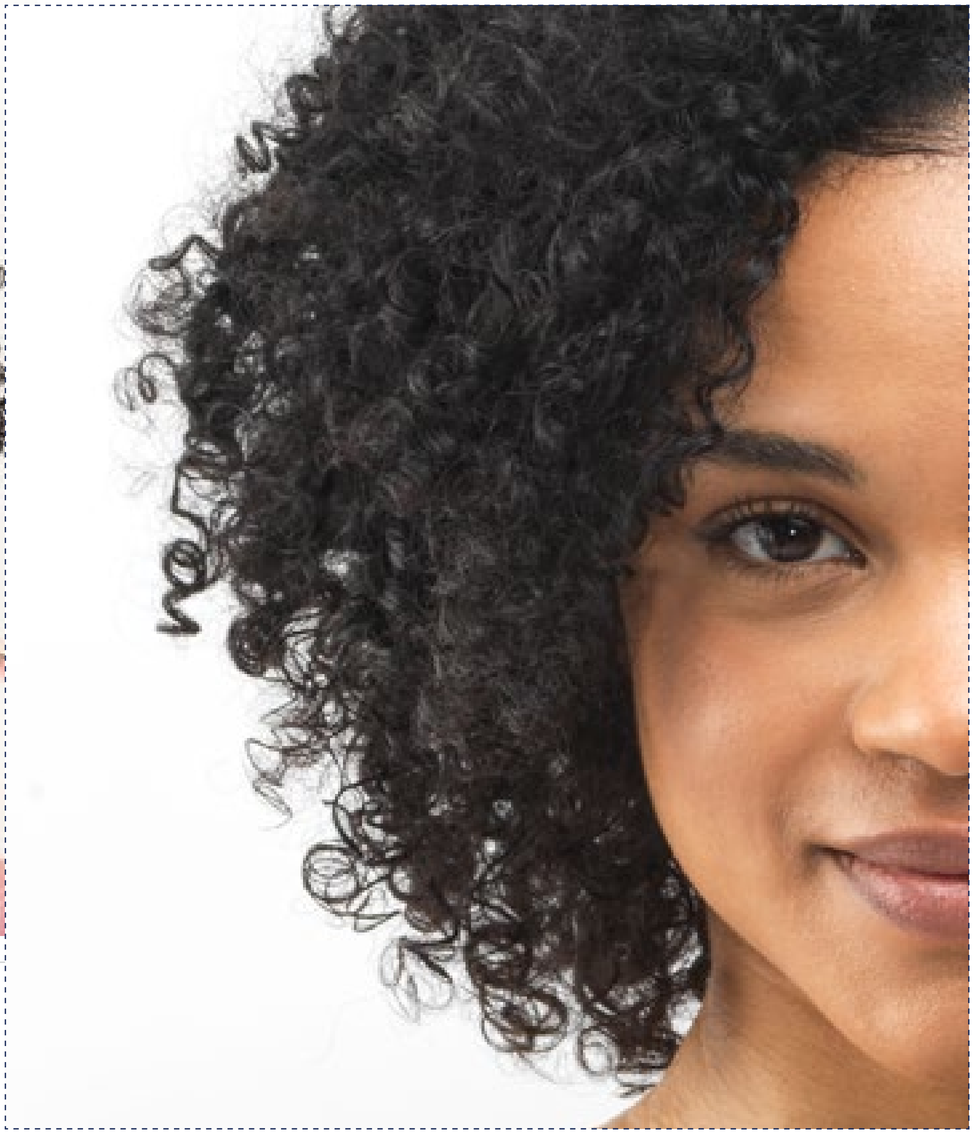
// Hairstyle

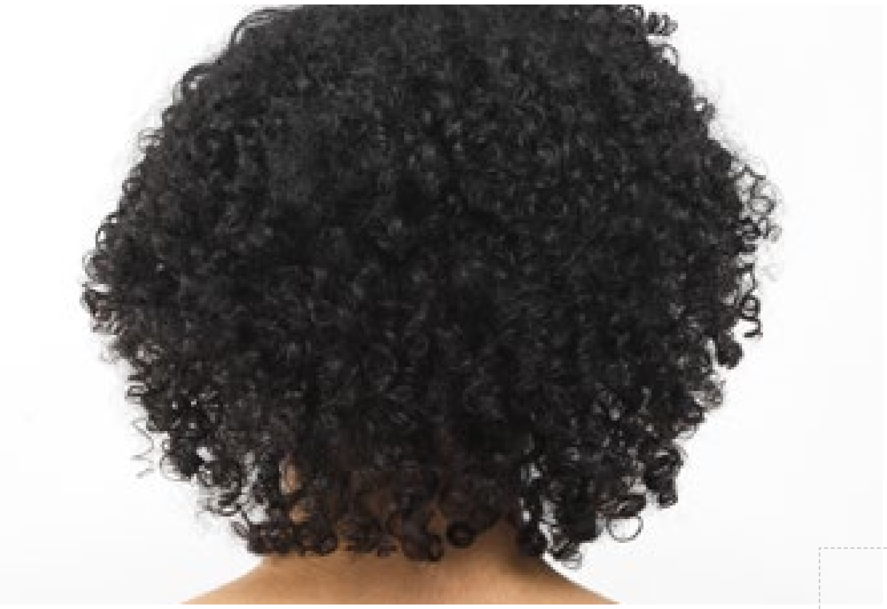
Curly Bob

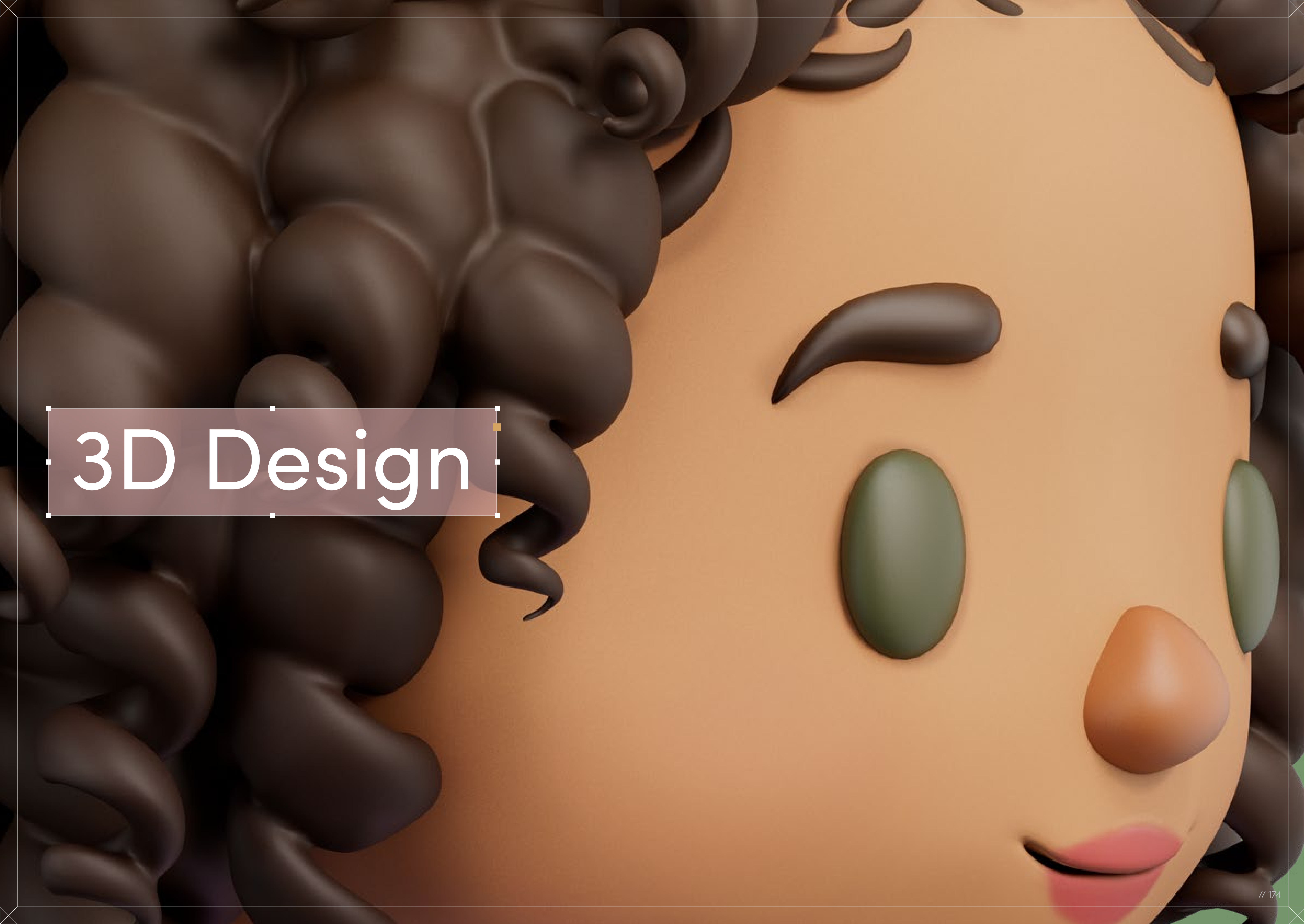
Style Background: This hairstyle is a classic bob shape with natural curls or textured waves. The defining characteristic of a Curly Bob is, in fact, the use of beautifully curly or coiled hair, and this style embraces and enhances those natural patterns. Natural hair styled in this way has seen a resurgence in recent years as several celebrities and online influencers have rocked the style.

Download Images



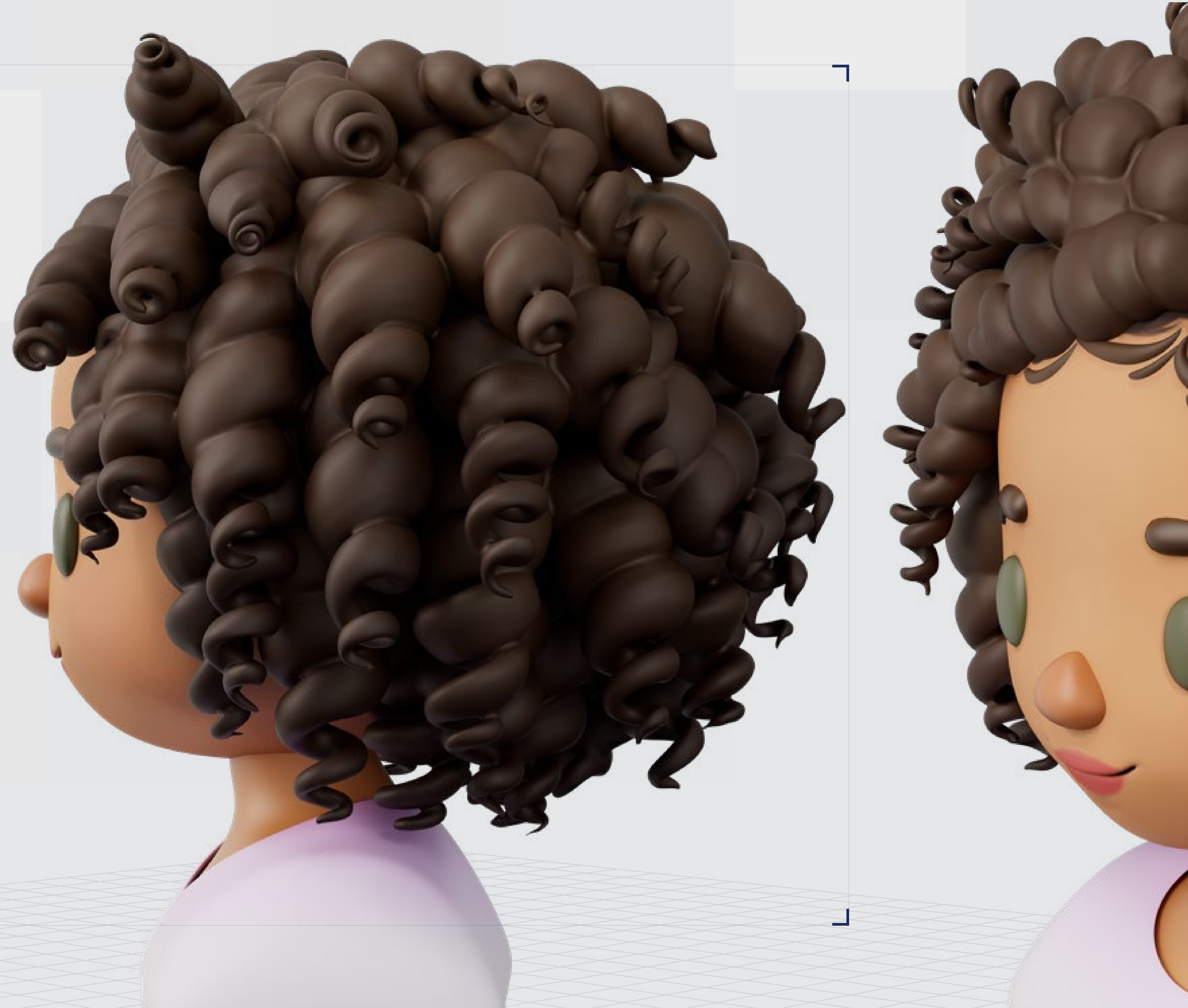






3D Design

Curly Bob



Digital Artist:
Chrissy Powell

Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

// Curly Bob

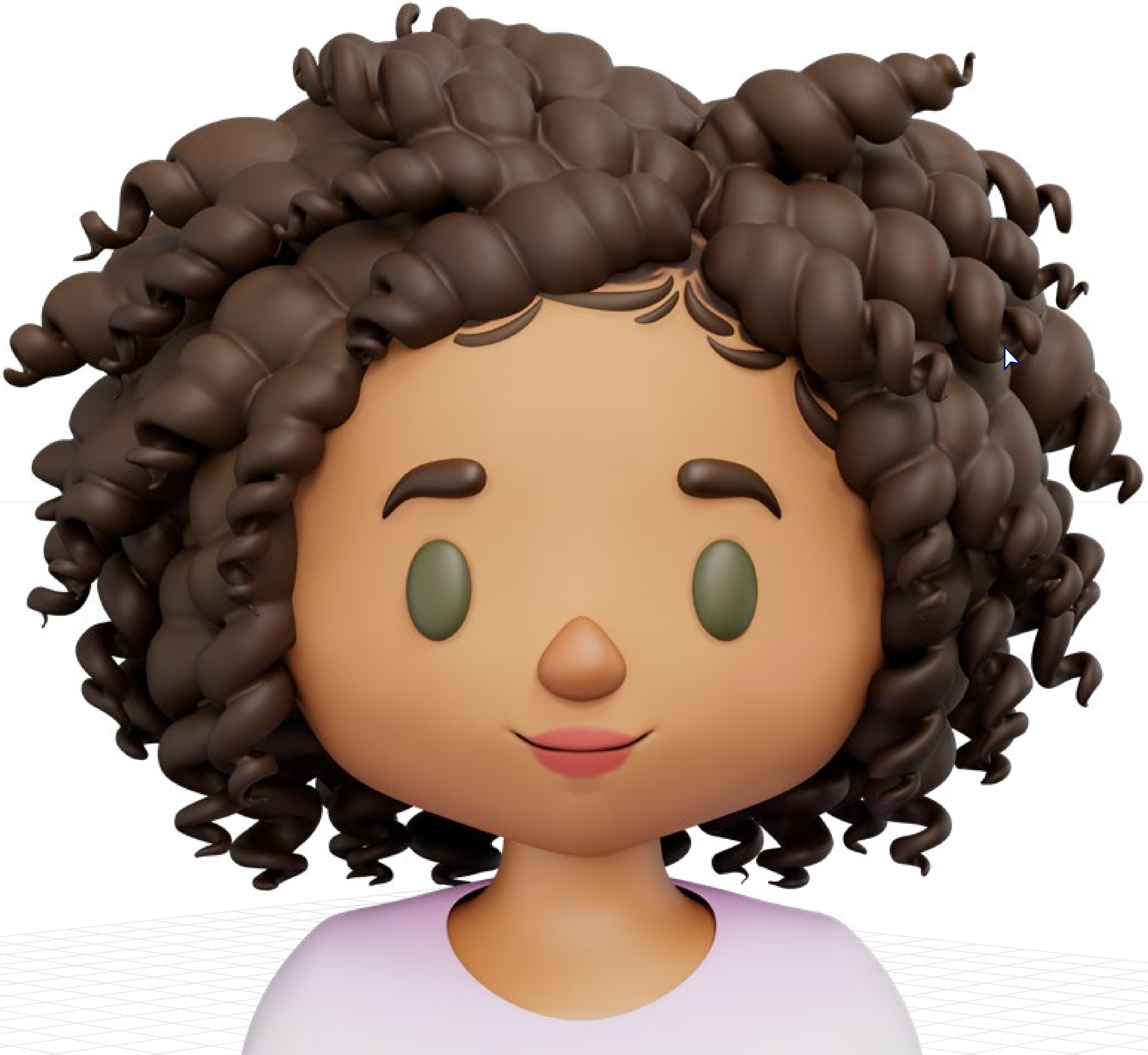
3D Design

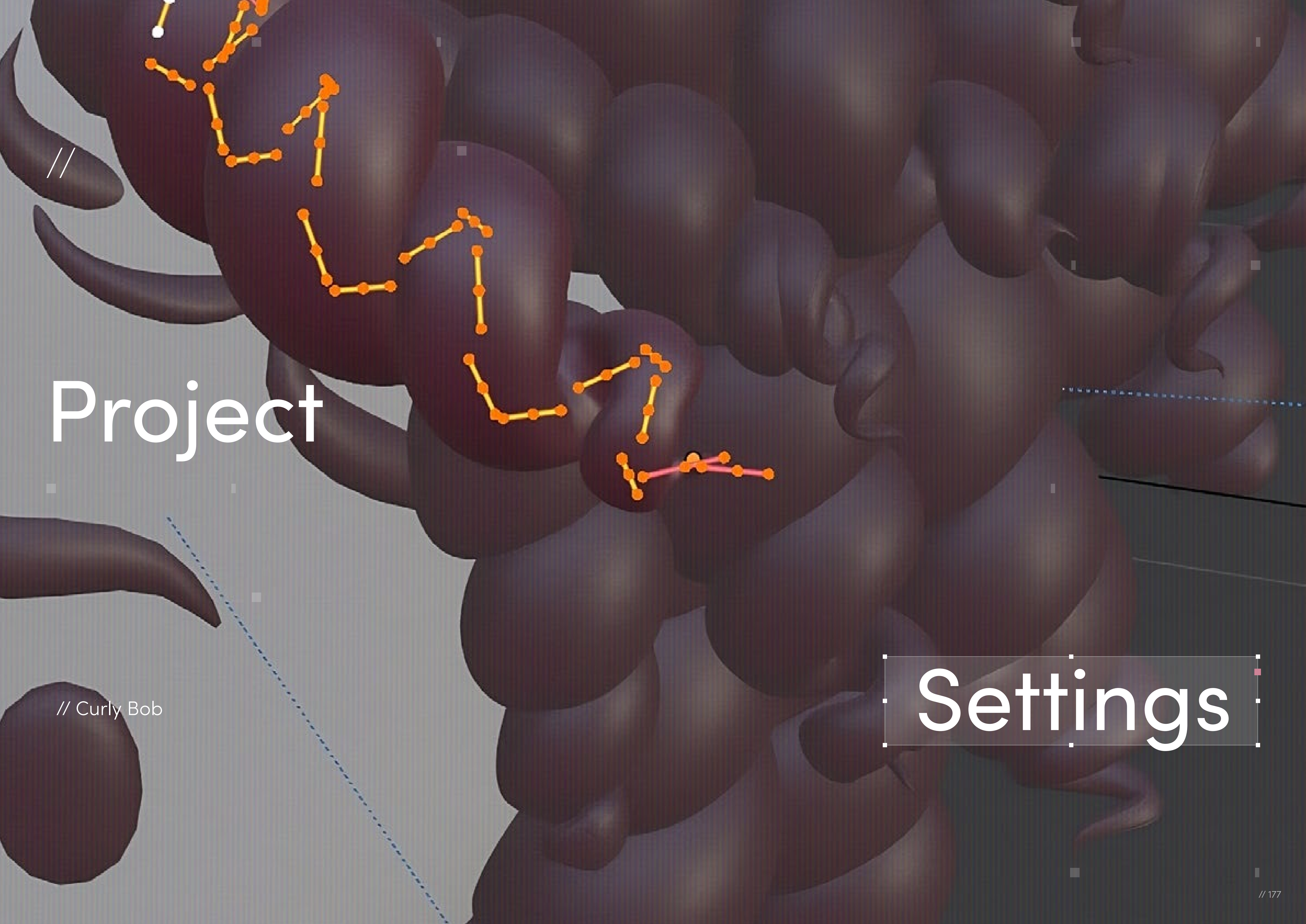
Curl Type Modeled: **3A**

// 01

// 02

Download 3D Render





//

Project Settings

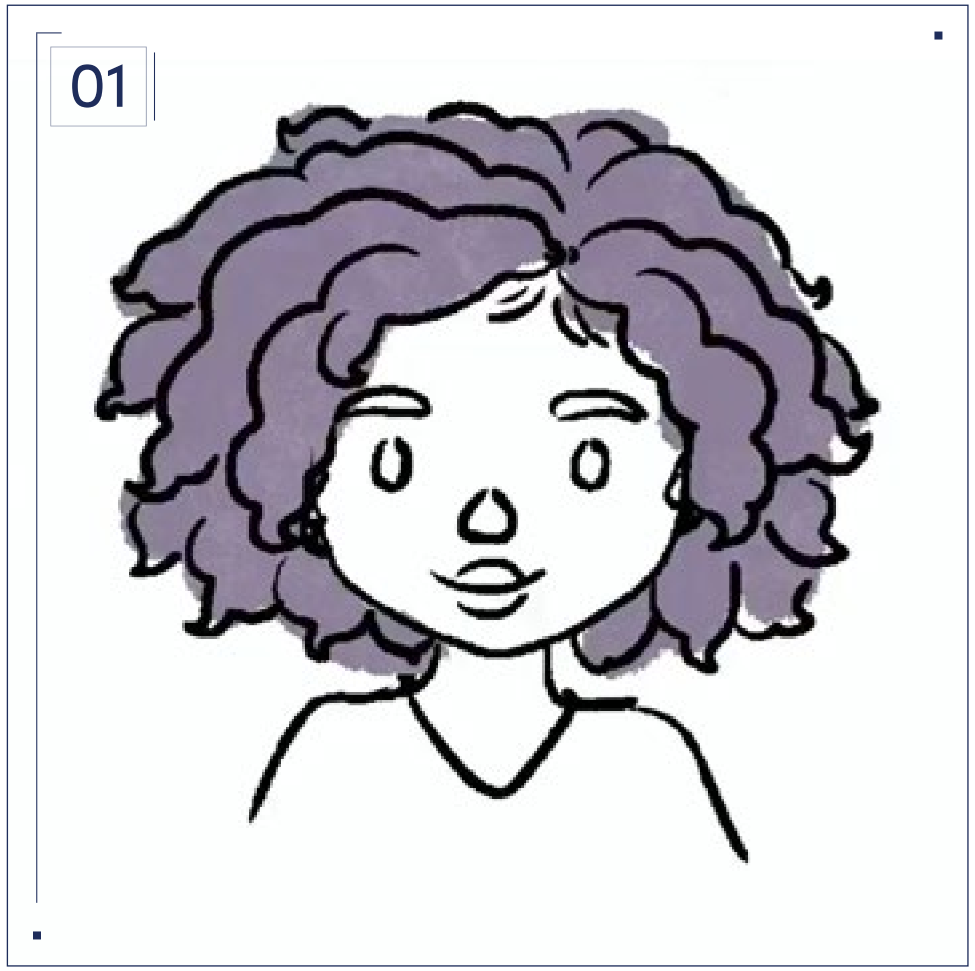
// Curly Bob

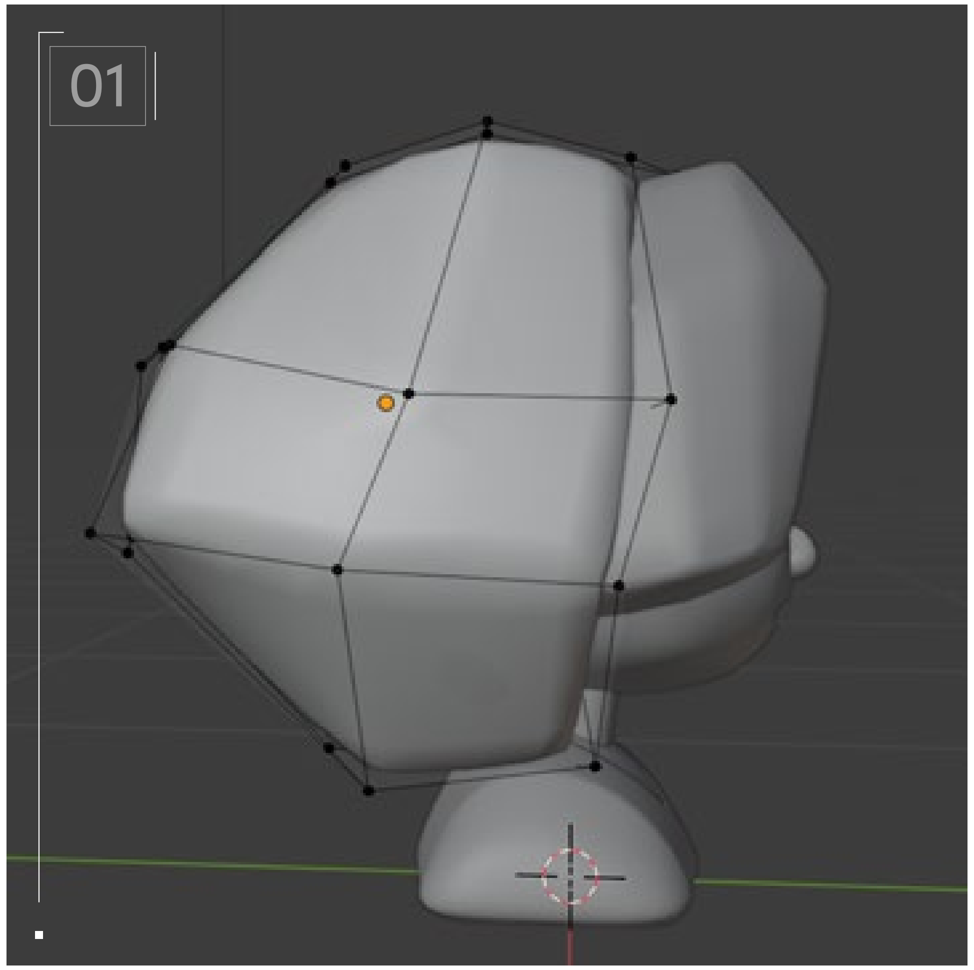
// Project Settings

Program Guide

Software: Blender
Render Engine: Procreate (2D)
Additional Plug-Ins: Illustrator (2D)

Step-by-Step Guide





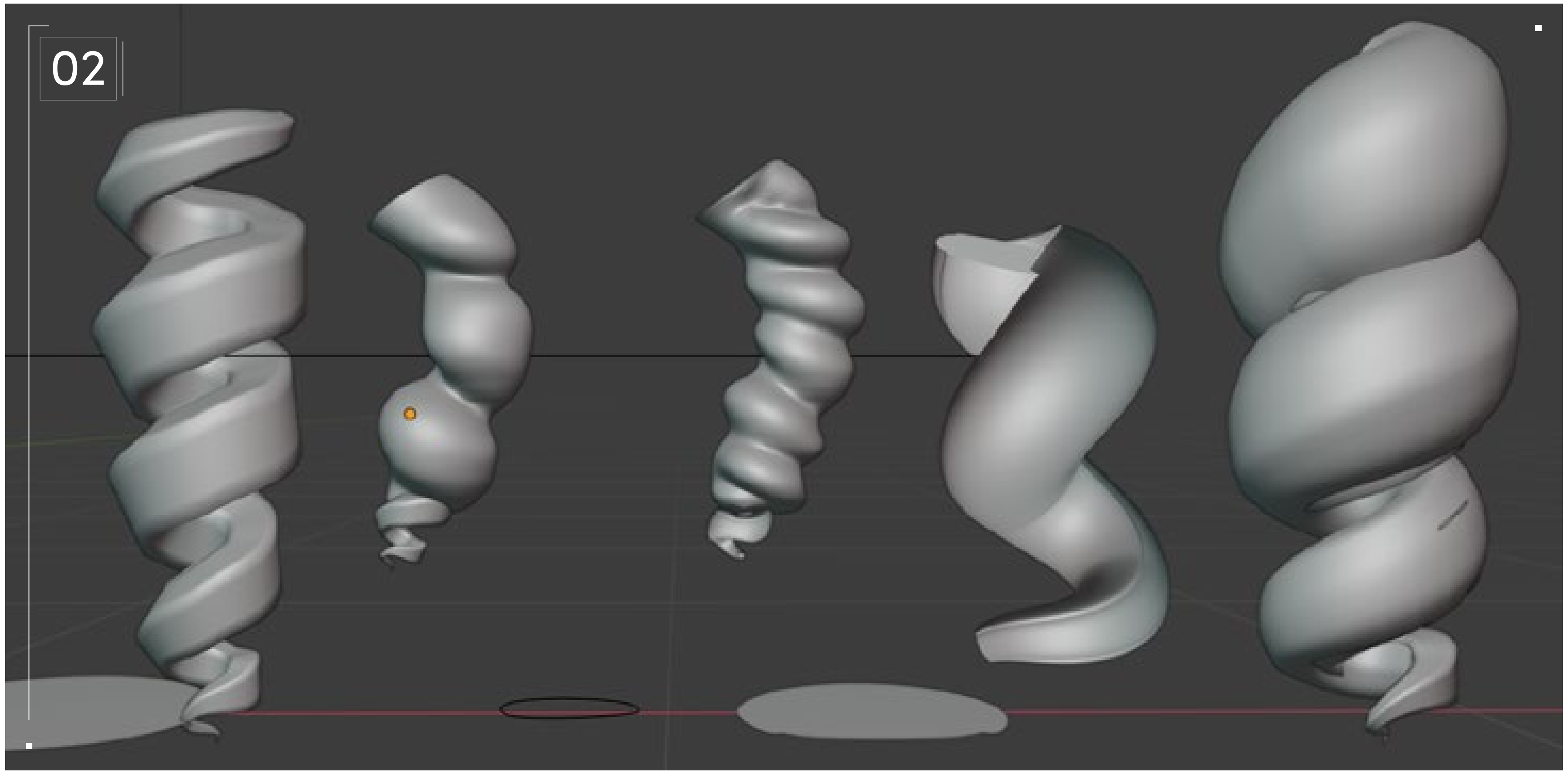
There are barely any tutorials for Black hair textures so every new project is an experiment for me. Since I come from a 2D illustration background, I always start with a rough sketch. I am entirely self-taught in 3D modeling so sketching helps me to quickly figure out my approach to stylization.

For this model, I experimented with a few different ways of portraying curls. This curl pattern is on the looser side, so I wanted to focus on the curls themselves and make them the star of the hairstyle.

Blocking is straightforward. I used primitives (cube) to block out the overall shape, taking care to consider the silhouette from all angles, not just the front and back. Extra geometry was added as needed. I did not block with curves because I just wanted to visualize the overall volume.

I also determined which parts of the hair I would create with texturing versus geometry at this stage.

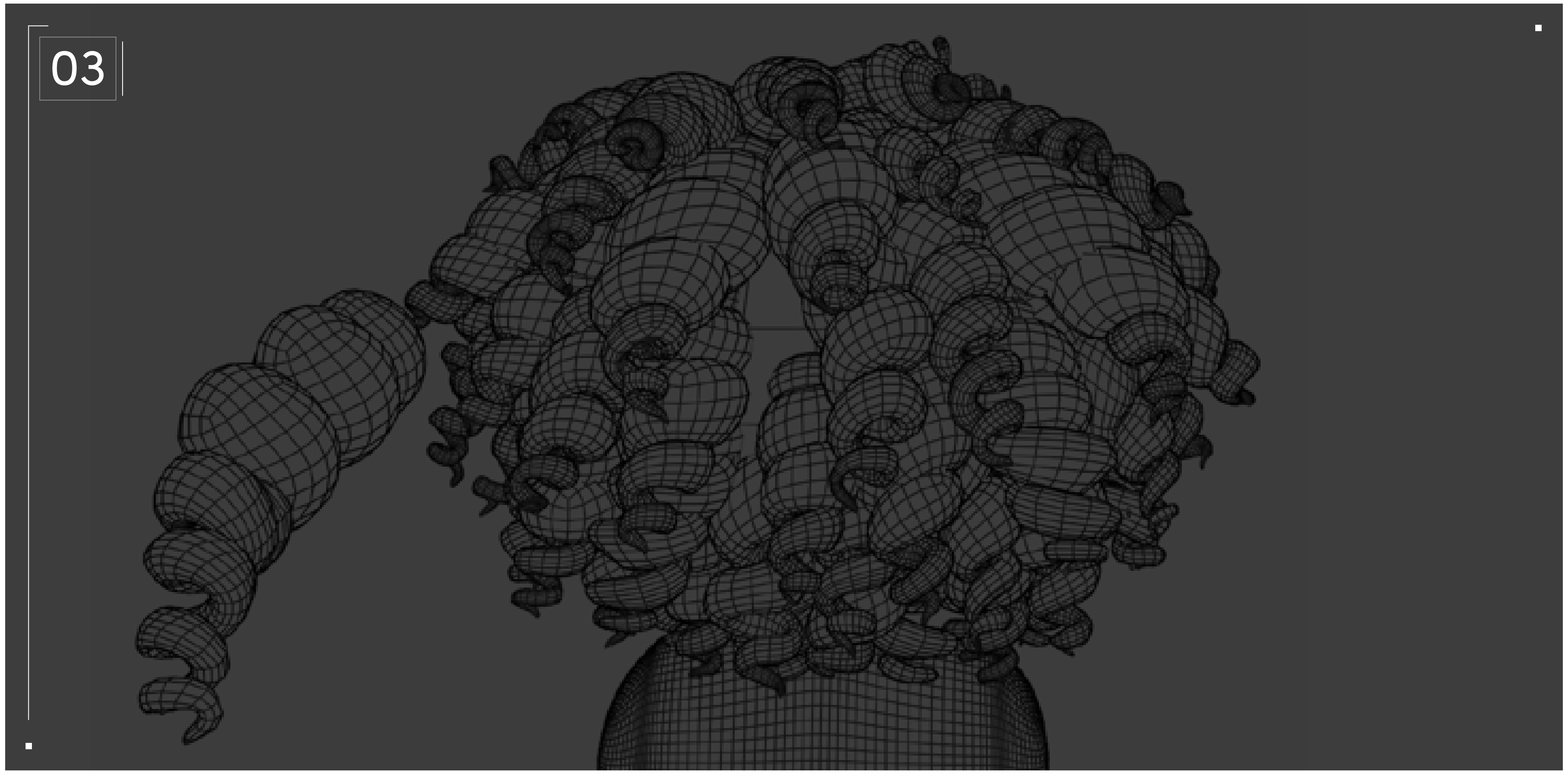
Step-by-Step Guide



Next, I began to create some tests for the 3D curls. I have not attempted this looser style of natural curl before, so testing was indispensable. For this style I tried extruding geometry, twisting curves, adding geometry to curve profiles, and finally settled on a combination of two techniques: a large tapered spiral curve for the body of the curl, with a smaller spiral curve at the ends.

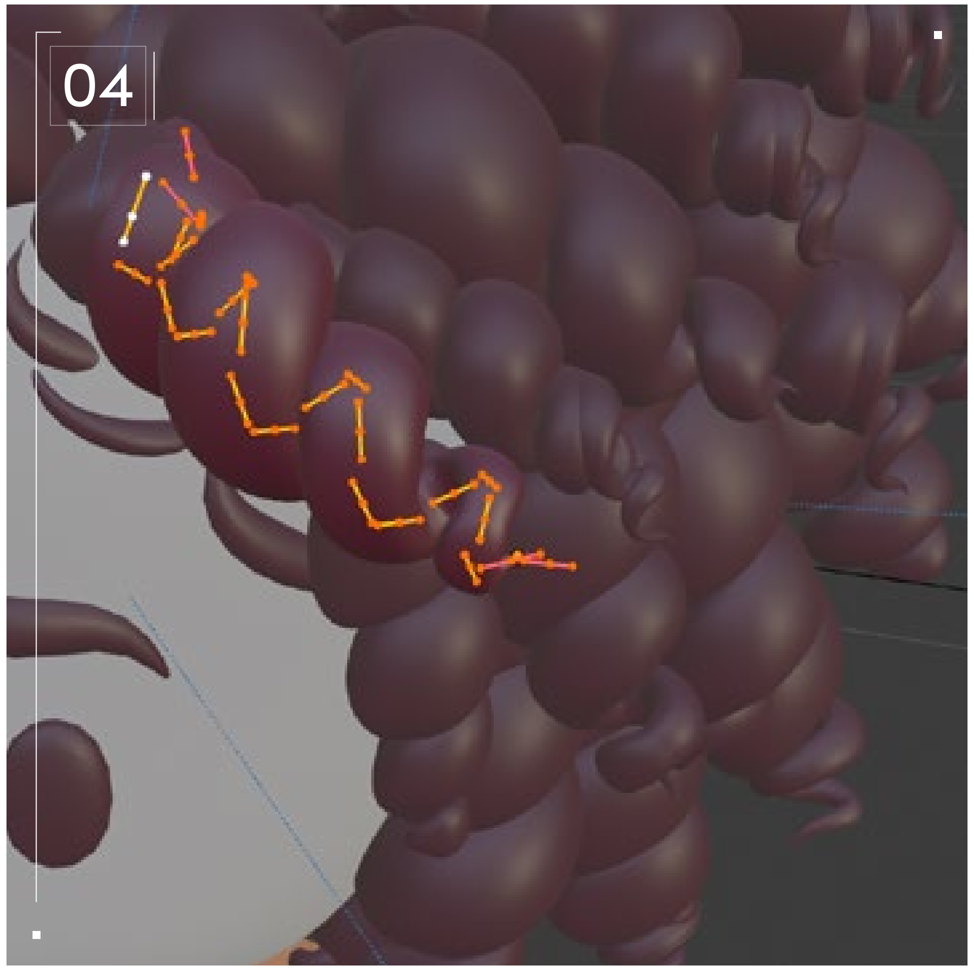
I used two spirals to be able to adjust the thickness and amount of detail I wanted for each strand, particularly at the ends. I will say this method did not produce the cleanest geometry but the shape was exactly what I wanted. You can probably also achieve this with geometry extrusion and an array modifier but, like I said, this was an experiment!

Step-by-Step Guide

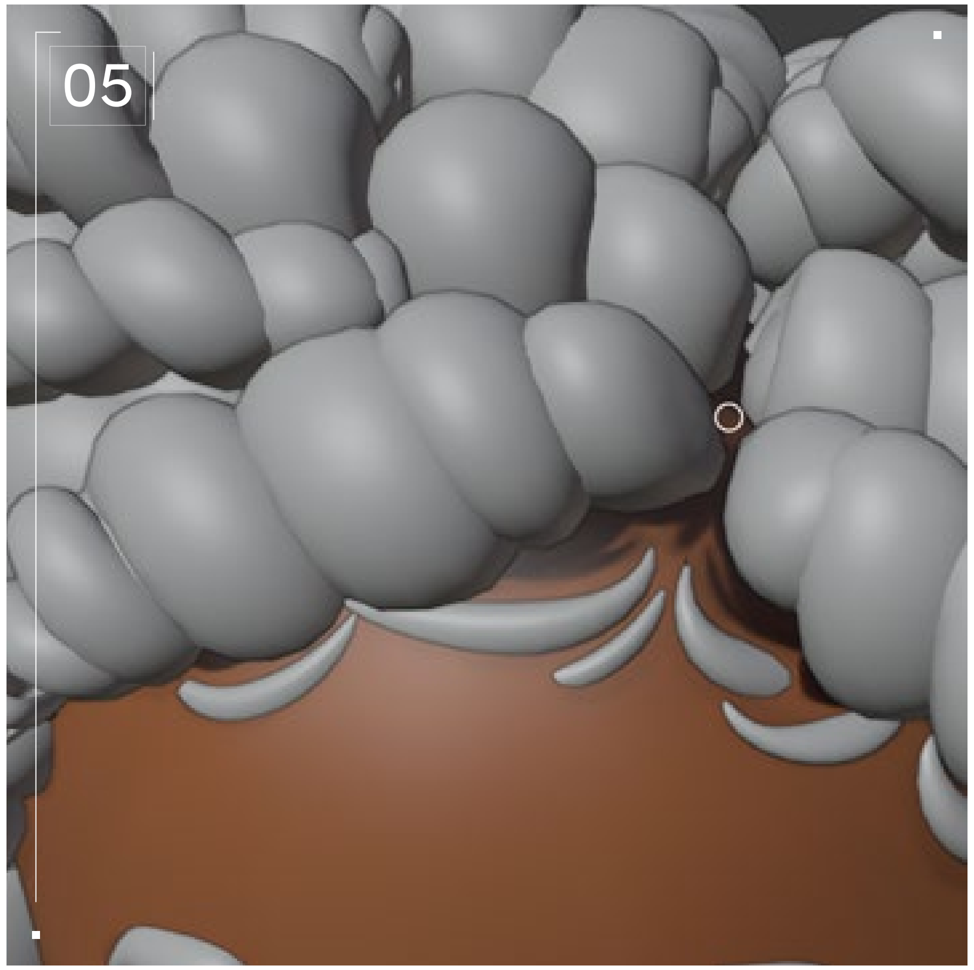


Once I created three or four curls I liked, I manually placed them around the scalp. The curls were individually adjusted in scale, curve, and length to create a more natural and interesting composition. I feel the manual placement of curls is necessary because the final model has life and buoyancy to it, even while not in motion. For different curl textures, I would need to experiment, but changing the overall size and density of the curls would be a start.

Step-by-Step Guide

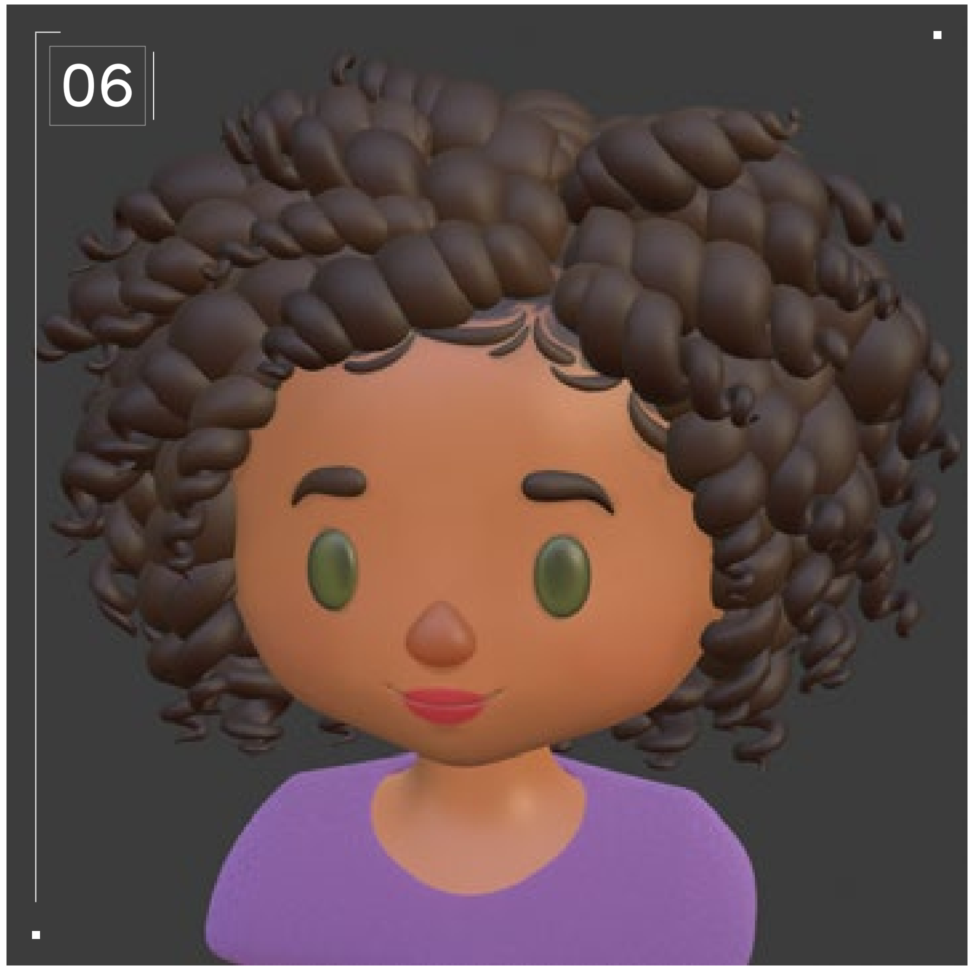


The spiral curls were then joined by section (to decrease load on my processor), remeshed, and retopologized. I did a few experiments here to see how the procedural shading would look on various sections of hair before settling on the final organization. It's subtle but I really think it adds something to such a simplified design. Overall, I did not pay attention to polycount, but you can remove some of the unseen sections of curls if needed.



Baby hair was created using regular curves with a basic taper profile. Please use multiple references for baby hair.

Not all baby hair is the same, nor is it interchangeable across all styles, as you can see with the variety of baby hair in all of our models.

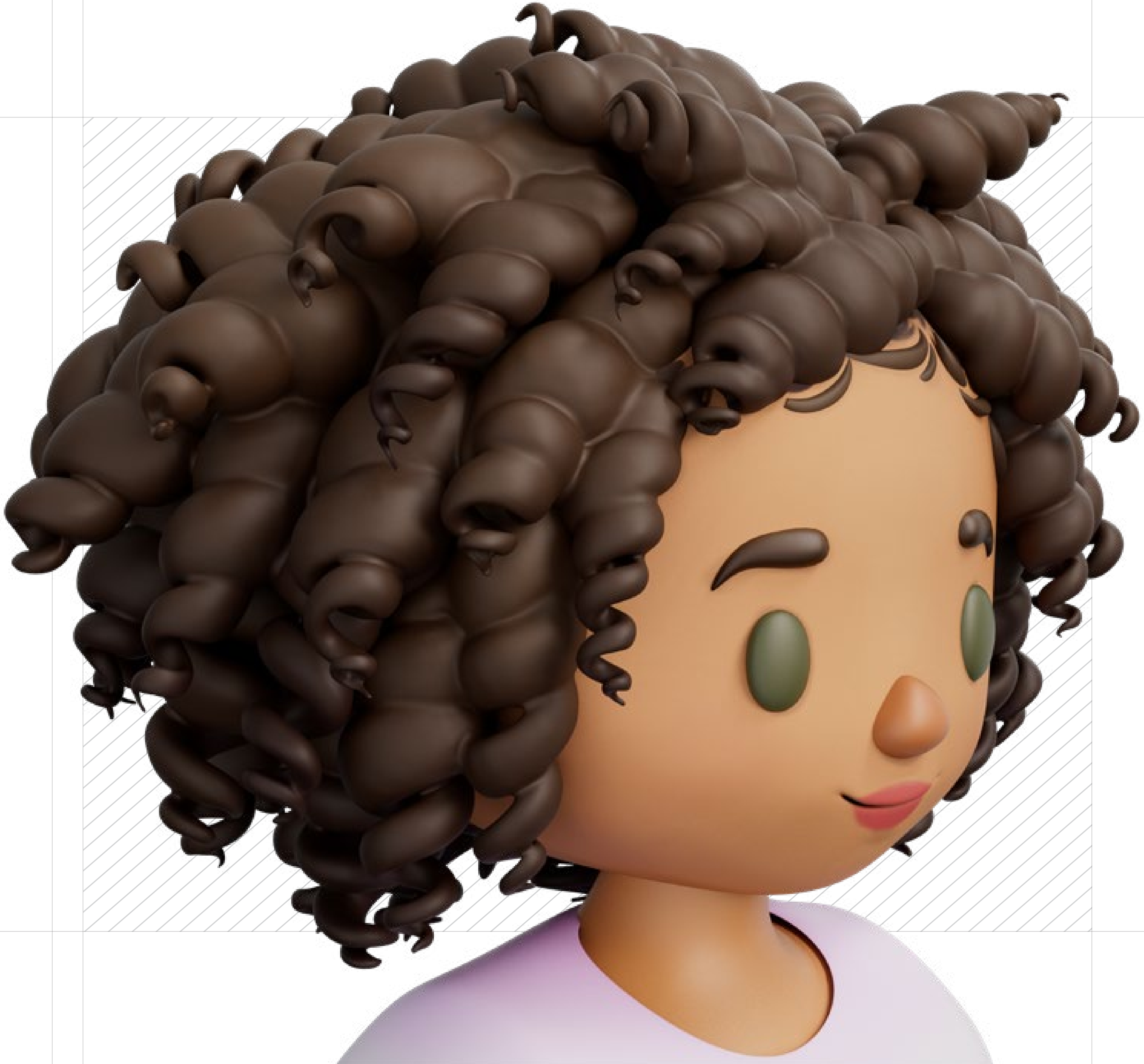


Texturing was kept simple and toy-like. I used procedural textures to give a subtle gradient and texture to the hair.

Technical Watchouts

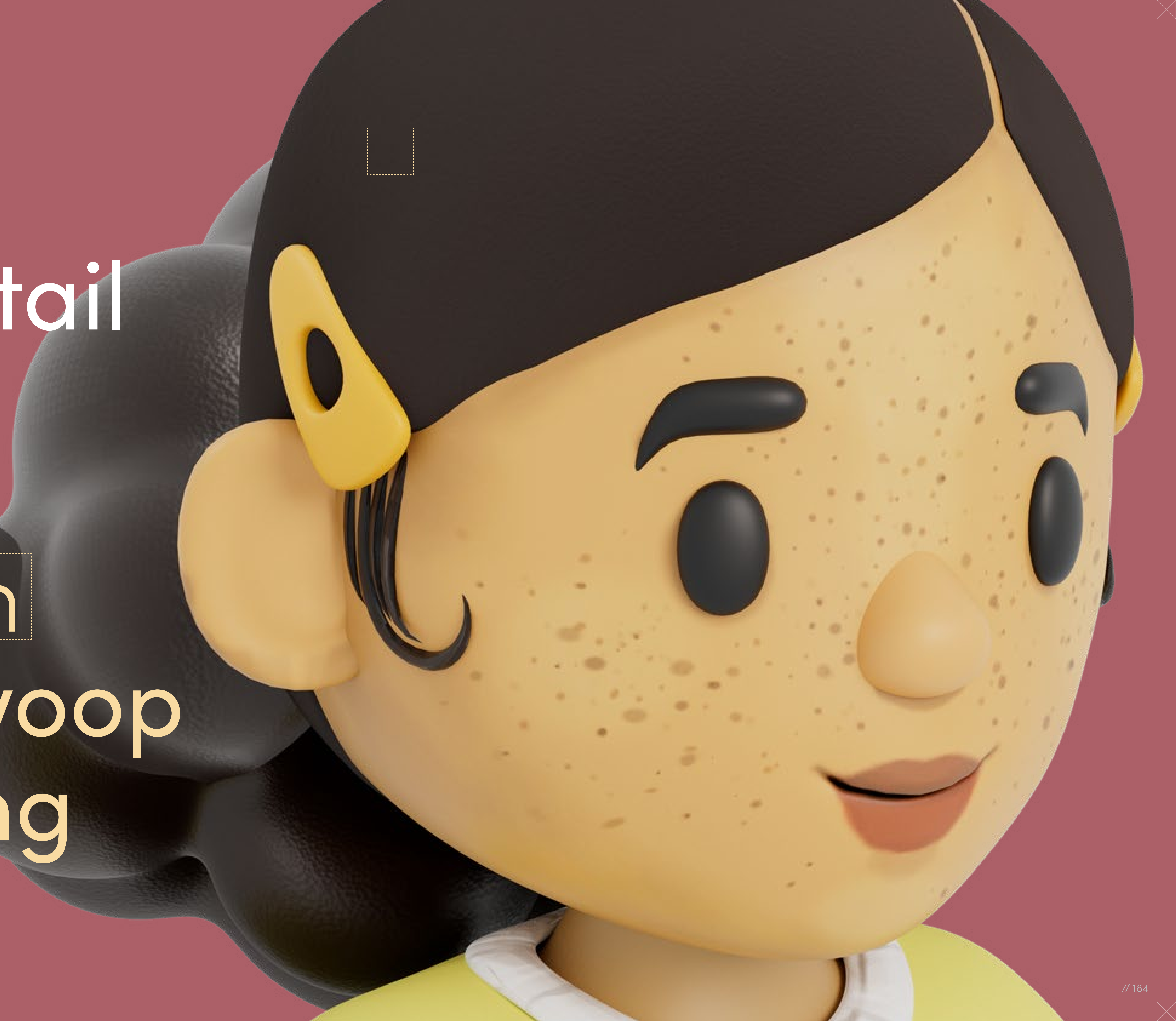
I purposely went with a stylized, simple look for my models because stylization is one of the hardest things to do effectively — you need to intimately understand your subject and all of its qualities. The first course of action for any natural hair creation is collecting a large database of references. One or two will not be enough if you aren't well versed in Black hair textures. I have had natural hair my whole life but I still don't know everything there is to know about all textures and styles. However, I do consider myself, and the other developers featured in this project, specialists in Black hair stylization! We are able to apply what we know from years of lived experience to any new style challenges and I believe that's what makes our stylization so distinct!

3D natural hair often goes awry by focusing too much on the wrong details. For example, a very common stylized natural hairstyle I see is a big smooth sphere with a curly texture applied to it. This is reminiscent of a '70s Afro and is often a hairstyle that comes to mind when people who are not familiar with curly hair are asked to think of a style. The problem is, not many people wear their hair perfectly round like that anymore — especially not with the muttonchop sideburns I see so often in games. It has become an outlier instead of the norm and will immediately come off as lazy and insensitive. To accurately portray natural hair in a modern and faithful way, you have to also consider how you will stylize the overall silhouette of the hair in a way that both makes sense and that most people currently wear it.



Low Ponytail

with Swoop Bang

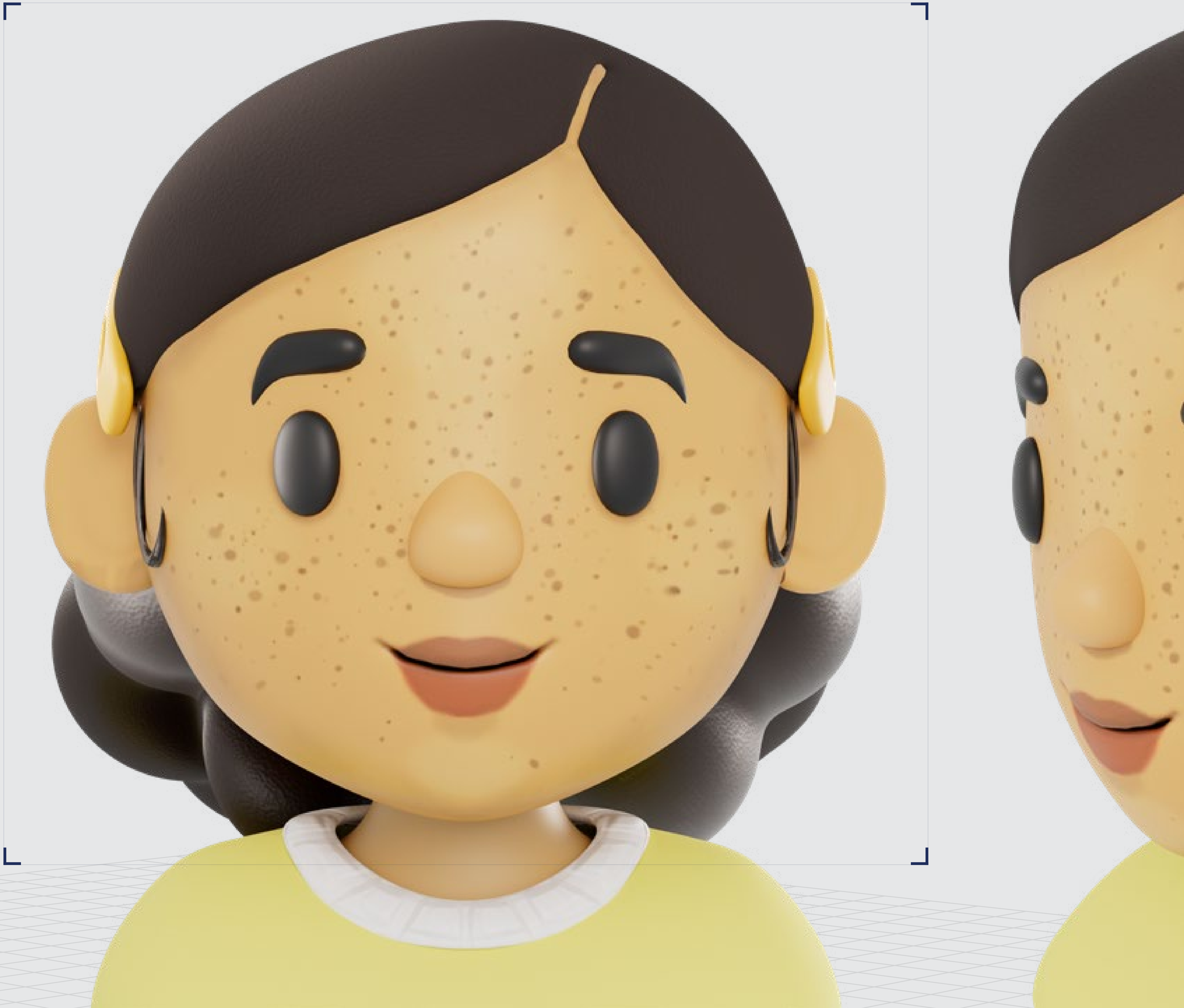


// Hairstyle

Low Ponytail with Swoop Bang

Style Background: Since much of Black haircare is focused on length retention, Swoop Bangs allow you to achieve the aesthetic of a bang without having to actually cut your hair. Hair of any texture can be brushed flat across the forehead using styling products, such as gels, to keep it in place and give the appearance of a bang.

Digital Artist:
Chrissy Powell



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

// Low Ponytail with Swoop Bang

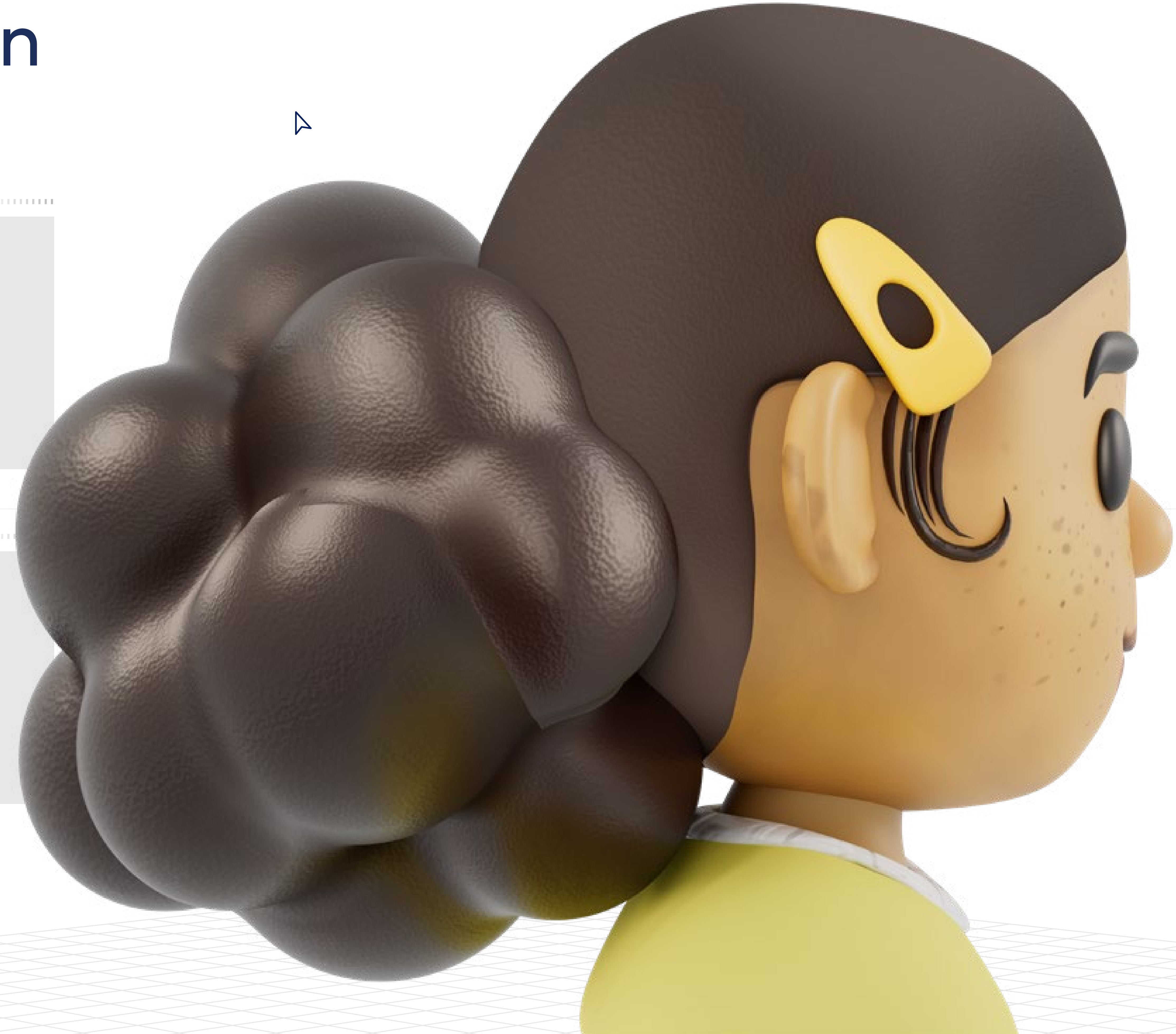
3D Design

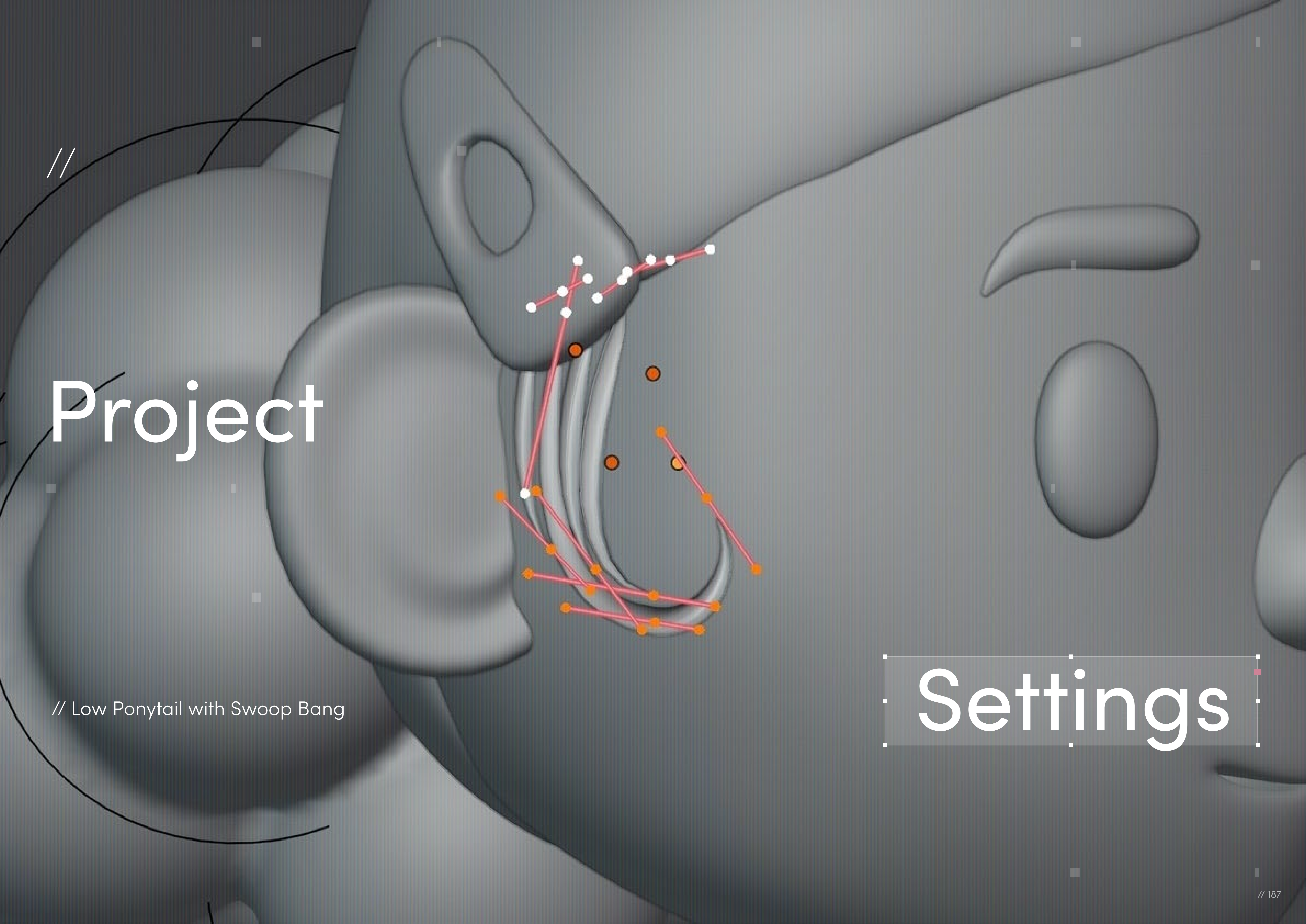
Curl Type Modeled: **4A/4B**

// 01

// 02

Download 3D Render





//

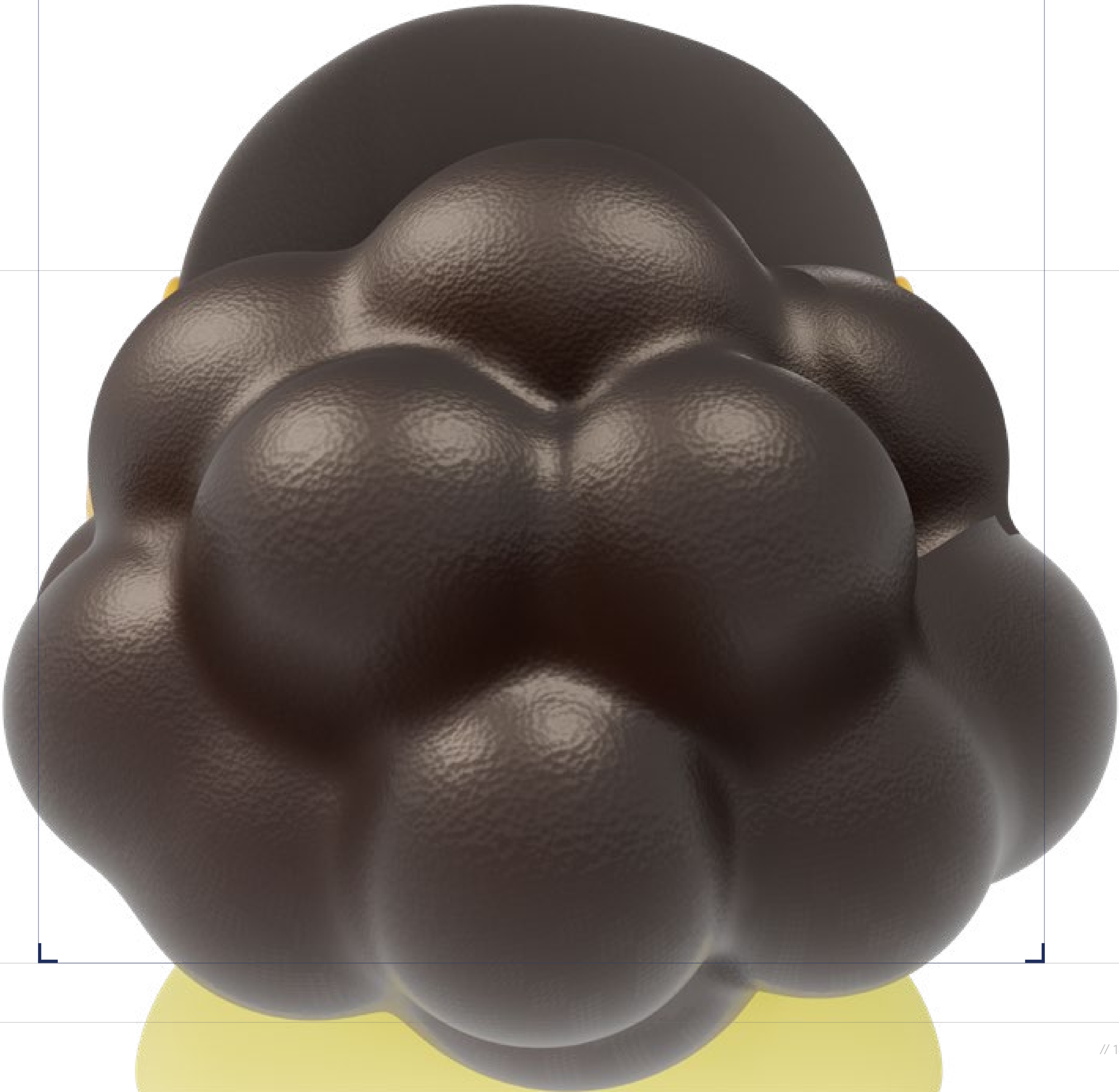
Project Settings

// Low Ponytail with Swoop Bang

// Project Settings

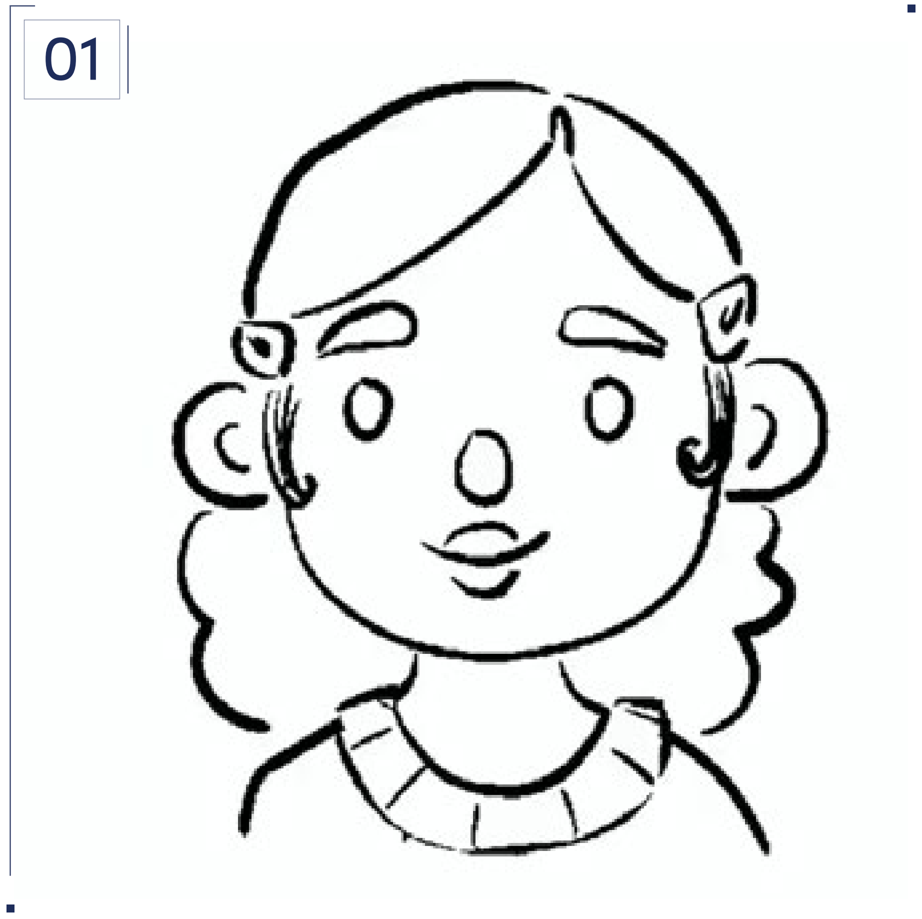
Program Guide

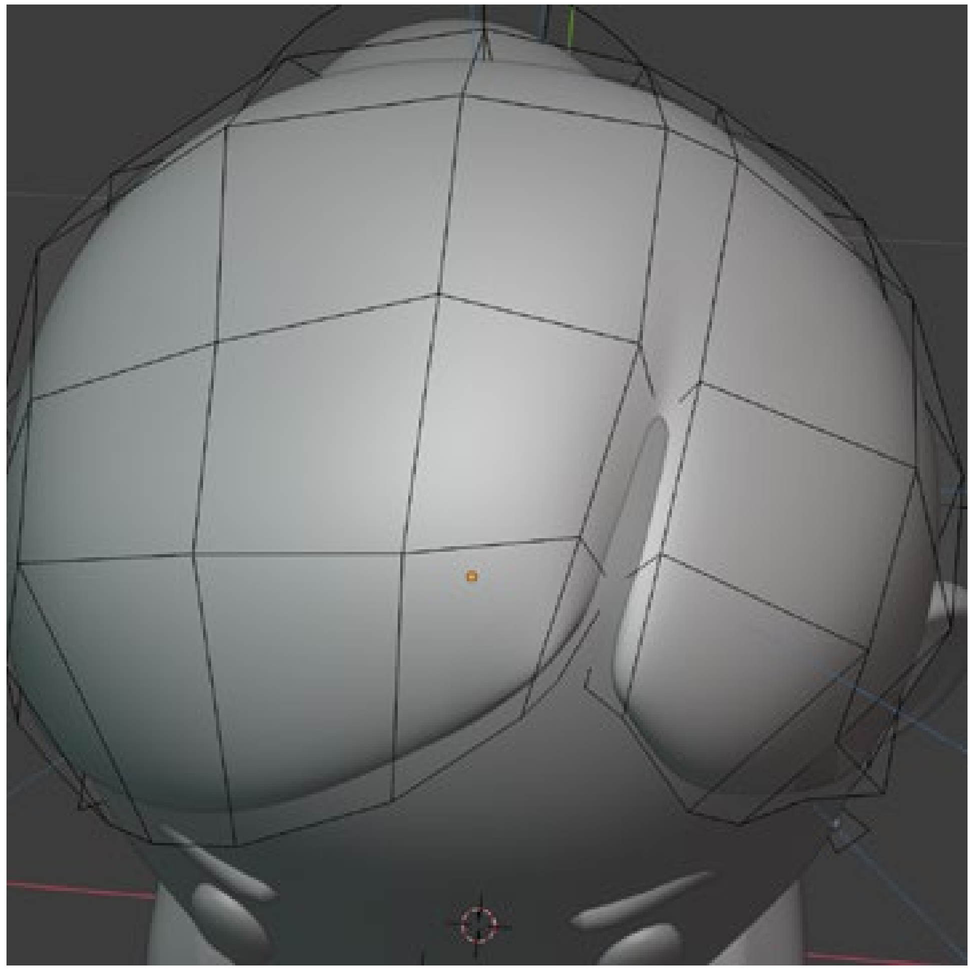
Software: Blender
Render Engine: Procreate (2D)
Additional Plug-Ins: Illustrator (2D)



Step-by-Step Guide

01





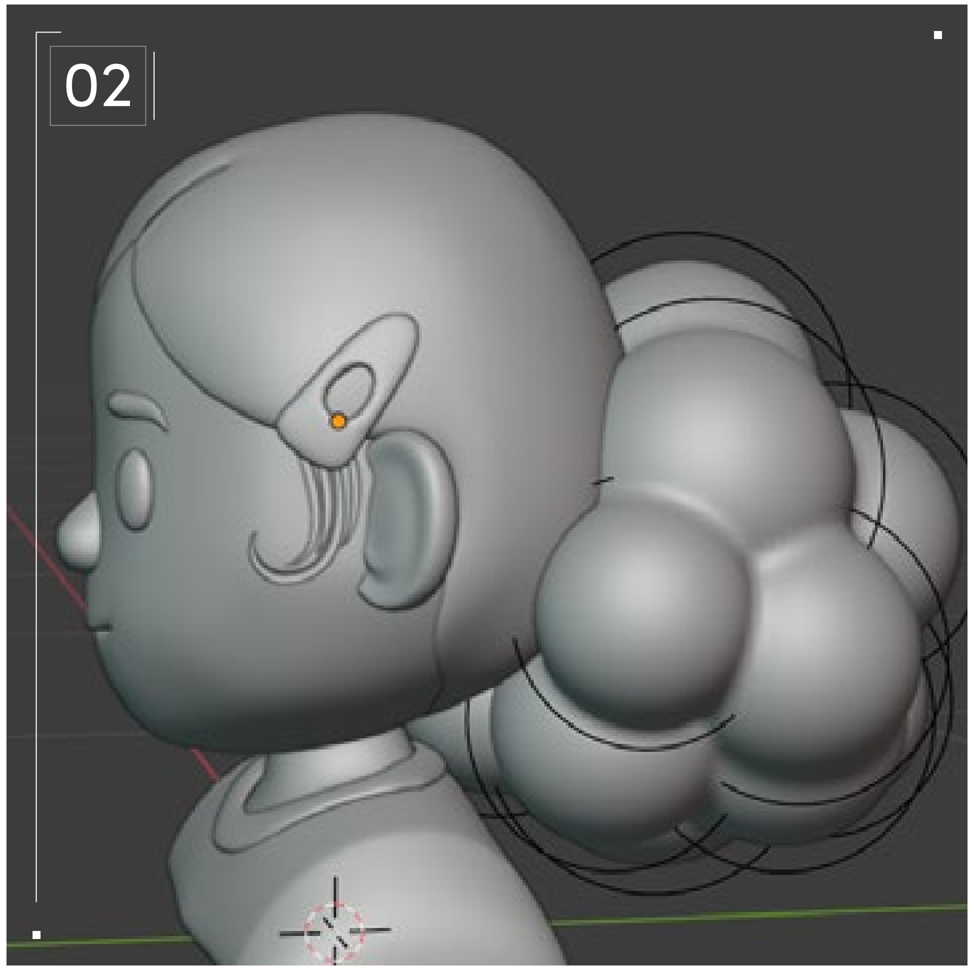
This hairstyle is a great staple to have in any character customization lineup and yet I never see it. It's cute, simple, and can be done on any curly hair texture! I think a lot of people would love to see this style available to them, especially with customizable hair clips!

Since I come from a 2D illustration background, I always start with a rough sketch. Sketching helps me to quickly figure out my approach to stylization. This one was pretty straightforward and did not require much experimentation.

For different curl patterns, the front of the style will remain mostly the same but the back Puff will be stylized differently. I cannot give much more detail here as it will greatly depend on the curl pattern you are going for. Experiment with adding different shaped curls and altering the Puff's overall silhouette to match your references. Hair clips can be added to enhance the style and bridge the eventual change in geometry between the swoop and the baby hair.

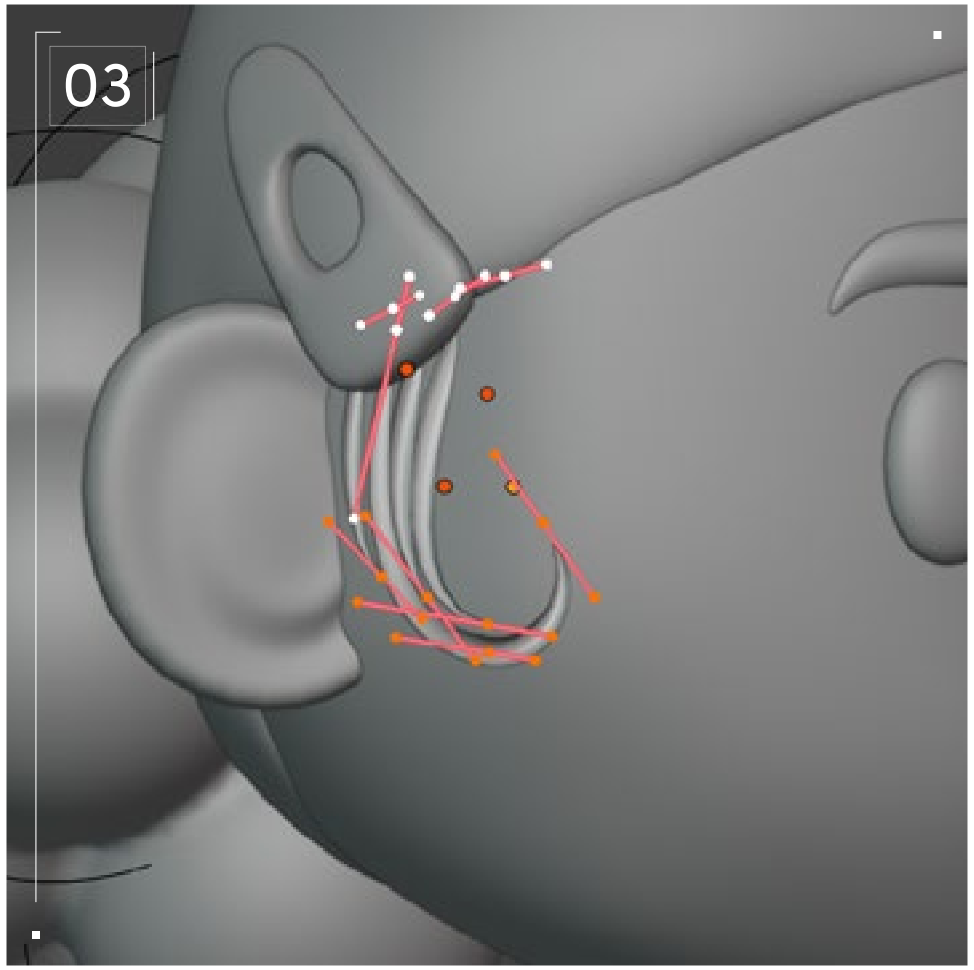
Blocking is straightforward. I used primitives to block out the overall shape, taking care to consider the silhouette from all angles, not just the front and back. Extra geometry was added as needed.

Step-by-Step Guide



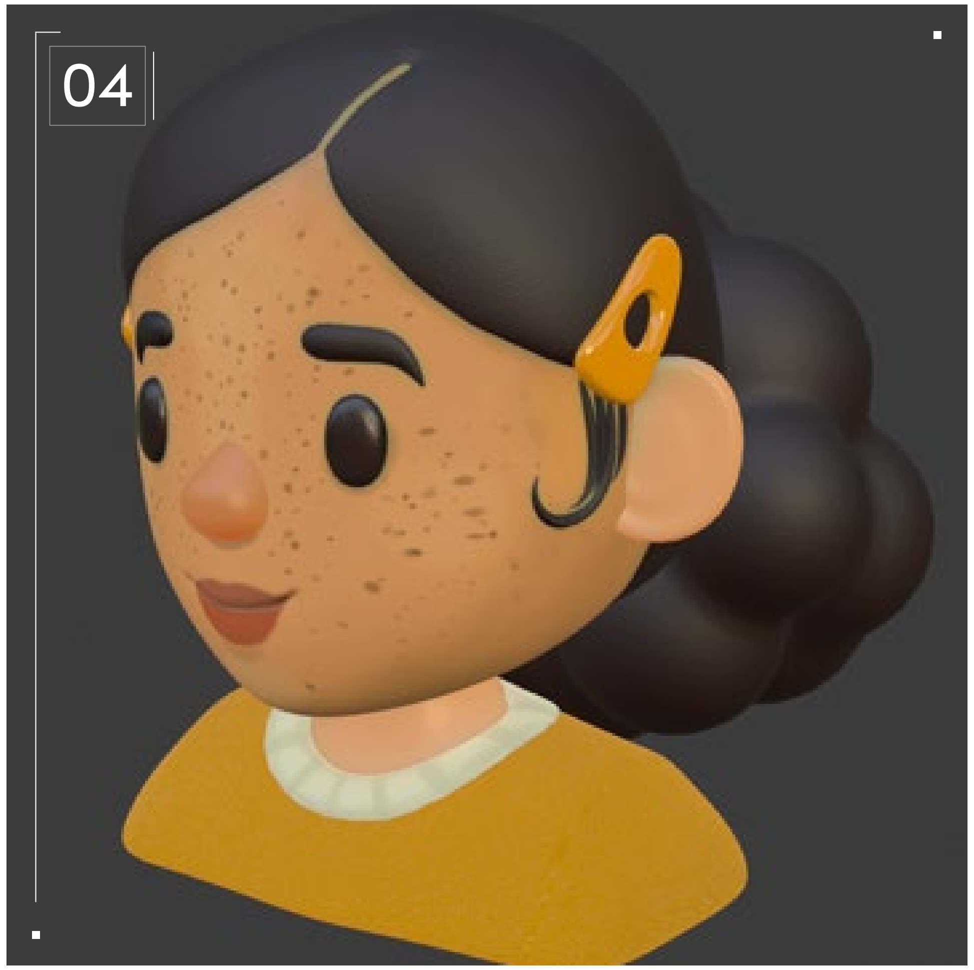
Next, I duplicated the scalp and used this as a base for the swoop by adding a solidify modifier. To make the parting more defined I added extra geometry to the head solely around the part, and extruded/sculpted that.

I created the Puff Ponytail using the Metaball Primitive. It could also be done with spheres/round cubes/etc. — any manner of techniques! What matters is keeping the silhouette interesting and believable. The hair was then remeshed and retopologized like normal.



I created baby hair using regular curves with a basic taper profile. Please use multiple references for baby hair.

Not all baby hair is the same nor is it interchangeable across all styles, as you can see with the variety of baby hair in all of our models.



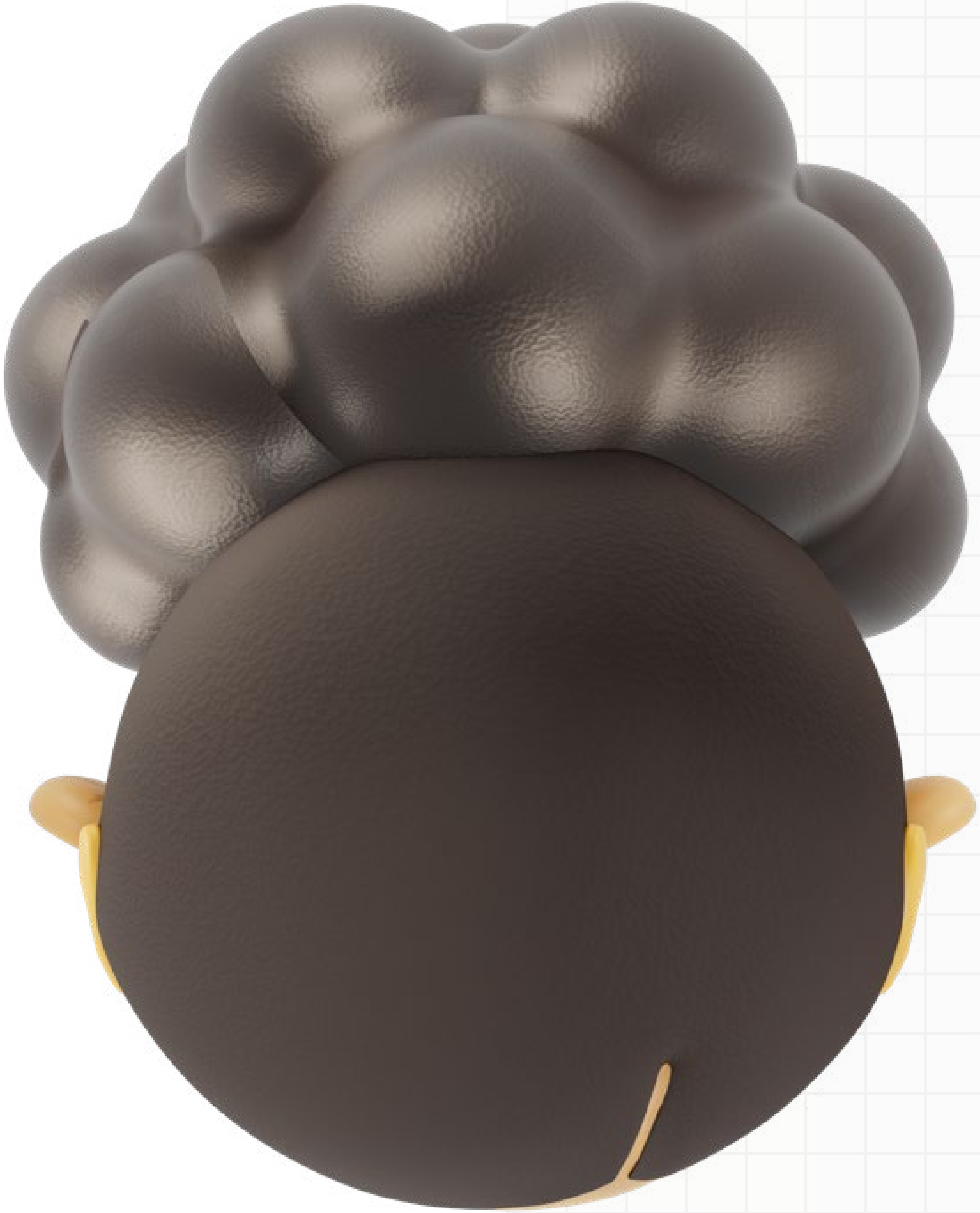
I kept texturing simple and toy-like. I used procedural textures to give a subtle gradient and texture to the hair. The face was hand-painted in Blender. I mainly work with procedural textures or just block out colors for 3D print. However, it was fun and I plan to be doing more of it in the future!

Overall, this style was not particularly difficult to create! All you really need is a change of focus and some time to experiment. I really believe experimentation and stylization are the keys here. It's not always about breaking your toolset; it's what you choose to use those tools on and how intentional you are about the important details.

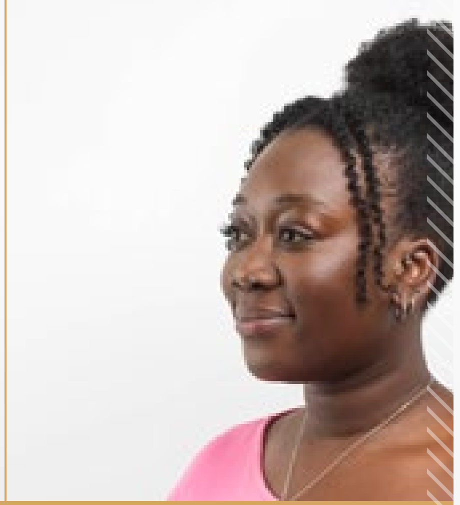
Technical Watchouts

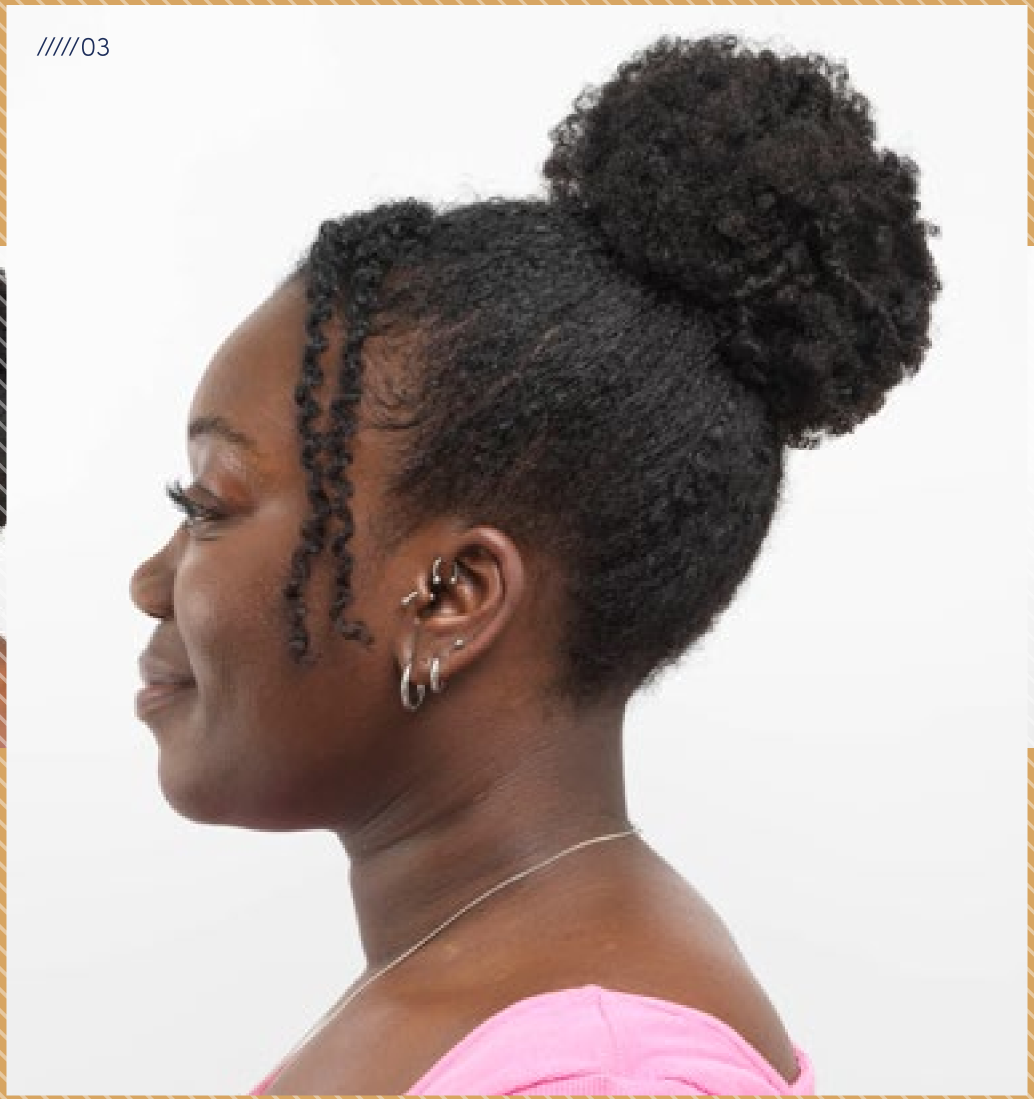
There are a few key points to this style: making the swoop look like it is hair brushed and stretched along the scalp into this style, the placement of the part, the stylization of the baby hair, and the stylization of the Puff in the back. I wanted to highlight the swoop and baby hair, so I went with a simple stylization of the back Puff. Despite its simplicity, I find it very charming and I hope that I see more bubbly Afro stylization where applicable.

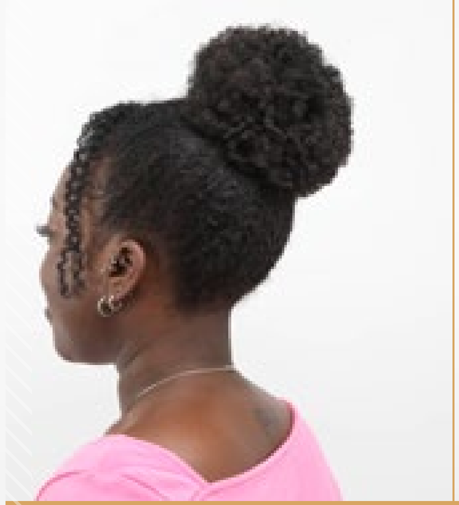
Another note: I rarely see baby hair or sideburns styled correctly on 3D characters. They should be delicate and make sense for each particular hairstyle. There is not a one-size-fits-all baby hair or sideburn styling. There are multiple ways to express these styles and they each add a bit of flair and expressiveness to the character! A library of baby hair and sideburn examples will definitely be helpful but make sure to keep the overall hairstyle in mind.



Afro Puff with



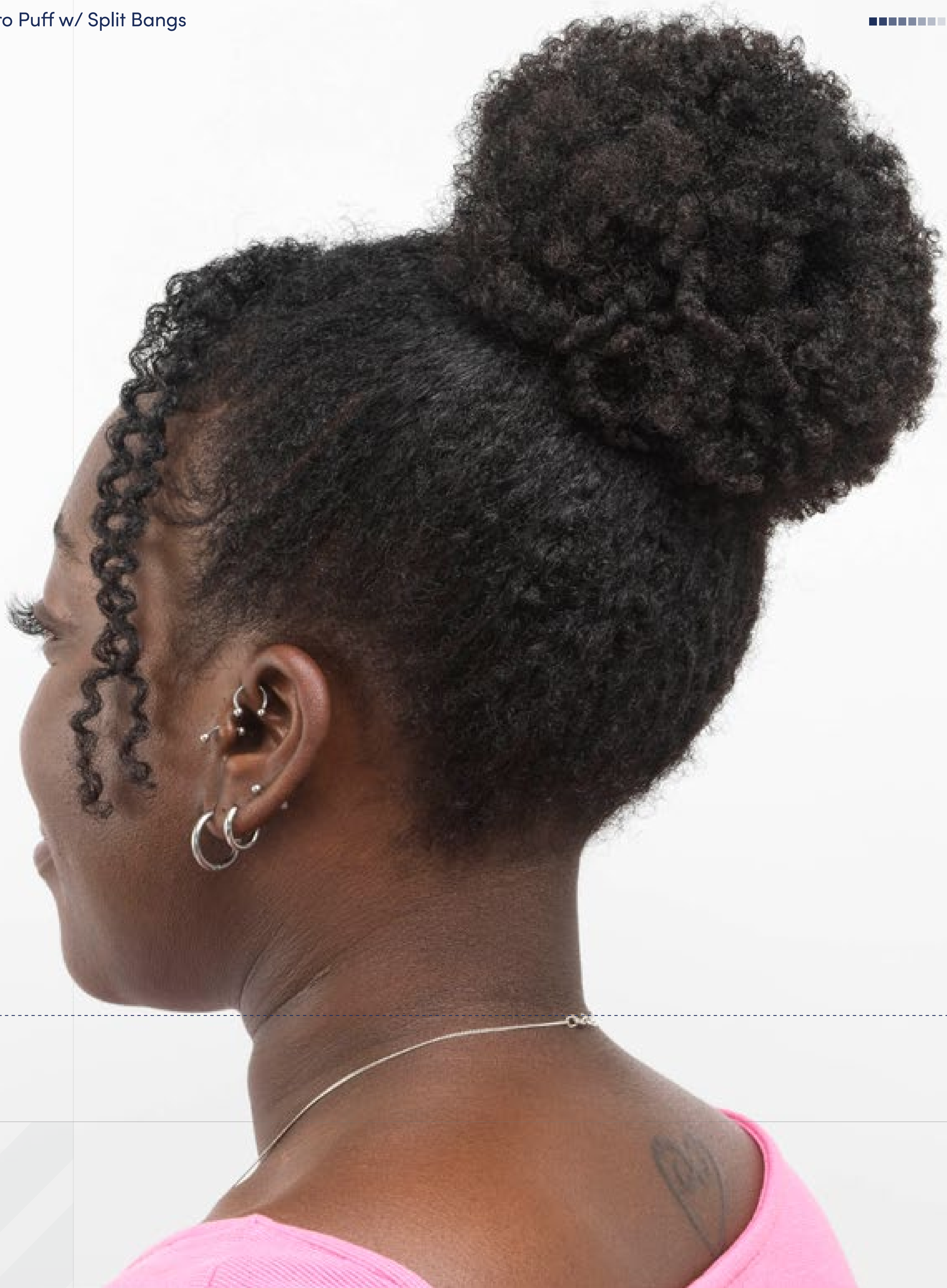




Split Bangs

Download Images

// Afro Puff w/ Split Bangs

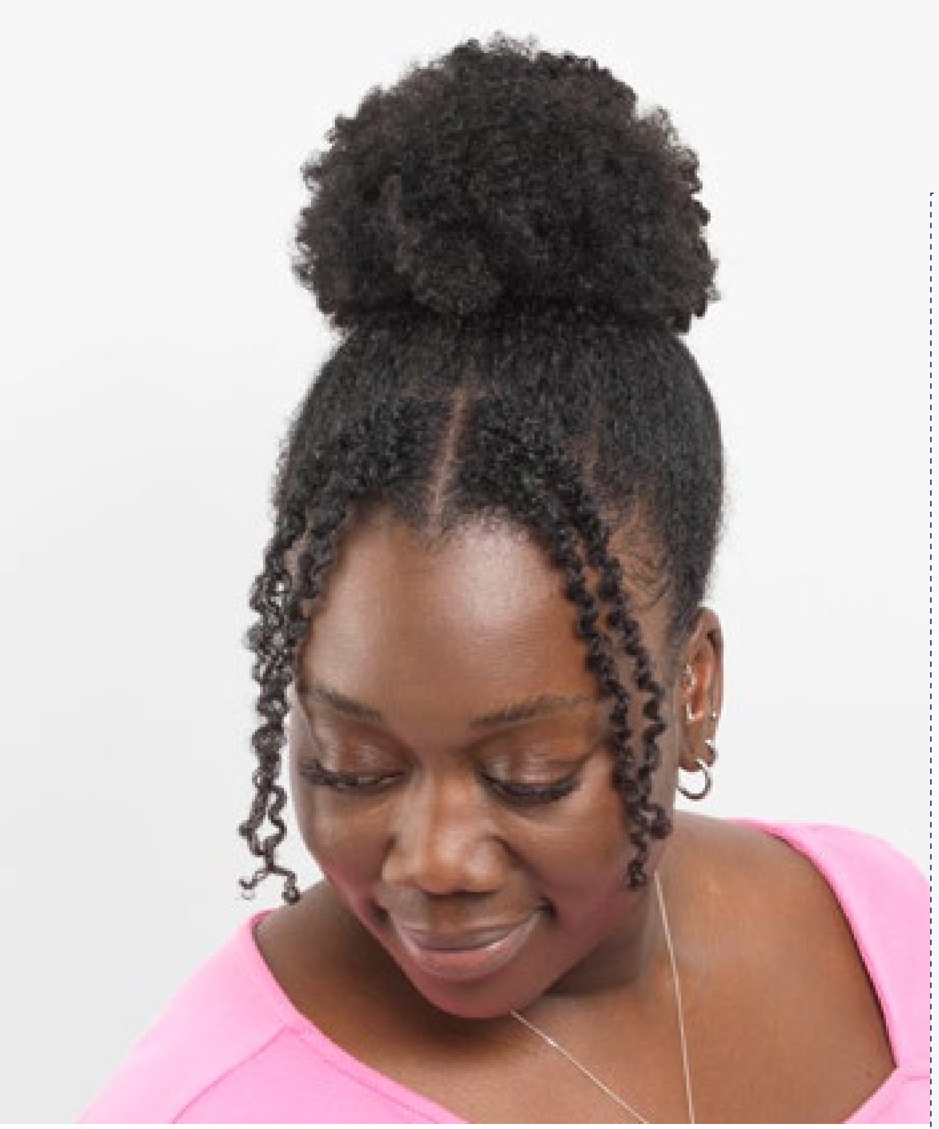


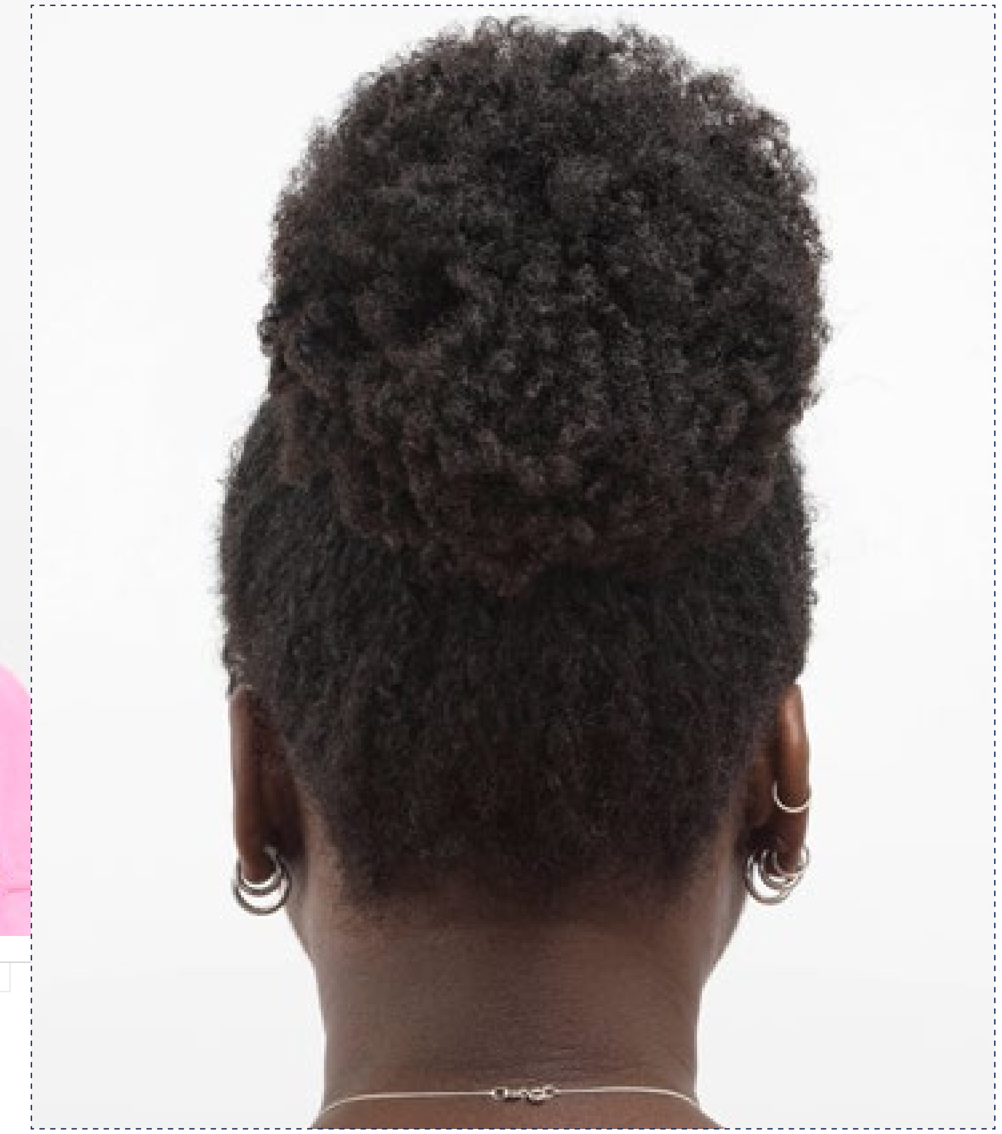
// Hairstyle

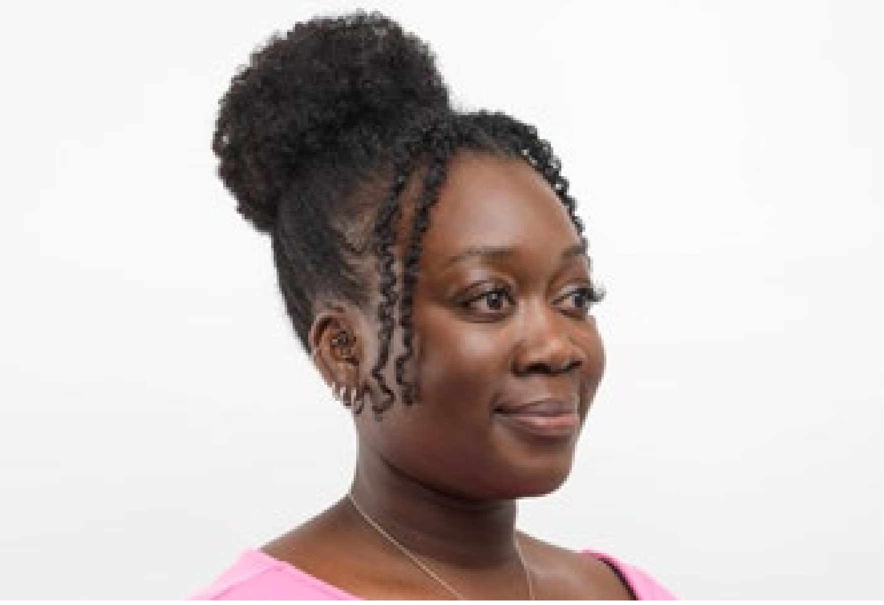
Afro Puff with Split Bangs

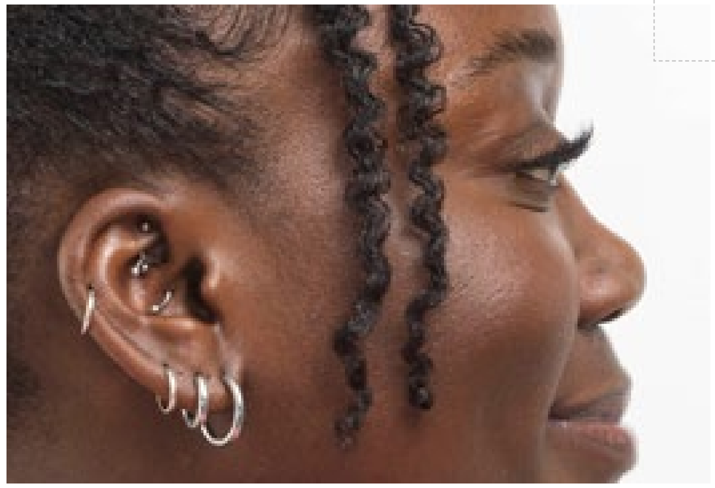
Style Background: Like the ponytail, the Afro Puff is named after the shape some hair textures take when pulled back and tied at the base of the scalp. Because dense curly-coily hair tends to defy gravity, the Afro Puff is simply what some hair does naturally when gathered together. They can be styled in different ways, but the classic look is to have your hair split down the middle with a large Afro Puff on either side of your head. People may choose to have more than two Afro Puffs or have a zigzagging parting to add their own unique spin to this iconic style.

Download Images

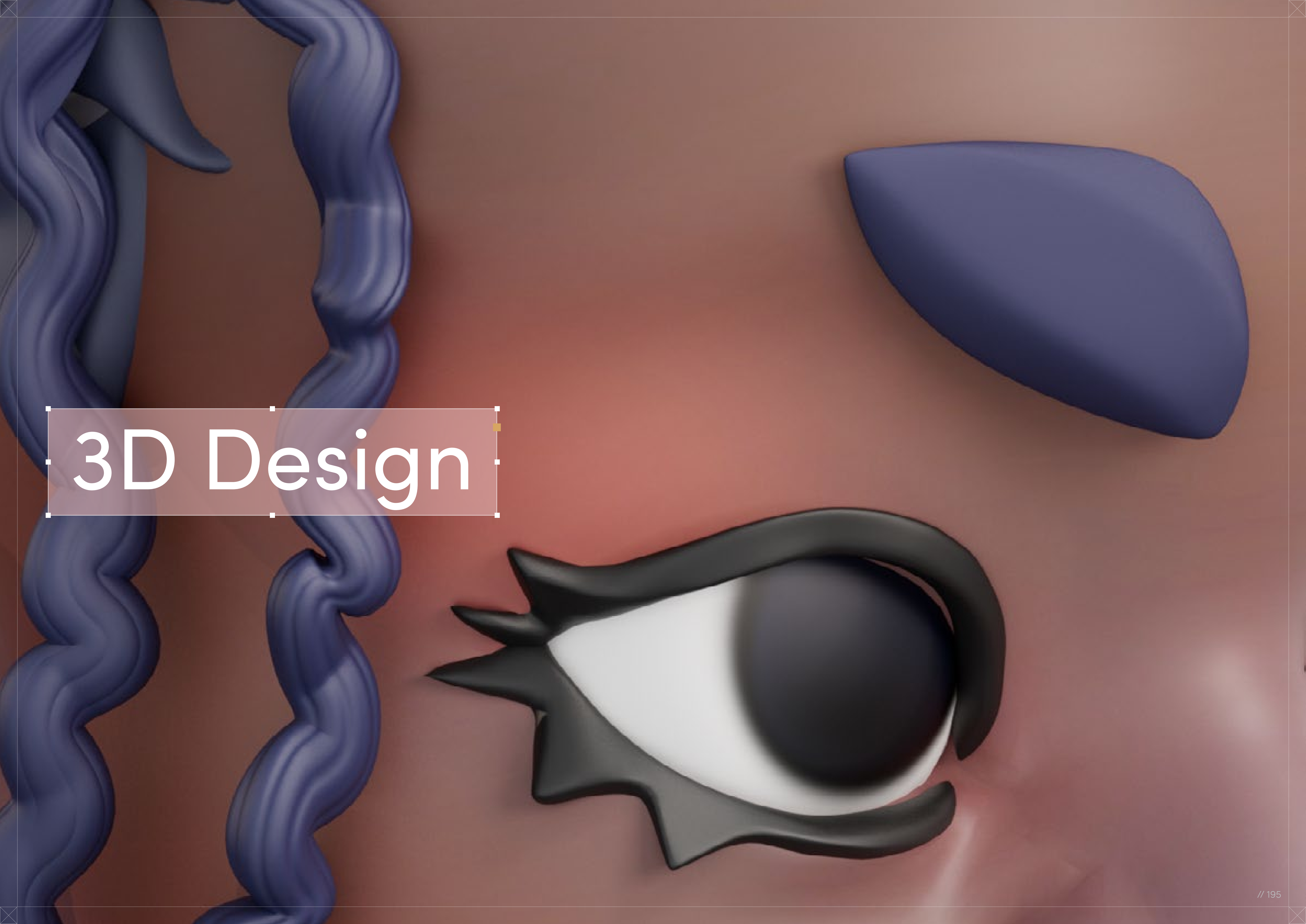




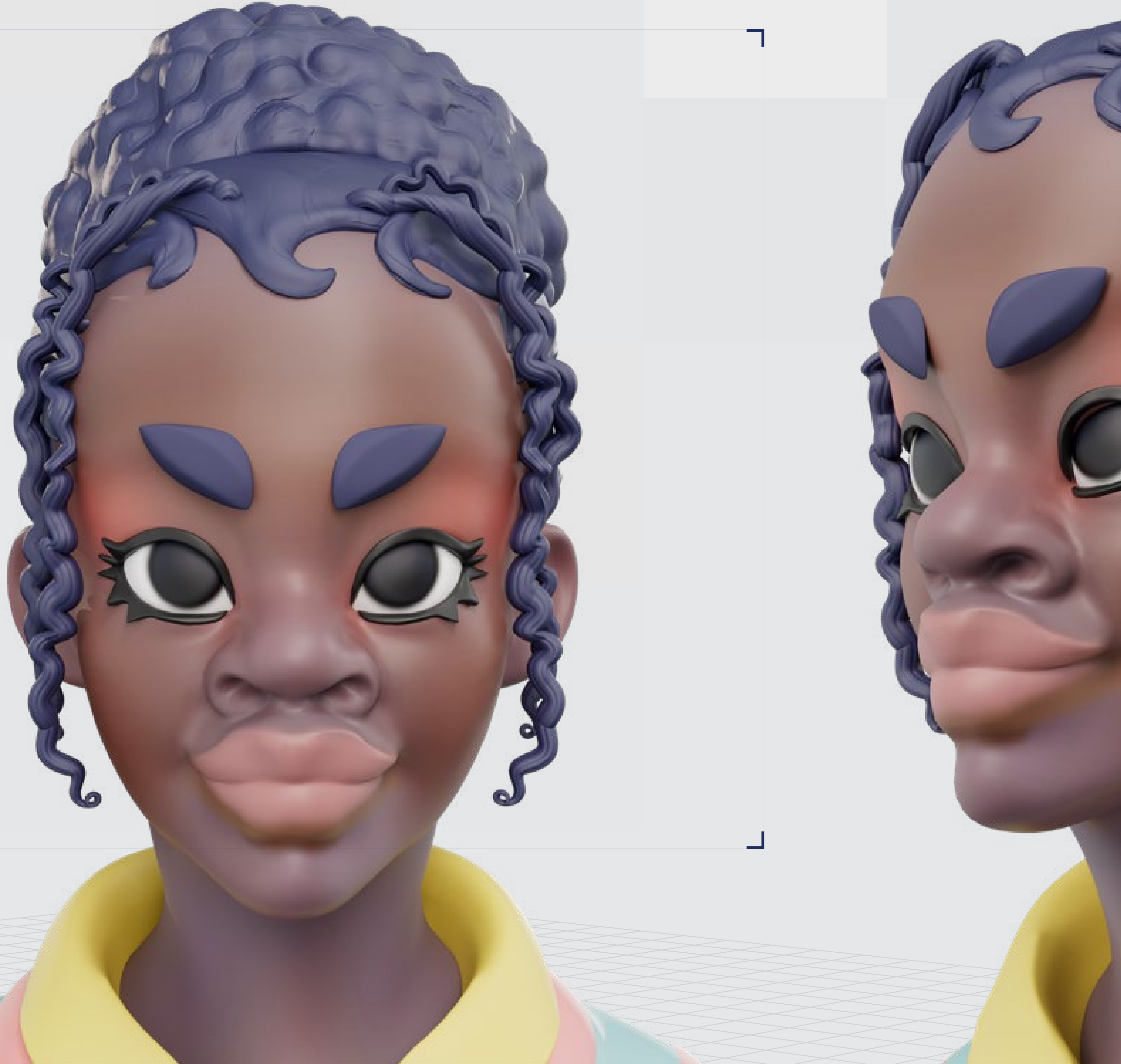




3D Design



Afro Puff with Split Bangs

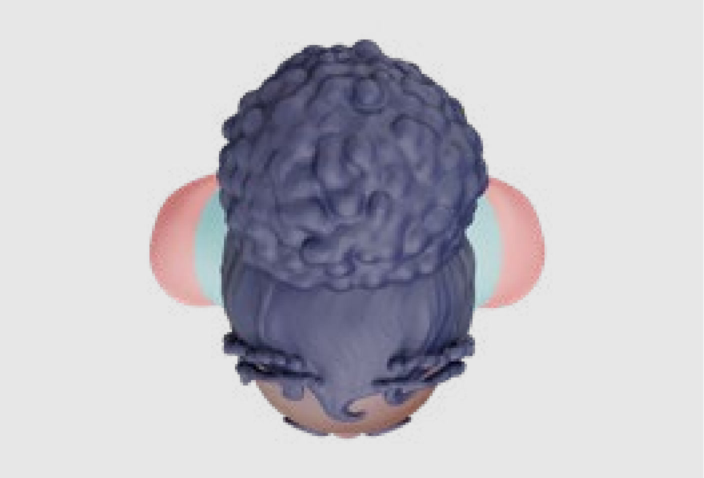


Digital Artist:
Adésayo Adéoyé (Ade)

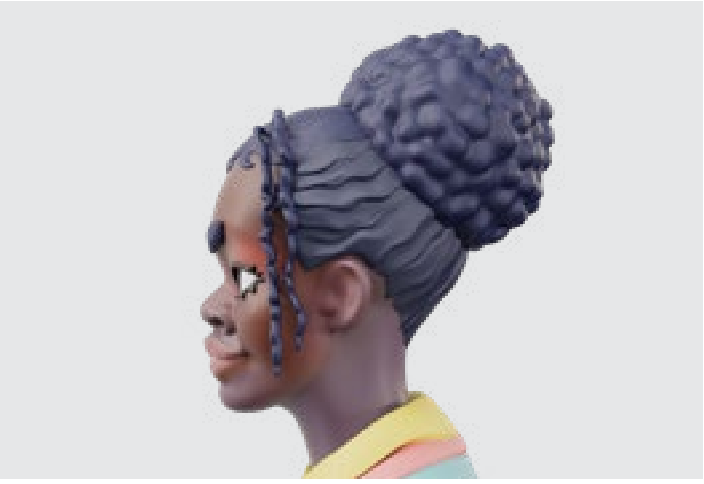
// Afro Puff with Split Bangs

3D Design

Curl Type Modeled: **4B**

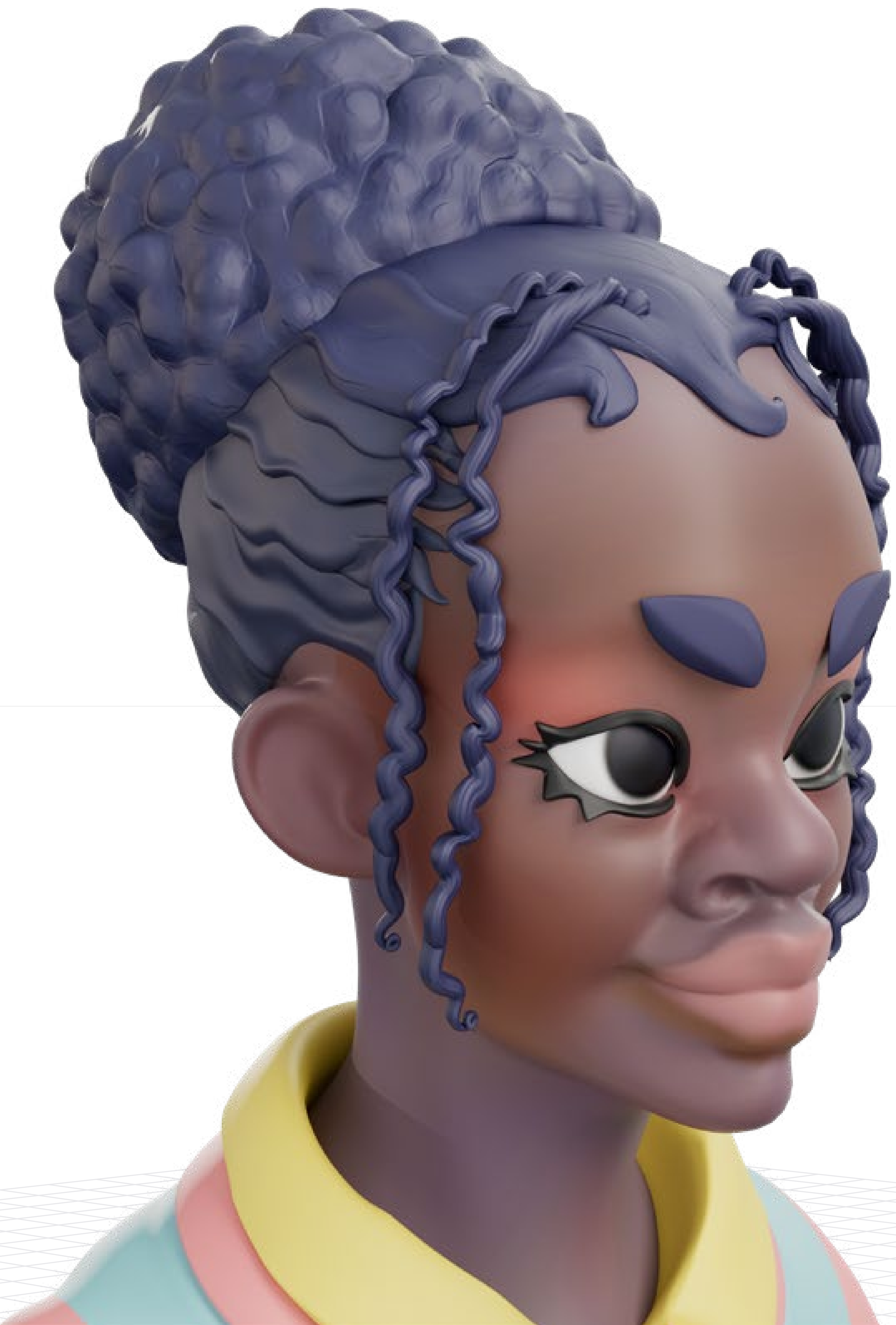


// 01



// 02

Download 3D Render

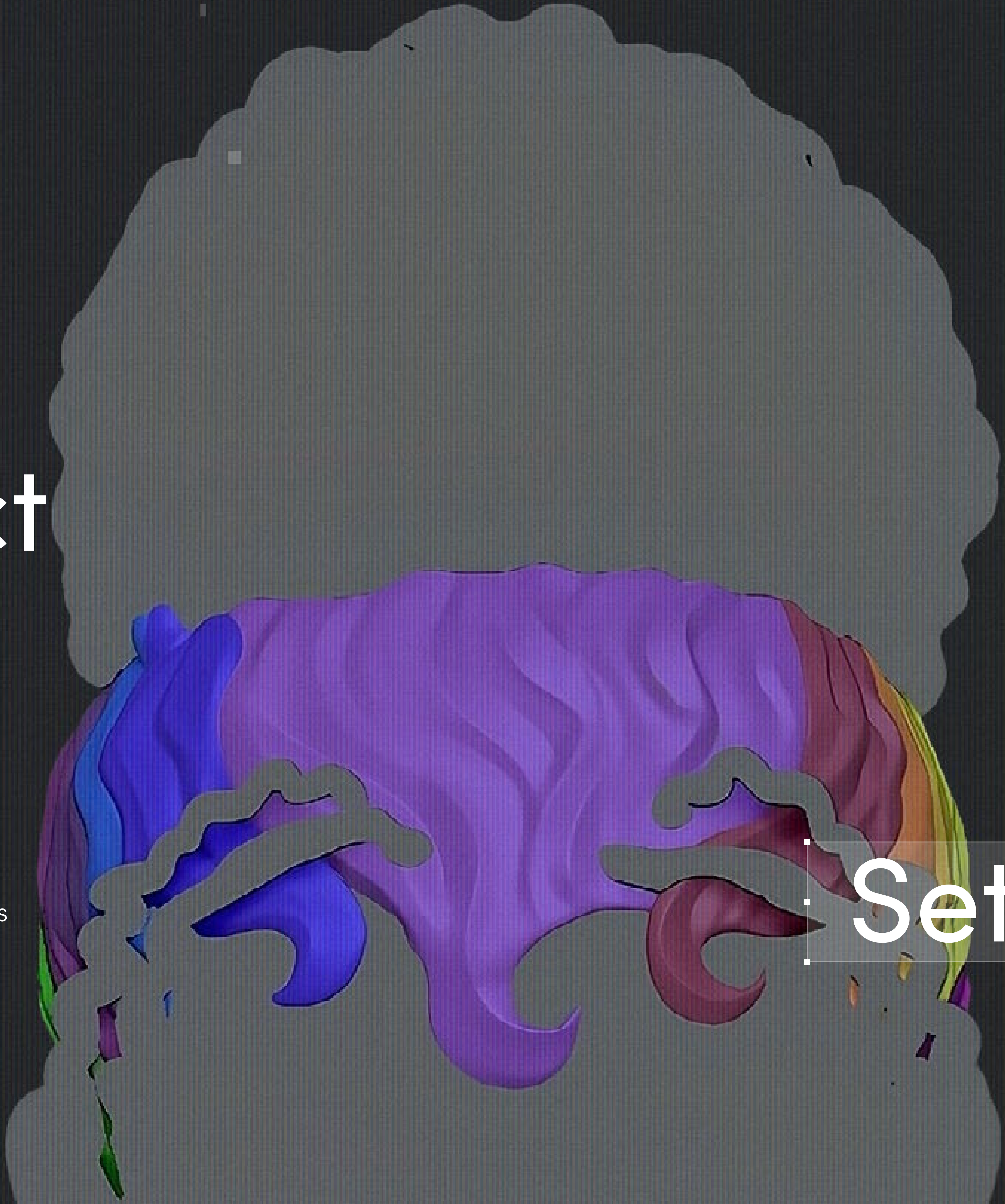


Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

Project Settings

// Afro Puff with Split Bangs



// Project Settings

Program Guide

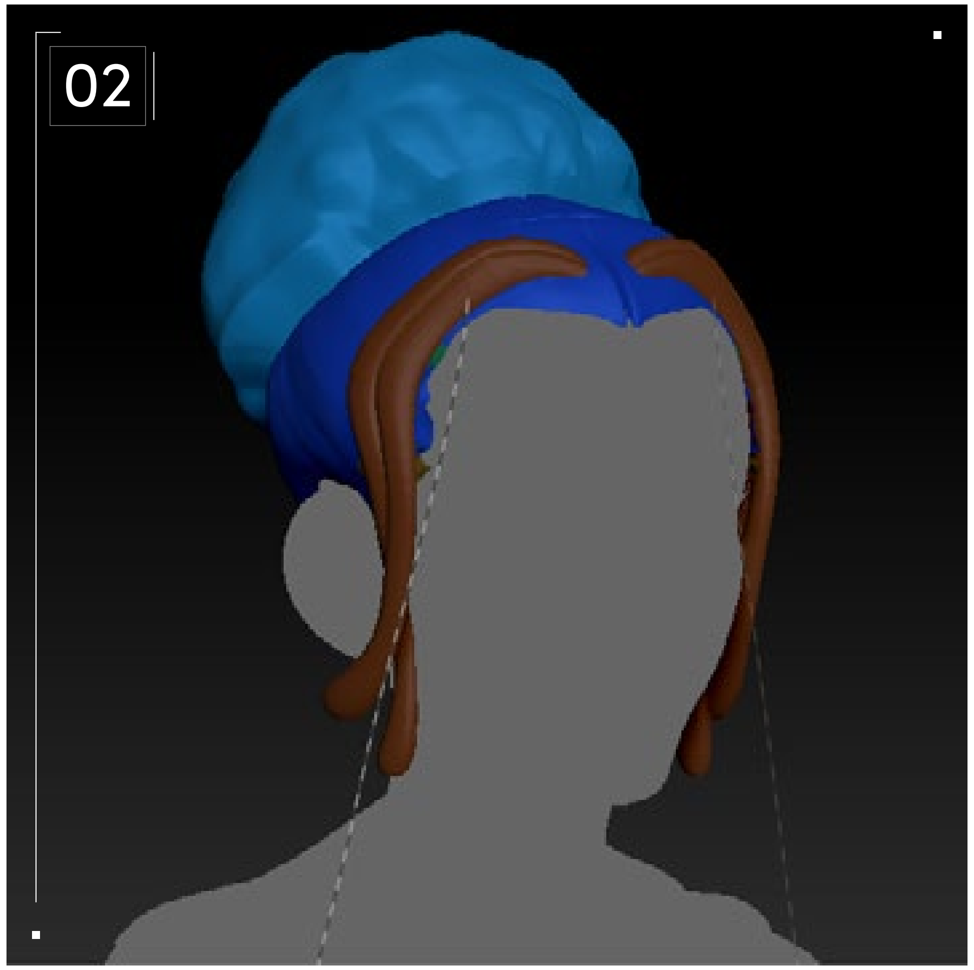
Software: ZBrush, Blender
Render Engine: Adobe Substance Painter
Additional Plug-Ins: Marmoset Toolbag

Step-by-Step Guide

01

I start every project with as many references as I can find. In this case, we were given images of models from a good range of angles. I used those as the base of the hair; it guided my early blocking-out stages. Then I assembled other references, both fictional and real, to guide my more stylistic choices.

02



// Blocking

First, I needed to block out the hair model using ZBrush. At this stage I'm aiming to get the general shape of the hair and see how the different sections interact. I started this style with a Dynameshed sphere set to the lowest resolution to make the model easier to manipulate.

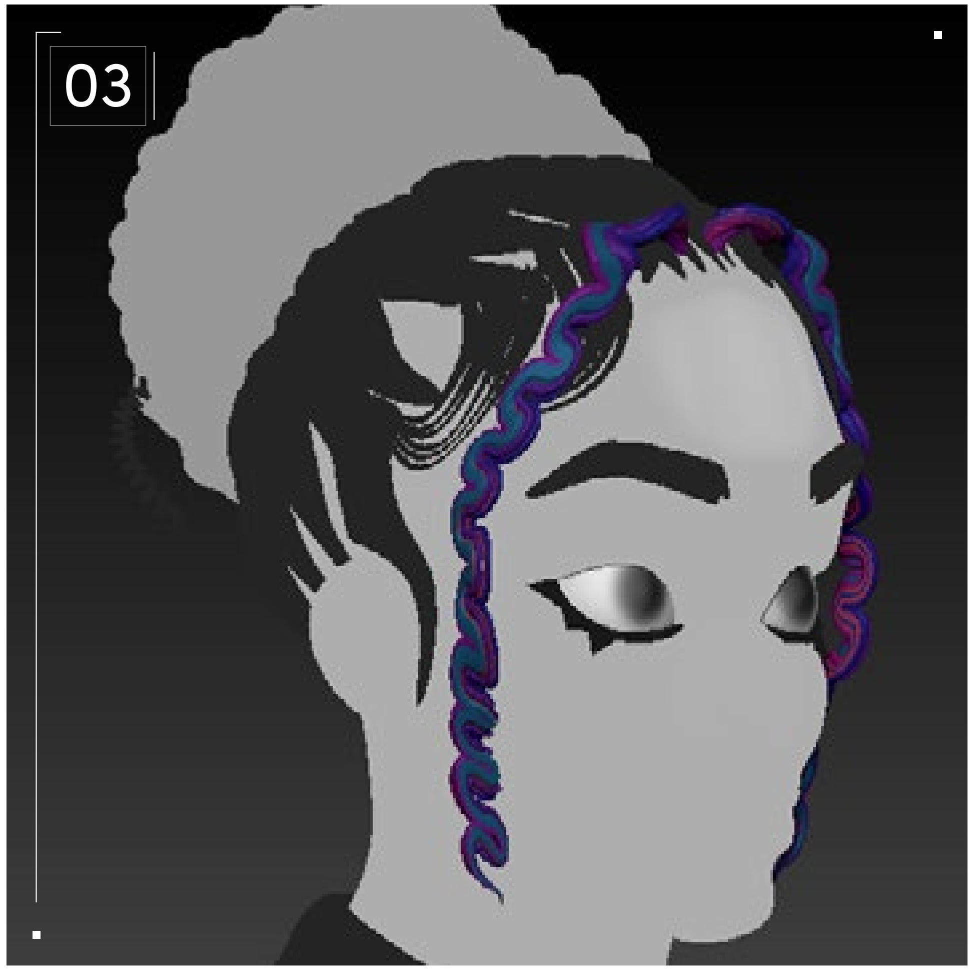
I broke the style into separate pieces, starting with a sphere, and used the Move brush and the ClayBuildup brush to define the basic forms, the Puff, the curls, and the hair between them.

ZBrush has a few ways of doing this, but I usually prefer to mask the areas I want to separate and then select "polygroups from mask" in the Polygroups subpalette. Then I started blocking out some of the details, such as the baby hairs. I had never tried to do them on a model before, but I always really liked how they looked so I was excited for the challenge. I ended up with quite a few different iterations of the baby hairs for the model. My style is usually pretty simplistic, but I wanted to try to capture some of their intricate details, which caused a pretty big dissonance in the sculpt between the baby hairs and the rest of the more simplistic and cartoony aspects of the model. In the end, I realized I could either make the rest of the model more detailed or I could simplify the baby hairs. I chose the latter.

I also split the Afro Puff into more parts. I liked how the model's hair looked, the way her hair was taut as it led into her Afro Puff, while maintaining a very slick, wavy texture. I used the ZModeler brush to bevel loops on the base mesh and then I cut out the faces. I then went to the Polygroups subpalette and used "auto groups." Then I did some more refining and got a decent base mesh for the next phase.

Next, I tried something a little new (to me at least) and used ZBrush's 3D layers. When I began sculpting the detail, I made a new layer in the Layer subpalette and made sure it was recording — although I think it does so automatically. After my issues with the baby hairs, I became pretty worried about making the hair a little "too much" in places and messing up the cohesion of the piece. So, with layers I was able to increase or decrease the strength of the brush strokes I made while sculpting the details, and that allowed me to tweak large parts of the mesh all at once on the fly.

Step-by-Step Guide



// Detailing

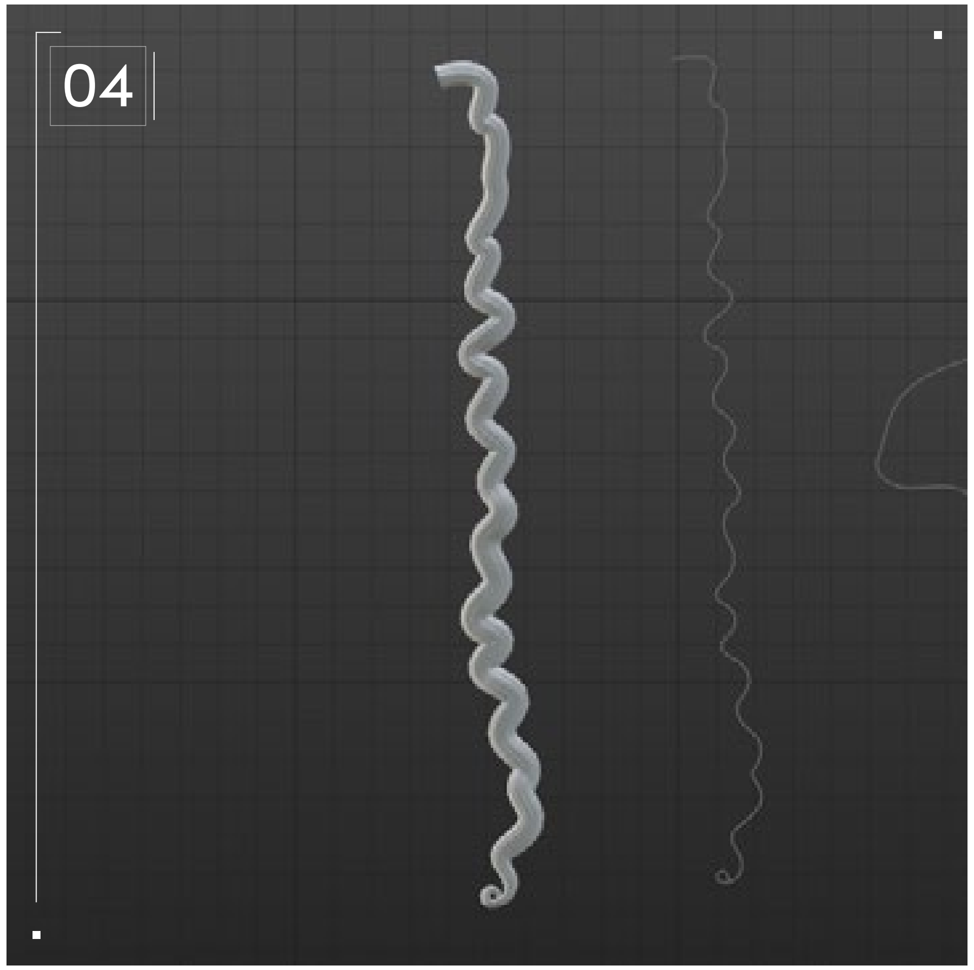
The Puff part of the Afro Puff was handled in the same way as the Fade on page 137.

For the hair between the Puff and the curls, I tried to sculpt some curly grooves into them and made the boundaries of each section less straight. This was achieved with the Dam Standard brush as well as the pinch brush. I would usually sculpt the grooves with the Dam Standard brush and then refine those shapes with the pinch brush. While using both brushes, I made sure to turn on Lazy Mouse so I could get smoother shapes. I then used the Move Topological brush to adjust the different sections individually, making them wavier and overlapping. After I was satisfied with the amount of variation in the sections, I then used the 3D layers again to adjust the strength of the grooves I had sculpted until I was satisfied with how they looked.

The curls were another place where I did quite a bit of iterating. At first, I sculpted them a bit more like how I usually sculpt braids, but this approach didn't work, and also for this model I didn't want things to be too busy. An approach that needs quite a few separate meshes melded together wouldn't be sufficient. I couldn't get the shape I wanted for the curls in ZBrush, so I went into Blender instead. In Blender I made a cube, then I merged all the vertices into one and placed that at the base of the curl on the reference.

I then extruded from that single vertex and repeated this as I tried to follow the shape of the curls. I did this with a few different angles of the curls and then selected the one that I liked the most. I then converted the mesh into a curve and extruded a curve circle along the curl curve I had just made. When I was satisfied with the shape, I then exported it back to ZBrush, where I used the bend curve modifier in the 3D gizmo to shape it to my liking and in line with the reference.

Step-by-Step Guide

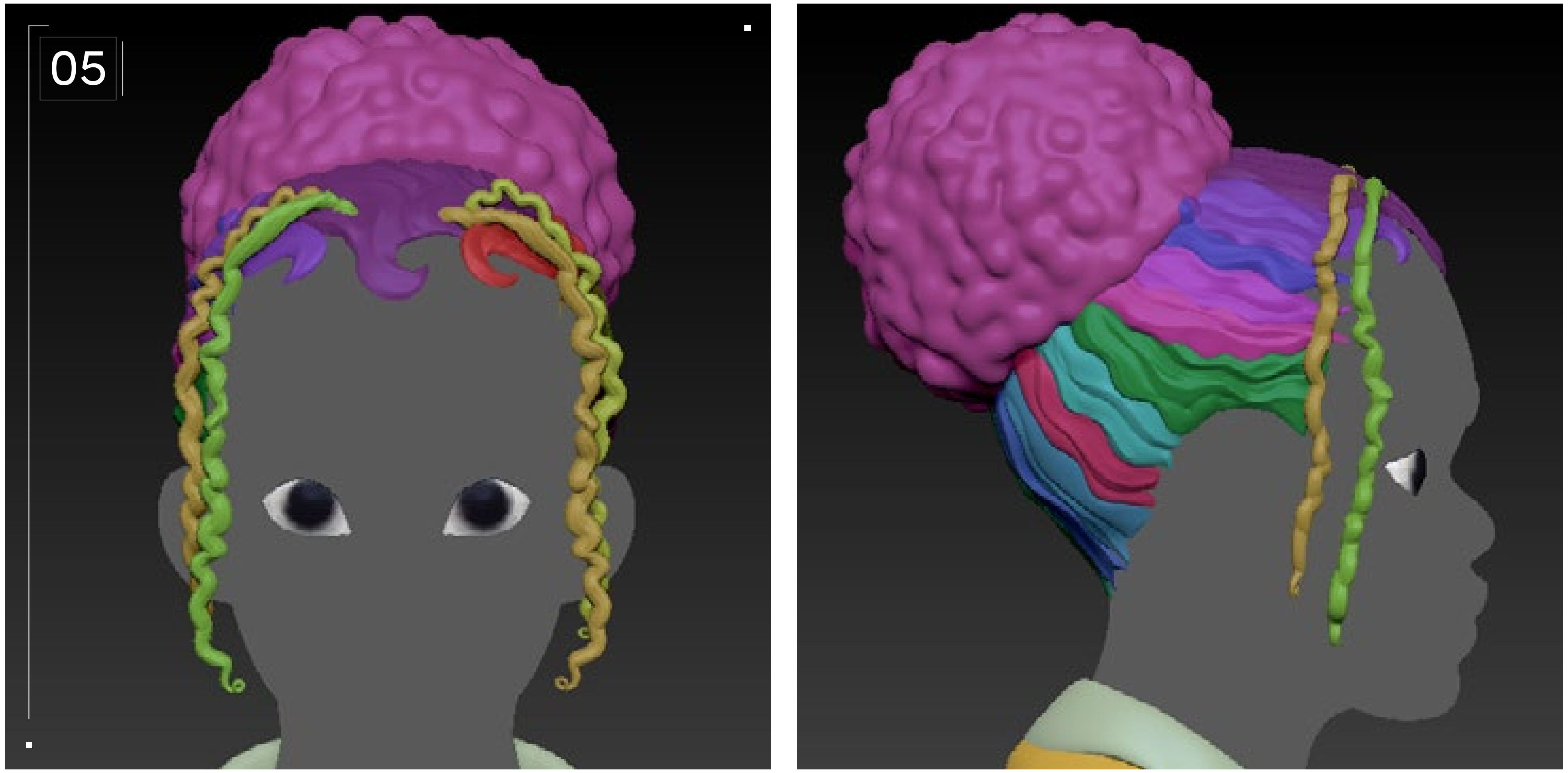


// Baking

When retopologizing my model I made sure to keep them as separate meshes so that I could bake each part without any intersecting geometry between the different parts. I then used the decimation master plug-in that's the default in the ZBrush plug-ins menu on the high-poly version of my meshes. As some of these ZBrush files can get pretty wild in terms of polycount, it helps to decimate them first as this allows you to retain detail while cutting down on geometry.

I then exported my high-poly and low-poly meshes, assembled them, and named them appropriately with the "_high" and "_low" suffixes. I exported them from Blender as OBJ and baked them in a Substance Painter. I baked all the usual maps — AO, Normal, for the final texture and curvature, group id, height, etc. — to help with generating masks in Substance Painter.

Step-by-Step Guide



// Textures/Materials

I textured the style in Substance Painter. The model had cooler tones for her skin, so I chose a cool but deep indigo. The texturing process was pretty simple. I painted my based colors, used the blur filter to make them blend better, and I tried to use masks to create some subtle variation in the roughness and color of the hair.

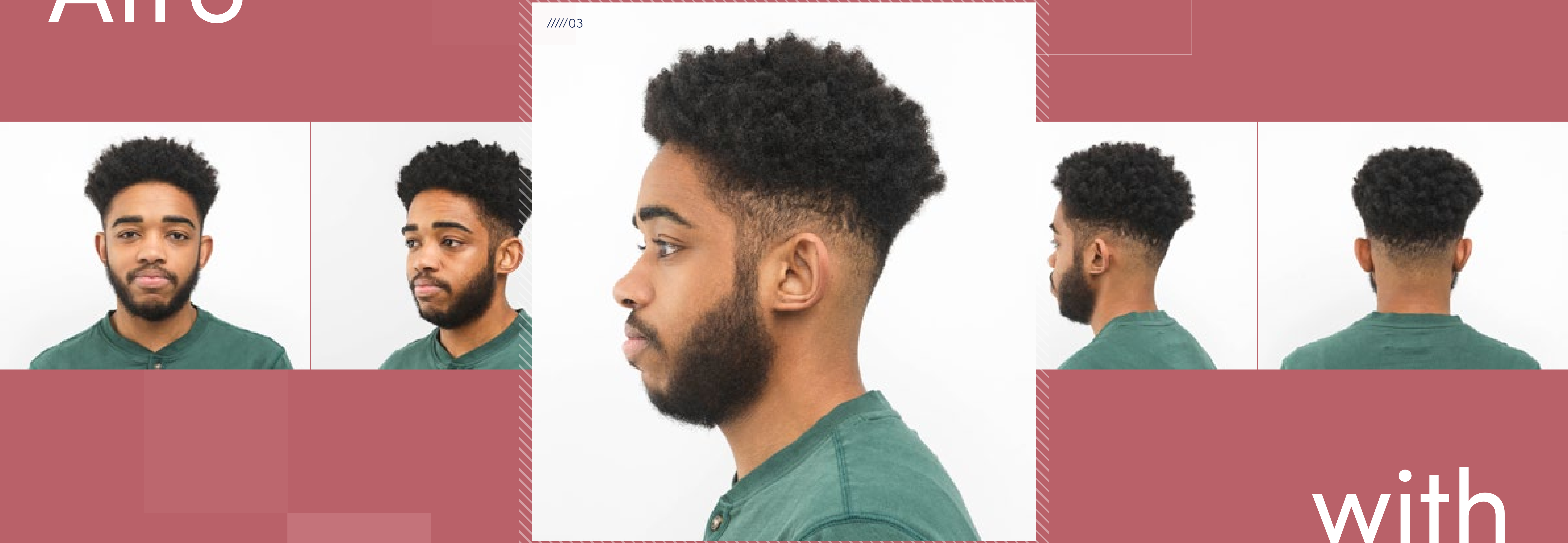
After tweaking values to my liking, I was satisfied with the colors. In Blender I tried to add a subtle gleam to the hair, so in the node tree of the material, I had a mix node as my roughness. The node was mixing the hair's roughness with a dark grey, and the alpha was driven by a Fresnel. I just tuned that to my liking. This was done in Blender but it's a pretty basic material and I'm certain it can be replicated in basically any 3D software with node-based materials.

Finally, all that was left was to render the model with the hair. I learnt quite a bit during this process. I had never worked on a style like the Afro Puff before and I also haven't met many people with that hairstyle in real life, so it was an enlightening experience.

Afro

Afro

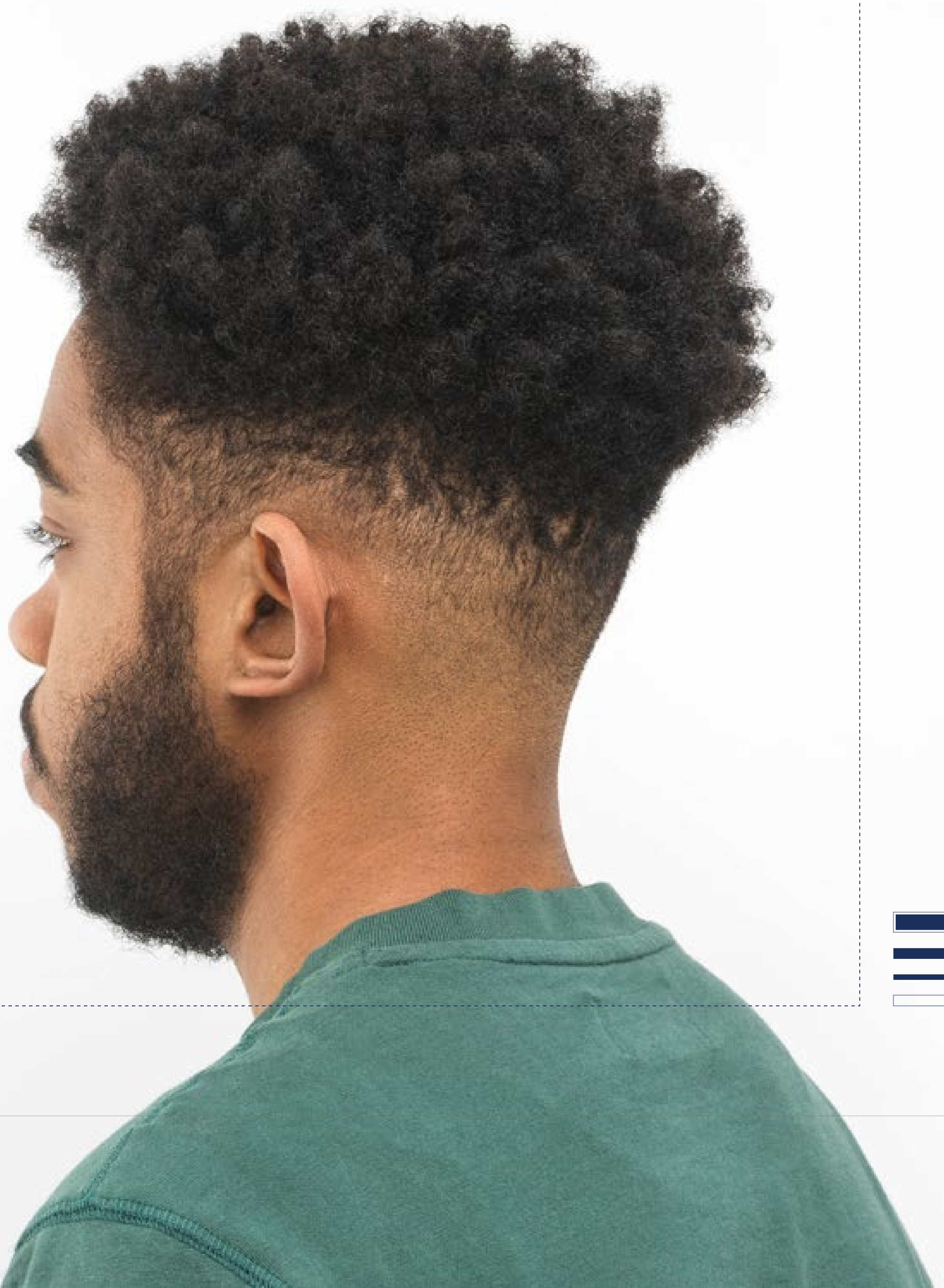
Afro



with Fade

Download Images

// Afro w/ Fade

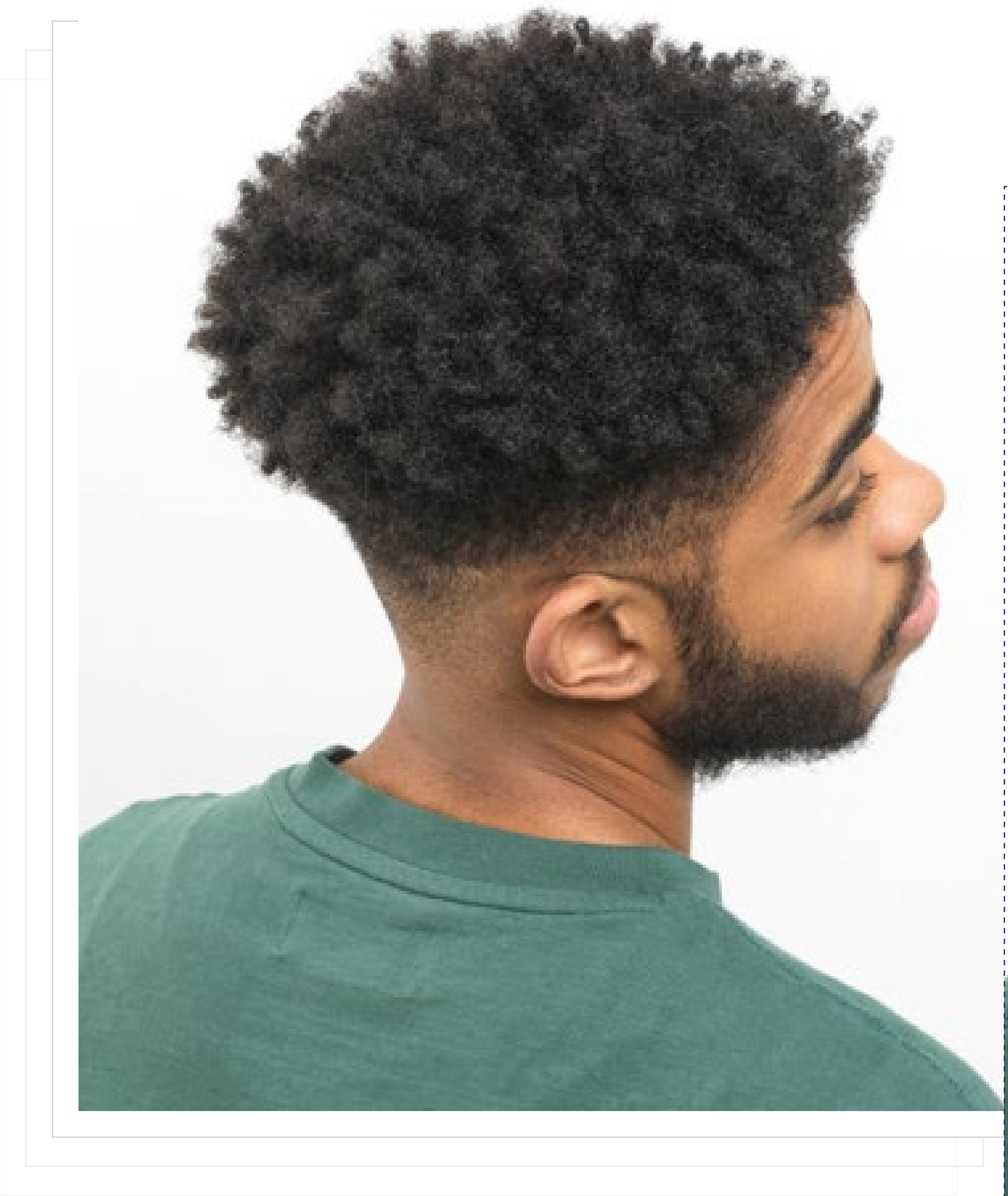


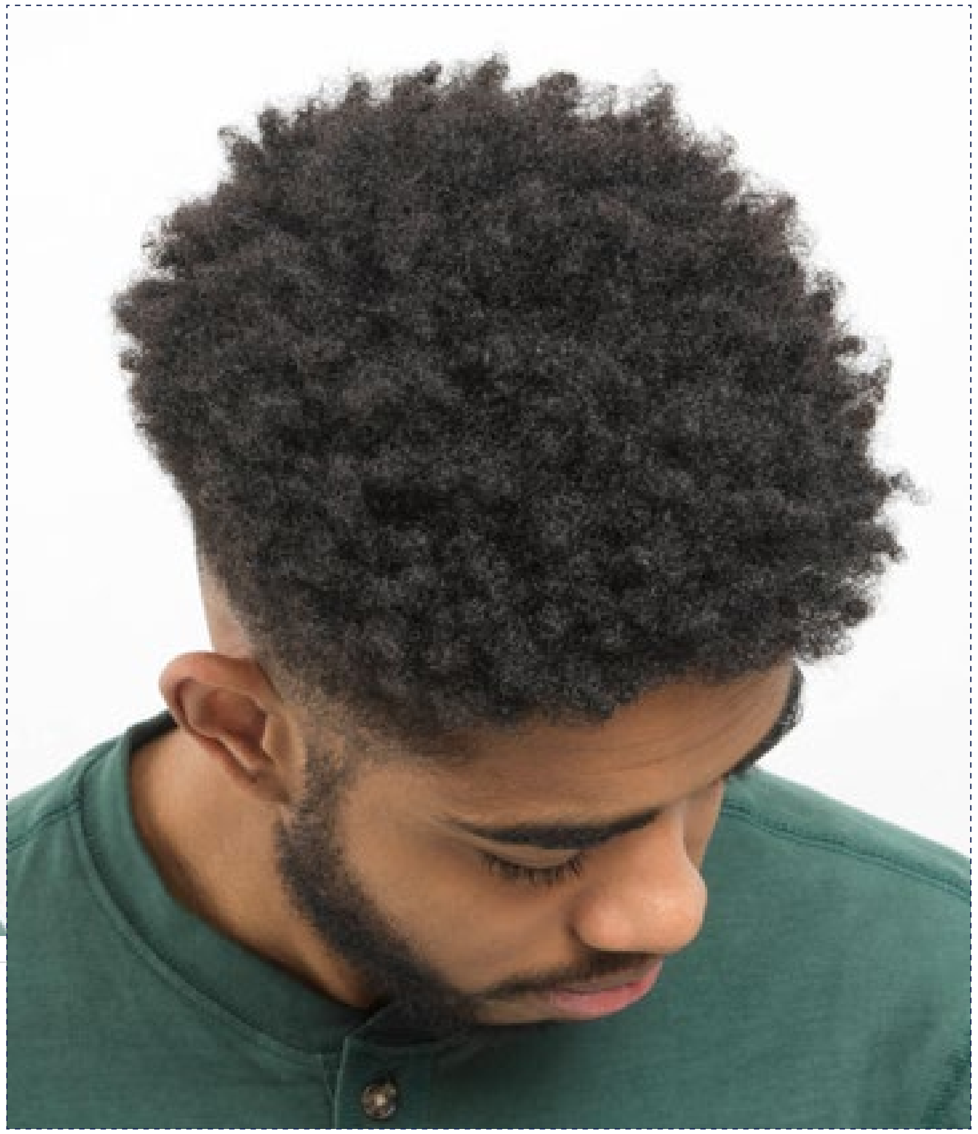
// Hairstyle

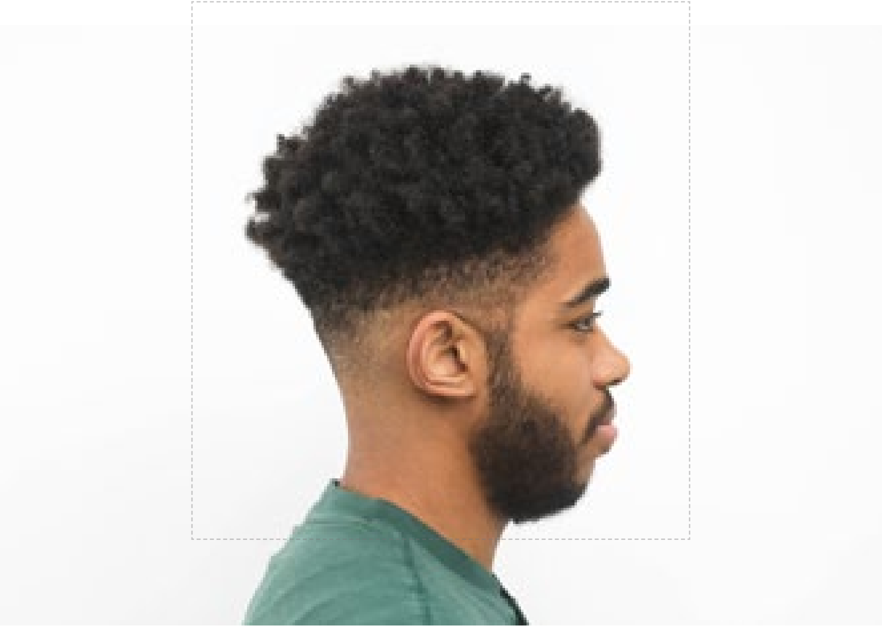
Afro with Fade

Style Background: This versatile haircut can be styled in a number of ways and works for a range of curly or kinky hair types and lengths. The Fade frames the face with sharp, angular lines while on top the afro can be trimmed and shaped to create dynamic looks.

Download Images

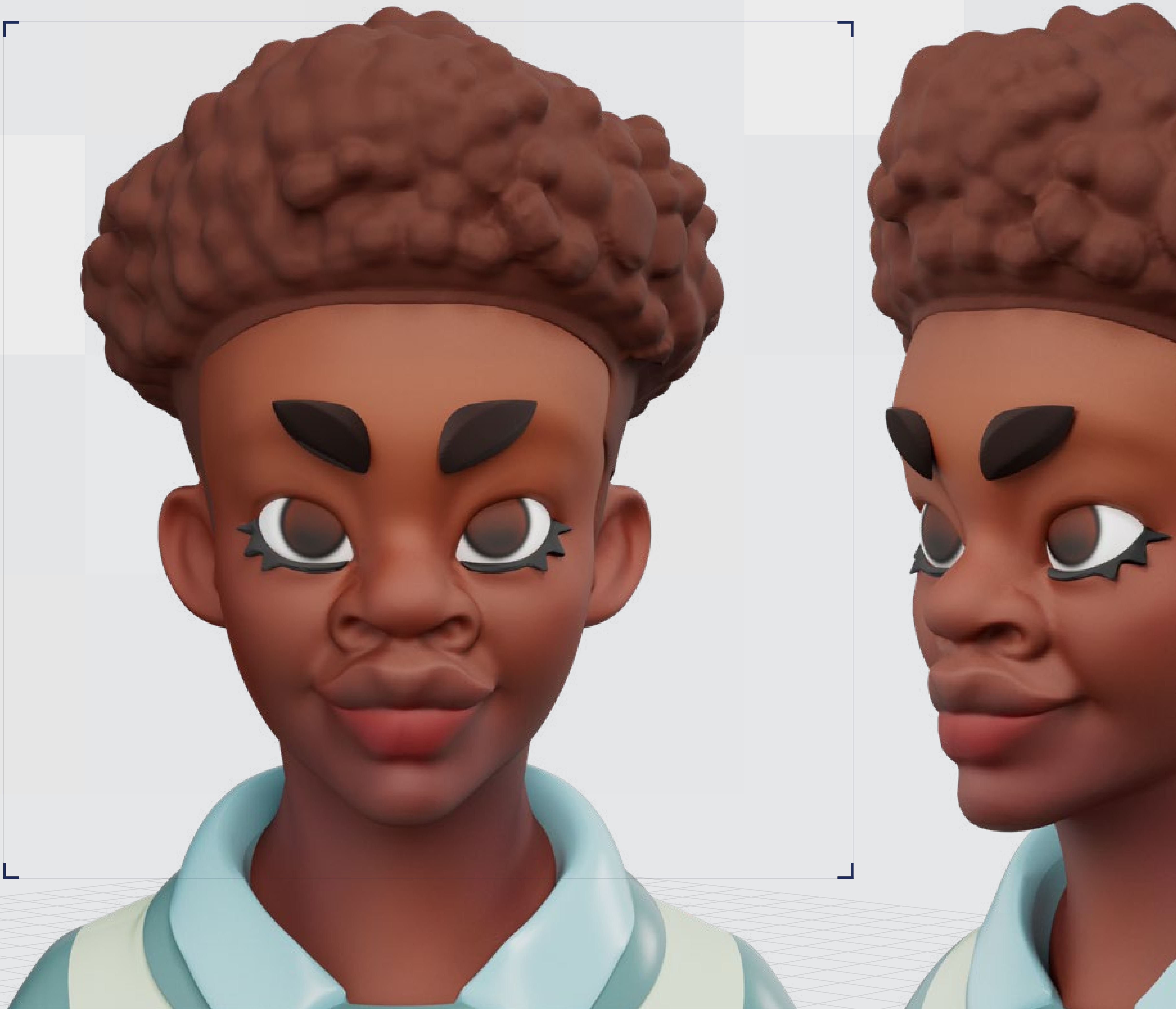






3D Design

Afro with Fade

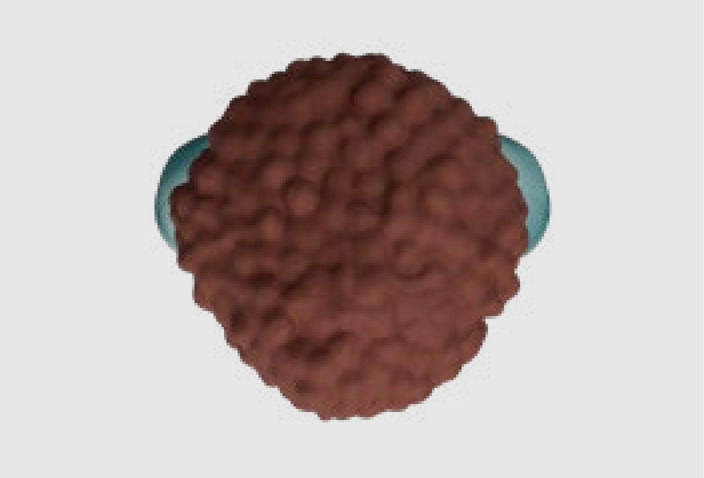


Digital Artist:
Adésayo Adéoyé (Ade)

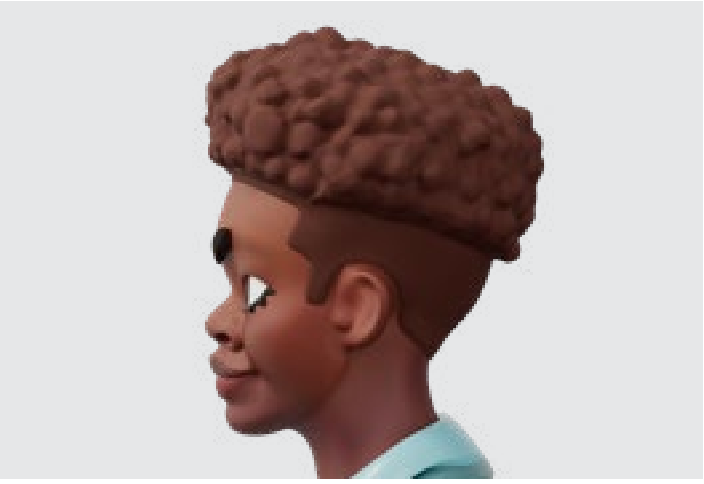
// Afro with Fade

3D Design

Curl Type Modeled: **4A/4B**

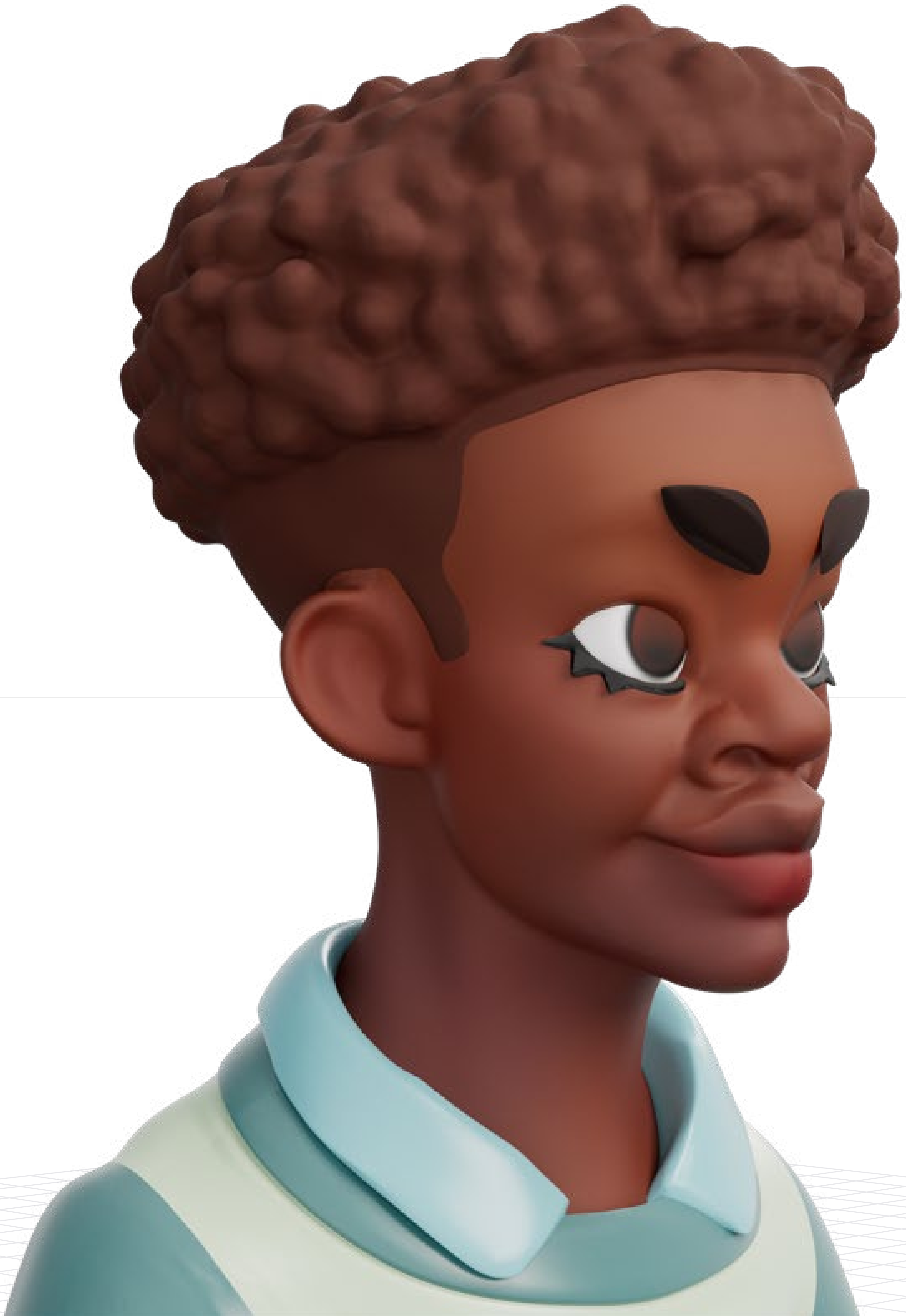


// 01



// 02

Download 3D Render



Backfaces:	N/A
Smoothness:Instance:	N/A
Display Layer:	N/A
Distance From Camera:	Render_LYR
Selected Objects:	65.994

//

Project Settings

// Afro with Fade

// Project Settings

Program Guide

Software: ZBrush, Blender
Render Engine: Adobe Substance Painter
Additional Plug-Ins: Marmoset Toolbag

// Project Settings

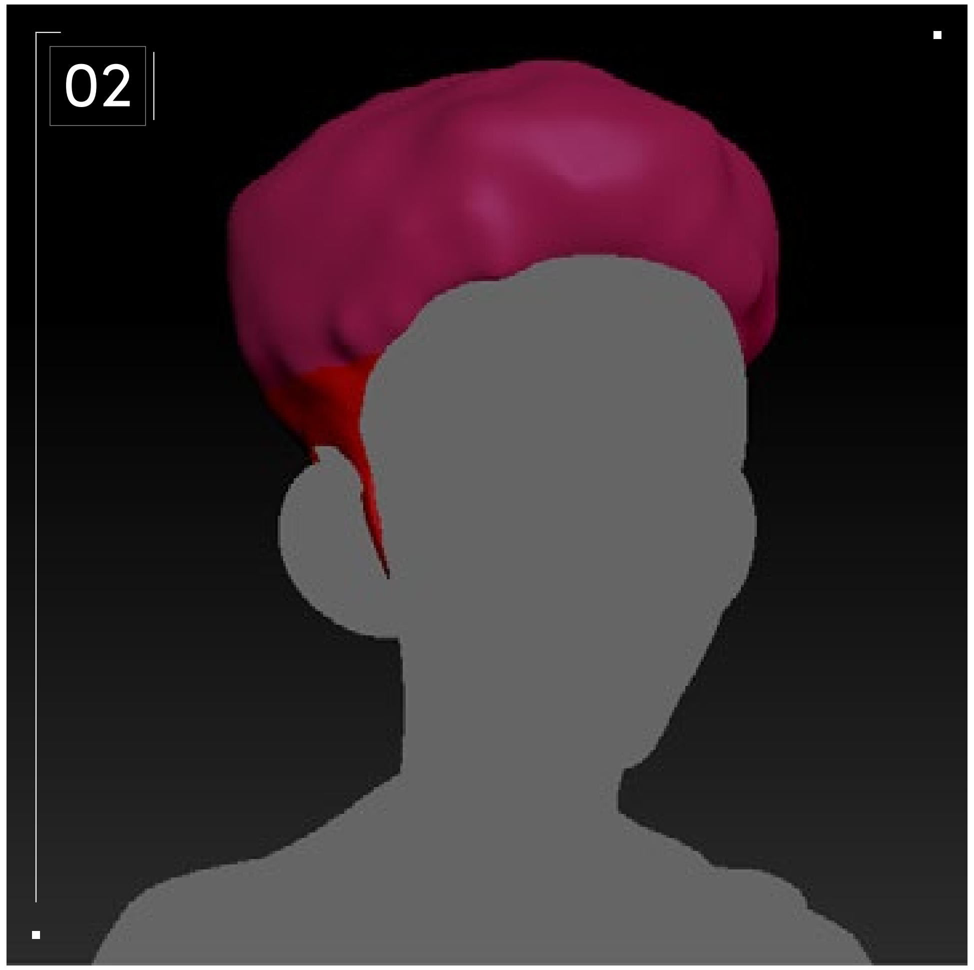
Step-by-Step Guide

01

// Reference images

I start every project with as many references as I can humanly find. In this case, we were given images of models from a good range of angles. I used those as the base of the hair as it guided my early blocking-out stages. Then I assembled other references, both fictional and real, to guide my more stylistic choices.

02



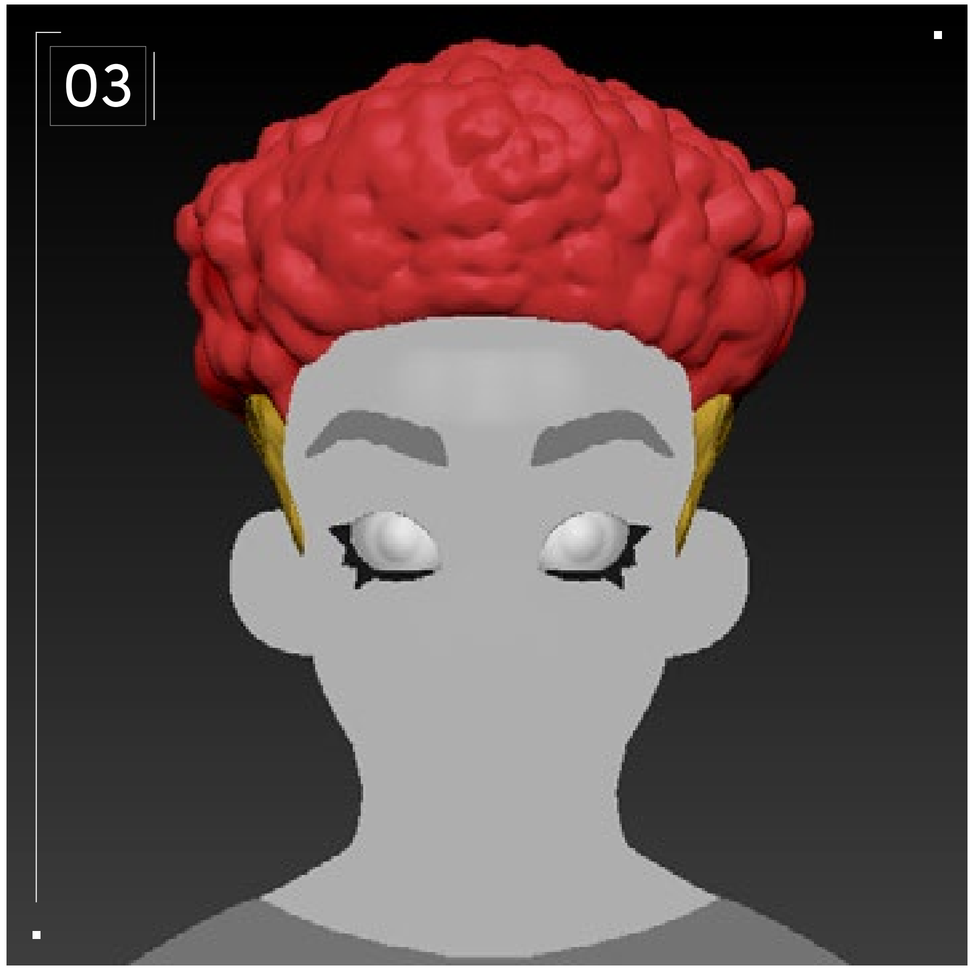
// Blocking

First, I needed to block out the hair model using ZBrush. At this stage I'm aiming to get the general shape of the hair and see how the different sections interact. I started this style with a Dynameshed sphere set to the lowest resolution to make the model easier to manipulate. Then I broke it into separate pieces. In the case of the Afro Fade, this was pretty simple: the Afro and the Fade.

ZBrush has a few ways of doing this, but I usually prefer to mask the areas I want to separate and then select Polygroups from mask in the Polygroups subpalette. Then I started blocking out some of the details, such as the baby hairs. I had never tried to do them on a model before, but I always really liked how they looked so I was excited for the challenge. I ended up with quite a few different iterations of the baby hairs for the model. My style is usually pretty simplistic, but I wanted to try to capture some of their intricate details, which caused a pretty big dissonance in the sculpt between the baby hairs and the rest of the more simplistic and cartoony aspects of the model. In the end, I realized I could either make the rest of the model more detailed or I could simplify the baby hairs. I chose the latter.

// Project Settings

Step-by-Step Guide



// Detailing

I used Zremesher and then subdivided them a few times. I would go between the different subdivision levels, adding details while on the highest subdivision level, and switch back to the lowest subdivision level to refine the shape more easily.

Next, I tried something a little new (to me at least) and used ZBrush's 3D layers. When I began sculpting the detail, I made a new layer in the Layer subpalette and made sure it was recording — although I think it does so automatically. After my issues with the baby hairs, I became pretty worried about making the hair a little "too much" in places and messing up the cohesion of the piece. So, with layers I was able to increase or decrease the strength of the brush strokes I made while sculpting the details, and that allowed me to tweak large parts of the mesh all at once on the fly.

For the Fade part of the Afro Fade, I refined the shape of the hairline, but I wanted to keep it pretty simple to contrast the Afro. For the Afro, I also wanted something pretty simple, so I got to work and used the ClayBuildup brush to make round shapes of varying sizes. I try to add a decent amount of variation when I'm detailing, so I would periodically turn on and off my symmetry so things wouldn't look too uniform on the Afro. I also used the inflate brush to shrink and enlarge parts of the hair. When I was semi-satisfied with the Afro, I then paused the layers recording and adjusted the strength of the brush strokes I had just done to my liking; then I was satisfied.

// Project Settings

Step-by-Step Guide

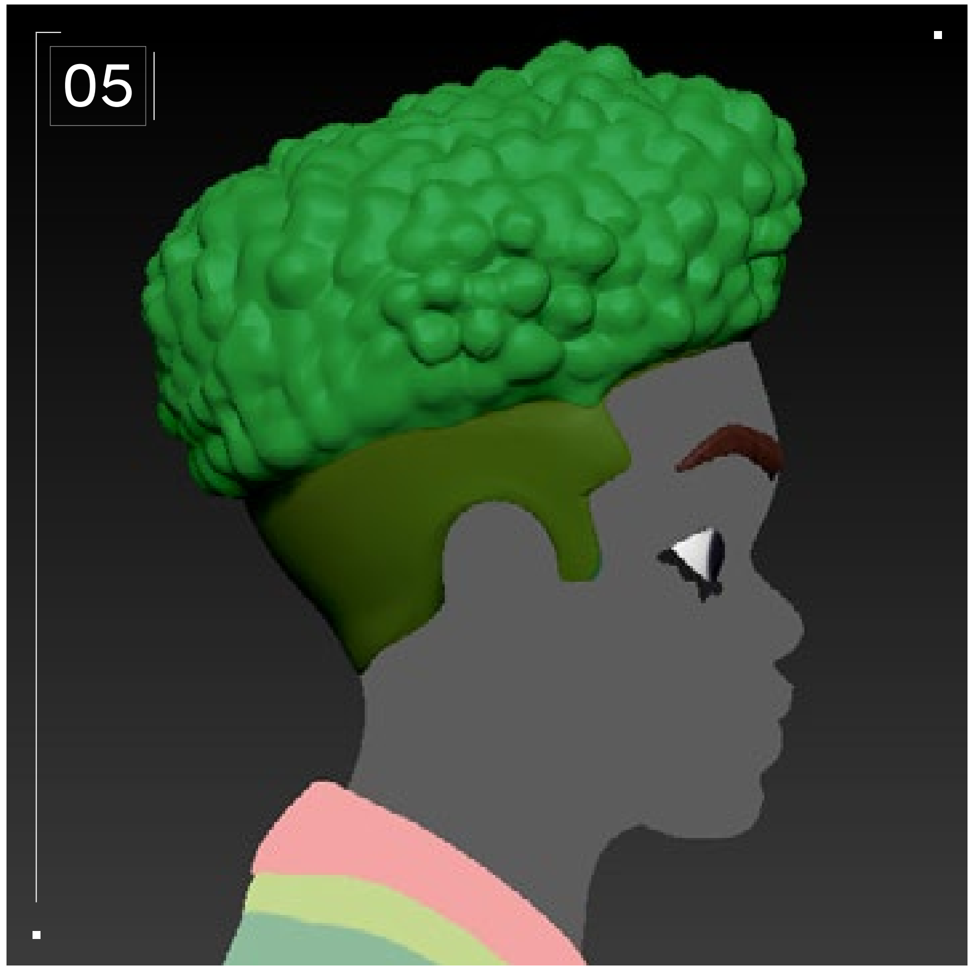
// Baking

When retopologizing my model I made sure to keep them as separate meshes so that I could bake each part without any intersecting geometry between the different parts. I then used the decimation master plug-in that's the default in the ZBrush plug-ins menu on the high-poly version of my meshes. As some of these ZBrush files can get pretty wild in terms of polycount, it helps to decimate them first as this allows you to retain detail while cutting down on geometry.

I then exported my high-poly and low-poly meshes, assembled them, and named them appropriately with the "_high" and "_low" suffixes. I exported them from Blender as OBJ and baked them in a Substance Painter. I baked all the usual maps — AO, Normal, for the final texture and curvature, group id, height, etc. — to help with generating masks in Substance Painter.

// Project Settings

Step-by-Step Guide



// Textures/Materials

I textured the style in Substance Painter. I wanted the colors of the style to go well with the skin tones I had picked out for my characters. I went with a warmer complexion and reflected that in the hair by choosing similarly warm colors. The texturing process was pretty simple. I painted my based colors, used the blur filter to make them blend better, and I tried to use masks to create some subtle variation in the roughness and color of the hair.

After tweaking values to my liking, I was satisfied with the colors. In Blender I tried to add a subtle gleam to the hair, so in the node tree of the material, I had a mix node as my roughness.

The node was mixing the hair's roughness with a dark grey, and the alpha was driven by a Fresnel. I just tuned that to my liking. This was done in Blender but it's a pretty basic material and I'm certain it can be replicated in basically any 3D software with node-based materials.

Finally, all that was left was to render the model with the hair.

Lead Contributors

A.M. Darke

Founder of the Open Source Afro Hair Library

A.M. Darke (she/he) is a critical game maker and professor of Games and Playable Media at the University of California, Santa Cruz. She teaches in the departments of Performance, Play, and Design; Digital Arts and New Media; and Critical Race and Ethnic Studies. Professor Darke is the founding director of The Other Lab, an intersectional feminist research space for experimental games and new media, and the Center for Black Virtuality.

A.M. is an internationally recognized media artist, award-winning game developer, and influential public scholar. In 2020 she founded the Open Source Afro Hair Library, a free resource of 3D assets celebrating Black hair textures and styles, developed in collaboration with Black 3D artists from around the world. In addition to his creative scholarship in the arts, Professor Darke has also won awards for her groundbreaking work in computer graphics, helping to develop the first algorithm dedicated to the simulation of Afro-textured hair.

A.M. describes himself as a gender-chaotic Black woman who values comfort, agency, and direct communication. She's the CEO of Darke Digital, a playful media studio and consultancy located in Oakland, California.

Lead Contributors

Nai'Vasha Grace

Celebrity Hairstylist

Nai'Vasha is a bicoastal hairstylist who specializes in natural hair, hair color, wigs, and men's grooming. Her celebrity clientele includes Alicia Keys, Tracee Ellis Ross, Yara Shahidi, Lupita Nyong'o, Elaine Welteroth, Laura Harrier, Aurora James, and Serena Williams. Nai'Vasha has collaborated with top photographers such as Scott Trindle, Yelena Yemchuk, Nagi Sakai, Bruce Weber, Bjorn Iooss, and Renell Medrano on shoots appearing in major publications such as *Vogue*, *Vanity Fair*, *Marie Claire*, *ELLE*, *W*, *Harper's Bazaar*, and *Allure*. Additionally, Nai'Vasha worked on major campaigns for Aerie, Google, Chanel, Bloomingdale's, Tory Burch, and Tiffany & Co.

Lead Contributors

Isaac Olander

3D Artist

Isaac is the CEO of TALLGRAN studios and has extensive experience in game design and 3D graphics. He's had the privilege of working for world-renowned VFX studios and has since founded his own successful 3D production company. Throughout Isaac's career, he's gained valuable insight into the many pitfalls and endless possibilities of 3D design with the mission to deliver exceptional 3D content from creative vision to final rendered product.

Issac feels honored to play an active role in a cause that he truly believes in. He is personally grateful to the many talented artists who have selflessly shared their knowledge before him. Their generosity has shaped his artistic journey and enabled him to reach where he is today. As such, Isaac feels incredibly lucky to have the chance to give back to the artistic community with this guide, by sharing his knowledge of 3D as well as his personal experiences.

Artist Bios

Sajhmori

3D artist

Sajhmori, or Saj for short, is a 28-year-old self-taught 3D character artist from the U.S. They left their first career as an aerospace engineer to develop characters for film and VFX. Now Saj mostly makes fan art and tailors their work to the video game industry.

Malika

3D artist

Malika is a 25-year-old character artist from Belgium. She is mainly a character designer and illustrator but has recently started working as a character artist for video games — The Wagadu Chronicles, to be precise — where she is mainly working on hair, clothes, and accessories.

Although Malika is more into fantasy and extravagant styles, given her love for all things hair and fashion, she is really happy to be part of the project.

H.D. Harris

3D artist

H.D. is an artist living in Michigan who was driven by Midwestern life to imagine worlds full of color, soul, and energy, even though they had to find the building blocks for those worlds themselves. They live with their partner and a small herd of furry animals, all of which keep them from drowning in the deep end when diving into the waters of creation.

Diandra Rose - Superfuchsia

3D artist

Through Diandra's art, she shares her journey as a self-taught 3D artist, drawing inspiration from the iconic aesthetics of the '90s and '00s video vixens to the tastemakers of today to reshape the narrative of Black women in art. Hours of meticulous research and archival work form the foundation of Diandra's creative process. Combining her background in photography and digital art with a passion for fashion and beauty, she has crafted her own fantasy that celebrates the many facets of Black femininity beyond the ever-present bald, thin, ambiguous woman.

Artist Bios

Adésayo Adéoyé (Ade)

3D artist

Adésayo is a 22-year-old Yoruba, self-taught 3D artist who dabbles in character and environment art.

Given he has been in full-environment art mode, Adésayo was surprised to be approached for this project, but as a lover of Dove soap, he was very happy and honored to contribute.

Chrissy Powell

3D artist

Chrissy is a 2D/3D artist located in Atlanta, Georgia, whose passion is highlighting softness within the Black community through their love of all things cute. She specializes in vector illustration, character design, hard surface modeling for 3D printing, and stylized 3D characters/props.

Chrissy is truly honored to be featured alongside so many talented Black artists. Although she is relatively new to 3D, it has reignited their creativity and believes there's much yet to be explored in Black hair.

Thank You

//

We believe that every single person has the right to see their beauty represented in virtual worlds with respect, care, and accuracy.

We believe that every single person has the right to see their beauty represented in virtual worlds with respect, care, and accuracy. The importance of natural hair, protective styles, and the deep cultural relationships between them and those who have it, cannot be overstated.

To misrepresent Black hair textures is to misrepresent every single person who has it.

So, we are proud to play a part in the ongoing fight to make sure natural hair is treated with love, respect, and equity — especially in mediums like video games, where the realization of oneself and one's beauty are truly limitless.

We thank every contributor who has given their artistry and personal experience to make this guide a reality. And we sincerely hope this guide serves you well in creating accurate Black hair textures and protective styles in their many beautiful forms.

Dove Mission Statement

At Dove, care has always been at the heart of everything that we do.

We believe in **care that goes beyond what our products deliver.** Since our inception, **we have strived to make beauty a source of confidence, and not anxiety.** We do this by **taking action to make tangible change.**

For women, that's challenging narrow beauty ideals and harmful stereotypes to set a new standard for the authentic, diverse, and inclusive representation of real beauty. And for the next generation, building their self-esteem, especially girls through the Dove Self Esteem Project — the world's biggest self-esteem education programme which has already reached over 90 million young people across 142 countries since it began in 2004.

But we cannot drive that change alone. We need to partner with like-minded companies, NGOs, individuals and even competitors to make a difference. Driving collective actions means delivering long-standing impact. With our partners we are continually committed to creating a more caring and compassionate world.

In 2019, we co-founded the CROWN Coalition in partnership with the National Urban League, Color of Change, and Western Center on Law and Poverty to help support the passage of the CROWN Act and create a more equitable and inclusive experience for Black women and girls. The CROWN Act, which stands for Creating a Respectful and Open World for Natural hair, is legislation that prohibits race-based hair discrimination based on hair texture and protective styles such as braids, locs, twists, and knots in the workplace and K-12 public and charter schools.

Today, The CROWN Coalition now includes over 100 community and advocacy organizations working toward real, actionable change in support of the CROWN Movement.

Most recently, we've taken our mission to the virtual world. Real Virtual Beauty is a series of commitments challenging the representation of beauty online and changing education for self-esteem in the virtual realm.

With the support of the Centre for Appearance Research and Unreal Engine Education, we launched the Real Virtual Beauty Training Course, designed to teach game developers how to bring variety to the appearance and functionality of digital characters.

We are encouraging thousands of developers to be part of a bigger movement to improve the well-being of players by raising their self-esteem, elevating their idea of how they fit in in the world, and broadening representation.

Code My Crown is our latest initiative to raise the bar in terms of accurate, positive and respectful representation in the gaming industry. We are proud to play a small part in the ongoing fight to make sure textured hair is treated with love, respect, and equity. We'll always continue to champion authentic representation of the Black community, both on our platforms and others.

Actions speak louder than words, and we take our responsibility and scale as the world's largest beauty brand seriously. **Let's Change Beauty.**

OSAHL Mission Statement

The Open Source Afro Hair Library provides free access to high-quality 3D models of Black hair textures and styles. One of the most important aspects of cultural identity, Black hairstyles have come to symbolize pride, personal expression, and political resistance. Our mission is to radically expand the images we see of Black hair on-screen, and the opportunities Black creators have behind the scenes. More than a resource, we are a growing community of Black artists and allies building a queer, feminist, creative community to celebrate Black hair in three dimensions.

open source afro hair library

Dove | open source afro hair library

This is a guide to the creation and celebration of Black hair textures and protective styles in video games.

It features instructions, personal insights, and detailed references, created by people within the Black community, for anyone to to create characters in the digital world.

No single document can capture the history and beauty of Black hair textures and protective styles. But through this body of work, our goal is tohelp us create the change we want to see.

Lifted Curls: A Model for Tightly Coiled Hair Simulation

ALVIN SHI*, HAOMIAO WU*, and JARRED PARR, Yale University, USA
A.M. DARKE, University of California, Santa Cruz, USA
THEODORE KIM, Yale University, USA

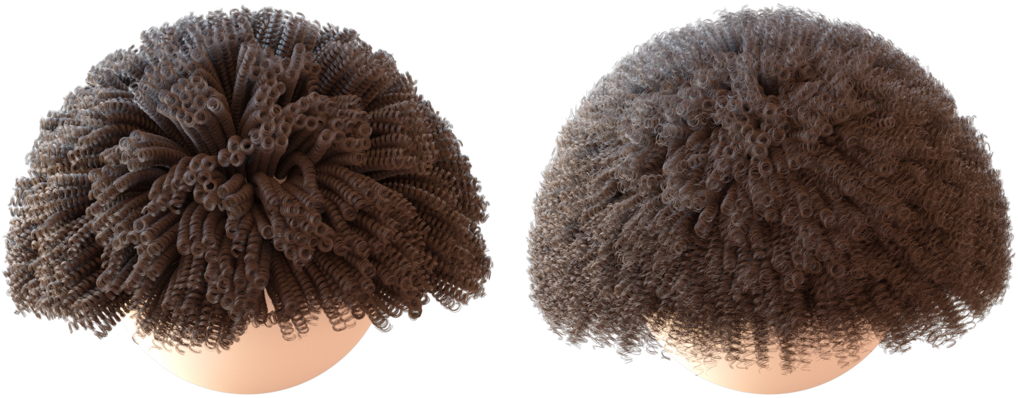


Fig. 1. We efficiently simulate tight curls on a hairball. 200K hairs generated from 2000 simulated wisps naturally form a "clumped" look (left), while 8000 simulated wisps (right) produces a more "frizzy" or "picked out" look. Both were run with $\Delta t = 1/30$ s, and respectively took 17.7 and 57.0 seconds per frame.

We present an isotropic, hyperelastic model specifically designed for the efficient simulation of tightly coiled hairs whose curl radii approach 5 mm. Our model is robust to large bends and torsions, even when they appear at the scale of the strand discretization. The terms of our model are consistently quadratic with respect to their primary variables, do not require per-edge frames or any parallel transport operators, and can efficiently take large timesteps on the order of ~1/30 of a second. Additionally, we show that it is possible to obtain fast, closed-form eigensystems for all the terms in the energy. Our eigenanalysis is sufficiently generic that it generalizes to other models. Our entirely vertex-based formulation integrates naturally with existing finite element codes, and we demonstrate its efficiency and robustness in a variety of scenarios.

ACM Reference Format:
Alvin Shi, Haomiao Wu, Jarred Parr, A.M. Darke, and Theodore Kim. 2023. Lifted Curls: A Model for Tightly Coiled Hair Simulation. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 2, Article 1 (August 2023), 19 pages. https://doi.org/10.1145/3606920

*Joint first authors

Authors' addresses: Alvin Shi, alvin.shi@yale.edu; Haomiao Wu, haomiao.wu@yale.edu; Jarred Parr, jarred.parr@yale.edu, Yale University, 51 Prospect Street, New Haven, CT, 06511-8937, USA; A.M. Darke, darke@ucsc.edu, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA, 95064, USA; Theodore Kim, theodore.kim@yale.edu, Yale University, USA.

2577-6193/2023/8-ART1
https://doi.org/10.1145/3606920

1 INTRODUCTION

We present a fast, robust, isotropic hyperelastic model specifically designed for the simulation of *tightly coiled* strands. Our goal is an efficient model that can simulate a full head of hair containing tight curls with a radius approaching ~5 millimeters (Fig. 1). Such hair is known as Types 3-4 [Walker 1997], Types V-VIII [De La Mettrie et al. 2007], kinky, coily, or Afro-textured hair.

Prior models for *curly* hair [Iben et al. 2013; Selle et al. 2008] tend to be designed for coils that are Type 2 [Walker 1997] or Types III-IV [De La Mettrie et al. 2007], which are significantly looser than the coils we examine. We are only aware of one other work in graphics that attempts a full-head simulation of hair within the "tightly coiled" regime [Bertails et al. 2006].

Cosserat-based models [Bergou et al. 2010; Pai 2002] tend to be demonstrated on straight or wavy hair, where a single hair strand is well-approximated by twenty [Gornowicz and Borac 2015] to thirty [Daviet 2020; Kaufman et al. 2014] line segments. Significantly denser discretizations, e.g. ~100 segments per hair, are needed to resolve the spatial frequencies we focus on here, but when large bends and torsions appear at scales near the strand discretization, they tend to exacerbate well-known numerical difficulties [Cirio et al. 2014; Gornowicz and Borac 2015] in existing models. The cost of increasing the strand resolution to obviate these issues becomes prohibitive.

We instead present a model specifically designed for tightly coiled hair. In particular, we robustly account for large bends and torsions near the discretization resolution by assuming that the rest state of each strand is *non-straight*: that is, the initial vertices are not co-planar. This assumption allows us to forgo the usual Cosserat approach [Pai 2002] of per-edge frames. Distinct from *twist*, which is defined with respect to edge frames, we compute *torsion* directly from vertex positions. Since hair is a thin strand with one clamped end, we have found this to be sufficient .

We refer to our model as *Lifted Curls*, because it consists of a 1D *stretching* energy, a bending term analogous to a 2D *membrane* energy, and a torsion term that recalls a 3D *volume* energy. In particular, the 3D term allows the strand to robustly resolve angle ambiguities that appear during large deformation. We target large, efficient timesteps on the order of $1/30$ seconds, such as those used in fully implicit production simulators [Kim and Eberle 2022]. These schemes require energy Hessians with all their negative eigenvalues removed. To this end, we obtain compact, closed-form eigensystems for every term in our energy, which allows the eigenvalues to be efficiently filtered. Our analysis is general, and can be used to improve the stability of previous hair models [Iben et al. 2013; Selle et al. 2008].

Our contributions are as follows:

- A fast, robust hyperelastic energy for tightly coiled hair, formulated as a 1D stretching energy, a membrane-like 2D bending energy, and a volume-like 3D torsion energy.
- Generic, closed-form eigensystems for strand-based stretching energies.
- Generic, closed-form eigensystems for unitary, angle-based isotropic bending energies.
- A closed-form 3D eigensystem for our torsion energy.

Our formulation operates solely on vertex positions, and is straightforward to incorporate into existing cloth and volume simulators.

2 RELATED WORK

Hair simulation is a classic computer graphics topic going back to at least Anjyo et al. [1992]. Most modern treatments incorporate Cosserat's theory of rods, which was first introduced to graphics by Pai [2002], and extended to include dynamics by Spillmann and Teschner [2007]. A key component of Cosserat theory is an orthonormal basis of directors defined along the strand, which allows torsion to be tracked even when the initial strand is straight.

Perhaps the most popular Cosserat-style model is Discrete Elastic Rods [Bergou et al. 2008], which formulates its directors using parallel transport, and tracks a twist-free Bishop frame. An implicit version was also proposed [Bergou et al. 2010], and has appeared in three film production systems [Gornowicz and Borac 2015; Lesser et al. 2022; Thyng et al. 2017]. This is the main model we will compare against, in particular the open source implementation from the original authors (Bergou) provided in Jawed et al. [2018]. While popular, practitioners have reported challenges with the model, including the need to smooth away singularities [Gornowicz and Borac 2015], and instabilities with BDF-2 under. a single Newton step [Eberle 2022]. We find that these problems are exacerbated in the case of tightly coiled hair, when large bends and torsions appear at the same spatial scale as the underlying discretization. Our model is designed to address these limitations.

We present an *isotropic* strand model, seemingly at odds with our goal of simulating tightly coiled hair, as curliness commonly correlates with hair containing elliptical (anisotropic) cross-sections [Bryant and Porter 2012]. However, recent studies have found that the distribution of different cell groups in a follicle is the actual dominant mechanical factor [Wortmann et al. 2020] that produces curliness (not anisotropy)‘, so an isotropic strand should suffice.

Position-Based Dynamics (PBD) approaches based on Cosserat theory have also been proposed. Umetani et al. [2015] used ghost points to track edge orientations, while Kugelstadt and Schömer [2016] used quaternions. Deul et al. [2018] accelerated solves on tree-structured assemblies, while Soler et al. [2018] showed how to introduce orientation into Projective Dynamics. Our contribution is orthogonal to this class of solvers, which could potentially benefit from our novel strand energy.

Spring-mass approaches also forgo the Cosserat theory, and have had success in two film production systems [Iben et al. 2013; Ward et al. 2010]. These approaches all use explicit time integration, which introduce significant timestep restrictions. Selle et al. [2008] uses implicit damping and explicit material forces, so the restrictions remain.

Volume methods have been employed to resolve dynamics in these simulations [McAdams et al. 2009], but specifically for straight hair, not the tight curls that we target. Hair Meshes [Yuksel et al. 2009] is a volumetric approach to styling hair, but is designed for, and demonstrated on, straight or wavy hair. The accompanying simulation method [Wu and Yuksel 2016] targets hair that is well-approximated by cloth, i.e. very flat, straight hair. We instead target tightly coiled hair.

Higher order spatial methods such as super-helices [Bertails et al. 2006] have their own successes, especially when their quadratic complexity becomes linear using a recursive solver [Bertails 2009]. We instead target spatially linear elements, as these are standard in film production [Kim and Eberle 2022; Lesser et al. 2022; McAdams et al. 2011]. In lieu of resolving linear vs. quadratic spatial discretization mismatches and iterative vs. recursive solver mismatches, we can instead simply inject our new FEM forces into an existing solver.

3 LIFTED CURLS ENERGY

In this section, we present our *Lifted Curls* energy. The stretching and bending terms will be familiar from other models, but the torsion term will be less familiar. We will write each in a way that facilitates eigenanalysis in the next section. The overall energy is:

$$\Psi_l = \Psi_s + \Psi_b + \Psi_t \tag{1}$$

where $\Psi_{s,b,t}$ respectively refer to stretching, bending, and torsion.

Notation: We adopt the following conventions. Scalars are unbolded, lowercase letters (k), vectors are bolded lowercase letters ($\mathbf{k}$), and matrices are bolded uppercase letters ($\mathbf{K}$). The current position of a vertex i is denoted with $\mathbf{x}_i$, while the original rest-space position has an overbar, $\bar{\mathbf{x}}_i$. Edges between vertices are denoted $\mathbf{e}_i$.

3.1 Stretching Energy

We use the stretching energy common to most strand models [Bergou et al. 2008; Sánchez-Banderas et al. 2020; Sueda et al. 2011]:

$$\Psi_s = \frac{\mu_s}{2}\left(I_s - 1\right)^2, \tag{2}$$

but write it using an $I_s \in \Re$ stretching invariant, defined using the 1D deformation gradient $\mathbf{d}$. Given the current positions $\mathbf{x}_0$ and $\mathbf{x}_1$ of an edge, and their rest-state positions $\bar{\mathbf{x}}_0$ and $\bar{\mathbf{x}}_1$, the deformation gradient is:

$$\mathbf{d} = \frac{\mathbf{x}_1 - \mathbf{x}_0}{\|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_1\|}. \tag{3}$$

The stretch invariant is then $I_s = \|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^\top \mathbf{d}}$.

3.2 Bending Energy

3.2.1 Energy Formulation. We use an isotropic, angle-based bending energy commonly found in other strand models [Cirio et al. 2014; Sánchez-Banderas et al. 2020; Sperl et al. 2022; Sueda et al. 2011]:

$$\Psi_b = \frac{\mu_b}{2}\left(\theta - \theta_0\right)^2. \tag{4}$$

We define θ in terms of three consecutive vertices on a strand, $\mathbf{x}_0$, $\mathbf{x}_1$, and $\mathbf{x}_2$. The edges between them are denoted

$$\mathbf{e}_0 = \mathbf{x}_0 - \mathbf{x}_1 \qquad\qquad \mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1, \tag{5}$$

which we will see in §4.2.1 becomes a *cloth-like* formulation. We measure the angle between the edges as $\theta = \text{acos}\left(\frac{\mathbf{e}_0^\top \mathbf{e}_1}{\|\mathbf{e}_0\|\|\mathbf{e}_1\|}\right)$, where θ_0 is the angle between the rest-state edges. This differs slightly from the usual definitions based on"turning angle" $\approx \pi - \theta$, but will facilitate analysis later. Robust-seeming atan2(y, x) formulations do not apply here, because y is always positive. In the absence of a third orienting vector (e.g. the hinge edge in cloth bending), the $(0, \pi)$ vs. $(-\pi, 0)$ ambiguity cannot be resolved. If acos returns $\theta = \pi/4$, it is unknown whether the angle should be $\pi/4$, or if an extreme deformation has occurred and the angle should actually be $-\pi/4$. We will resolve this issue using our torsion energy in §3.3.

3.2.2 Relationship to DER. For two reasons, we avoid tan-based energies [Bergou et al. 2008] like $\Psi_{\tan} = \frac{\mu_b}{2}\left(\tan\left(\frac{\theta}{2}\right) - \tan\left(\frac{\theta_0}{2}\right)\right)$. First, as others have observed, the singularity at $\theta = \pi$ introduces robustness issues. Cirio et al. [2014] reported undesirably large forces appearing when simulating yarn, and instead advocated for Eqn. 4. Gornowicz and Borac [2015] also found the large forces undesirable, and smoothed them with an approximate linearization (their Fig. 4). Second, tan-based energies yield a near-linear response for straight rods where $\theta_0 \approx 0$, but for tight curls where θ_0 begins to approach π, the tan asymptote starts to dominate. The force response, even near the rest state, becomes unappealingly exponential.

3.3 Torsion Energy

3.3.1 Energy Formulation. We introduce a novel torsion energy

$$\Psi_t = \frac{\mu_t}{2}\left(\tau - \tau_0\right)^2, \tag{6}$$

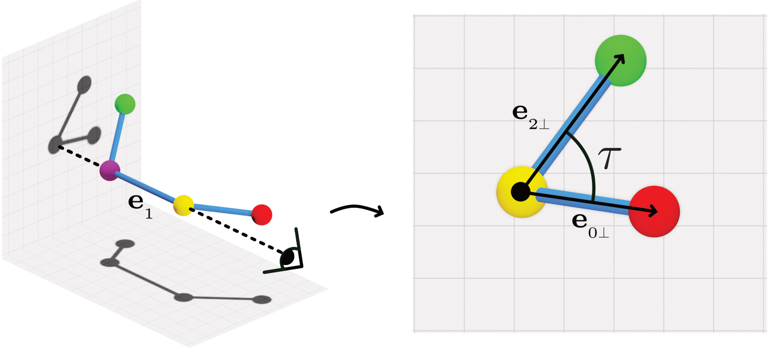


Fig. 2. Geometry of our torsion energy Ψ_t along a tetrad of vertices. We project the vertices onto the plane orthogonal to the middle edge, $\mathbf{e}_1$ (left), and obtain the projected edges $\mathbf{e}_{0\perp}$ and $\mathbf{e}_{2\perp}$ (right). The torsion angle τ is then the in-plane angle between these edges.

which is applied to each *tetrad* of vertices along a strand, and produces a *volume-like* formulation. Following Eqn. 5, the edges between the vertices are

$$\mathbf{e}_0 = \mathbf{x}_0 - \mathbf{x}_1 \qquad \mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1 \qquad \mathbf{e}_2 = \mathbf{x}_3 - \mathbf{x}_2.$$

The torsion τ along edge $\mathbf{e}_1$ is computed by projecting $\mathbf{e}_0$ and $\mathbf{e}_2$ onto its orthogonal plane (see Fig. 2):

$$\mathbf{e}_{0\perp} = \mathbf{e}_0 - \frac{\mathbf{e}_0^\top \mathbf{e}_1}{\|\mathbf{e}_1\|^2}\mathbf{e}_1 \qquad \mathbf{e}_{2\perp} = \mathbf{e}_2 - \frac{\mathbf{e}_2^\top \mathbf{e}_1}{\|\mathbf{e}_1\|^2}\mathbf{e}_1. \tag{7}$$

We can then assemble a deformation gradient-like matrix

$$\mathbf{W} = \left[\begin{array}{c|c|c} \mathbf{e}_{0\perp} & \mathbf{e}_{2\perp} & \mathbf{e}_1 \end{array}\right] \in \mathfrak{R}^{3\times 3} \tag{8}$$

where the torsion angle τ is defined as

$$\tau = -\mathcal{S}\left(\det \mathbf{W}\right) \cdot \operatorname{acos}\left(\frac{\left(\mathbf{e}_{0\perp}\right)^\top \mathbf{e}_{2\perp}}{\|\mathbf{e}_{0\perp}\|\|\mathbf{e}_{2\perp}\|}\right) \tag{9}$$

and $\mathcal{S}(x)$ is a sign function:

$$\mathcal{S}(x) = \begin{cases} -1 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{10}$$

Similar to bending, τ_0 denotes torsion at the rest state. The acos term resembles θ from bending, but measures the torsion induced by the two edges *orthogonal* to the bending direction. This removes a troublesome, non-physical nullspace from the system, because any triplet $\mathbf{x}_{0,1,2}$ rotated about the axis $\mathbf{e}_0$ yields the same bending energy. The addition of $\mathbf{x}_3$ and Ψ_t instead ensure that one particular rotation becomes energetically preferred.

The $\mathcal{S}\left(\det \mathbf{W}\right)$ resolves the $(0, \pi)$ vs. $(-\pi, 0)$ ambiguity. Four non-co-planar vertices form a tetrahedron, so negative volume becomes well-defined. Similar to how $\det \mathbf{F}$ is used to detect when a volumetric element has inverted [Smith et al. 2018], an negative $\det \mathbf{W}$ unambiguously signals that extreme deformation has occurred, and that the strand has torqued outside the $(0, \pi)$ range. Instead of using tan-based barriers [Bergou et al. 2008] to prohibit this ambiguity, our energy is well-defined over $(-\pi, \pi)$. Our approach is closely related to atan2(y, x), as $\det \mathbf{W}$ corresponds to the triple product for y.

3.3.2 Relationship to DER. Our torsion force can disappear if the simulation coerces the strand into a line (i.e. $\|\mathbf{e}_{0\perp}\| = \|\mathbf{e}_{2\perp}\| = 0$). However, this is consistent with DER [Bergou et al. 2008], where the twisting force is defined in terms of the binormal. This also disappears when neighboring segments become parallel, as the cross-product of parallel segments is a zero vector. In this configuration, both models assume that stretching and bending forces will dominate.

While our energy assumes a non-straight *rest state*, it handles straight *deformed states* robustly. As can be seen in §4.3 of the supplement to Panetta et al. [2019], this more difficult for DER-like parallel transport-based methods, because differentiating the parallel transport operator produces a tan term which becomes singular as the two edges approach each other, and also introduces matrix asymmetries that must be approximated. In contrast, differentiating our energy yields a well-behaved force function. We will show in §5.1.4 that this allows us to take timesteps that are orders of magnitude larger than DER.

3.3.3 Physical Interpretation. Our core assumption that the initial configuration is *non-straight* ensures that torsion becomes visible to the vertex positions, because it guarantees that the initial τ_0 is always well-defined. Since we are simulating thin strands of hair, we can assume only one end of the strand is fixed, and that the relative cross-sectional area is small. We can justify discarding the material frames because as cross-sectional inertia approaches zero, so does the relative twist angle (see §5 in [Bergou et al. 2008]). Twist becomes almost entirely characterized by torsion, so our energy captures the predominant force. These assumptions might complicate the simulation of other phenomena such as plectonemes or torque transducers, but this is acceptable since our focus is on hair. We will see in §5.1 that these assumptions considerably improve numerical robustness.

Outside of graphics, torsion has been defined as the angle between the normals of two consecutive triangles [Banchoff 1982]. However, this measure is not orthogonal to the bending direction, and generates overlapping forces similar to the parasitic phenomena observed in spring-mass approximations. Our formulation instead produces torsion forces that are, by construction, orthogonal to bending.

4 EIGENANALYSIS OF LIFTED CURLS

Large, efficient timesteps require the matrix of force gradients to be positive semi-definite. One well-known method for ensuring this is to filter the per-element eigenvalues [Teran et al. 2005], but if this is done numerically, its running time can rival that of the PCG solve. See §5 in our supplemental document for detailed timings.

We instead follow the analytic approach of Smith et al. [2019] and Lin et al. [2022]. We show that it is possible to construct the analytic eigendecomposition of each term in our energy and efficiently filter the energy Hessians. Our generic analyses can also be applied to other energies.

4.1 Stretching Eigenanalysis

We start with the stretching energy (Eqn. 2) by applying a set of transformations that reveal its eigenstructure. Our analysis will show that a simple strategy in fact corresponds to the exact analytic filter: *under compression, fall back to Gauss-Newton.*

4.1.1 Stretching Eigensystem. The force and force gradients along an edge are computed as

$$\mathbf{f} = -l\frac{\partial \Psi_s}{\partial \mathbf{x}} \qquad\qquad \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = -l\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2}, \tag{11}$$

where $\mathbf{f}$ is the force along the edge, $\mathbf{x} \in \Re^6$ is a vector of the two vertex positions, and l is the length of the original edge. We want the analytic eigensystem of $\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2} \in \Re^{6\times 6}$.

However, $\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2}$ contains a rank-3 nullspace that corresponds to rigid translation. In order to isolate the non-null components, we rewrite the Hessian terms of the deformation gradient $\mathbf{d}$ (Eqn. 3):

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{d}}{\partial \mathbf{x}}^{\top} \frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} \frac{\partial \mathbf{d}}{\partial \mathbf{x}} \tag{12}$$

where $\frac{\partial \mathbf{d}}{\partial \mathbf{x}} = \begin{bmatrix} -1 & 1 \end{bmatrix} \otimes \mathbf{I}_{3\times 3}$, the $\otimes$ denotes a Kronecker product, and $\mathbf{I}_{3\times 3}$ is an identity matrix. Without loss of generality, we can now focus on obtaining the analytic eigensystem of $\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} \in \Re^{3\times 3}$.

We next expand in terms of the stretch invariant I_s from §3.1

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} = \frac{\partial^2 \Psi_s}{\partial I_s^2} \frac{\partial I_s}{\partial \mathbf{d}} \frac{\partial I_s}{\partial \mathbf{d}}^T + \frac{\partial \Psi_s}{\partial I_s} \frac{\partial^2 I_s}{\partial \mathbf{d}^2} \tag{13}$$

and apply the identities $\frac{\partial I_s}{\partial \mathbf{d}} = \frac{\mathbf{d}}{I_s}$ and

$$\frac{\partial^2 I_s}{\partial \mathbf{d}^2} = \frac{1}{I_s} \frac{\partial \mathbf{d}}{\partial \mathbf{d}} + \mathbf{d} \frac{\partial I_s^{-1}}{\partial \mathbf{d}} = \frac{1}{I_s} \left(\mathbf{I}_{3\times 3} - \frac{1}{I_s^2} \mathbf{d}\mathbf{d}^{\top} \right), \tag{14}$$

to arrive at the final generic expression:

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} = {\color{red}\left(\frac{\partial^2 \Psi_s}{\partial I_s^2} - \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \right) \left(\frac{\mathbf{d}}{I_s} \right) \left(\frac{\mathbf{d}}{I_s} \right)^{\top}} + {\color{blue}\frac{1}{I} \frac{\partial \Psi_s}{\partial I_s} \mathbf{I}_{3\times 3}}. \tag{15}$$

Close examination of this equation reveals the underlying eigensystem. The blue term is a scaled identity, so its eigenvectors are arbitrary, and all three eigenvalues are $\frac{1}{I} \frac{\partial \Psi_s}{\partial I_s}$. The red term is an outer product, so its eigenvector is $\mathbf{d}/I_s$, which is a normalized version of $\mathbf{d}$. The red coefficient subtracts the $\frac{1}{I} \frac{\partial \Psi_s}{\partial I_s}$ eigenvalue in the $\mathbf{d}/I_s$ direction away from the blue term, and replaces it with $\frac{\partial^2 \Psi_s}{\partial I_s^2}$. Therefore, the generic eigenpairs $(\lambda_i, \mathbf{q}_i)$ must be:

$$\lambda_0 = \frac{\partial^2 \Psi_s}{\partial I_s^2} \qquad \mathbf{q}_0 = \frac{\mathbf{d}}{I_s} \tag{16}$$

$$\lambda_1 = \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \qquad \mathbf{q}_1 = \mathbf{d}_{\perp} \tag{17}$$

$$\lambda_2 = \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \qquad \mathbf{q}_2 = \mathbf{d}_{\times}. \tag{18}$$

The last two eigenpairs span an arbitrary subspace, so we use $\mathbf{d}_{\perp}$ and $\mathbf{d}_{\times}$ to denote two vectors orthogonal to $\mathbf{q}_0 = \mathbf{d}/I_s$ (See Fig. 3).

Similar to Smith et al. [2019], our generic analysis applies to *any* energy written in terms of I_s. The eigenvectors remain the same regardless of the energy, and only the eigenvalue expressions change. For our specific Ψ_s (Eqn. 2), the analytic eigenvalues are:

$$\lambda_0 = \mu_s \qquad \lambda_{1,2} = \mu_s \left(1 - \frac{1}{I_s} \right). \tag{19}$$

The first eigenvalue is always positive, and the second two immediately suggest a filtering strategy.

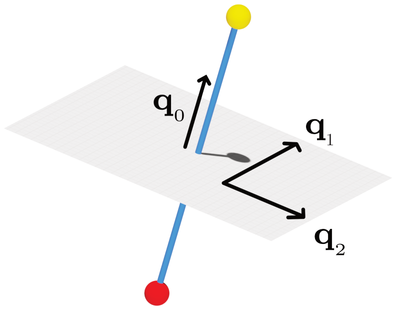


Fig. 3. Eigenvectors of strand stretching energies. The $\mathbf{q}_0$ vector is unique and its eigenvalue is always positive-definite. Non-unique eigenvectors $\mathbf{q}_1$ and $\mathbf{q}_2$ span the orthogonal plane and become indefinite under compression. *All* strand stretching energies conform to this eigenstructure.

4.1.2 Filtering Strategy and Discussion. The $\lambda_{1,2}$ eigenvalues only go negative under edge compression ($I_s < 1$), which corresponds to bifurcation under buckling. In this case, filtering the negative eigenvalues is equivalent to discarding the blue term from Eqn. 15, as well as $\frac{1}{I_s}\frac{\partial \Psi}{\partial I_s}$ from the red term.[1] All that remains is the outer product, $\frac{\partial^2 \Psi}{\partial \mathbf{d}^2} \approx \frac{\partial^2 \Psi}{\partial I_s^2}\left(\frac{\mathbf{d}}{I_s}\right)\left(\frac{\mathbf{d}}{I_s}\right)^\top$, which corresponds to the commonly used Gauss-Newton approximation [Choi and Ko 2002; Zehnder et al. 2021] for the Hessian. Our analysis shows that this filtering strategy is in fact exact: *for compressed edges, just compute the outer product.*

Our analysis generalizes to higher-order energies, such as St. Venant-Kirchhoff-like stretching

$$\Psi_4 = \mu_s (I_s - 1)^4, \tag{20}$$

where the eigenvalues become:

$$\lambda_0 = 3\mu_s (I_s - 1)^2 \qquad\qquad \lambda_{1,2} = \mu_s \frac{(I_s - 1)^3}{I_s}. \tag{21}$$

The indefiniteness condition ($I_s < 1$) remains the same, so again the Gauss-Newton strategy is an exact filter.

4.2 Bending Eigenanalysis

We next perform an eigenanalysis of our bending energy, Eqn. 4. As with stretching, we will first project off irrelevant nullspaces, but we will then see that the main challenge is analyzing the Hessian of θ, the angle between the two strand edges.

4.2.1 Cloth-like Formulation. We start by observing that since the bending energy involves three vertices, it can be thought of as a triangle of cloth. We can then rewrite Eqn. 4 into a more cloth-like form by arranging its two edges into a matrix:

$$\mathbf{E} = \left[\begin{array}{c|c} \mathbf{e}_0 & \mathbf{e}_1 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{x}_0 - \mathbf{x}_1 & \mathbf{x}_2 - \mathbf{x}_1 \end{array}\right] \in \Re^{3\times 2}. \tag{22}$$

This resembles the deformation gradient for cloth [Kim 2020], but no material space pullback is needed, as it is already encoded in θ_0. By introducing the standard basis directions $\mathbf{u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and

[1] While we are filtering the *Hessian* to maintain robustness, buckling is still present in the *forces* of the overall system.

$\mathbf{v} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, the angle formula can be rewritten as:

$$\theta = \operatorname{acos}\left(\frac{\mathbf{u}^\top \mathbf{E}^\top \mathbf{E}\mathbf{v}}{\|\mathbf{E}\mathbf{u}\| \cdot \|\mathbf{E}\mathbf{v}\|} \right) \tag{23}$$

We can *vectorize* $\mathbf{E}$ using the vec $(\cdot)$ operator [Golub and Van Loan 2013; Kim and Eberle 2022], and adopt the convention of a vectorized matrix becoming the lowercase version of the original, e.g.

$$\mathbf{e} = \operatorname{vec}(\mathbf{E}) = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \end{bmatrix}. \tag{24}$$

Similar to Eqn. 12, we then use $\mathbf{e} = \operatorname{vec}(\mathbf{E})$ to write

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} \in \Re^{9\times 9}. \tag{25}$$

We can now directly analyze $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} \in \Re^{6\times 6}$ while ignoring the rigid translation nullspace. Similar to Eqn. 13, we rewrite $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2}$ in terms of a scalar invariant, but this time use θ in lieu of I_s:

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} = \frac{\partial^2 \Psi_b}{\partial \theta^2} \frac{\partial \theta}{\partial \mathbf{e}} \frac{\partial \theta}{\partial \mathbf{e}}^\top + \frac{\partial \Psi_b}{\partial \theta} \frac{\partial^2 \theta}{\partial \mathbf{e}^2}. \tag{26}$$

The red term is an outer product of $\frac{\partial \theta}{\partial \mathbf{e}}$, so its eigensystem is known. The main challenge is determining the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{e}^2}$.

4.2.2 *Eigensystem of θ.*

The analysis becomes simpler when the edges are normalized[2], so we form the *tangent* matrix

$$\mathbf{T} = \left[\begin{array}{c|c} \frac{\mathbf{e}_0}{\|\mathbf{e}_0\|} & \frac{\mathbf{e}_1}{\|\mathbf{e}_1\|} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{t}_0 & \mathbf{t}_1 \end{array} \right] \tag{27}$$

and denote its vectorization $\mathbf{t} = \operatorname{vec}(\mathbf{T})$. Using the normalization

$$\mathbf{N} = \begin{bmatrix} \frac{1}{\|\mathbf{e}_0\|} & 0 \\ 0 & \frac{1}{\|\mathbf{e}_1\|} \end{bmatrix} \otimes \mathbf{I}_{3\times 3}, \tag{28}$$

we can then write:

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} = \mathbf{N}^\top \left(\frac{\partial^2 \Psi_b}{\partial \theta^2} \frac{\partial \theta}{\partial \mathbf{t}} \frac{\partial \theta}{\partial \mathbf{t}}^\top + \frac{\partial \Psi_b}{\partial \theta} \frac{\partial^2 \theta}{\partial \mathbf{t}^2} \right) \mathbf{N}. \tag{29}$$

The challenge is now to find the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$.

Fortunately, we have now transformed the Hessian into a form that has a *very* simple structure. It can be written in terms of the tangents $\mathbf{t}_0$ and $\mathbf{t}_1$, the binormal $\mathbf{b} = \mathbf{t}_0 \times \mathbf{t}_1$, and their cross products $\mathbf{b}_0 = \mathbf{t}_0 \times \frac{\mathbf{b}}{\|\mathbf{b}\|}$ and $\mathbf{b}_1 = \mathbf{t}_1 \times \frac{\mathbf{b}}{\|\mathbf{b}\|}$. The complete analytic eigenpairs $(\lambda_i^\theta, \mathbf{q}_i^\theta)$ of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$ are:

$$\lambda_0^\theta = \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_0^\theta = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ \mathbf{b} \end{bmatrix} \tag{30}$$

$$\lambda_1^\theta = \frac{\cos\theta + 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_1^\theta = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \tag{31}$$

$$\lambda_2^\theta = -1 \qquad \mathbf{q}_2^\theta = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{t}_0 + \mathbf{b}_0 \\ \mathbf{0}_3 \end{bmatrix} \tag{32}$$

[2] The general case of non-normalized edges introduces complexities that are extensively analyzed in Wu and Kim [2023].

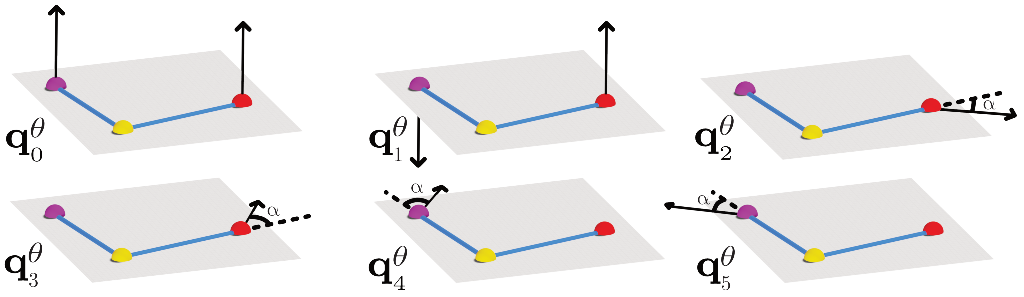


Fig. 4. The six eigenvectors of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$. The $\mathbf{q}^\theta_{0,1}$ vectors capture out-of-plane motions, while $\mathbf{q}^\theta_{2,3}$ represent in-plane stretches along the $\mathbf{t}_0$ yellow-red edge. Each is an in-plane $\alpha = \frac{\pi}{4}$ rotation of that edge. The $\mathbf{q}^\theta_{4,5}$ vectors are analogous rotations of the $\mathbf{t}_1$ yellow-purple edge.

$$\lambda^\theta_3 = 1 \qquad \mathbf{q}^\theta_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{t}_0 - \mathbf{b}_0 \\ \mathbf{0}_3 \end{bmatrix} \tag{33}$$

$$\lambda^\theta_4 = -1 \qquad \mathbf{q}^\theta_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{t}_1 + \mathbf{b}_1 \end{bmatrix} \tag{34}$$

$$\lambda^\theta_5 = 1 \qquad \mathbf{q}^\theta_5 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{t}_1 - \mathbf{b}_1 \end{bmatrix} \tag{35}$$

where $\mathbf{0}_3 \in \Re^3$ denotes a 3-vector of zeros. A Matlab/Octave script that verifies these expressions is provided in the supplemental materials. These eigenvectors can be interpreted relative to the plane spanned by $\mathbf{t}_0$ and $\mathbf{t}_1$ (see Fig. 4). The $\mathbf{q}^\theta_{0,1}$ vectors span the directions *orthogonal* to this plane, the $\mathbf{q}^\theta_{2,3}$ vectors are $\pm\frac{\pi}{4}$ rotations of $\mathbf{t}_0$ within that plane which form a local planar basis about $\mathbf{t}_0$, and $\mathbf{q}^\theta_{4,5}$ form an analogous planar basis about $\mathbf{t}_1$.

4.2.3 Bending Eigensystem. With the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$ in hand, we must now determine the eigensystem of

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{t}^2} = \frac{\partial^2 \Psi_b}{\partial \theta^2} \frac{\partial \theta}{\partial \mathbf{t}} \frac{\partial \theta}{\partial \mathbf{t}}^\top + \frac{\partial \Psi_b}{\partial \theta} \frac{\partial^2 \theta}{\partial \mathbf{t}^2}. \tag{36}$$

Then we can compute $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} = \mathbf{N}^\top \frac{\partial^2 \Psi_b}{\partial \mathbf{t}^2} \mathbf{N}$ using Eqn. 29. Since $\mathbf{N}$ is a diagonal matrix of positive scalars, filtering the eigenvalues of $\frac{\partial^2 \Psi_b}{\partial \mathbf{t}^2}$ is sufficient to guarantee that $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2}$ is positive-semidefinite (PSD). Applying Eqn. 25 then ensures that $\frac{\partial^2 \Psi_b}{\partial \mathbf{x}^2}$ is PSD.

Eqn. 36 is a rank-one update to $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$, so we can use the approach of Kim [2020] and apply the Bunch-Nielsen-Sorensen (BNS) formulae [Bunch et al. 1978] to compute how the $\frac{\partial \theta}{\partial \mathbf{t}} \frac{\partial \theta}{\partial \mathbf{t}}^\top$ update shifts the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$. While BNS is a numerical approach, our transformed system is now sufficiently simple that it can be applied analytically.

Applying BNS to an $\Re^{6\times 6}$ matrix generally involves solving a 6th order polynomial, but we make two key observations. First, the update vector is orthogonal to $\mathbf{q}^\theta_0$ and $\mathbf{q}^\theta_1$. Second, the remaining eigenvalues $\lambda^\theta_{2\ldots 5}$ are all ±1. Together, these *reduce the 6th order polynomial to a quadratic*, and allow the rank-one updated eigensystem to be written down in closed form. The complete derivation is in the supplemental materials, and the final eigensystem is

$$\lambda_0 = \frac{\partial \Psi}{\partial \theta} \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_0 = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ \mathbf{b} \end{bmatrix} \tag{37}$$

$$\lambda_1 = \frac{\partial \Psi}{\partial \theta} \frac{\cos\theta + 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_1 = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \tag{38}$$

$$\lambda_2 = \frac{\partial^2 \Psi}{\partial \theta^2} + \sqrt{\frac{\partial^2 \Psi}{\partial \theta^2}^2 + \frac{\partial \Psi}{\partial \theta}^2} \qquad \mathbf{q}_2 = \begin{bmatrix} \frac{\partial \Psi}{\partial \theta}\mathbf{t}_0 - \lambda_2 \mathbf{b}_0 \\ \frac{\partial \Psi}{\partial \theta}\mathbf{t}_1 + \lambda_2 \mathbf{b}_1 \end{bmatrix} \tag{39}$$

$$\lambda_3 = \frac{\partial^2 \Psi}{\partial \theta^2} - \sqrt{\frac{\partial^2 \Psi}{\partial \theta^2}^2 + \frac{\partial \Psi}{\partial \theta}^2} \qquad \mathbf{q}_3 = \begin{bmatrix} \frac{\partial \Psi}{\partial \theta}\mathbf{t}_0 - \lambda_3 \mathbf{b}_0 \\ \frac{\partial \Psi}{\partial \theta}\mathbf{t}_1 + \lambda_3 \mathbf{b}_1 \end{bmatrix} \tag{40}$$

$$\lambda_4 = \frac{\partial \Psi}{\partial \theta} \qquad \mathbf{q}_4 = \begin{bmatrix} \mathbf{t}_0 + \mathbf{b}_0 \\ -\mathbf{t}_1 + \mathbf{b}_1 \end{bmatrix} \tag{41}$$

$$\lambda_5 = -\frac{\partial \Psi}{\partial \theta} \qquad \mathbf{q}_5 = \begin{bmatrix} \mathbf{t}_0 - \mathbf{b}_0 \\ -\mathbf{t}_1 - \mathbf{b}_1 \end{bmatrix} \tag{42}$$

where the last four eigenvectors are written in unnormalized form. As in the stretching case, the specific expressions for the eigenvalues change, but the form of the eigenvectors remains the same, regardless of the energy. The first two again correspond to out-of-plane motion, the middle two are now in-plane oscillations, and the last two parameterize in-plane translations. The specific eigenvalues for our bending energy Ψ_b (Eqn. 4) are:

$$\lambda_0 = \mu_b \left(\theta - \theta_0\right) \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \lambda_1 = \mu_b \left(\theta - \theta_0\right) \frac{\cos\theta + 1}{\|\mathbf{b}\|} \tag{43}$$

$$\lambda_2 = \mu_b \left(1 + \sqrt{1 + (\theta - \theta_0)^2}\right) \qquad \lambda_3 = \mu_b \left(1 - \sqrt{1 + (\theta - \theta_0)^2}\right) \tag{44}$$

$$\lambda_4 = \mu_b(\theta - \theta_0) \qquad \lambda_5 = -\mu_b(\theta - \theta_0). \tag{45}$$

As with $\frac{\partial^2 \theta}{\partial \mathrm{t}^2}$, Matlab/Octave scripts are provided in the supplemental materials that verify the correctness of these expressions.

4.3 Torsion Eigenanalysis

The torsion energy deals fundamentally with the angle between two vectors, so we can reuse components of our bending eigenanalysis. We start with the vectorization $\mathbf{w} = \mathrm{vec}\left(\mathbf{W}\right)$ of Eqn. 8, and construct a torsion Hessian analogous to Eqns. 12 and 25 $\frac{\partial^2 \Psi_t}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} + \left(\frac{\partial \Psi_t}{\partial \mathbf{w}}\right)^\top : \frac{\partial^2 \mathbf{w}}{\partial \mathbf{x}^2}$, but the double-contraction is equivalent to $-\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$, which we describe further in the supplement. This then yields:

$$\frac{\partial^2 \Psi_t}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta. \tag{46}$$

Similar to bending, we can now directly examine $\frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2}$, where rank-3 translation nullspace has been removed. A slight wrinkle appears because $\frac{\partial \mathbf{w}}{\partial \mathbf{x}}$ seemingly contains new, non-trivial entries, but we show in the supplement that force contributions result to zero while the Hessian gains an easy-to-handle second term. Without loss of generality, we can use the expression:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & \mathbf{e}_0^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 - 1 & -\mathbf{e}_0^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 & 0 \\ 0 & \mathbf{e}_2^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 & -\mathbf{e}_2^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 - 1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \otimes \mathbf{I}_3. \tag{47}$$

While $\frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2}$ appears to be rank-9, the third column of $\mathbf{W}$ only participates in the $\mathcal{S}\,(\det \mathbf{W})$ term of τ. This becomes a constant under differentiation, except for a Dirac delta at zero which we can be filtered [Kim et al. 2019]. Thus, we can deflate away another rank-3 subspace to obtain a torsion-specific edge matrix:

$$\mathbf{E}_t = \mathbf{W} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{48}$$

Using the vectorization $\mathbf{e}_t = \text{vec}\,(\mathbf{E}_t)$, the first term in the Hessian now becomes:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial \mathbf{e}_t}{\partial \mathbf{w}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{e}_t^2} \frac{\partial \mathbf{e}_t}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{x}}. \tag{49}$$

The edge matrix $\mathbf{E}_t$ is structurally identical to the bending edge matrix $\mathbf{E}$ (Eqn. 22). All our prior analysis of $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2}$ (Eqn. 26) can be directly reapplied. The tangent-space eigenvalues for Ψ_t are then:

$$\lambda_0 = \mu_t\,(\tau - \tau_0)\,\frac{\cos\tau - 1}{\|\mathbf{b}\|} \qquad \lambda_1 = \mu_t\,(\tau - \tau_0)\,\frac{\cos\tau + 1}{\|\mathbf{b}\|} \tag{50}$$

$$\lambda_2 = \mu_t\left(1 + \sqrt{1 + (\tau - \tau_0)^2}\right) \qquad \lambda_3 = \mu_t\left(1 - \sqrt{1 + (\tau - \tau_0)^2}\right) \tag{51}$$

$$\lambda_4 = \mu_t(\tau - \tau_0) \qquad \lambda_5 = -\mu_t(\tau - \tau_0). \tag{52}$$

The full $\frac{\partial^2 \Psi_t}{\partial \mathbf{e}_t^2}$ is reconstructed using Eqns. 28 and 29, and $\frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}}$ obtained with Eqn. 49. PSD-ness is preserved throughout because both transformations left- and right-multiply using the same rectangular matrix. Even if a matrix contains negative singular values, they self-cancel during the second multiply. Thus, the same filtering process can be applied to $-\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$ from Eqn. 46, albeit with the *positive* eigenvalues filtered, due to the leading negative. Further details on $\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$ are provided in the supplement.

The eigenanalysis of our energy is now complete.

5 IMPLEMENTATION AND RESULTS

For linear system assembly and solution, we used Eigen [Guennebaud et al. 2010], but wrote custom routines to accelerate sparse matrix assembly. We used Eigen's conjugate gradient solver, but similar to previous works [Daviet 2020; Fei et al. 2017; Lesser et al. 2022] added a per-strand block-diagonal preconditioner. We found a full Cholesky decomposition of each strand block to be highly effective, as the small block sizes (~300 × 300) allowed factorizations and back-solves to parallelize cleanly. This accelerated solves by between 6× and 1130×, bringing solver times to parity with system assembly and collision processing (Table 3). Detailed timing breakdowns are §5 of our supplementary document. All linear system solves were run to $\epsilon = 1 \times 10^{-8}$ on a 4-core, 4.6 GHz Intel Xeon W-2225 with 128 GB memory.

We used position-based BDF-1 with 3 Newton iterations for time integration. Kinematic collisions were handled with constraint projection [Baraff and Witkin 1998; Tamstorf et al. 2015], and all other collisions were resolved using penalty forces. Material parameters were set using Young's modulus (E) and Poisson's ratio (ν). The strand constants were set as:

$$\mu_s = E\pi r^2 \qquad \mu_b = \frac{E}{4}\pi r^2 \qquad \mu_t = \frac{E}{4(1+\nu)}\pi r^4. \tag{53}$$

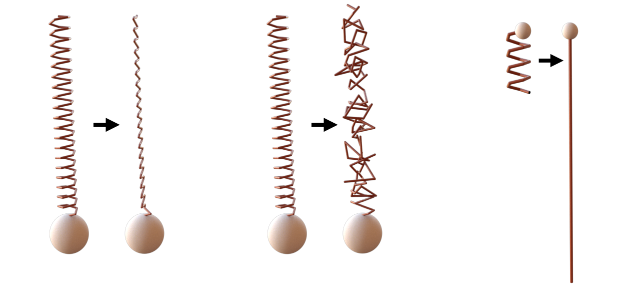
Table 1. We compress a coil (Fig. 5a) and simulate at $\Delta t = 1/30$ using Discrete Elastic Rods (DER) and Lifted Curls (LC). Multiple Hessian filtering strategies were applied to DER, including no filtering, Gauss-Newton for bend/twist/stretch, Gauss-Newton for bend/twist but full filtering for stretch, and full filtering for bend/twist/stretch. Past 40% compression, all DER filtering strategies failed. Our LC energy always recovers, even at 99.999% compression.

Curl	Discrete Elastic Rods (DER)				Lifted Curls
Compression	Unfiltered	Gauss-Newton (GN)	GN + Stretch Filter	Filtered	(Ours)
30%	unstable	unstable	stable	stable	stable
40%	unstable	unstable	stable	unstable	stable
60%	unstable	unstable	unstable	unstable	stable
99.999%	unstable	unstable	unstable	unstable	stable

5.1 Results

5.1.1 Kinematic Tests. We ran kinematic displacement tests comparing our Lifted Curls (LC) energy to the implicit implementation of Discrete Elastic Rods (DER) [Bergou et al. 2010] provided by the original authors (Bergou) [Jawed et al. 2018]. Unless stated otherwise, the tests were performed on a coil with 100 vertices, spaced 6 mm apart, with 1 mm radius strands forming curls of 5 mm radius. Collisions and gravity are deactivated. All simulations used $\Delta t = 1/30$ s, 3 Newton iterations, $E = 3.9e9$ g/cm s^2, $\nu = 0.48$ and $\rho = 1.3$ g/cm^3.

To facilitate comparison, we tested multiple Hessian filtering strategies for DER. These included *Unfiltered* (the original code), *Gauss-Newton* where only the outer product terms were used, *GN + Stretch Filter* where bending and twisting were approximated using their outer product terms but the stretching term was fully filtered, and *Filtered* where all Hessian terms were filtered.



(a) Compression Test (b) Jitter Test (c) Straighten Test

Fig. 5. Kinematic displacement tests comparing the stability of our model and Discrete Elastic Rods.

5.1.2 Compression Test. In Table 1 and Fig. 5a, we compressed a coil by 30% along the x-axis (inset, left). LC quickly recovered, as well as the *GN + Stretch Filter* and *Filtered* versions of DER (see Table 1 and video). Under 40% compression, LC recovered while only the *GN + Stretch Filter* strategy succeeded. Beginning at 60% compression, all DER strategies failed. For this case, the *GN + Stretch Filter* strategy succeeded if timestep was set to $\Delta t = 1/3{,}000$, a **100**× difference in stability. The *Filtered* strategy succeeded at $\Delta t = 1/300{,}000$, a **10,000**× difference in stability. We ran compression tests up to 99.999%, and our LC energy always recovered with $\Delta t = 1/30$.

We found that DER implementations commonly set the maximum number of Newton iterations to anywhere from 50 [Jawed et al. 2018] to 10,000 [Fei et al. 2019]. If we view Newton iterations as

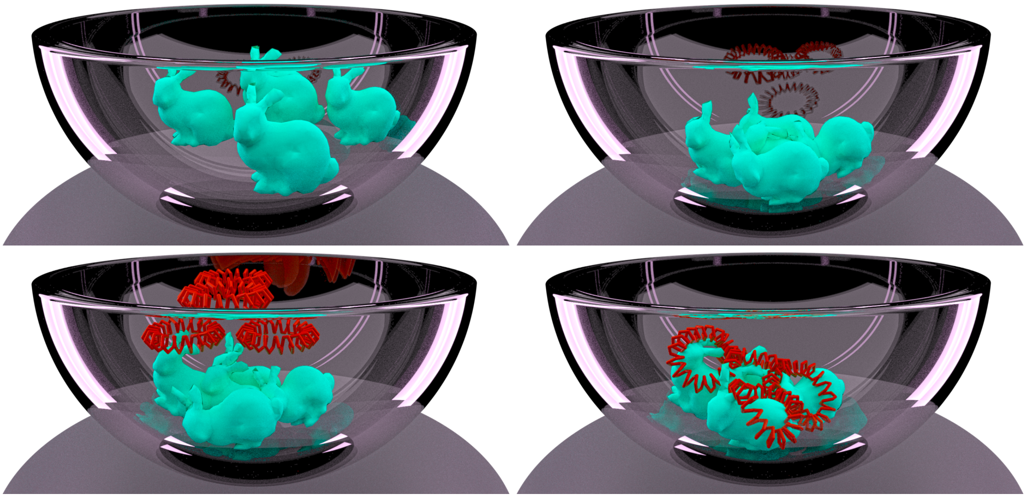


Fig. 6. Our strand model integrates cleanly with existing simulators. We drop five volumetric bunnies into a bowl, followed by four spiral hair ties. The hair ties are simulated using our Lifted Curls energy, and entangle realistically with the bunny ears.

approximate substeps in time, the numbers roughly align with the stability behavior we observed. However, these Newton iteration counts are outside our acceptable performance range.

We hypothesize that *GN + Stretch Filter* yielded slightly better stability because most of the stiffness lives in the stretching term, so it must be carefully filtered. In contrast, if bending and twisting are far from equilibrium, the outer product yields a more conservative descent direction, so convergence becomes slower but more stable.

5.1.3 Jitter Test. In Table 2 and Fig. 5b, we kinematically jittered each vertex of the coil by varying amplitudes. When each vertex was jittered by 1 mm, both LC and DER quickly recovered (see video). For larger jitters (2 mm) all DER filtering strategies diverged. We tested up to a jitter of 5 mm, where LC quickly recovered. The *GN + Stretch Filter* DER strategy again only stabilized after setting $\Delta t = 1/3{,}000$, while the *Filtered* strategy performed slightly worse than in previous tests and needed a timestep of $\Delta t = 1/3{,}000{,}000$.

5.1.4 Straightening Test. To show that our energy only requires the *rest state* to be non-planar and handles straight *deformed states* robustly, we kinematically straightened a coil of 20 vertices (Fig. 5c). Gravity was enabled for this example. LC recovered under $\Delta t = 1/30$, while all DER filtering strategies failed (see video). Consistent with prior tests, the *GN + Stretch Filter* strategy only succeeded at $\Delta t = 1/3{,}000$, while *Filtered* succeeded at $\Delta t = 1/300{,}000$.

5.1.5 Strands, Shells, and Volumes. Past systems have unified volumes, shells, and strands [Chang et al. 2019; Martin et al. 2010], but we found integrating our LC energy into an existing FEM solver was straightforward. If linear FEM forces are already represented as couplings between the vertices of simplicies, our approach simply maps additional strain energies onto existing infrastructure. Fig. 6 shows a preliminary integration of our strand model with an existing volume solver. The simulator already handled edge-edge contacts, so strand collisions just become another instance of this contact case.

Table 2. We jitter the vertices of a coil (Fig. 5b) and simulate at $\Delta t = {}^{1}/_{30}$ using Discrete Elastic Rods (DER) and Lifted Curls (LC). Multiple Hessian filtering strategies were applied to DER (see Table 1 and §5.1.2). When jitter amplitudes exceed 1 mm, all DER filtering strategies fail. Our LC energy recovers, even more for much larger 5 mm jitters.

Jitter Amplitude	Discrete Elastic Rods (DER)				Lifted Curls (Ours)
	Unfiltered	Gauss-Newton (GN)	GN + Stretch Filter	Filtered	
0.5 mm	unstable	unstable	stable	stable	stable
1 mm	unstable	unstable	stable	unstable	stable
2 mm	unstable	unstable	unstable	unstable	stable
5 mm	unstable	unstable	unstable	unstable	stable

5.1.6 Hairball Test. Similar to previous works, we place hairs along the top of a 15 cm radius sphere. Previous works modelled straight or wavy hairs with thirty [Daviet 2020; Kaufman et al. 2014], fifty [Michels et al. 2015], or sixty [Han et al. 2019] segments, but we found that at least 100 vertices are needed to adequately sample tight curls. We generated hairs using the parameters from §5.1.1, consistent with the measurements from De La Mettrie et al. [2007].

We adopted the *wisp* (a.k.a. clump) approximation common in film production [Butts et al. 2018; Thyng et al. 2017], where each simulated strand is a proxy for multiple hairs [Bertails et al. 2003; Choe et al. 2005; Ward and Lin 2003]. Tightly coiled hair commonly forms such wisps (see inset photo), and we linearly decreased the wisp radius approaching the tip to maintain consistency with real-world references. We uniformly distributed 200K hairs between the wisps, recomputed mass and μ_* in Eqn. 53 at each vertex according to the wisp radius, and set $\Delta t = {}^{1}/_{30}$.

DER exploded with this Δt, so we only show LC results in Fig. 1 and 7. We simulated $E = 6e^{8}$ $^{g}/_{cm\,s^2}$ for 2000 wisps to account for voids between hairs, making it softer than the $E = 1e^{9} - 1e^{11}$ $^{g}/_{cm\,s^2}$ range of single-strand stiffnesses from the literature [Bertails et al. 2005; Daviet 2020]. Similarly, we reduced density by 97% to $\rho = 0.033$ $^{g}/_{cm^3}$. Based on sample measurements, the 2000 wisp simulations used a base wisp radius of 3.54 mm, and tip radius of 0.71 mm. For the 4000 and 8000 wisp simulations, we proportionally reduced the radii while progressively doubling E, as hairs cluster along the centerline. This yields $E = 4e^{10}$ $^{g}/_{cm\,s^2}$ for 128K wisps (i.e. the resolution of individual hairs), which is within previously reported ranges.

The appearance of tightly coiled hair is qualitatively different from the straight of wavy hair that usually appears in computer graphics, so we provide photo references in Fig. 8. We compare the "clumped" and "picked out" looks from our simulations against the tightly coiled hair from the photos of Leal Alexander [2023]. For the clumped look, our results qualitatively match the cylindrical curls in the photo. The diffuse appearance of the picked out look is also reflected in our results. Both photos feature hair from the the same model, so the visual differences arise from coil coherence.

5.2 Limitations and Future Work

We have presented a robust model for tightly coiled hair. Our energy is consistently quadratic in its primary variable, making it analogous to ARAP [Sorkine and Alexa 2007] or co-rotational [Etzmuß et al. 2003] models for shells and volumes. Similar to those models, ours yields a linearly compliant look, so increasing non-linearity while retaining robustness is future work.

Fig.	Wisps	DoFs	Collision Handling	System Assembly	PCG	Total Time
1,7	2000	806K	35.4%	32.3%	32.4%	17.7s
7	4000	1.61M	35.9%	33.6%	30.5%	27.6s
1,7	8000	3.22M	34.9%	33.9%	31.2%	57.0s

Table 3. Timings for $\Delta t = 1/30$ timesteps of BDF-1 with 3 Newton iterations. Collision Handling includes detection and response, and System Assembly is time spent in matrix assembly, minus collisions.

Our energy assumes isotopic, non-straight hair, so an anisotropic, quadratic model for straight hair is still future work. While our model was designed for curly hair, we have found that it is also stable for straight hair ($\theta_0 \approx \pi - 1e^{-8}$ and $\tau_0 \approx 1e^{-8}$). Consistent with other works [Sánchez-Banderas et al. 2020], the main source of instability is when edges become very short and the stretching energy becomes ill-conditioned.

The scaling behavior of tightly coiled strands under continuous collision detection (CCD) [Kaufman et al. 2014] and frictional contact [Daviet 2020] is also worth further investigation. In lieu of our wisp approach, Eulerian-on-Lagrangian methods [Sueda et al. 2011] could instead be used to resolve highly correlated curls. Grooming tightly coiled hair to send to our simulator also remains challenging [Ogunseitan 2022; Patrick et al. 2004]. Finally, exploring the "looks" of hair that are possible, beyond the "clumped" and "frizzy" looks in Fig. 1, is an interesting direction for future investigation.

ACKNOWLEDGMENTS

We thank the reviewers for their comments. This work was supported by Adobe, the Bungie Foundation, Teng and Han Family Fund, and National Science Foundation (IIS-2132280). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Leal Alexander. 2023. *Instagram*. https://www.instagram.com/curlygallal/

Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. 1992. A simple method for extracting the natural beauty of hair. In *Proc. of SIGGRAPH*. 111–120.

TF Banchoff. 1982. Global geometry of polygons. III. Frenet frames and theorems of Jacobi and Milnor for space polygons. *Rad Jugoslav. Akad. Znan. Umjet* 396 (1982), 101–108.

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proc. of SIGGRAPH*. 43–54.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Trans. Graph.* 29, 4 (2010), 1–10.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. *ACM Trans. Graph.* 27, 3 (2008), 1–12.

F. Bertails. 2009. Linear time super-helices. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 417–426.

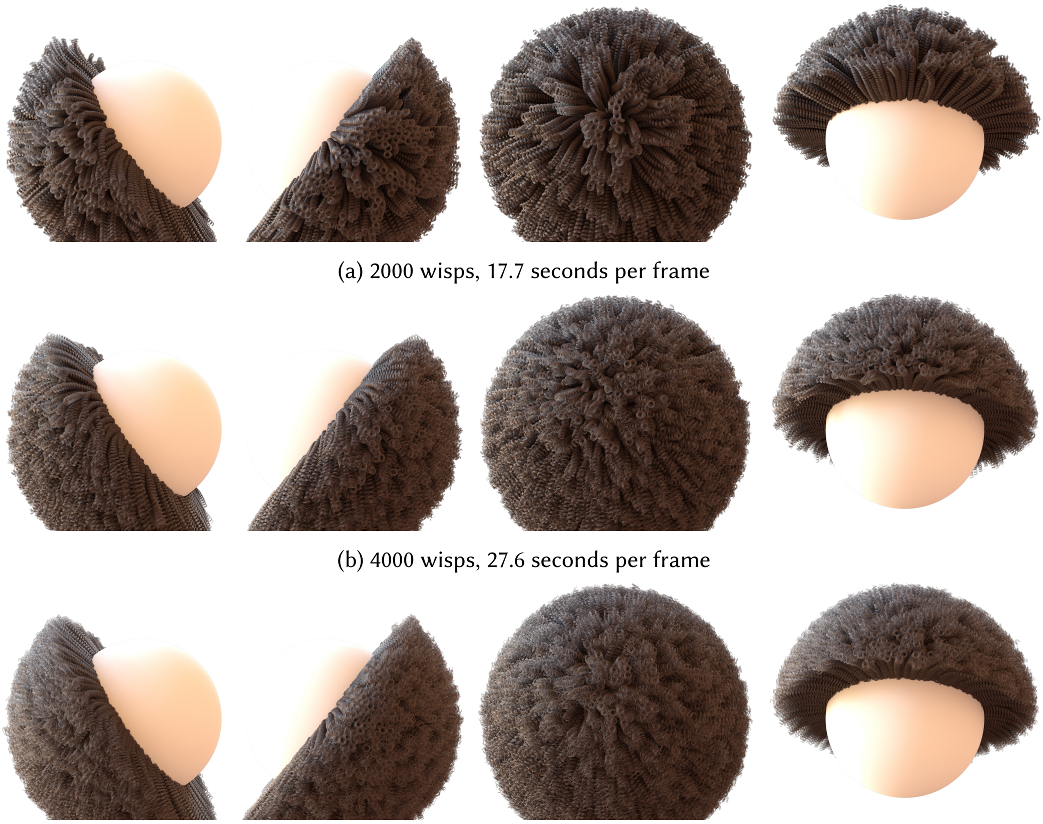
F. Bertails, B. Audoly, MP Cani, B. Querleux, F. Leroy, and JL Lévêque. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph.* 25, 3 (2006), 1180–1187.

F. Bertails, B. Audoly, B. Querleux, F. Leroy, JL Lévêque, and MP Cani. 2005. Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics short papers*. Eurographics.

F. Bertails, TY Kim, MP Cani, and U. Neumann. 2003. Adaptive wisp tree-a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proc. of Symposium on Computer Animation*.

H. Bryant and CE Porter. 2012. Hair Ethnicity. In *Practical Modern Hair Science*, T. Evans and RR Wickett (Eds.). Allured Business Media, Chapter 6, 193–222.

James R Bunch, Christopher P Nielsen, and Danny C Sorensen. 1978. Rank-one modification of the symmetric eigenproblem. *Numer. Math.* 31, 1 (1978), 31–48.



(a) 2000 wisps, 17.7 seconds per frame

(b) 4000 wisps, 27.6 seconds per frame

(c) 8000 wisps, 57.0 seconds per frame

Fig. 7. **Left to right:** hairball rotations to the left, right, towards the viewer, and away from the viewer. **Top to bottom:** 2000 wisps naturally form a "clumped" look. Maintaining total hairs at 200K, while increasing wisps to 4000 and 8000, gives progressively more diffuse, "picked out" looks. Images are high-res; please zoom in to see details.

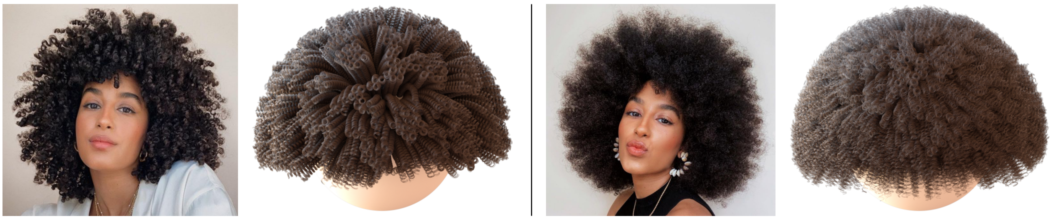


Fig. 8. **Left pair:** Comparison to real-world "clumped" look. **Right pair:** Comparison to real-world "picked out" look. Images are from Alexander [2023].

A. Butts, B. Porter, D. Van Gelder, M. Hessler, V. Krishna, and G. Monheit. 2018. Engineering full-fidelity hair for Incredibles 2. In *ACM SIGGRAPH Talks*. 1–2.

J. Chang, F. Da, E. Grinspun, and C. Batty. 2019. A Unified Simplicial Model for Mixed-Dimensional and Non-Manifold Deformable Elastic Objects. *Proc. of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–18.

Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. 2005. Simulating complex hair with robust collision handling. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 153–160.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but responsive cloth. *ACM Trans. Graph.* 21, 3 (2002), 604–611.

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. 2014. Yarn-level simulation of woven cloth. *ACM Trans. Graph.* 33, 6 (2014), 1–11.

Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Trans. Graph.* 39, 4 (2020), 61–1.

R. De La Mettrie, D. Saint-Léger, G. Loussouarn, A. Garcel, C. Porter, and A. Langaney. 2007. Shape variability and classification of human hair: a worldwide approach. *Human biology* 79, 3 (2007), 265–281.

C. Deul, T. Kugelstadt, M. Weiler, and J. Bender. 2018. Direct position-based solver for stiff rods. In *Computer Graphics Forum*, Vol. 37. 313–324.

D. Eberle. 2022. Personal Communication.

Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. 2003. A fast finite element solution for cloth modelling. In *Proc. of Pacific Graphics.* 244–251.

Y. Fei, C. Batty, E. Grinspun, and C. Zheng. 2019. A multi-scale model for coupling strands with shear-dependent liquid. *ACM Trans. Graph.* 38, 6 (2019), 1–20.

Y. Fei, H. Maia, C. Batty, C. Zheng, and E. Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Trans. Graph.* 36, 4 (2017), 1–17.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations.* JHU press.

G. Gornowicz and S. Borac. 2015. Efficient and stable approach to elasticity and collisions for hair animation. In *Proc. of the Symp. on Digital Production.* 41–49.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

X. Han, T. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. 2019. A hybrid material point method for frictional contact with diverse materials. *Proc. of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–24.

Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic simulation of curly hair. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 63–71.

M Khalid Jawed, Alyssa Novelia, and Oliver M O'Reilly. 2018. *A primer on the kinematics of discrete elastic rods.* Springer.

D. Kaufman, R. Tamstorf, B. Smith, J. Aubry, and E. Grinspun. 2014. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. Graph.* 33, 4 (2014), 1–12.

T. Kim. 2020. A finite element formulation of Baraff-Witkin cloth. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 1–9.

T. Kim, F. De Goes, and H. Iben. 2019. Anisotropic elasticity for inversion-safety and element rehabilitation. *ACM Trans. Graph.* 38, 4 (2019), 1–15.

T. Kim and D. Eberle. 2022. Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH Courses.* 1–259.

T. Kugelstadt and E. Schömer. 2016. Position and orientation based Cosserat rods.. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 169–178.

S. Lesser, A. Stomakhin, G. Daviet, J. Wretborn, J. Edholm, N. Lee, E. Schweickart, X. Zhai, S. Flynn, and A. Moffat. 2022. Loki: a unified multiphysics simulation framework for production. *ACM Trans. Graph.* 41, 4 (2022), 1–20.

Huancheng Lin, Floyd M Chitalu, and Taku Komura. 2022. Isotropic ARAP energy using Cauchy-Green invariants. *ACM Trans. Graph.* 41, 6 (2022), 1–14.

S. Martin, P. Kaufmann, M. Botsch, E. Grinspun, and M. Gross. 2010. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph.* 29, 4 (2010), 1–10.

A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph.* 28, 3 (2009), 1–6.

A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* 30, 4 (2011), 1–12.

D. Michels, P. Mueller, and G. Sobottka. 2015. A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes. *Computers & Graphics* 53 (2015), 136–146.

Sofya Ogunseitan. 2022. Space Rangers with Cornrows: Methods for Modeling Braids and Curls in Pixar's Groom Pipeline. In *ACM SIGGRAPH Talks.* 1–2.

Dinesh K Pai. 2002. Strands: Interactive simulation of thin solids using cosserat models. In *Computer graphics forum*, Vol. 21. Wiley Online Library, 347–352.

Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.* 38, 4 (2019), 1–15.

Deborah Patrick, Shaun Bangay, and Adele Lobb. 2004. Modelling and rendering techniques for african hairstyles. In *Proc. of Afrigraph.* 115–124.

RM Sánchez-Banderas, A. Rodríguez, H. Barreiro, and MA Otaduy. 2020. Robust eulerian-on-lagrangian rods. *ACM Trans. Graph.* 39, 4 (2020), 59–1.

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. *ACM Trans. Graph.* 27, 3 (2008), 1–11.

B. Smith, F. De Goes, and T. Kim. 2018. Stable neo-hookean flesh simulation. *ACM Trans. Graph.* 37, 2 (2018), 1–15.

B. Smith, F. De Goes, and T. Kim. 2019. Analytic eigensystems for isotropic distortion energies. *ACM Trans. Graph.* 38, 1 (2019), 1–15.

Carlota Soler, Tobias Martin, and Olga Sorkine-Hornung. 2018. Cosserat rods with projective dynamics. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 137–147.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proc. of Symposium on Geometry processing*, Vol. 4. 109–116.

G. Sperl, RM Sánchez-Banderas, M. Li, C. Wojtan, and MA Otaduy. 2022. Estimation of yarn-level simulation models for production fabrics. *ACM Trans. Graph.* 41, 4 (2022), 1–15.

Jonas Spillmann and Matthias Teschner. 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer animation.* 63–72.

S. Sueda, GL Jones, D. Levin, and D. Pai. 2011. Large-scale dynamic simulation of highly constrained strands. *ACM Trans. Graph.* 30, 4 (2011), 1–10.

R. Tamstorf, T. Jones, and S. McCormick. 2015. Smoothed aggregation multigrid for cloth simulation. *ACM Trans. Graph.* 34, 6 (2015), 1–13.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 181–190.

M. Thyng, C. Evart, T. Jones, and A. McAdams. 2017. The art and technology of hair simulation in Disney's Moana. In *ACM SIGGRAPH Talks.* 1–2.

N. Umetani, R. Schmidt, and J. Stam. 2015. Position-based elastic rods. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 21–30.

Andre Walker. 1997. *Andre Talks Hair.* Simon & Schuster.

K. Ward and MC Lin. 2003. Adaptive grouping and subdivision for simulating hair dynamics. In *Proc. of Pacific Graphics.* IEEE, 234–243.

K. Ward, M. Simmons, A. Milne, H. Yosumi, and X. Zhao. 2010. Simulating Rapunzel's hair in Disney's Tangled. In *ACM SIGGRAPH Talks.* 1–1.

FJ Wortmann, G Wortmann, and T Sripho. 2020. Why is hair curly?—Deductions from the structure and the biomechanics of the mature hair shaft. *Experimental dermatology* 29, 3 (2020), 366–372.

H. Wu and T. Kim. 2023. An Eigenanalysis of Angle-Based Deformation Energies. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation.*

Kui Wu and Cem Yuksel. 2016. Real-Time Hair Mesh Simulation. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Redmond, Washington). 59–64.

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair Meshes. *ACM Trans. Graph.* 28, 5 (2009).

J. Zehnder, S. Coros, and B. Thomaszewski. 2021. SGN: Sparse Gauss-Newton for Accelerated Sensitivity Analysis. *ACM Trans. Graph.* 41, 1 (2021), 1–10.