Assignment #3 - Final Project - Rube Goldberg Contraption

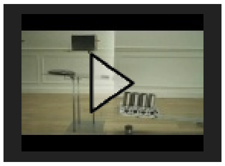
Due Apr 28 by 5:30pm **Points** 180 **Submitting** a file upload

In this assignment you will implement a simple game/tech demo that highlights a complete rigid body physics engine.

This will be the third, and final graded assignment for the course. This assignment will count for 20% of the course grade.

Overview

Rube Goldberg machines are playful demonstrations of carefully setup environments: A single force on an object in the environment creates a chain reaction that is entertaining, and perhaps useful. See:
https://www.youtube.com/watch?v=_ve4M4UsJQo (https://www.youtube.com/watch?v=_ve4M4UsJQo)



(https://www.youtube.com/watch?v=_ve4M4UsJQo)

For this assignment you will demonstrate the capabilities of your rigid body physics engine by simulating a Rube Goldberg contraption.

You have complete artistic freedom in the setup of the simulation. The only requirements are that it demonstrates all of the features that we have studied during the course:

- Rigid body equations of motion:
  - Linear & angular velocity
  - Forces and torques
- Collision Detection for rigid bodies
  - Boxes & Spheres
  - Broad-phase detection
  - Narrow-phase detection & contact generation
- Collision Resolution for rigid bodies
  - Linear & angular velocity resolution
  - Penetration resolution
  - Resting contact & friction

Requirements/Grading (180 points / 60 bonus)

- Implementation of mathematics library [20 points]
  - Vector [5 points]
  - Matrix [5 points]
  - Quaternion [10 points]
- Implementation of Euler integrator [20 points]
  - Force accumulation [5 points]
  - Torque accumulation [5 points]
  - Inertia tensor application [10 points]
- Force Generators for Rigid Bodies [10 points]
  - Gravity [5 points]
  - Spring/Bungees [5 points]
  - Other force generator [Bonus: 5 points]
- Collision Detection [40 points]
  - Implementation of collision detection acceleration structure [Bonus: 20 points]
    - If you do implement this option, make sure to adequately demonstrate it! Hint: Demonstrate many simultaneous bodies!
  - Narrow-phase collision detection: [40 points]
    - Sphere-sphere contact generator [5 points]
    - Sphere-plane contact generator [5 points]
    - Sphere-box contact generator [5 points]
    - Box-plane contact generator [5 points]
    - Box-Box contact generator [20 points]
  - Graphical display of collision locations [Bonus: 5 points]
- Contact resolution [30 points]
  - Velocity Resolution
    - Linear velocity resolution [10 points]
    - Angular velocity resolution [10 points]
  - Interpenetration Resolution

    - Linear projection interpenetration resolution [10 points]
    - Nonlinear projection interpenetration resolution [Bonus: 20 points]
- Resting Contact Support [10 points]
- Friction Support [Bonus: 10 points]
- Good Tech-Demo Features [10 points]
  - Reset Button, Game State Display, Debug Info
- Professionalism of submission [10 points]
- Creativity of demonstration [10 points]
- Blog Post
  - Gameplay Video [5 points]
  - Write Up [5 points]
- Presentation [10 points]

Submission Process:

We will use GitHub as the primary submission means for this assignment. Your work will be submitted via canvas.

- A zipped exe for your work that will run on any Windows 7/8 machine. (Make sure to test on 'naked' machines so that you do not have any surprise dependencies)
- A link to your blog post where you've described your work - learnings, challenges, relevant implementation details.
- A link to a commit on GitHub where we will code review your work.

Notes:

- There is intentionally some ambiguity in the assignment description. Use your best judgement in designing your demo.
- Please feel free to reach out to me for help, but also use your peers. You should help each other, but always make sure that the work that you submit is your own.
- Be mindful of the grading breakdown. Focus on aspects of your simulation that will get you the best return on grades, and then expand into the more challenging implementations.
- Keep things simple, and focus on the physics simulation first!