Concordia University

COMP 477 - 6311

Homework 2

Prof. Tiberiu Popa

Fall 2017

Purpose

This homework will give you the opportunity to acquire practical experience with key-frame animation and interpolation. This is a programming assignment which requires you to write a program that allows the user to create and play key-frame animations for a given character. The key-frames will be interpolated using four different methods that you have seen in tis class: matrices, Euler angles, quaternion (linear) and quaternion (SLERP).

You will first create a visual modelling tool that allows the user to intuitively pose a character in a given frame. Then you register this pose to a given key-frame and move on to the next key-frame. Once multiple key-frames have been modelled, the animation can be played using the following three methods presented in class: SLERP, linear interpolation of the Euler angles, Matrix linear interpolation. The animations can be loaded and saved in a standard file format described below.

Submission

The assignments must be done individually. On-line submission is required - a hard copy will not be read or evaluated. Submission deadline is wed Nov 1st, 1:00pm EST. No late submissions allowed.

What's given to you already

You can use as a starting point your solution for H1 or the codebase provided in moodle.

Expected behaviour

Your modelling program should have two modes: an editing mode and a animation mode. Pressing the key *m* toggles between these modes.

In the editing mode, the + and - key increment and decrement the active key-frame that is being modelled. Key-frames start at *0* and the active key-frame increment/decrement is not cyclic. When active frame is 0 the - key has no effect. When the active frame is the last in the list, the + key create a new keyframe copying the transformations from the previous frame. All keyframes are equally spaced in time. By default (i.e. when the program is started without any arguments) there is only one key-frame in the list.

In the animation mode, the + and - key increment and decrement the active frame. So if, for instance, the animation has two key-frames at times 0 and 10 repeating strokes of the the + key will interpolate the animation from frame 0 to frame 10 using increments of 1. The animation mode has a playback sub-mode that is toggled using the *p* key that play the animation continuously. The speed of the animation can be controlled using the keys *j* and *k*.

Your program has to have load and save functionality. If the program is ran with an argument, this argument represents an animation file that has to be loaded by default. Otherwise, by pressing keys *l* and *s* the program should load the animation from a file *load.anim* and, respectively, save the current animation to a file *save.anim*. In the case of a load command, the current keyframes should be discarded.

You are required to generate 3 animation saved as: *wave.anim* , *breakdance.anim* and *dance.anim* . The animations should match the names: *wave.anim* should depict a simple hand waving animation, *breakdance.anim* should depict a breakdance sequence and *dance.anim* should depict a dance of your choice.

Changes of the system state such as editing mode, current key-frame or current frame have to be signaled by a message to the standard output console. (You will be shown in the lab how can you add a console window to your project).

You have to implement four interpolation modes. By default the interpolation mode is matrix interpolation. Pressing the keys *1*, *2*, *3*, *4* sets the interpolation mode to matrix, Euler angles, quaternion (linear), quaternion (SLERP), respecively.

Practical issues

The first part of this homework is modeling a key-frame pose. You can assume the skeleton file corresponds to the one from the previous assignment. You can use the geometry and the weights from the previous homework.

The format for your animation file is as follows: each line corresponds to one frame. The first number is the key-frame (can be any value). The following numbers are a group of 17 transformations: rotations of each bone with respect to its local coordinate frame. The rotations should be represented as quaternions, therefore there are 4 numbers for each transformation. In total each line contains $17 \times 4 + 1 = 69$ numbers. The key-frames can be assumed to be in ascending order. You can assume that the root transformation is always identity.

Note that the codebase provided uses matrices to represent transformation. For this homework it is easier if you store them as quaternions so we recommend that you write your own quaternion class. If not up to 3 our of 10 marks will be deducted. You will then have to write the conversion from quaternion to matrices and Euler angles. Note, however, that, if you implement it correctly, you do not actually need the reverse conversion.

Submission Deliverables:

1. Submit well-commented source code including README files, data files, if any, along with the Project/Workspace files, and any other instructions for compiling/ building/running your program. The source code must be submitted electronically in a zip format with all the required files (example: Hwk2_IDNumber.zip) to https://fis.encs.concordia.ca/eas/ Submit this zip file as "Homework 1". Please do not e-mail the submission.

2.The homework has to compile and run out of the box on the lab computers. If it does not, a grade of zero will be awarded. In very special circumstances, the homework can be demoed on the student laptop, but this is only with the explicit consent of the instructor. Please ask first. In any case the homework should still be submitted as mentioned above.

3.A brief write-up about your program, showing the hierarchic structure used to do this homework, usage instructions and any special features that you would like us to consider.

4. The three animation files described above.

For this homework you are not allowed to use an external quaternion class, you must implement your own.