# 3D Physics – Modern C++ 3D Physics Simulator

Sponsor: Student self-direct, no sponsor.

Target: Windows, Linux, MacOS

## Background

Many indie and research projects need a fast, modular 3D physics core with a clean C++ API. This project delivers a physics-first simulator that emphasizes correctness, determinism, and extensibility, while staying lightweight and dependency-minimal.

## Project Needs

1. Core simulation
   * Rigid and soft bodies, sleeping, continuous and discrete collision detection.
   * Stable time-stepping with fixed-tick scheduler, additional optional precise integrator for more realistic simulations.
2. Collision detection
   * Broad and narrow phase with advanced mesh colliders for convex and concave shapes.
   * Ray casting and ray tracing for queries, picking, and debug visualization.
3. Constraints
   * Distance, hinge, slider, ball-socket, fixed, cone-twist.
   * Rope and cloth via Position-Based Dynamics or sequential impules.
4. Shapes
   * Box, sphere, cylinder, convex hull.
   * Custom polygons/polyhedra and triangle meshes.
5. Units & safety
   * Strongly typed physical quantities with a units library to prevent unit errors.
6. OpenGL testbed (up for change)
   * GLFW windowing, GLEW loading, GLM math.
   * Custom shading, lighting.
7. Modern C++ design
   * C++20 modules, efficient and scalable API.

## Project Scope

The scope of this project is a physics engine that can be used for game development, scientific visualization, demonstrations, or simulations.

## Deliverables

1. Functional prototype: Playable OpenGL sandbox with sample scenes, buildable environments using stacks, ropes, ragdolls, vehicles, etc.
2. Engine library: Public C++ API with documented headers and examples.
3. Test suite: Unit and regression tests, scenario scripts, determinism checks.
4. Benchmarks: Collision phase stress tests, stability, and speed.
5. Demo media: Short videos and screenshot from the OpenGL harness.

## Assumptions and Dependencies:

* C++23 or newer, CMake >= 3.28.
* OpenGL 4.1+, GLFW, GLEW, GLM, units library.
* No external game-editor dependency; focused on physics core and testbed.