Assignment 2 - Particles and Springs

Hide Assignment Information

Instructions

Spring Mass System

[25 Marks]

In this assignment, your task is to develop a spring-mass system and a corresponding simulator that support an arbitrary number of particles. The particles can be connected with springs in arbitrary ways and are subject to gravity. There is a flat ground, oriented along the y-axis, that you should model particle collisions with using penalty forces. The simulator should support the following integration methods: Forward Euler, Symplectic Euler, and Verlet.

You must implement the interface commands below. None of these commands need to work *after* the user has started the simulation. Additionally, during grading, these commands will only be called (during a particular invocation of the application) in the order in which they are given here (see instructions and clarifications, as well as, attachments). The attachments include these commands and when opened in the TCL/TK window they will run.

1. system <sys_name> dim <Number of Particles>
   This command initializes the particle system to hold up to the given number of particles. Particles may be initialized to the origin of the world, or may not appear until added by the next command.
2. system <sys_name> particle <index> <mass> <x y z vx vy vz>
   This command sets a position, mass, and velocity for a given particle.
3. system <sys_name> all_velocities <vx vy vz>
   This command sets the velocity of all particles.
4. simulator <sim_name> link <sys name> <Number of Springs>
   This links the simulator to a particular particle system and initializes it to work with a given number of springs.
5. simulator <sim_name> spring <index1> <index2> <ks> <kd> <restlength>
   This sets up a given spring. If the rest length is a negative number, the system should automatically set the rest length of the spring to the distance between the corresponding particles at the time the command is given.
6. simulator <sim_name> fix <index>
   This command nails particle <index> to its current position.
7. simulator <sim_name> integration <euler|symplectic|verlet> <time step>
   This changes the integration technique used by the given simulator and sets the time step of the integration.
8. simulator <sim_name> ground <ks> <kd>
   Sets the parameters of the penalty forces applied to particles that try to go underground.
9. simulator <sim_name> gravity <g>
   Sets the acceleration due to gravity, in unit length per unit time squared.
10. simulator <sim_name> drag <kdrag>
    Sets the global drag (friction) coefficient (Fi = -kdrag vi). The command expects a positive number

Marking scheme

- **[4]** Particle system
- **[4]** Script commands
- **[3]** Forward Euler

- **[3]** Verlet
- **[3]** Symplectic Euler
- **[4]** Ground collision penalty forces
- **[1]** Gravity
- **[1]** Locking of a particle's position
- **[2]** Particle and spring drawing
- **[-4]** Missing readme. Include your **own** readme, **clearly** detailing what you've finished and/or have not finished. If you do not, we will remove marks as indicated.

Instructions & Clarifications

1. Make sure you follow the exact syntax specified in the assignment. We will be

Submit Cancel

vector or quaternion libraries. However, if you have one you may use it. GLM is **very** useful for A1 and A2. https://github.com/g-truc/glm
3. Do not use spaces in file names.
4. Clean your project before submitting it. It removes unnecessary, very large files that VS creates. **You may have to manually remove the ".vs" folder.**
5. Zip your cleaned solution and submit the zip archive file.
6. Name your zip file [Last name]_[First name]_A2.zip
7. If the code doesn't compile or doesn't run you will get 0 points. No partial points will be given.
8. If there are input files required and your code doesn't parse the input files correctly you will get zero points.
9. An example of a portion of the completed assignment can be seen here: https://www.youtube.com/watch?v=r-kUX048JX4
10. **Write a readme file with a description of what you've done, what's missing, or any extra features that you may have added.** We will test your projects primarily with the scripts that are posted online, so make sure it works for those. As usual, we will NOT test your code with malformed or nonsensical commands. If you couldn't get your project to work with our scripts, you may submit scripts of your making that do something we can look at (let us know in the readme though). Do this only as a LAST RESORT!

Performance Advice

- Use an appropriate time step. If the simulation explodes, use 0.0001 for debugging reasons. When everything is working fine, the time step should be in the range (0.001, 0.01), preferably closer to 0.01.
- Try to draw points using GL_POINTS instead of drawing spheres for each particle. You can even add an if-statement to use points or spheres depending on how many particles there are.
- Double-check that you don't accidentally have a memory leak and that you are not re-allocating variables that do not need to be re-allocated.
- Double-check that you don't accidentally copy large amounts of data unnecessarily.
- Another possible reason for performance problems is the console output. Writing to the console can be expensive. Commenting out your console output may improve performance.

General Advice

1. You may want to implement additional commands to facilitate debugging.
2. Make sure you properly clean up the scene, before instantiating any new objects to avoid crashes.
3. Work incrementally!

Additional resources for assignment

- See attachments. You can load TCL files by right-clicking the TCL/TK command line window (TkCon) and Selecting File -> Load File.

Examples

Complex Full Euler Examples

0:00 / 0:15

Simple Spring Symplectic 0.01

0:00 / 0:03

Simple Spring Symplectic 0.01 with Kd set to 0.1 (very underdamped)

0:00 / 0:27

Due on Mar 8, 2024 11:59 PM

Attachments

- cube_complex.tcl (2.13 KB)
- cube_complex_full.tcl (2.06 KB)
- cube_simple.tcl (2.16 KB)
- many_particles.tcl (589 Bytes)
- one_particle.tcl (305 Bytes)
- simple_spring.tcl (380 Bytes)

Download All Files

Submit Assignment

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File   Record Audio   Record Video

Comments