**CS 499 Milestone Three: Narrative**

**Category Two: Algorithms and Data Structures**

**Artifact Description**

The artifact I selected is a 3D scene application that I originally built in CS-330: Computational Graphics and Visualization. It’s an interactive project that renders a virtual environment with objects like a laptop, phone, and desk, and it uses textures, lighting, and shading to make the scene more realistic. The project was developed with OpenGL, GLFW, and C++.

**Justification for Inclusion**

I chose this artifact for my ePortfolio because it gave me a meaningful opportunity to apply algorithms and data structures to solve a practical problem. While the original version of the project worked well visually, it lacked realism since the camera could freely pass through objects. To fix this, I added an Axis-Aligned Bounding Box (AABB) collision detection system.

This enhancement highlights two key skills:

* **Data structures**: I created a bounding box structure to capture the minimum and maximum 3D coordinates of needed objects, which defined their space in the scene.
* **Algorithms**: I wrote a collision detection routine that checks whether the camera’s new position would intersect any bounding box. If a collision is detected, the movement is blocked, otherwise the camera moves as expected.

**Course Outcomes Achieved**

This work met the outcomes I set in Module One:

* **Outcome 3**: I designed and evaluated a solution using algorithmic principles. The AABB method is a trade-off. It’s less precise than more advanced methods, but it’s much faster and easier to implement. For this project, that balance of simplicity and performance was exactly what I needed.
* **Outcome 4**: I applied sound techniques to extend the project by adding collision detection into the graphics pipeline. This showed I could build on an existing codebase in a way that added new, useful functionality.

Since the enhancement aligned well with my goals, I didn’t need to adjust my original outcome-coverage plan.

**Reflection on the Process**

Working on this enhancement taught me a lot about choosing the right solution for the problem. At first, I experimented with more complex methods, like bounding spheres, but they added unnecessary complexity and didn’t really match my mostly rectangular objects. Switching to AABBs turned out to be the right call, it was simple, efficient, and effective.

The main challenge was making sure the collision logic worked smoothly with the camera’s movement without breaking anything else. I tackled this by developing incrementally: I added bounding boxes for one object at a time, visualized them to make sure they were correct, and then tied them into the collision system. This step-by-step approach helped me catch issues early and kept the code stable.

In the end, the process reinforced the value of iterative testing and maintainable design. The project feels much more polished now, and it gave me a concrete way to apply data structures and algorithms in a graphics context.