Session project

Building a video game physics engine

Goals

Through implementation, understand the main concepts of a physics engine. These goals will be achieved gradually through the design of a game engine. The project will be evaluated through four iterative phases.

Specifications

The physics engine must be coded in C++ and run on the Windows operating system. Although it is strongly recommended to incorporate a GUI and 3D rendering into your project, you should, at a minimum, clearly display the results of your simulation.

The project will be done**in teams of 2 or 3**. Any plagiarism will automatically result in a grade of zero.

Weighting and evaluation

Each phase will be evaluated independently during four meetings with the teacher. These meetings will take place in the weeks following the delivery of the code, either via an online meeting or during class periods.

Your source code will also be evaluated and must meet the criteria of good programming practices: readable, structured and documented code.

The weighting of each phase is divided as follows:

• Phase 1 – Particle simulation – Part 1:	15%
• Phase 2 – Particle simulation – Part 2:	25%
• Phase 3 – Rigid body simulation:	30%
• Phase 4 – Collision management:	30%

Each assessment is based on:

- (80%) The functionalities requested in each phase;
- (20%) Additional features (Game elements, 3D engine, etc.);
- Your grade will be weighted according to the effort individually put into the project.

Deliverables and timelines

For each phase, you must submit:

- Your source files, without the compiled files.
- Any other files necessary for compiling and running your program;
- A "ReadMe" file explaining how to compile and run your program;

- A development journal, where we find the difficulties encountered and the justification for the different choices made during development and your contribution to the project. You must also include the references that you used (book chapters, internet pages visited, etc.).

The documents must be**deposited in a Git repository (GitHub)**accessible to the teacher. The commit of the submitted version must be clearly identified and dated before the phase submission deadline.

Description of the different phases of the project

Phase 1: Particle simulation – Part 1:

The goal of this phase is to create a simple physics engine for particle management based on the concepts seen in classes. In particular, you must:

- Implement a Vector3D class with all the necessary methods.
- Implement a Particle class including the appropriate attributes.
- Implement an integrator to update the position and velocity of each particle every frame.
- Make a short game to demonstrate the functionalities of your Physics engine.

Phase 2: Particle simulation – Part 2:

The goal of this phase is to continue the development of the physics engine created during phase 1 by adding the notions of forces and constraints seen in class. Your engine will thus allow the management of clusters of particles. In particular, you must:

- Implement generators for the following forces: gravity, friction and springs**. Several types of springs must be implemented.**
- Implement a detection system. This system must detect the following types of collisions: interpenetrations, rods and cables, contacts at rest.
- Implement a pulse-based collision resolution system.
- Play a little game where you move a cluster of particles to demonstrate all the functionalities of your Physics engine.

Phase 3: Management of rigid bodies by adding rotational physics

The goal of this phase is to specialize the physics engine of phases 1 and 2 by adding rotational physics, thus implementing the concepts seen in class. In doing so, the engine will allow the management of rigid bodies. In particular, you must:

- Implement the Matrix3, Matrix4 and Quaternion classes with the relevant attributes and methods.
- Implement a RigidBody class with the relevant attributes and methods.
- Adapt your physical integrator, your force generators to RigidBody type objects.

- Play a short game using objects of different shapes to demonstrate all the features of your Physics engine. Among other things, you must clearly demonstrate the impact of forces and moments of force on your object (rotational and linear physics).

Collision management should not be implemented in this phase.

Phase 4: Addition of a basic collision resolution system

The goal of this phase is to add a simple collision detection system to the physics engine of phase 3.

To do this, you must, in particular:

- Implement a collision detection system (broad phase) using one of the structures seen in class (eg Grids, BSP, BVH, etc.).
- Implement a collision generation system (narrow phase) which generates collision data (CollisionData).
- Implement all classes relevant to the realization of this project: tree, Contact, CollisionData, Primitive, Plane, Sphere, Box, etc.
- Make a short game demonstrating objects colliding to show all the features of your Physics engine. You must, among other things, ensure that during a collision the contact data is clearly displayed (for example, by pausing your game).