CSC8503: Practical Tasks - Week 1

Summary

These are the day-by-day practical tasks which you are expected to explore during demonstrated lab sessions during Week 1 of this tutorial series. Unlike previous modules, due to the multiple lectures required per day to cover all material, these tasks are listed by day, rather than lecture.

Day 1

Today the concept of Physics Engine was introduced, along with a refresher on Newtonian dynamics. The practical tasks suggested are intended to help you get a feel for the purpose of the material you'll explore over the coming week.

- Explore the framework provided, paying particular attention to its place within the game engine as a whole.
- Play around with some of the framework's default variables, and see what effect changes to those values have on the behaviour of objects within it.
- See if you can add another entity to the test environment (for example, another sphere).

Day 2

Today you covered numerical integration and constraints. These principles, as with most early elements of this module, have broader application than simply game physics. Indeed, most mechanical aspects of game programming involve constraint-based solvers.

- Implement a calculator for the Semi-Implicit Euler Method
- Implement a calculator for the Midpoint Method
- Spawn a sphere at your camera location with a keystroke, which has a fixed acceleration in the direction of the camera. Update its position accordingly (remember the point about scale within the lecture - ensure the motion of the sphere, and its size, are appropriately scaled)
- Implement a distance constraint
- Attach a constraint between the entity you added yesterday, and an arbitrary point in the world
- Extension: Explore the properties you observe during execution if you deactivate the drift constraint

Day 3

Today you explored collision detection, both in the context of broad phase culling and narrow phase detection.

- By now, you should feel comfortable adding new entities to the test environment
- Implement a simple sphere-sphere collision check for non-spherical entities
- Use that sphere-sphere check to cull impossible collision pairs
- Indicate that a collision has been detected by changing the colour of the objects involved

Day 4

Today you were introduced to collision manifolds as an enhancement to the existing collision data.

- Using NCLdebug, draw collision surface areas for objects
- Investigate decomposition of concave objects into multiple convex objects
- Extension: Explore the minimum number of contact points needed to accurately handle various face collisions

Day 5

Today we concluded our physics tutorials, introducing collision response and global solvers for constraint-based systems.

- Explore collision response implementation within the framework
- Vary the Baumgarte constant and time-step - what do you observe? Why?
- Explore different frictional constants
- Stack objects
- Extension: Explore the Jacobi iteration solver - why would this solution often be better for GPU computation?