Louka Fortin-Sirianni
40210881

07/04/2024

COMP 371 – Project Report

## Introduction

The semester-wide OpenGL project has proven to be very stimulating and quite challenging at the same time. Before going into the details, let us reiterate the purpose of this project

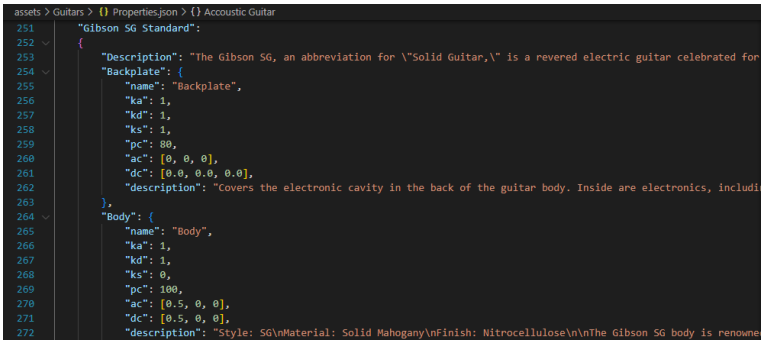
To begin with, the motivation for the project stemmed from my love for rock. I wanted to make something that would showcase the essence of this music genre, and that would give lessons on the specifics of the main instruments. Thus, I went for an educative Guitar Showcase. Guitars come in many different formats with different parts and different materials. I needed to settle on four main objectives.

1) Shading the parts of the instrument differently
    a. Different surfaces, different colors and feel
2) Realistic finishing textures shading
    a. Explore different finishes on the guitars
3) Layering the components of different guitars
    a. Render three replicas
    b. Independently layer the parts
4) UI supports interacting with guitar parts
    a. Allow user to move the parts
    b. Informative GUI
    c. Clickable parts

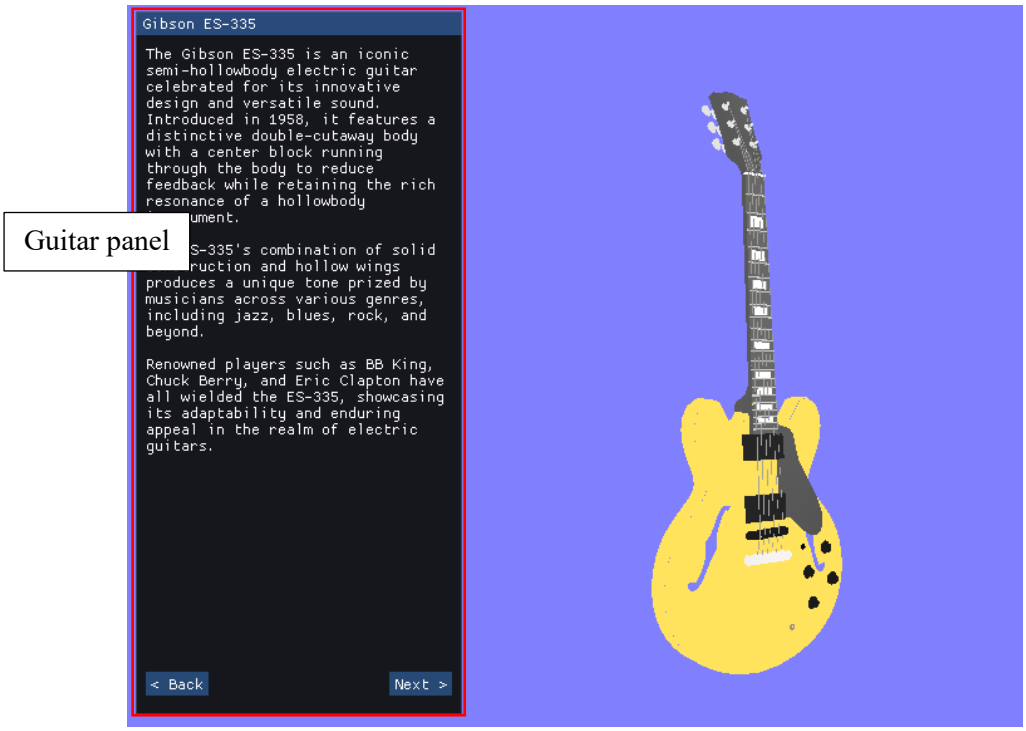
## Achievements

The most important task before starting to program the showcase application was to gather guitar wavefront files and to layer their components. Blender worked wonderfully – though it took me a long time – to help me accomplish this task, allowing me to select entire parts, merge them, and finally rename them. I also found three different replicas. My objects had their parts layered, therefore objective (3) was considered over, as all operations further on depend on the layering.

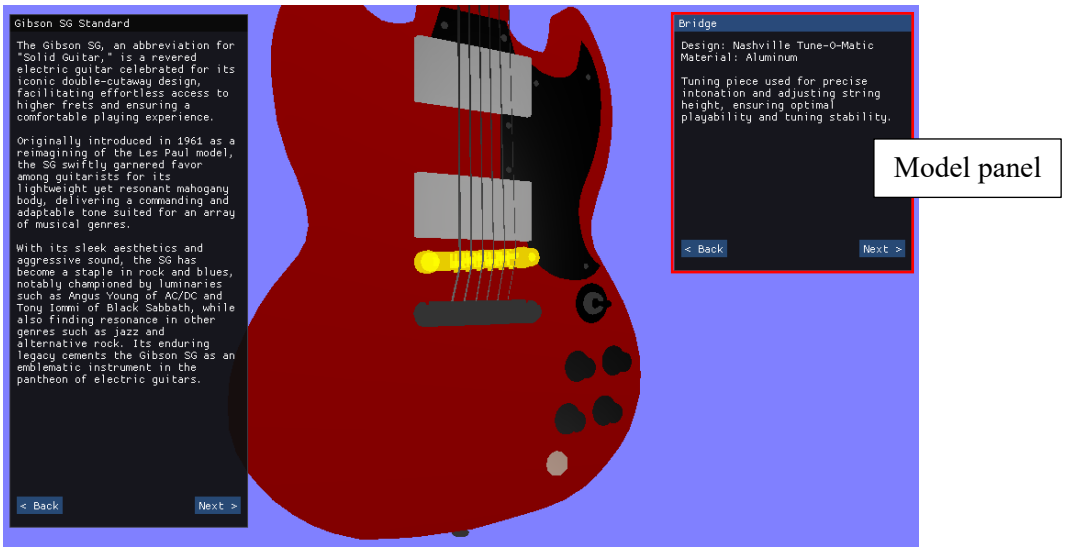
Using a JSON file, mapped the parts to their lighting properties, descriptions, and names, manually. The application reloads properties with debug commands which helped me adjust the colors and properties of each part. Eventually, I would write a shader that is given these uniform colors, which concludes objective (1).



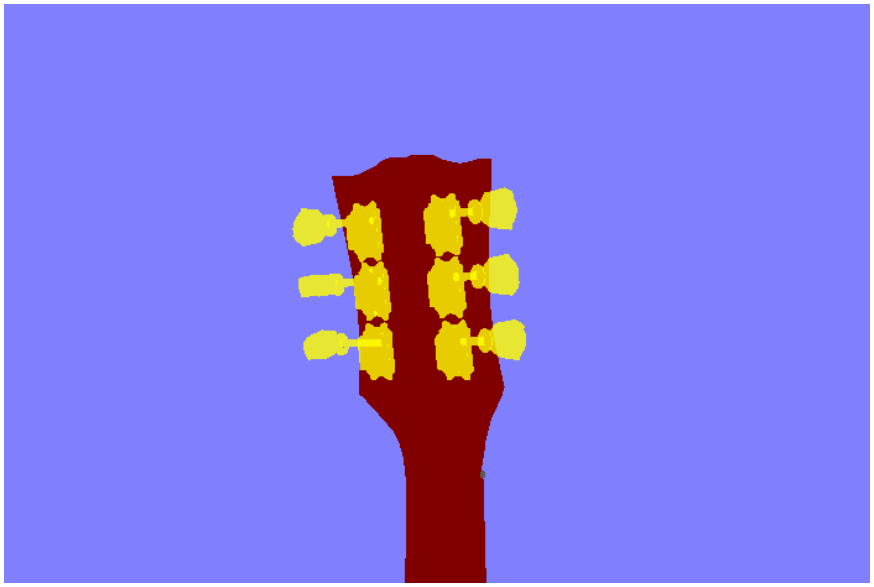
Next on the list was the GUI. At first this seemed to be the most difficult objective to achieve, but the library Dear ImGui made it very simple. Following my information design in the JSON property file, I allocated a first window on the left corner of the application to display the name of the guitar and its description. The “Back” and “Next” buttons allow the user to change the current guitar.




A second part of the GUI consists in selecting parts of the guitar by clicking on them. We will cover the Picking strategy in a moment. Each part has a name and a description, which can be displayed just like in the Guitar panel. I made the choice to add a smaller window on the left to encapsulate part interactions.



As for the picking strategy that was mentioned earlier, each part of the guitars was given a unique code in the range [1, 255]. The reason behind this range is that if the user were to click the image, the application would draw the scene offscreen. Using a specific shading where each model fragment is given the color RGB(model.code, model.code, model.code), we can check which pixel is clicked and automatically get the reference to the model. On click, the model is “picked”, and the original shader changes the model color to yellow and lower its alpha to 80%.



Picked Tuning Machines on Gibson SG Standard

**Limitations**

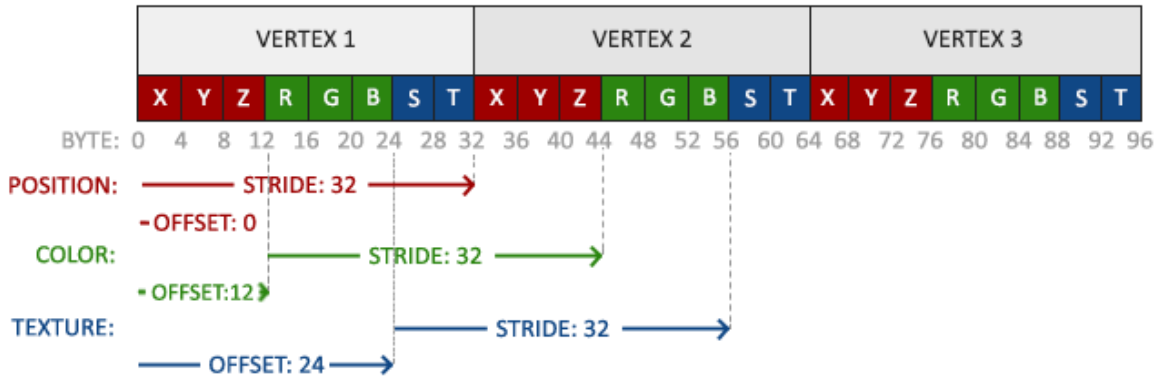
Here are some paths that were not taking in consideration for the final implementation.

First off, my vision of the GUI shifted while I was building this program. I originally aimed to have movable guitar parts, but I ended up choosing not to add this feature. It felt like there was no good way to make this idea appealing. Usually, very few guitar parts are moveable, and it was not exactly the main vision I had. What could have been a good idea is to scrap the informative part of the program and allow to strum the strings.

I had issues with the shading part of the program. The models are very small, and the camera and light are close to the guitars, which may have caused small numerical problems for the GPU. Also, the image textures were discarded as an idea.

**Challenges**

There were many times during the semester where I spent more than a couple of days figuring out how to implement certain parts of the program. First, I had issues with setting the VBO of my objects. The original approach used an EBO to keep track of the vertex information. However, I learned that each EBO element indexes all the position and normal buffer objects the same, and I could not be bothered to refactor the buffers I parsed. The final approach uses a singular interleaved VBO per model.




Interleaved VBO

As a big challenge, we can also count the layering of the guitar parts. This was one of the most time-consuming tasks due to the number of parts per guitar. The Gibson ES-335 and the Gibson SG Standard had around 15 parts each. Writing information and the colors took quite a while. In Blender, however, most of the parts were labeled intelligibly, which helped me merge some meshes.

Finally, the picking. Model selection was the scariest part of all but felt underwhelming in the end. I first investigated framebuffers and offscreen rendering, which led me to initialize a picking framebuffer. This led me to debugging something that virtually had no solutions. For some reason, the custom framebuffer was never bounded. I must have been doing it the wrong way. Lack of documentation and resources around framebuffers pushed me to consider another solution: render with the picking program in the default buffer. This simplified the procedure, and I was able to pick individual models.

**Conclusion**

The motivation for this project stemmed from not knowing how to end my studies in software engineering. In the past few months, I took a big interest in computer graphics while working on this OpenGL project. I learned tremendously about the graphics pipeline by implementing parts of it with the help of graphics APIs like GLM.

The most challenging part was to find suitable guitar models and to layer their parts of unique shading and display of information. The mathematical part of the project was almost trivial because I had the chance learn about 3D transformations and light equations during class time. One thing that made me reflect the most was the picking of the models and shading these models accordingly. I am quite proud of my accomplishments, and next semester, I will take on the following class in the series: Introduction to Game Development. I believe that following this path of studies is promising and I will be sure to learn tons of new fun concepts as I step forward in that field.

**References**

Useful links and libraries that were consulted in the making of the project.

[1] Dear ImGui. https://github.com/ocornut/imgui

[2] Json parsing. https://github.com/nlohmann/json

[3] Shading demo. https://www.cs.toronto.edu/~jacobson/phong-demo/

[4] OpenGL tutorials. https://learnopengl.com/

[5] Picking. https://www.lighthouse3d.com/tutorials/opengl-selection-tutorial/