**Computer Graphics 2023 project**

**DVD-Logo**

author: Louis Rives-Lehtinen, xrives00

# Project Goal

* Objective: Replicate the iconic never ending bouncing DVD logo animation but in a 3D environment.
* Purpose: None specific, I just thought that it would be an entertaining idea especially that I grew up watching DVDs.

# Used Technologies

1. OpenGL and GLFW: Utilized for creating the windowed OpenGL context and managing graphics rendering.
2. GLEW (OpenGL Extension Wrangler Library): Employed for managing OpenGL extensions and ensuring compatibility.
3. C++ Programming Language: The primary language for implementing the application, handling logic, and integrating various components.
4. GLFW: GLFW (Graphics Library Framework) is an open-source, multi-platform library for creating windows with OpenGL contexts and managing input, providing a simple and consistent interface for handling various tasks in graphics programming.
5. GLAD: GLAD (GL Extension Loader) is a C/C++ library used for loading OpenGL extensions easily and efficiently.
6. Blender: Utilized for 3D modeling and exporting the DVD logo object in the OBJ file format.
7. Shader Programming (GLSL): Custom shaders were created for handling vertex and fragment shading in the rendering pipeline.
8. Visual Studio: The integrated development environment used for coding, testing, and debugging the application.

# Used Resources

1. The DVD logo image was directly taken from its website: <https://bouncingdvdlogo.com/>
2. The CD image was taken from this page: <https://en.wikipedia.org/wiki/File:CD_icon_test.svg>; then it was converted to an svg file using this website:

<https://www.adobe.com/express/feature/image/convert/png-to-svg>

1. Blender was used to convert the svg images into meshes; Thus I was able to modelize the 3d shape of my objects and then convert them into an obj file.
2. <https://www.youtube.com/watch?v=FFxoQlUPca8>

# Most Important Achieved Results

Describe three things that are best about your project. It is best to show and comment images, in the worst case use text to describe.

1. My object: I think that my object is an accurate representation of what the 3D DVD logo would be.



1. The effective bounces of logo on the walls

# Controlling of Created Program

**Window Controls:**

* *Close Window:* The user move the camera using the arrows and move the objective of the camera using the mouse.

**Animation Control:**

* Automated Animation: The 3D DVD logo animation is automated and continuously runs without direct user intervention. The rotation of the logo is controlled within the code to mimic the perpetual bouncing motion.

**Shader Interaction:**

* Shader Parameters: While not directly controlled by the user, the shaders dynamically impact the visual output. Shader parameters, such as lighting, colors, and transformations, are set within the C++ code to achieve the desired effects.

**Configuration:**

* Shader Files: The program relies on external shader files (e.g., fragment.glsl, geometry.glsl, vertex.glsl) for defining the rendering pipeline. Any modifications to shaders should be done in these files.

# Experience with the Selected Platform and Tools

In this project, I primarily utilized Visual Studio for development, and the overall experience was relatively straightforward and easy to understand.

As for OpenGL, the complexities of the graphics library became apparent. OpenGL's intricate nature can be quite challenging for beginners, as it requires a deep understanding of various concepts, including shaders, vertices, and object files. While this complexity provides immense flexibility for developers, it also adds to the learning curve.

And my prior knowledge of C++ made it easier for me to navigate the complexities of OpenGL and understand the concepts and functionalities associated with this graphics library.

# What Was the Biggest Challenge

The most significant challenge during the project was managing and debugging the shaders. Since shaders are integral to the rendering process in OpenGL, any issues within them directly impacted the visual output. The intricacies of GLSL (OpenGL Shading Language) required meticulous attention to syntax, variable types, and overall logic. Also since I could not change my learning agreement on time, the computer graphics courses/labs were overlapping with one of my other course, thus I was not able to come to all lectures, and I had to learn most of the technical details during the conception of the project.

# Experience Gained From the Project

1. OpenGL and Graphics Programming:

Deepened Understanding: The project enhanced my knowledge of OpenGL, Blender and graphics programming, especially in the context of 3D animations. Beforehand I have any knowledge about vertex manipulation, shaders, and rendering techniques.

1. Integration of External Assets:

Working with 3D Models: Incorporating a 3D model of the DVD logo into the project involved understanding and integrating external assets. This experience broadened my skills in handling diverse file formats and external resources in a graphics programming context.

1. Shader Development :

Understanding how to manipulate lighting, colors, and transformations in shaders.

# Autoevaluation

Evaluate your solution in individual categories (0 – nothing done, despair, 100% - perfection). A project might easily get full score even when some components of this autoevaluation are very low. Declaring values close to 100% in all or many categories might be a sign of misunderstanding of the problematic. An important skill (seen from this exercise) is to evaluate the strengths and weaknesses of one’s own solution.

**Technical Design: 75%** (analysis, problem decomposition, selection of suitable tools, …)

I achived what I was aiming for although the logo is not as complex I’d like to (I tried adding the cd underneath but it would not work)

**Programming: 75%** (quality and readability of code, stability in runtime, generality of solution, reusability, …)

Could definitely be more optimized but I think it is decent.

**Usability of the Created Solution: 30%** (practical usability, aesthetic quality, user satisfaction, …)

Most likely not usable in external projects since the aesthetic is terrible and its goal would not really make sense. Altough the camera would be definitely reusable.

**Use of Resources: 70%** (use of existing code and data, use of literature, avoiding re-inventing the wheel, …)

I checked kind of everything, forums, youtube tutorials, AI… I even made different versions of the code using different libraries. (assimp, geGL, Glew, SDL, glm, glfw, glew).

**Time Management: 30%** (balanced maturity of project parts, level of hurry, missing parts of the solution, …)

The project could definitely be cleaner. The submit is in a hurry.

**Overall Impression: 70%** (amount of work, gained skills+experience, usefulness, selected goal, anything, …)

# The project, overall, was a fun experience despite not yielding any significant results after approximately 20 hours. Despite the lack of concrete results, I gained valuable knowledge about OpenGL, including shaders, vertices, and object files. While I may have fallen of my goal, I can reflect on the effort I put into the project even if it clearly was not enough within the given time constraint.

# Recommendation for Assigning Future Projects

Be able to chose his subject is a double edged-sword; It definitely stressful to start the project from scratch, although being able to chose the subject makes the project more entertaining.

# Recommendation for Future Students

This documentation will be made available to PGRe students in the next years. Help them select a good assignment/goal of the project and the platform. Warn them from problems that you faced, share with them your experience.

Watch out for the time constraint, do not chose something too ambitious if you do not have any experience on openGL.

# Miscellaneous

Anything else should be in the documentation? Write it here! If necessary, create a new chapter.