1 Programming

In this assignment you will write a program (with some C++ code already provided to you) to create and animate a jointed stick figure. The program takes as input text files descriping the figure and the keyframes, and outputs a text file with all the animation parameters for each frame. There will be two programs provided, one which replays the animation in OpenGL, and one which generates RenderMan output. You should study these programs to understand them—in subsequent assignments you will need to generate your own RenderMan output etc.

You should find the software, and associated README files, in the directories `stick_figure_*`.

The one file in which you need to add code is in `skeleton2d.cpp` in `stick_figure_preprocess`. Part of this is adding code to compute Catmull-Rom splines of the kinematic variables, and part is to do the forwards kinematic computation at each frame to figure out the geometry of each link and write that out to a file.

Note: the higher-order accurate boundary condition at the end for nonuniform knots in Catmull-Rom is the same as the one at the start (given in the slides) except with indices reflected, i.e.:

$$s_n = \left(\frac{t_n - t_{n-2}}{t_{n-1} - t_{n-2}}\right)\frac{y_n - y_{n-1}}{t_n - t_{n-1}} - \left(\frac{t_n - t_{n-1}}{t_{n-1} - t_{n-2}}\right)\frac{y_n - y_{n-2}}{t_n - t_{n-2}}$$

where I multiplied top and bottom of the fractions by -1 to make the t-differences positive.

The skeleton is a 2D model, where each link is a simple line segment of some specified length. Each link, except the root, has its first endpoint attached to its parent at some specified fraction along the parent. The kinematic variables are the (x,y) coordinates of the first endpoint of the root link, and then angles (given in degrees, not radians) for each link. For the root link, that angle is the angle from the vertical. For the other links, that angle is how much the link is rotated from its parent. See the provided source code in `skeleton2d.cpp` and the example input files for more details on how the code expects this information to be formatted.

The output of the preprocessor is a description of the geometry at each frame, i.e. the (x,y) coordinates of the two endpoints of each link in the model. The OpenGL program will read this in and present the animation interactively; the RenderMan program will read it in and produce a RIB file for each frame which can be rendered by a RenderMan compliant renderer to produce high-quality images.

2 Analysis: Kinematic Models

In this problem, you are to think about how to create a kinematic model of you. That is, think of all the bones you can move independently in your body. This doesn't include, for example, facial expressions, the bones in your ear, individual vertebrae, etc.: just stick to bones or groups of bones you can move independently from the rest of your body. As discussed in class, in reality most if not all joints in the human body can't be accurately modeled with rigid links and rotating joints—however, you can still do a very good job. **List all the degrees of freedom you think are necessary to accurately model yourself, including a diagram (or several diagrams) of your body to label where they are, and include any explanatory notes you need to make it clear what the degrees of freedom are.**

For example, your jaw has (approximately) three degrees of freedom: you can translate it in the plane parallel to the ground, and rotate it up and down. The rest of your head has no degrees of freedom until you get to the neck.

Hand this in to me in class, office hours, or under my office door before I get in the morning after it is due.