VSFX 705 / Exercise 3 / Spring 2013 / Fowler*

Exercise 3 – Data Parsing with Python

DATE DUE: Class 16
DATE ASSIGNED: Class 11

Goals:
This assignment will focus on the student becoming familiar with python and data parsing. The goal is to take a motion capture text file, which contains point positions (x,y,z) of motion capture markers and use python to separate the data and write a new file to be used in Houdini.

Requirements:
Given a motion capture data file in txt format, your python script will parse this file and write a new file to be used in Houdini.

Optionally, for those looking for more of a challenge, you may use bvh file format and take this data into the 3-D package of your choice. An excellent resource is at http://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html

Considerations:
There are excellent example of this process on Malcolm Kesson's 705 website.
In particular I have put two aside on my website for easy access and clarity. The two I have chosen are Nate Usiak (taking the data into Houdini) and Sorya Serei (taking the data into Maya). These are excellent starting points. You will be taking your file into Houdini. Note that it is not necessary to write out the velocity as show in Nate's example.

Suggestion for the optional exercise: start by using the same type of files that are provided and make sure your code is working with the txt version. This is the format used in the previous exercise examples. Then make your script so that it reads in bvh format instead. Bvh does not have marker data, but only skeleton data.

Submissions guidelines:
Create a directory named LastnameFirstnameExercise3

- you should develop an html or pdf page that contains a description, full code and reference to images. Websites are encouraged but not required and should be noted in a readme.
- LastnameFirstnameExercise3.mov – the resulting animation
- LastnameFirstnameExercise3.hipnc file – and Houdini otl file
- LastnameFirstnameExercise3.py files and any additional files named appropriately

Grading:
Clean efficient code and a cool visual are the emphasis.
Meeting the minimum specifications, 80%. To move your grade above 80% go beyond the specifications, demonstrate exploration and understanding.

Although you have been provided with examples, you are NOT TO HAND IN THE CODE GIVEN IN THE EXAMPLES. YOU ARE TO CREATE YOUR OWN SCRIPT. This must be properly commented and understood by you – the examples are only for reference. Handing in the exact code will result in a grade of zero.

Be creative, have fun. See Hints below.

Additional Notes for HOUDINI – This matches the example given. You don't need velocity – modify appropriately.

To take the resulting data into Houdini you must create a digital asset.

1. **File->New Operator Type** – brings up a dialogue box.
    - name it appropriately ie. fields Operator Name and Label
    - Operator Style must be set to **Python Type**
    - Network Type must be set to **Geometry Operator**
    - Save To Library should point to your local folder (don't forget to hand in the otl!)

2. Hit Accept – an **Edit Operator Type Properties** window appears
    In the **Basic** tab be sure to set Minimum Inputs and Maximum Inputs to zero.
        set the Label to be something unique (maybe include your name)
    In the **Parameters** tab set up two variables, one for the data files, one for velocity
        (these parameters will be used in the code tab – they must match
        I've used mocapFiles and velocityScale)
    In the **Code** tab place the script (courtesy of Nate Usiak found here)
        make sure your parameters for source_file and velocity_scale match the code
        ie. what is in quotes should be the names you used above

3. Throw down a geo node,
    bring in a "mocap" node (from your digital asset – hit tab and type in the name).
    Point the source_input parameter to your files
    make sure velscale is set to 1 and you are all set.

This is also on the website under Python in Houdini Examples