Concordia University

COMP 477 - 6311

Animation for computer games

Prof. Tiberiu Popa

Fall 2021

Purpose

This homework will give you the opportunity to acquire practical experience with physics simulation and time integration.

Submission

- The assignments must be done individually. On-line submission in EAS under Assignment 3 is required - a hard copy will not be read or evaluated.
- You must modify ONLY two files: submission/a3solution.h and submission/a3solution.cpp.
- You must submit a zip file containing ONLY these two files in a flat folder structure (i.e. no subfolders in the zip file).
- Please use .zip files and not other compressing formats extensions such as .rar.
- Failure to adhere precisely to these submission instructions will result in a grade of 0.
- The soft submission deadline is on Monday October 4th, 13:00. Hard deadline November 29th, 2021, 23:59.

Evaluation

- These two files mentioned above will be placed inside the codebase and your solution will be evaluated by running the codebase with your two files.
- The evaluation will be done demo style in a short 5min session with your TA. You should be prepared to answer questions about your implementation.

- Evaluation is based on the functionality of your submission (i.e. we will not look or debug the code) as well as the questions from the Instructor or TA.

- We run plagiarism detection software on the code you submit.

What's given to you already

You are given a simple spring 2D editor developed in QT5 using QTCreator. You are expected to use the same tools in developing your solution. This framework allows you to create, edit, load and save an arbitrary configuration of springs. The functionality of the editor is explained below. But similar to A2, the points are of two types: fixed and variable. The fixed points, marked by an "X", are constraint to those user location while the others can move during the simulation (i.e. they are the variables of the time integrator). The functionality of this framework will be presented in detail during the lab session. You are not allowed to use any external code or library other than what is given in the codebase.

Expected behaviour

The framework provides 3 operation modes that can be selected from the File/Set Mode menu.

- Select and Edit Joints. In this mode, using left mouse click and drag you can create new points or move the 2D position of existing points. Right mouse click deletes the point and all incoming or outgoing links. By default the joints are variable. By double clicking a point you can toggle between fixed and variable points. The fixed joints are marked with an X.

- Select and Edit Links. In this mode, using left mouse drag you can add springs between points. A mouse drag operation between two points that have a spring already will remove the spring. Unlike the previous assignments, the direction of the mouse drag should not change anything as the spring is symmetric (i.e. we do not have the hierarchical structure that we had for forward kinematics or even the inverse kinematics).

- Showtime. This is the mode you have use to implement the 2D physical simulation. Even if there is no interaction from the mouse there could still be movement so the TA will show you during the lab how to use the timer functionality to do the time integration. In this mode only joints that are variable can be selected and dragged. By clicking on a variable joint, this node temporarily becomes fixed until the mouse is released. Dragging the mouse should change the position of the selected points and update the positions of all other variable nodes based on the simulation.

More details about the codebase as well as expected behaviour will be provided during the lab session.

Parameters

By clicking on the "Parameters" menu command you will get a dialog box where you can set some of the parameters: gravity constant, point mass, damping coefficient, spring stiffness, time step, etc. Changing these parameters should have immediate effect (i.e. after pressing the OK button all these parameters should be updated in the simulation.

One of the most important parameters is the "Use Implicit Euler" check box. When unselected (by default) you should implement the explicit Euler method. When checked it should switch to implicit Euler. The implicit Euler, if implemented correctly, can give you up to 30% bonus marks. We will test your code with various parameters.

Evaluation notes

We will test your code with one connected component and at least one fixed point, but otherwise any configuration should work.

Implicit Euler

The implicit Euler method will require a root finding algorithm, a simple Newton-Raphson method should work. The algorithm in itself is very simple, but there is considerable complexity from the fact that your function is a multi-variate function (i.e. the root finding is in the spatial domain and not temporal) and from the fact that you need to extract and program the computation of the partial derivatives of a fairly complex high-dimensional function. You are allowed to use MAPLE or a software that does automatic differentiation, but it is plagiarism to get it from someone else even when the other person compute it using MAPLE. If you jse an automatic differentiation software you need to send me also the MAPLE sheet for this problem. As not too many people will go this route, should this be the case you can send it to me by e-mail. However, you can also program it by hand.

I am not sure how to state this any more clear: there cannot be any extension for this assignment, it will take a lot of time to code and debug, we will not debug your code so you should start as soon as possible.