

RIGIDBODY PHYSICS & INTERACTIVE FORCES

Proposal by George Mavroeidis for COMP 477

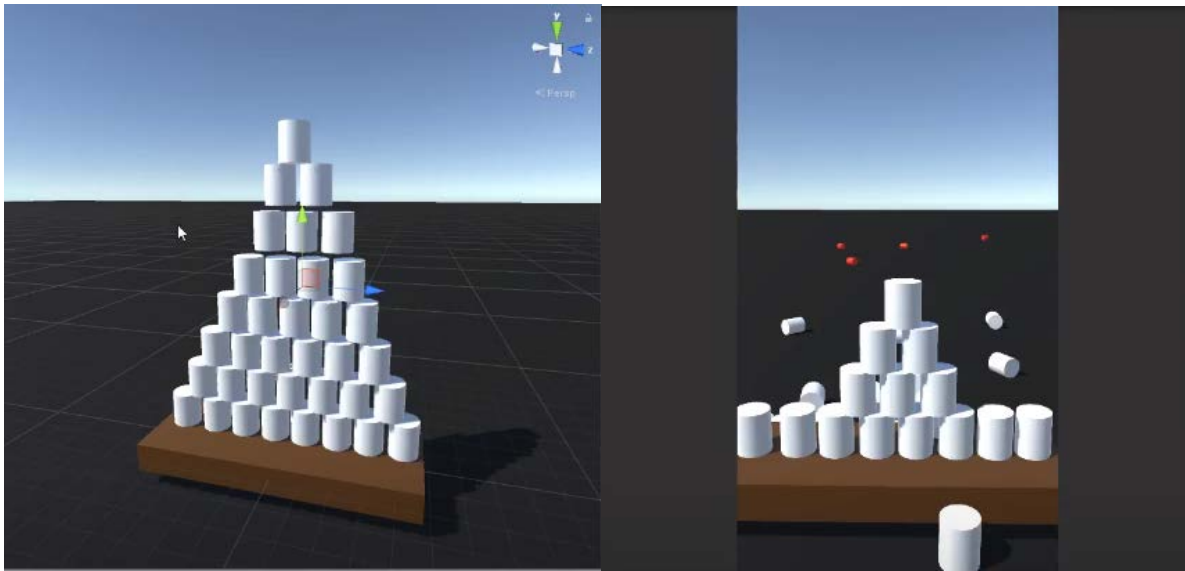
Prof. Tiberiu Popa

Motivation:

This project will be based on a fundamental physics engine feature that I have been working with for years. In 3D animation, I've created multiple projects that utilize rigid body physics that get affected by collisions. In video games, most of my projects involve rigid body components that involve some sort of reaction from collisions, like a ball bouncing off a wall.

Although it is a subject that has been discussed extensively, I am interested in testing additional features, such as constraints and soft body physics, which simulate different types of environments. In addition to that, specific mesh colliders will also be simulated, where a collider is formed at any imported object provided.

An example that inspired me for this project, was a Unity tutorial, where plastic cups are stacked up like a pyramid, and the user pushes them, resulting to a crumbling set of cups. I was inspired to replicate this in C++ from scratch, but with the elasticity of the cup as well.



Made in Unity by DitzelGames on Youtube

Objectives:

1. Create rigid body component that can be applied to the provided mesh
 - Implement the rigid body simulation update cycle
2. Apply basic Newtonian physics to the rigid body component: forces, gravity
3. Apply principles of mechanics: Linear & angular velocity, center of mass, linear & angular momentum, Inertia, net force, net torque
4. Collision detection and resolution with external geometry
 - Hard collision, impulses between collided objects using separating planes
5. Click-feature where the user adds a specific force (or push) to the stack of cups
6. Window that allows to enter parameters before demonstration
 - Gravity scale, mass of meshes, clicking force
7. Rendering: will add light source, textures on cubes and flat ground to facilitate visual representation
 - Will add coordinate system axis (X,Y,Z with RGB colors) for facilitating world perspective
8. Examples of highlighted classes to be made: Box, Rigidbody, Collider, Ground, Camera
9. Libraries and tools to be used: C++, OpenGL(glm library), Visual Studio
10. Objects on scene: ground, light source, camera, coordinate system, stand stool (optional, where cups stack up), ~six boxes, UI input board (OR file script that uses file I/O to acquire inputs)