FACIAL RIG

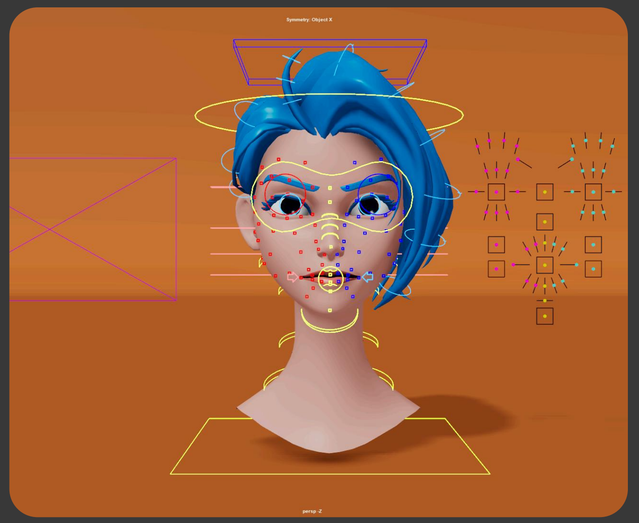
11/25/2025
Yanshu Liang
yanshu_liang@outlook.com

INTRO

Reference & Model Preparation

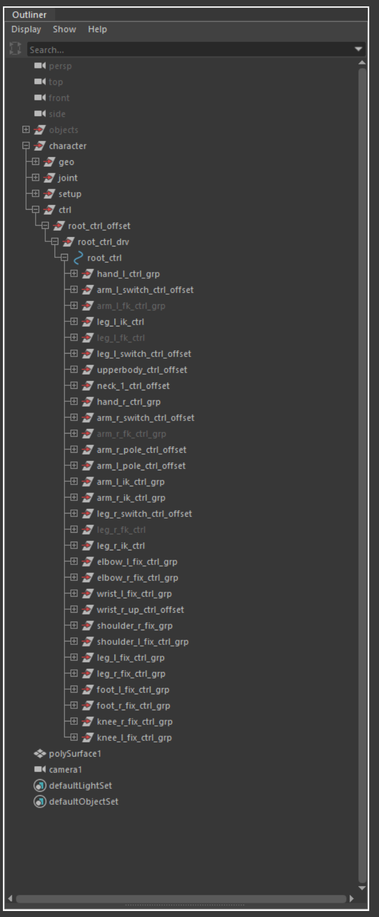
My interest in facial rigging began with Detroit: Become Human. In story-driven 3D games like this, characters communicate emotion through subtle and expressive facial movements, which rely heavily on a robust and flexible facial rigging system. With this inspiration, I started from standard facial rigging techniques and expanded them with more detailed controls and custom tools to support richer expressions and higher-quality animation.

Rigging Logic



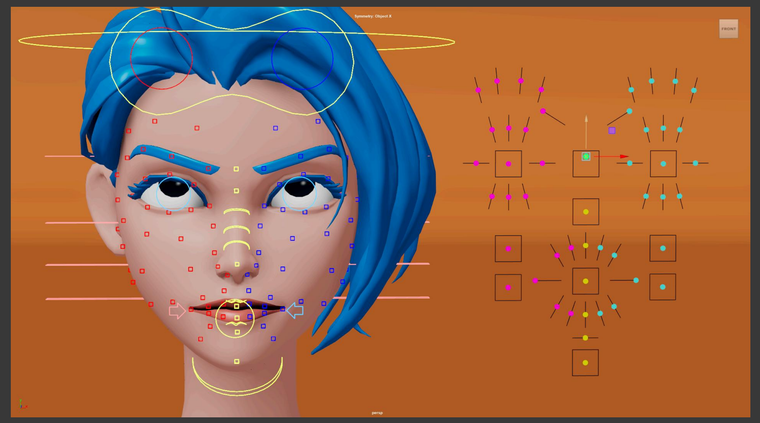
In this facial rigging project, I began by building a complete set of foundational facial joints and creating individual controllers for each of them. On top of this system, I added broader regional controls, such as an eye aim setup and a global mouth controller. At the same time, I created specific expressions using blendshapes to speed up the animation workflow. Finally, I integrated all joint-based controls and blendshape deformation into a single control panel through SDKs, making facial animation more streamlined and efficient.

Hierarchy designed for Facial rig.

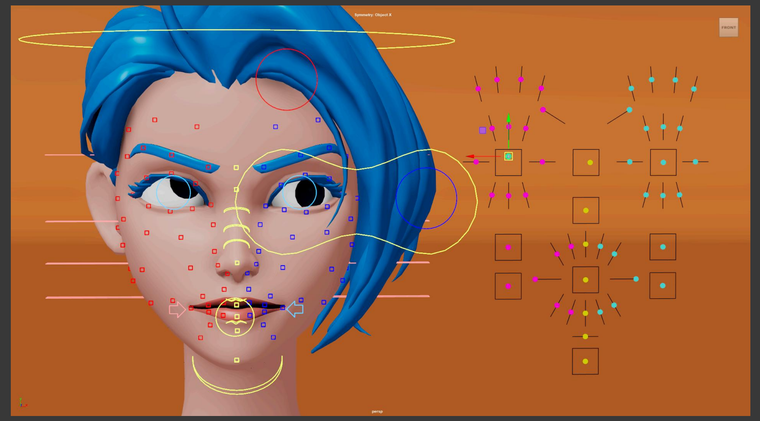


Eye Aim Controller

The aim control drives the overall orientation of both eyeballs. On top of this setup, individual eye controllers allow for fine-tuning adjustments for each eye, providing detailed control while maintaining a consistent global gaze direction.



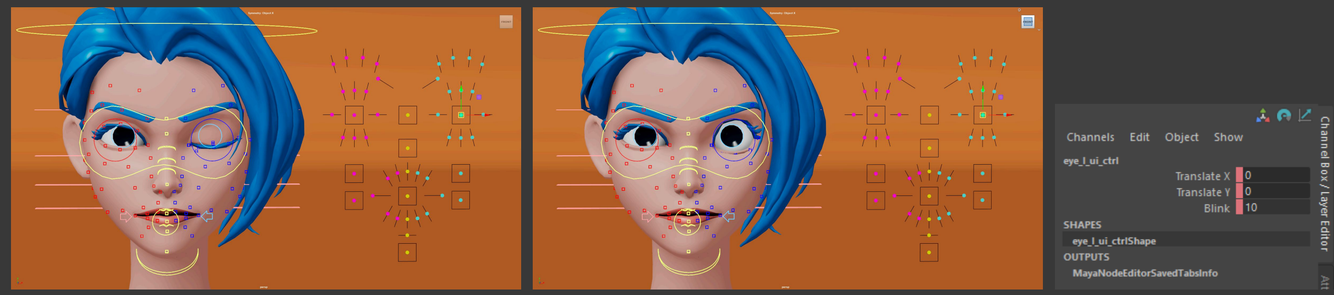
eye aim control



individual eye controller

Blink

A blink attribute was added to the eye controller in the control panel and connected to the original eyelid controllers. This setup allows the animator to easily close, open, or widen the eyes through a single parameter, improving both the efficiency and consistency of eye animation adjustments.



Individual eye control

The scale and translate attributes of the entire eye were connected to the eye controller, allowing the controller to directly manipulate the overall size and position of the eyeball. This setup provides flexible eye adjustments while maintaining a stable rig structure.

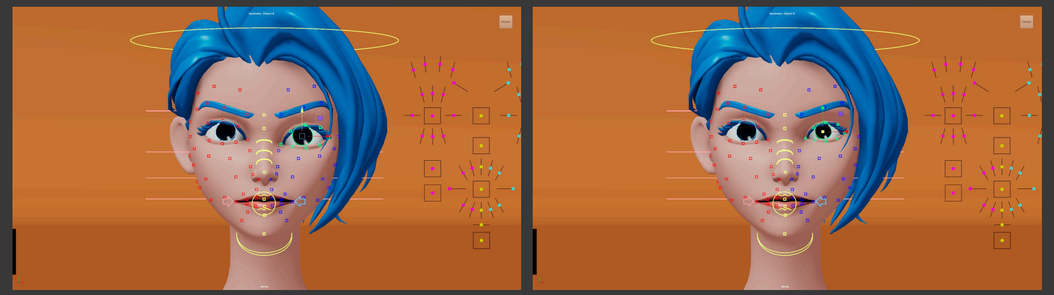


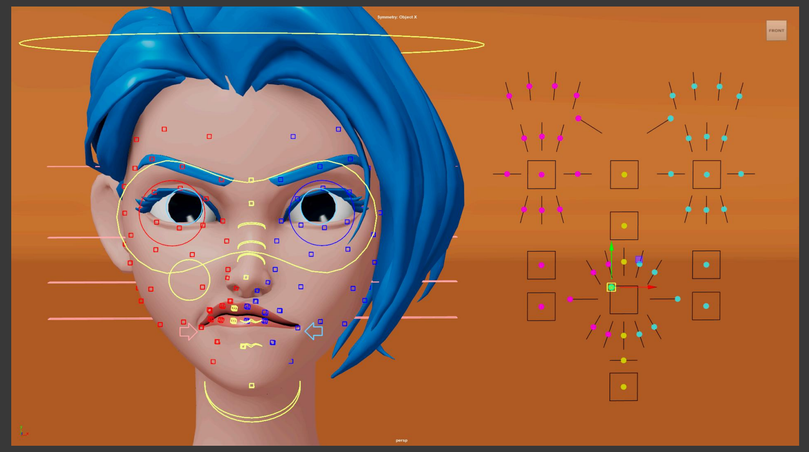
Figure 3

Mouth controller

Whole Controller

For the mouth rig, I created a master mouth controller. Each mouth control was placed inside a dedicated follow_control group, and the pivot of each group was moved to the center of the oral cavity. This setup allows all mouth controls to rotate around a shared central point, enabling cohesive and intuitive overall mouth manipulation.

Facial Tool Bag Plugin
__Center Control Tool__

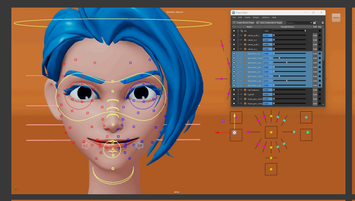


Use control panel to control the movement of the whole mouth

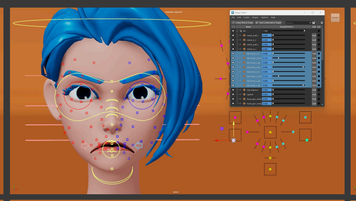
Blendshape System

To streamline the creation of specific mouth expressions, I sculpted a set of blendshape targets for the lip and mouth region and connected them to the facial controllers. Using Set Driven Keys, I established the drive relationships so that the control panel parameters can directly manipulate the mouth-corner expressions and related blendshape channels.
This setup provides a more intuitive and efficient workflow for shaping detailed facial expressions.

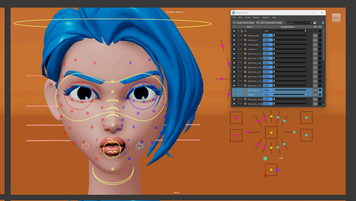
Mouth corner up



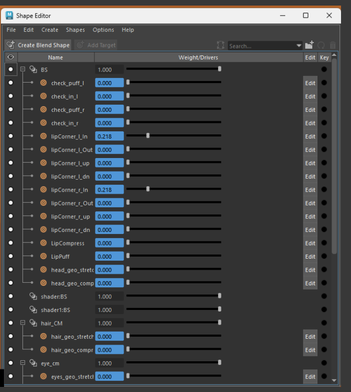
Mouth corner down



Pout



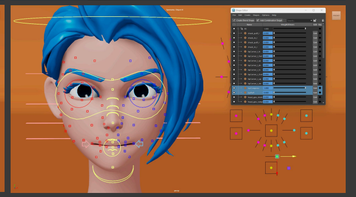
BS Shape Editor



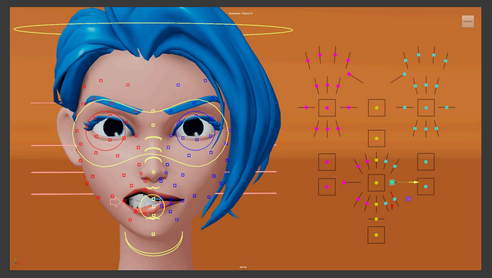
Mouth corner out

Mouth corner in

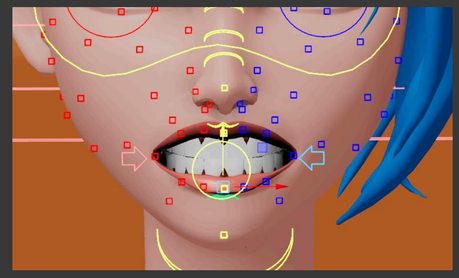
Lip press



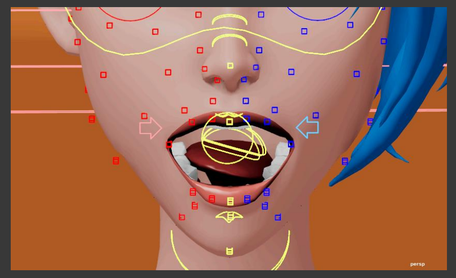
Sub-controllers



Zipper Mouth:
Controls the closure of the lips starting from the corners.



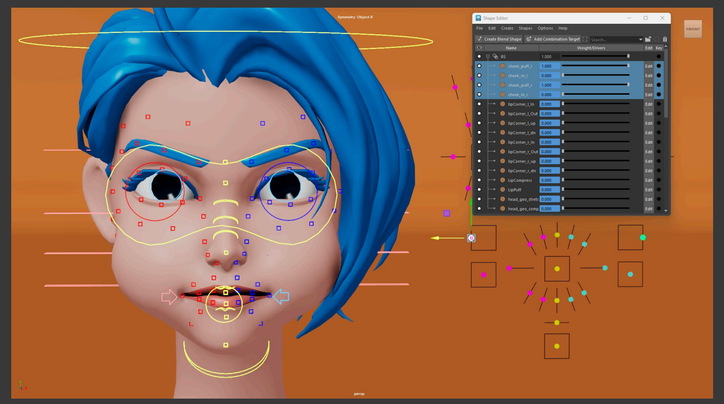
Separate Lip Controllers:
Independently control the movement of the upper and lower lips.

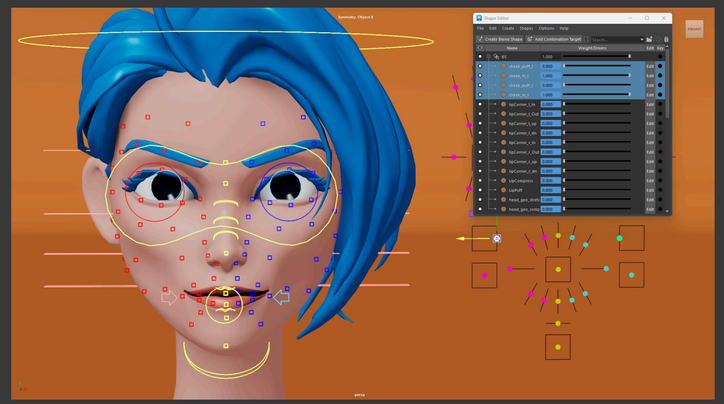


Tongue Controller:
Controls the movement of the tongue within the oral cavity.

Cheek

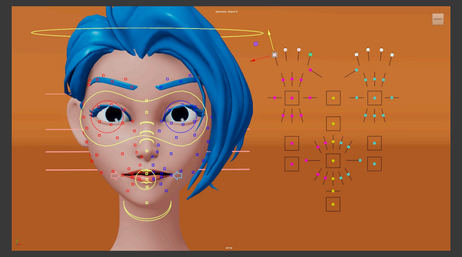
I created puff and cheek-suck blendshape targets for the cheek area and connected them to the control panel. This allows animators to easily adjust the degree of cheek inflation or contraction through the panel, enabling more detailed and expressive facial animation.

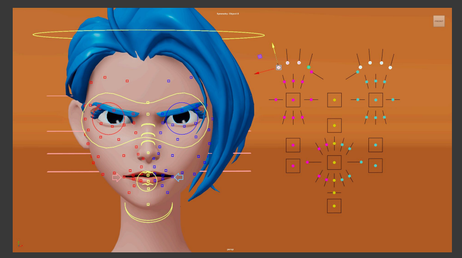




Eyebrows

Control Panel

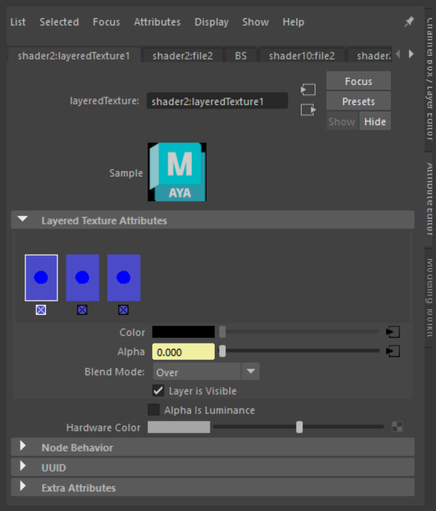




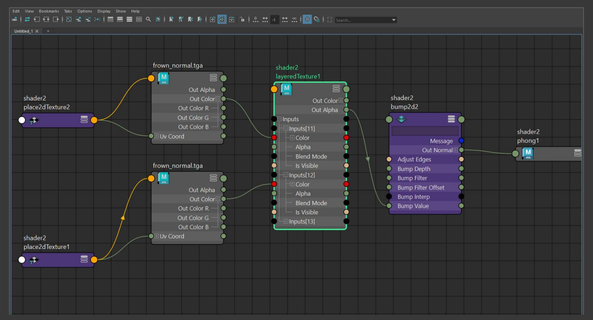
Control the eyebrow movement through the control panel.

normal map:
bake form high poly
Frown expression model

Frown

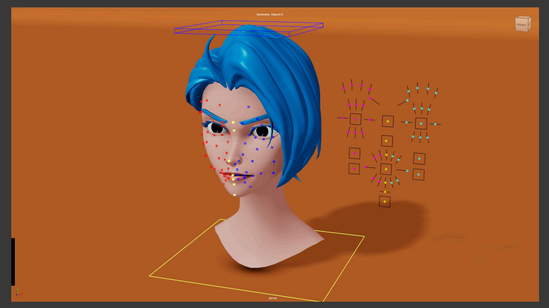


I connected two normal maps to a Layer Texture node and linked the blendshape node from the original model to the alpha channel of the Layer Texture. This setup allows one of the normal maps to appear or disappear automatically based on the activation of the blendshape.

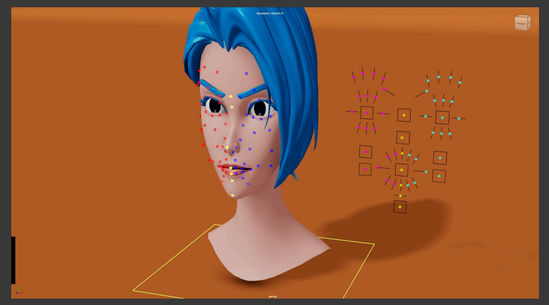


Head Stretch

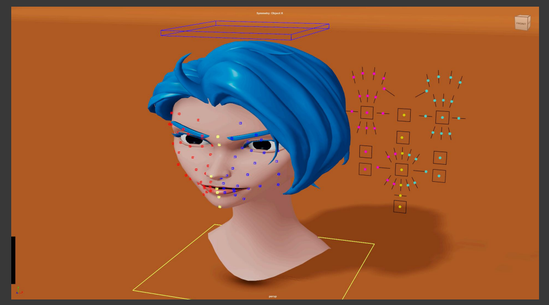
Using Maya’s built-in Lattice Deform function, I applied compress and stretch deformations to the entire model group and connected the output to the original model’s blendshape node. This setup allows for controlled overall compression and elongation of the model.



orinigal model



stretched model

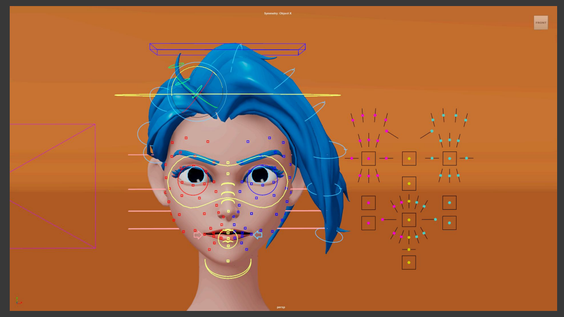


compressed model

This tool generates a set of “follow” controllers on the original mesh that deform along with it. It works by duplicating the selected controllers, creating offset and driver groups, and then using a closestPointOnMesh node to find the nearest vertex on the mesh. Each follow controller is constrained to that vertex, ensuring it moves accurately with the mesh’s deformation. This allows riggers to create controllers that naturally follow facial or body movements.

Hair Spline IK

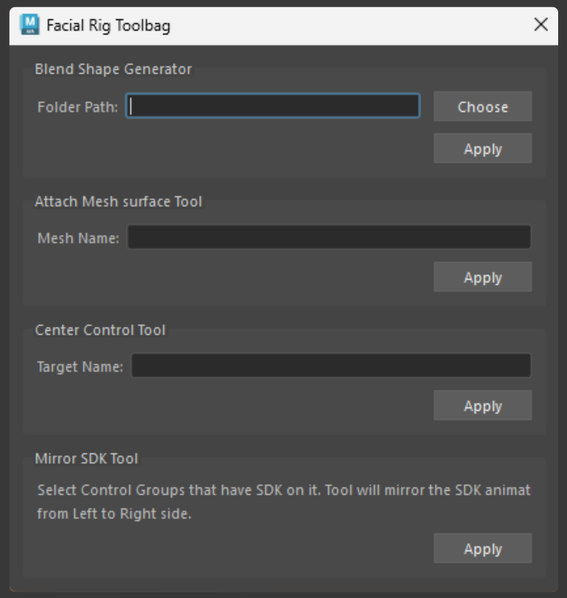
For the hair rig, I used a spline IK setup, which provides both FK and IK functionality. This approach allows the hair to twist and deform more naturally, resulting in smoother and more believable motion.

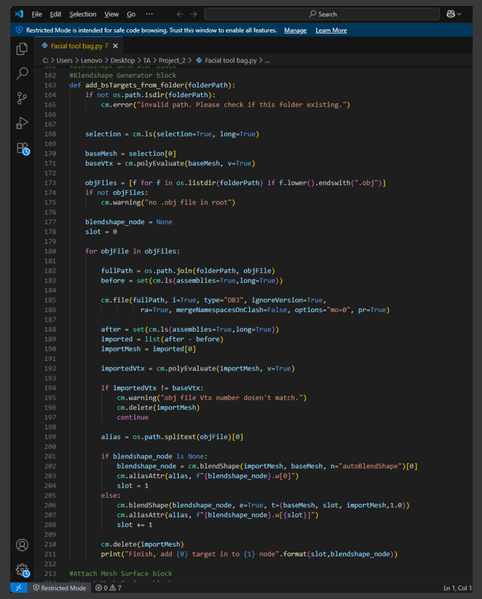


Plugin UI: Facial Tool Bag

Overview

Facial Tool bagis a Maya facial-rigging assistant toolset developed with PySide2 and Maya Python API.
It contains four core modules—BlendShape Batch Generator, Attach-to-Surface Controller Tool, Center Control Tool, and SDK Mirror Tool.
All features are integrated into a Qt interface inside Maya, designed to streamline facial rigging workflows, reduce repetitive tasks, and improve production efficiency.





Blend Shape Generator (BSG)

Imports OBJ files from a folder and creates blend shape targets on a selected base mesh, automatically naming each target.

Attach Mesh Surface Tool (AMS)

Attaches selected controllers to the surface of a target mesh, making them follow the nearest vertex for accurate deformation.

Center Control Tool (CC)

Centers selected controller groups on a target object and connects their translation and rotation to it.

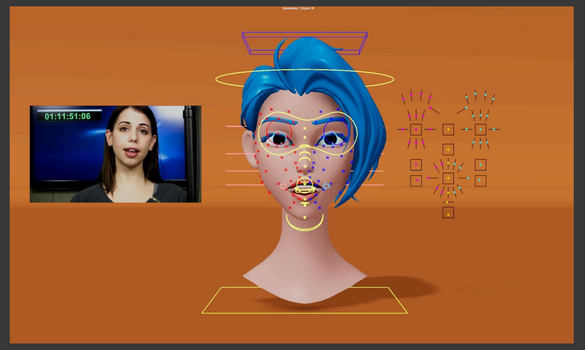
Mirror SDK Tool (MSDK)

Mirrors Set Driven Key (SDK) animations from one side to the other, with optional axis flipping for translation or rotation.

Motion Capture Retargeting

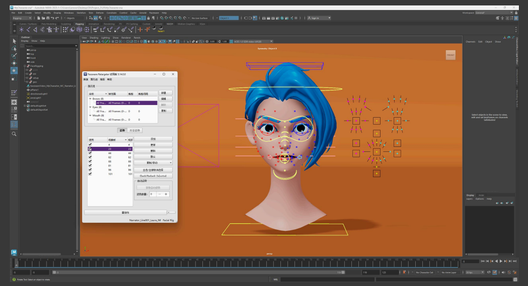
Overview

In the final stage, I utilized Faceware to assist with animation retargeting. Expression sets were first defined within Faceware, and specific keyframes (K frames) were marked. The system then automatically generated the retargeted animation, enabling efficient transfer of facial movements across different models or characters.



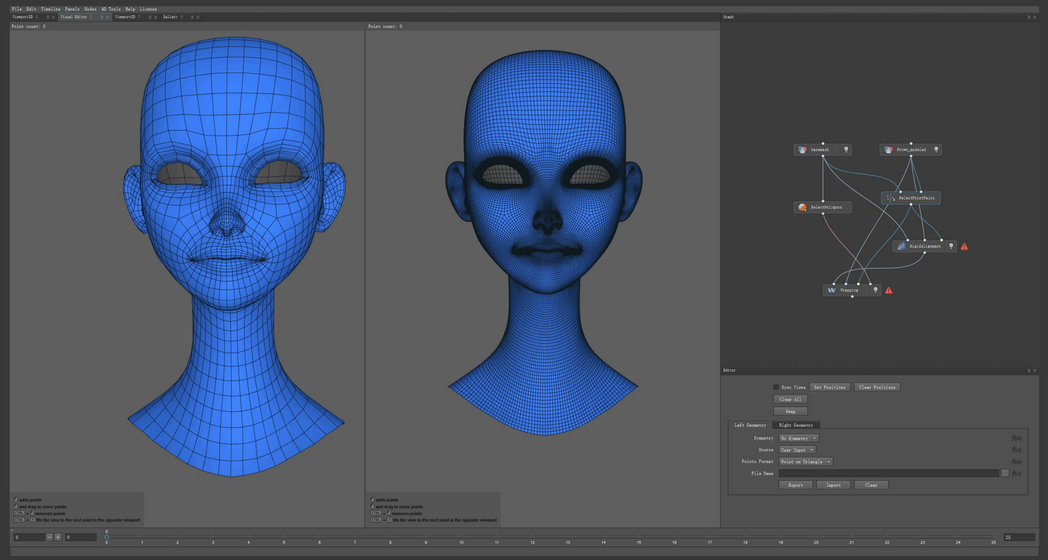
Match the motion capture files provided by Faceware

Faceware



Warp 4D

When creating blendshape heads for specific expressions, in addition to manual sculpting for fine-tuning, I utilized existing expressions from Warp4D for retopology, ensuring that the new expression models maintain the same polygon count as the original mesh. This approach preserves compatibility for animation driving while providing additional expression options and detailed control.



compressed The high-resolution frown expression was retopologized to match the polygon count of the low-resolution model while preserving the original shape as closely as possible.