

Artifact Description

The artifact is an enhanced version of a project from the CS330 Computer Visualizations course. Originally created as part of my coursework, it's a graphics programming project using OpenGL and GLFW to render dynamic graphical elements such as circles (balls), and bricks. My enhancement includes a user-controlled paddle, adding interactivity as well as improving the code base OOP design, significantly expanding on the original coursework.

Justification for Inclusion

I selected this enhanced artifact for my ePortfolio because it showcases my passion for graphics programming and demonstrates my ability to refactor and extend an existing codebase.

Specific components that highlight my skills include:

- **Graphics Programming:** Skills using OpenGL and GLFW to create and render dynamic graphical elements.
- **Code Refactoring:** Analyzing and restructuring the existing codebase, improving encapsulation, and enhancing code readability. Also, I added new features like a user-controlled paddle and realistic ball physics, which required extensive refactoring.
- **New Features addition:** Adding new features like a user-controlled paddle and realistic ball physics.

I improved the artifact in several ways:

- Added a user-controlled paddle, making the game interactive, which it wasn't before.
- Redesigned collision detection for more realistic ball bounces.
- Implemented game-ending conditions.
- Refactored code for better encapsulation and readability.
- Improved separation of concerns by reorganizing functions and classes.

Course Outcomes

- Designing and evaluating computing solutions: The project showcases problem-solving skills in graphics rendering and collision detection using algorithmic principles.
- Using innovative techniques and tools: Utilizing OpenGL and GLFW for rendering, along with modern programming practices like encapsulation and modular design.

Reflection on the Enhancement Process

I learned some valuable lessons during this enhancement:

- The importance of writing maintainable code, especially when working with legacy systems.
- The challenge of redesigning core systems, such as the collision detection mechanism and adding interactive elements.
- Something that I found to be very valuable, is iterative development in small chunks. I ran the game every time I made a change to the code, which is actually how game development tends to be.

I also encountered several challenges:

- Implementing text rendering, which required more complex libraries than initially anticipated as GLFW does not render text in a reasonable manner. It required that I create text out of geometrical elements or create a texture in an image editing software, which I have no expertise in using. I ultimately opted for a color system to indicate win-and-lose conditions.
- Balancing the desire to add new features with the need to maintain code quality and stability. At a certain point, I ran too wild with my scope of enhancements to the point where the program was unrecognizable as well as buggy.
- A glimpse into the reality of software development, where initial plans need to be adjusted based on emerging requirements and obstacles. I believe this was both a challenge and a lesson.

This project has been a great learning experience, giving a glimpse into real-life software development challenges, and how obstacles come up due to changes in requirements that can't be necessarily thought of before development begins. I believe this enhancement shows my ability to extensively modify a legacy codebase in a meaningful way, make use of advanced graphical libraries, and improve code quality through implementing software engineering best practices, as well as my ability to adapt when obstacles come up.

Screenshots:





