**Custom Physics Simulation**

My physics system attempts to simulate the interaction of rigidbodies in a 2d physics system.

**Improvements To Be Made**

There are many possible improvements that can be implemented to improve my physics library.

* The library uses a rudimentary form of broadphase, AABB comparisons. Replacing that with quadtrees or other more efficient broad phase algorithms would be much faster.
* Physics bodies and shapes are not stored sequentially in memory. Organising these in memory could significantly improve performance.
* A constraint solver would fix some instability problems
* Collision iterations do not currently do much in terms of increasing the stability of the physics system. Using a more advanced collision iteration system would improve stability significantly.
* Massive objects are unstable, especially when used in compound colliders.
* The library is currently dependant on glm. It wouldn’t be too hard to implement a basic math library to remove this dependency.
* The library does not use const correctness. It should really use const correctness.

**Third Party Libraries**

glm is the only library used for the physics library itself. The testbed, however, uses glm, freetype, glfw, imgui, and enet

**References/Influences:**

* Box2D
* My old physics implementation from last year
* AIE powerpoints
* Finn
* <https://en.wikipedia.org/wiki/Collision_response>
* <https://www.youtube.com/watch?v=ajv46BSqcK4>
* <https://www.youtube.com/watch?v=MDusDn8oTSE>
* <https://www.youtube.com/watch?v=0XQ2FSz3EK8>
* <https://stackoverflow.com/questions/31106438/calculate-moment-of-inertia-given-an-arbitrary-convex-2d-polygon>
* <https://dyn4j.org/2010/05/epa-expanding-polytope-algorithm/>
* <https://dyn4j.org/2011/11/contact-points-using-clipping/>
* <https://en.wikipedia.org/wiki/List_of_moments_of_inertia>
* <http://www.jeffreythompson.org/collision-detection/>