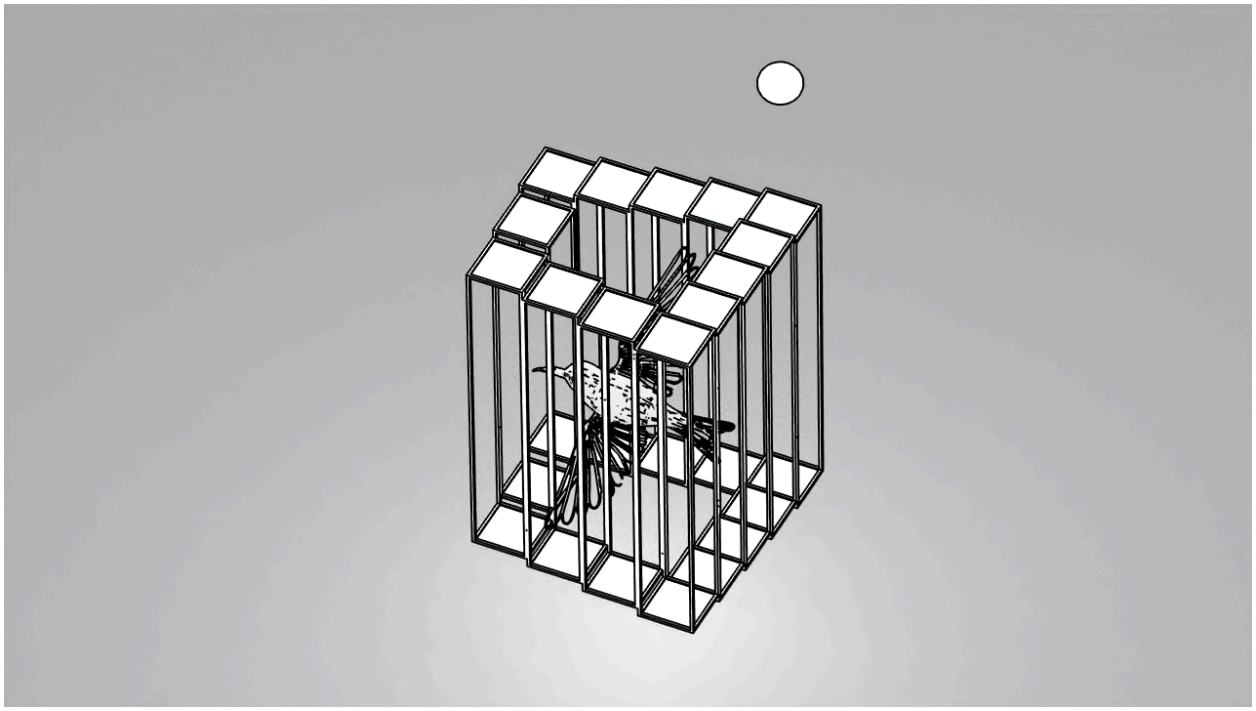
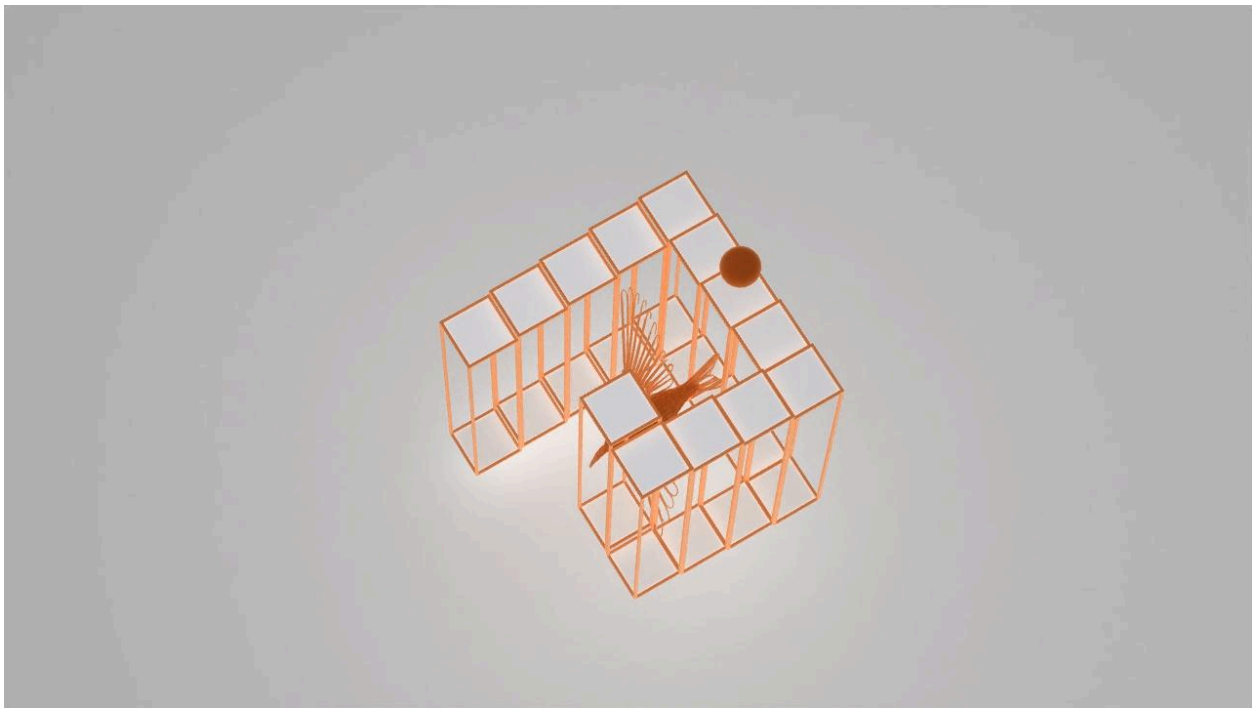
CGI Tools Assignment Documentation

Jessica Dong s5803453@bournemouth.ac.uk





Project Brief

Assignment option: Esher's Impossible Staircase
Project name: *Cage Within*

Cage Within is a 3D rendered illusion animation inspired by Escher's iconic *impossible staircase* structure, combined with recurring motifs from his work such as birds and illustrative graphic elements.

The project aims to express the idea that what confines us is often ourselves — the seemingly inescapable cage and endless loop do not truly exist.

The production pipeline involved **Maya**, **PyMEL**, and **Houdini**, and also explored how **AI-assisted generation tools** can support different stages of a full 3D workflow.

Project Theme

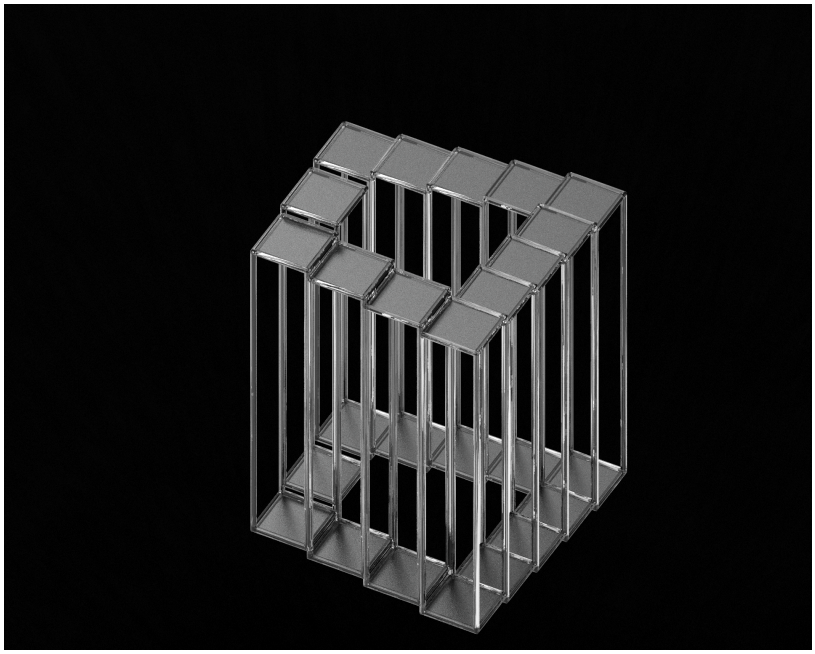
The theme of this project was gradually clarified during the development process. I rendered multiple images with different compositions, and eventually decided that the staircase "cage" combined with birds conveyed the concept most effectively.

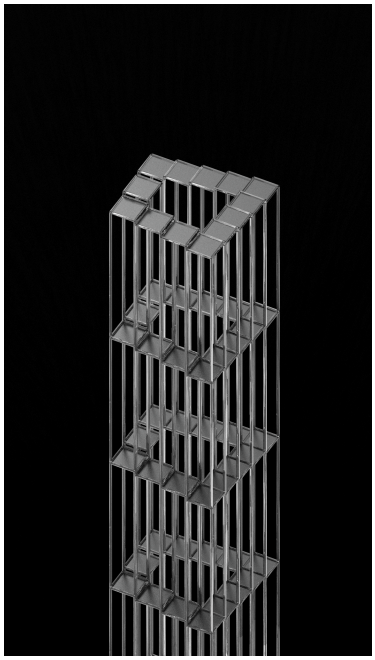
Many of Escher's works emphasize repetition and endless cycles, with birds and fish appearing as recurring motifs. In addition to the required **10-second fixed-camera render**, I originally planned to include a moving camera shot to reveal the illusion mechanism, as well as a bird flight animation. However, due to time constraints, these ideas were not fully realized.

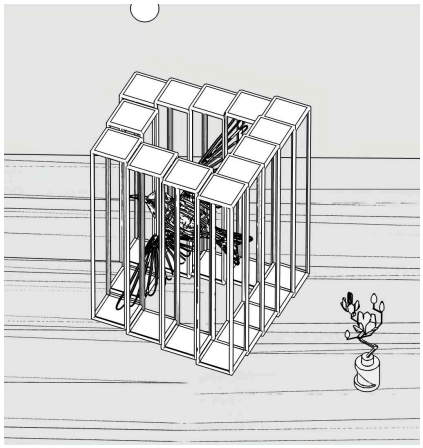
I intentionally avoided creating additional architectural elements. My priority was achieving a final render I was satisfied with. While experimenting with environmental assets, I found that illusion-based structures require extensive iteration on object placement and occlusion to achieve a visually pleasing result. Adding elements purely for complexity conflicted with the conceptual clarity of the piece.

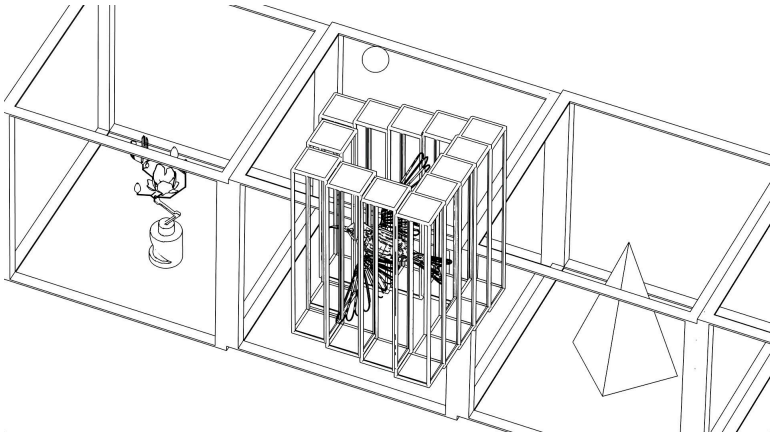
The current outcome represents the best balance I could achieve within the limited timeframe, while also allowing me to experiment with several new techniques I had not previously used.

Below are some of my rendering experiments:

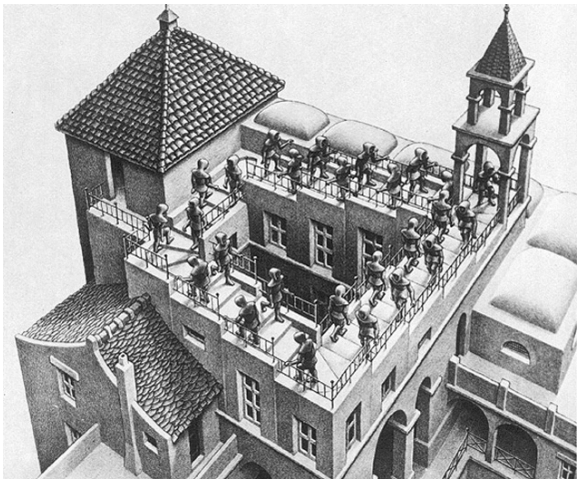








Inspiration and the Use of AI



Inspiration: Jinil Park's Drawing Series

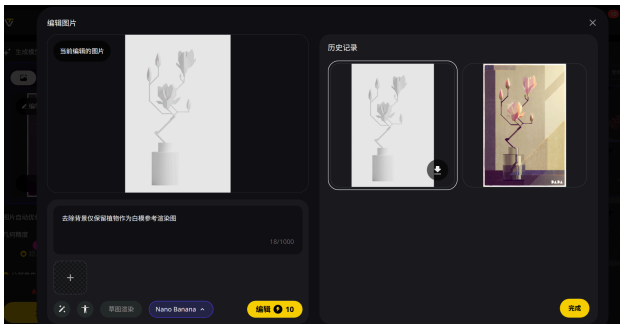
Jinil Park's furniture sculptures translate the visual qualities of two-dimensional sketches into three-dimensional space. I found this minimalist, line-based aesthetic to be highly compatible with Escher's impossible staircase structures, resembling principles explored in spatial composition studies.

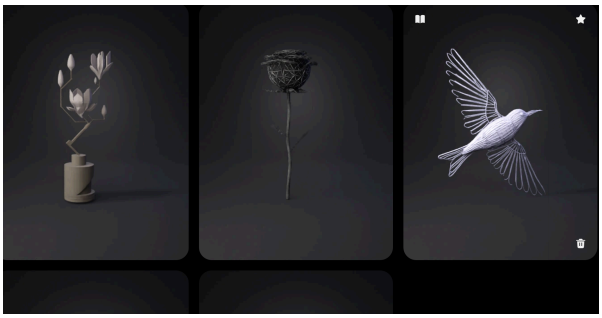
Based on this inspiration, I used **Gemini** to help generate prompt variations, then employed **Hunyuan 2.0 text-to-image generation** to produce visual references in a similar style. These images were subsequently converted into 3D models using **TripoAI**.

Prompt Example:

> *A landing bird with its wings half lifted, one foot raised, viewed from a 45-degree angle showing both wings, made of tangled black steel wires, inspired by Jinil Park's Drawing Series. The lines have varied thicknesses, some as thick as steel wire. The form is airy with cross-hatching textures. Pure white background, studio lighting, hyper-realistic wire texture.*

Result:





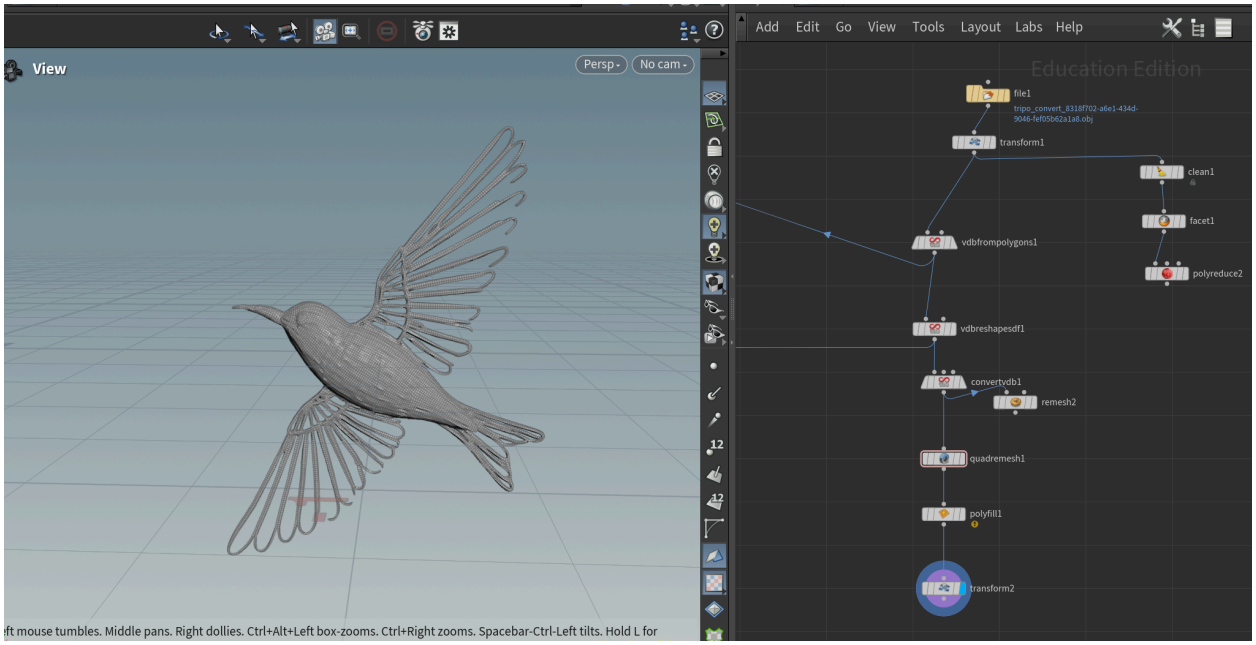
Problems and Solutions

AI-generated 3D models struggled to interpret the hollow, wireframe-like structure correctly, often producing solid geometry instead. To address this, I used **Nano Banana** for image editing to enhance negative space and reduce excessive line density before 3D generation.

The resulting models contained extremely high polygon counts, often reaching several hundred thousand polygons, making them impractical for use. I imported the assets into **Houdini** and applied **VDB remeshing** and **polygon reduction** workflows. I processed two models — a bird and a floral arrangement — although the floral model was not used in the final render.

This process helped me understand the basic pipeline of polygon reduction. However, my technical understanding is still limited, as significant mesh artifacts began to appear once the polygon count was reduced to around 10,000.

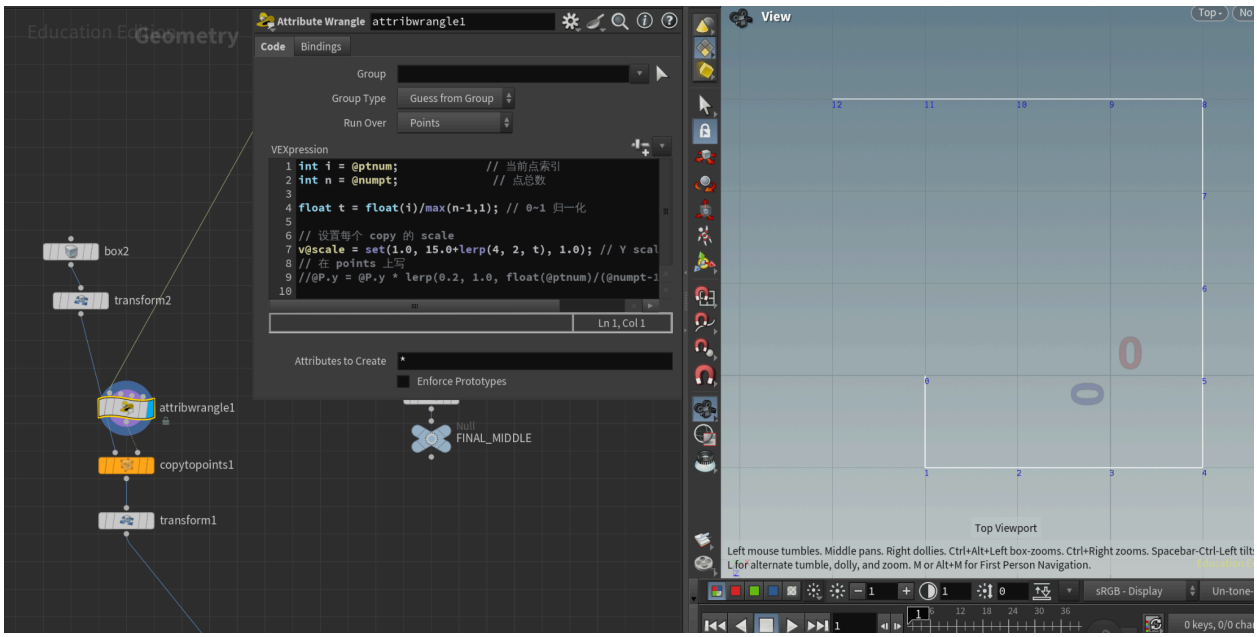
A more effective approach would be to convert the polygon mesh to VDB for reshaping, then perform reduction in stages. The model could also be split into regions with different density requirements, applying reduction selectively based on visual importance.

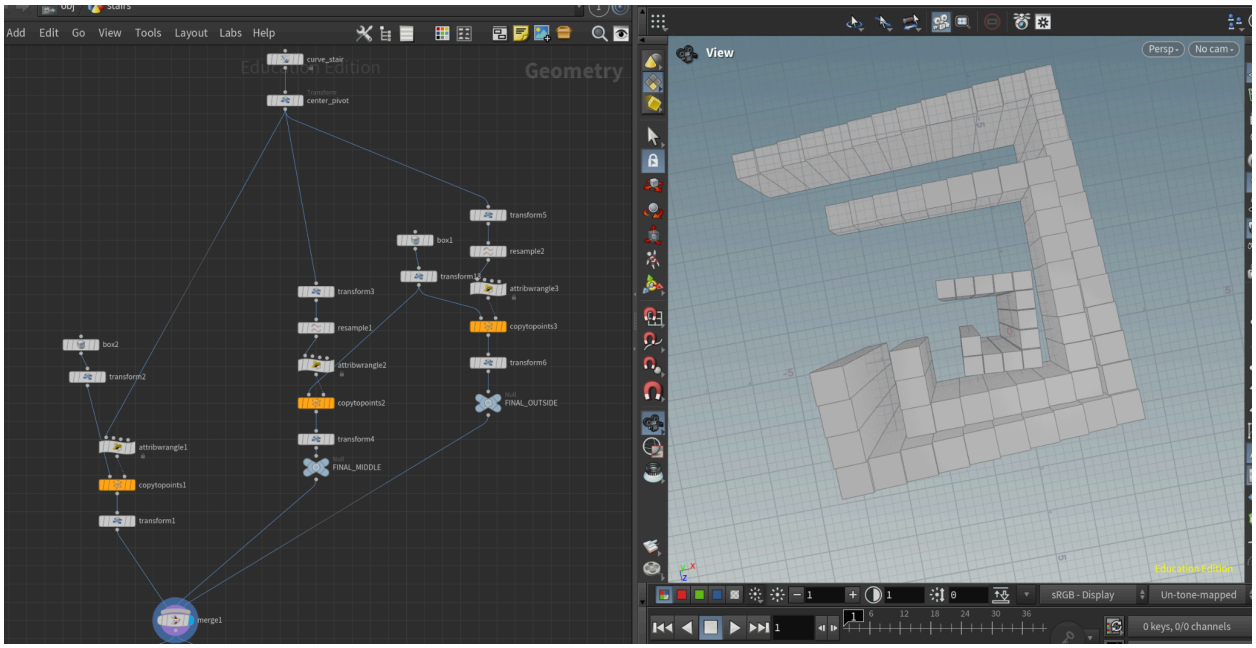


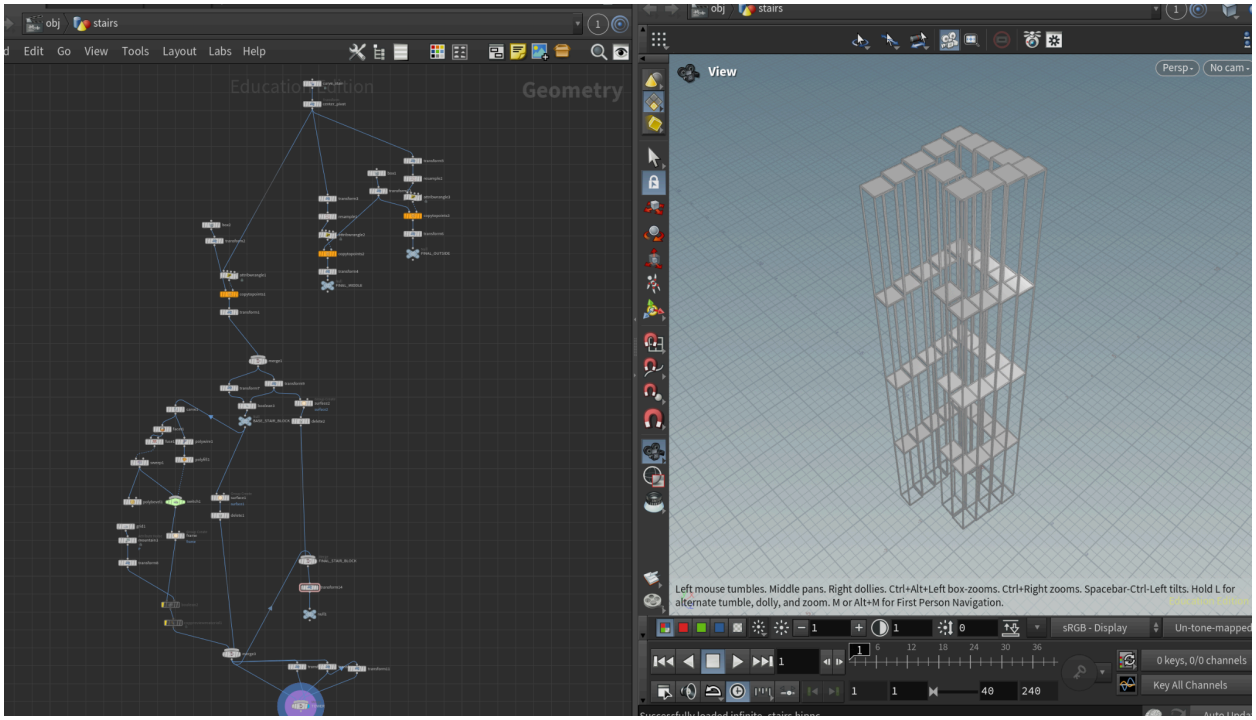
Generation of Stairs in Houdini

The staircase was generated procedurally in Houdini. A curve was first drawn to define the overall shape and proportion of the staircase layout. By resampling points along the curve, the number of steps and their density could be adjusted.

The step elevation and transformation were controlled using **@ptnum**, allowing each step to incrementally scale and translate upward.





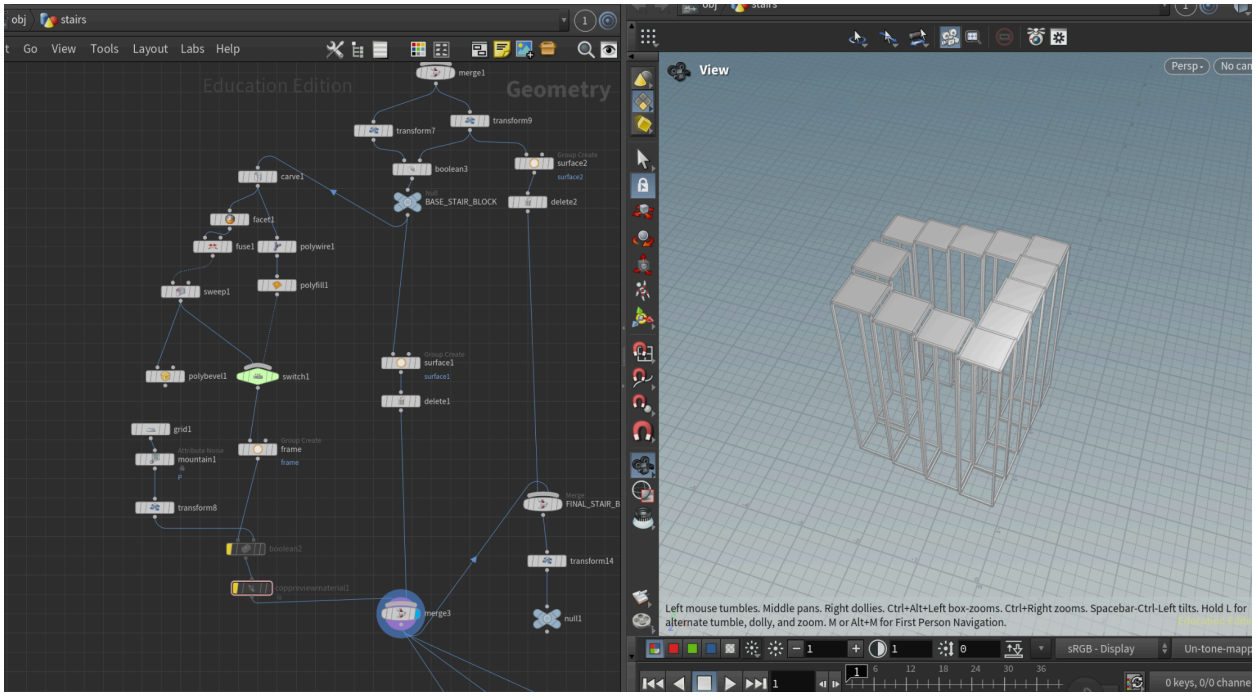


For the hollow structural effect, I experimented with two methods:

1. **PolyWire on a cube**, which generates cylindrical tubes along mesh edges. This method resulted in asymmetrical geometry, and produced gaps at corners when polygon counts were low.

2. **Sweep with a rectangular profile**, which produced cleaner results overall, but introduced overlapping edges at sharp corners.

Since the final render used a **Toon Shader**, each method had different visual strengths and weaknesses, and the final choice required balancing overall rendering quality.



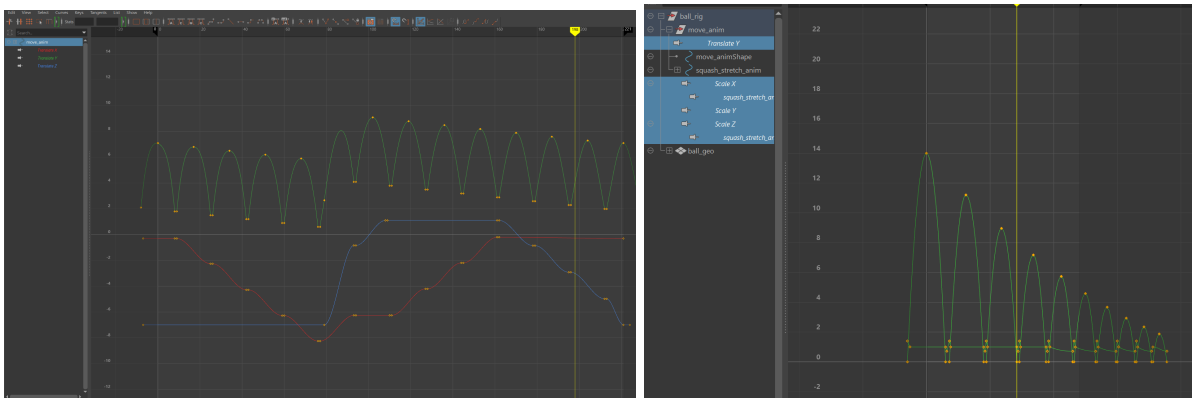
Bouncing Ball Animation

The vertical bouncing animation of the ball was generated using a **PyMEL script**. The script allows control over elastic deformation and ground contact time, enabling simulation of different material properties and energy decay behavior.

Because the ball needed to move along the staircase, a staircase height parameter was incorporated into the animation logic.

The animation principle was based on this reference video:
https://youtu.be/PhqA2m05OkI?si=gQQogK8Ilz1ntPvy

The implementation primarily relies on the relationship between time and the square root of height.



The full script is in the **bouncing_ball_animation.py**

To use the script, copy it to the script editor and run

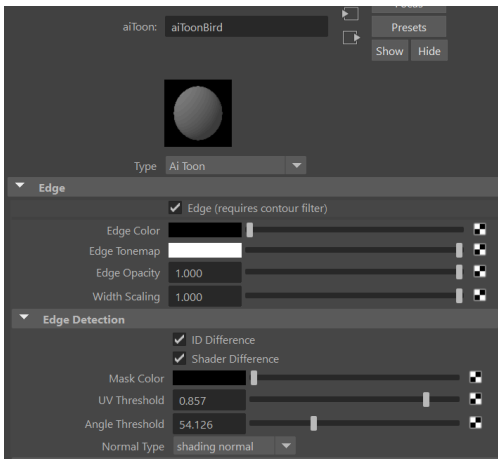
The scene must contain:

A transform node named **`move_anim`** Controls the vertical movement of the ball (`translateY`)
A transform or rig node named **`squash_stretch_anim`** Controls squash & stretch deformation (`scaleY`)

Shader and Rendering

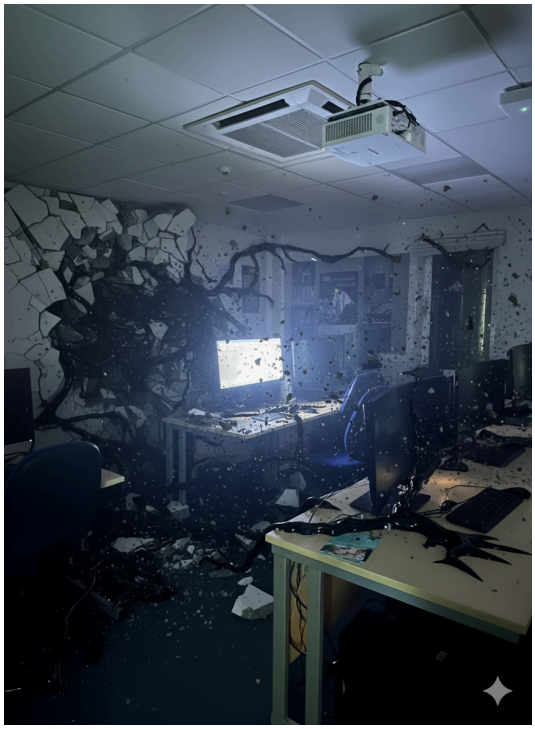
The hand-drawn visual style was achieved using **Arnold's AiToon Shader**. Several key parameters were adjusted to produce the current outline and shading effect.

Tutorial: https://youtu.be/UAoAiY8NZIk?si=Dulp8iTD0cz21sjh



Irrelevant but Fun

I did consider having some AI-generated visual reference of building structure, but they are not interesting. However, I made a picture of our lab being destroyed.




Reference

Furniture that looks like line drawings by Jinil Park
弾むボールのアニメーション Bouncing Ball Animation
How to use the aiToon Shader for Maya
Random Noise in Maya (Tips & Tricks)
ChatGPT for technical problem solving and report translating https://chatgpt.com/
Gemini for image prompt https://gemini.google.com/
HunYuan Image https://hy-image.org/
TripoAi for generating 3d model
https://studio.tripo3d.ai/3d-model/77ef9878-302d-46fa-b1ec-a33b951c0e3b?invite_code=5HS3G5