Final Project - Animation

COSC 3P98 – Computer Graphics

Daniel Conway & Justin Vieira

Overview

We decided to go with the Animation final project over any of the other because as GAME programming students, we feel that it would benefit us the most in our careers. By learning the basics of animation and modeling software, this will not only allow is to better communicate with artists and animators in game development positions, but also allow us to get better at 3d art if we wanted to make games solo.

Theme

As for the theme and feel of our animation, we decided early in the brainstorming process that we wanted to use video game models, and make them act in ways they wouldn't in their games. For instance, in our animation, the core gameplay loop of Super Mario 64 is Mario trying to get stars. In our animation, we kept the same goal, but made him move and use items in ways that he isn't able to in SM64. We feel that by connecting to a game and franchise that is very popular, and breaking expectations of how people expect Mario to move, that this allows us to connect with viewers faster than if we chose models and a setting nobody has previously seen.

Technical Specifications

For this animation, we used Blender 2.92.0 for making the animation, and rendered it using the Cycles renderer.

For video editing, exporting, and sound, we used MAGIX Video Pro X.

Models

In order to spend more time focusing on the animation, and skills we wanted to learn, we decided to use models that are available for free online. One place we looked was Adobe's Mixamo, which has dozens of fully rigged models, but decided to go with models-resource.com because as avid video game players (and GAME programming students), we thought that it would be more fun to use characters and a setting we could connect to. We looked at models across many video games franchises, but decided to go with Super Mario 64, because we felt that it could fit our theme detailed in the overview the best.

All the models on models-resource.com have been ripped directly from popular video games. In our case, most of them were good to use but a few posed issues. The main issue was with our Mario model, which was not rigged, skinned, and the UV's of the face textures were off. In a later section we will cover how we solved those issues. The only other model that was not ready to go was Peach's Castle, which required some moving of objects into their correct positions and resolving some z-clipping conflicts.

Below is a table of the models we used, who submitted them to models-resource.com, the link to the download page, and a preview of the model.

Model	Submitter	Link	Preview
Peach's Castle Exterior	alecpike	https://www.models-resource.com/nintendo_64/supermario64/model/503/	
Mario	Friedslick6	https://www.models-resource.com/nintendo_64/supermario64/model/1637/	
Power Star	alecpike	https://www.models-resource.com/nintendo_64/supermario64/model/1708/	
Trampoline	Kovkov	https://www.models-resource.com/mobile/mytamagotchiforever/model/25376/	
Green Shell	Peardian	https://www.models-resource.com/wii/supersmashbrosbrawl/model/10167/	

Pre-Animation Technical Work

As detailed above, we had to learn a few new skills in order to allow us to use the models we wanted to. The 3 main skills we learned was rigging, skinning, and UV mapping.

For rigging the Mario model, we followed a few different tutorials on Youtube, and modified them to fit our needs. Since Blender is such a popular and accessible program, there are more tutorials online than a paid software such as 3ds Max. The rigging went smoothly, and we learned the fundamentals of how bones work.

For the skinning process, we had to weight paint the model according to each bone in the rig we created. Blender has an auto weight paint tool, but it didn't suit our needs. It gave more influence to vertices farther away from their corresponding bones, and we wanted more control over what bones move certain vertices. We decided to manually weight paint the model, and only give influence to the area of the mesh surrounding the bone.

As for UV mapping, since Mario's eyes, mustache, hair, and a few other features are textures, we had hoped that when exporting as an fbx, they would keep their positional data. This was unfortunately not the case. To fix this, we had to open Blender's UV editor, and reposition the textures onto the mesh to the best of our ability.

Sound Design

For the sound design of this project, we decided that it would be better to limit ourselves to only use sounds that were present in the original Super Mario 64 games. This caused us to sometimes not have the sounds we wish we had, but required us to improvise and use the sounds how they weren't intended in the games. The music we used is also from SM64.

Below is a table outlining the sounds used, and where there are freely available online.

Sound	Uploader	Link
All SFX in SM64	Doorhenge	https://www.sounds-resource.com/nintendo_64/mario64/sound/1441/
Super Mario 64 Soundtrack - Slider	RadiatorRampardos	https://youtu.be/l7I8dYKeke8

Rendering

Initially, we chose Blender's lightweight renderer called Eevee, which took around 2 minutes to render the 66 second animation using a 3060ti GPU. The quality was not too bad, but we wanted to experiment further and see what it would our animation would look like using the heavy renderer called Cycles. This was unfortunate a very timely process, and took over 7 hours to render due to the added motion blur, denoising, and advanced lighting. We believe that this significantly improved our final result, and was a worthwhile choice.

Below are a few screenshot comparisons between the two renderers.

Eevee	Cycles