Human Motion as a Function of Body Parameters

William Fletcher Cole
wfcole@ucsc.edu

Abstract

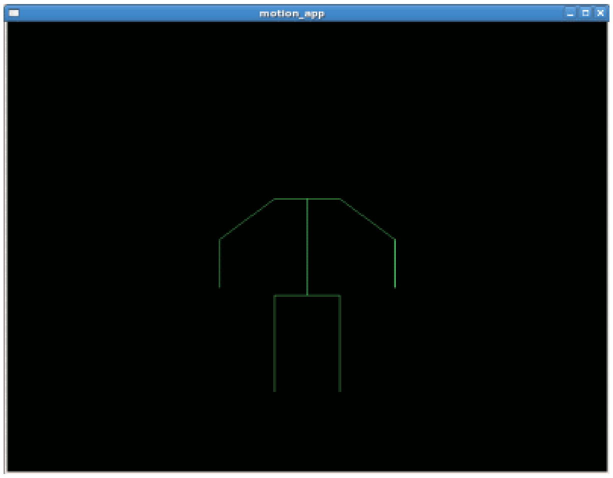
This project is an attempt at representing the effects that different varying parameters of human bodies can have on the motion of the body. The project will be presented as an application in which users will be able to modify the values of various parameters of the human body in order to observe the resulting effects on the model of the body and the impact on the animation the model is capable of running through.

Introduction

For the project I decided to focus on three body parameters which the user would be able to modify. I chose to allow the user to change the height, weight, and weight spread of the human model. The weight spread function allows the user to choose to put more weight toward the upper or lower portions of the body, allowing for a more varied representation of the human body. The user is able to change the parameters using arrow keys for the height and weight, and using number keys to effect the weight spread. At any time the user may choose to run the model through a simple demonstration which will help to demonstrate the significance of the impact that the edited parameters has on the model of the body. I chose to implement a jumping-jacks animation for the project because it is simple enough to implement using a programmatic approach to animation and its movment depends equally on all parts of the body so it allows for a comprehensive visualization of the various changes available to the user. While the height and weight parameters produce little variance in their results, the weight spread parameter allows for a much more varied model because the different areas of the body will begin to act differently during the jumping-jacks.

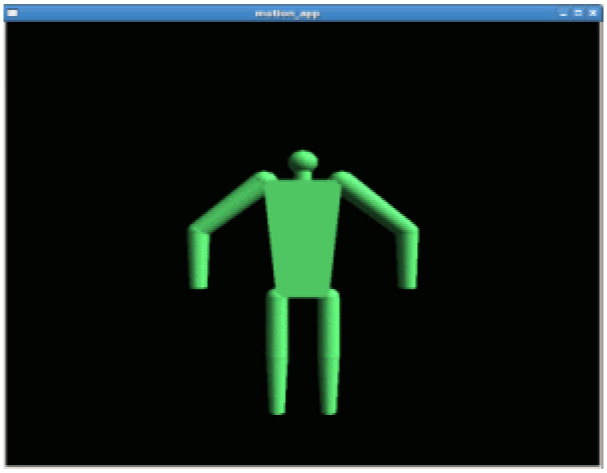
The Model

To implement the model of the human body I decided to just approach the task within Open GL using mainly primitives and some GL_QUADS rather than import a more detailed model from a 3D modelling program. This approach allowed me to edit the model on the fly from within the program more easily where as if I had imported a model I would be having to deal with a bunch of joints and such that might not have fit exactly into my setup within my code, and if I wanted to change some way with how I was setting up the hierarchical body-part structure or the effects of the animation I would have to import a new model or figure out how to re-adapt an old one. The drawing functions for the model appear in "drawfns.h/c" while they are used in "body.c".



The result is a not too detailed representation featuring spherical primitives for the joints and head, cylindrical primitives for the limbs, and a simple structure of GL_QUADS for the torso. As a result the model doesn't simulate the effects of the parameters and animation on more subtle aspects of the human body but still allows for a generalized conception of the overall impact of the parameters and the animation on the body.

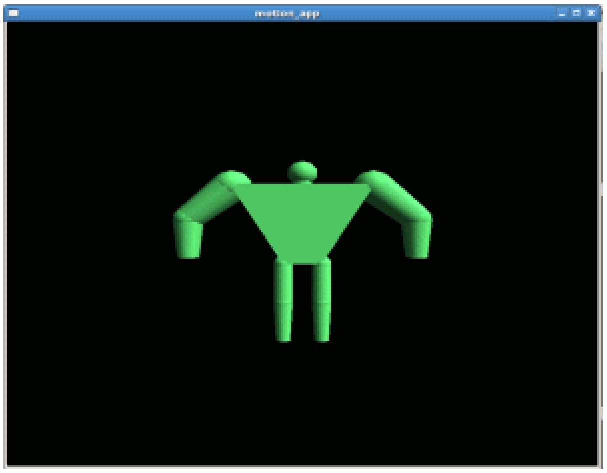
The implementation of the model in terms of structure consists of a series of body parts which build off each other in an hierarchical fashion. For example during the drawing of the model the upper torso is called first, which then pushes a matrix onto the stack which is now translated to its origin, which now allows the program to call the functions for the shoulders, translate to the position of the shoulders, then push another matrix onto the stack which sets up the functions for the arms until all of the upper body parts have been reached. Each individual body part only stores their position and orientation in respect to their parent body part, which allows for them to exist independently as part of the hierarchical structure. The representation of the model is drawn using the current positions, lengths and orientations of each body part stored in the structure for the model. The file "body.h" contains the struct for a body part demonstrating the necessary fields for the structure.



The Parameters

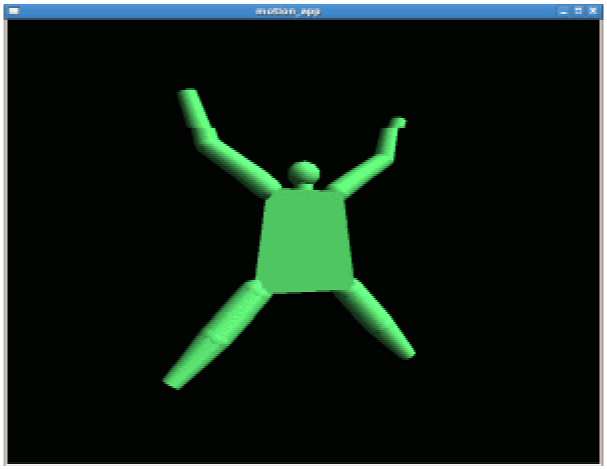
The user is able to edit the parameters for the height, weight and weight spread of the human model. When the user adjusts the height of the model the related aspects of each body part are adjusted appropriately. For instance the length of the torso is increased when the height of the body is increased, while the width of each body segment is increased when the weight is increased. However the current weight spread may dictate that the effect is greater on certain parts of the body than others.

The user modifies these parameters using key presses. The model is then updated in real time, even during the animation, to reflect the current state of the user specified parameters. Depending on its value, the weight spread decreases the width of either the lower or upper body parts, and increases the width of the other.



The Animation

I decided to approach the implementation of the jumping-jacks animation for the body in a programmatic fashion by keeping track of the current rotational values for each joint and updating them according to the current steps through the animation. During each step the effected joints have their rotational values edited according to the current parameters specifications as well as the current direction of the animation. During the first half of the jumping-jacks animation the joints are each rotating outwards and expanding while during the second half of the animation the joints rotate back to their original orientations. This approach simulates the movements associated with jumping-jacks reasonably well. I also implemented simple physics to demonstrate the jumping of the model. When the model starts each half of the animation it kicks off the ground with an initial velocity kept track of in “body.c”. There is always a constant force of gravity which accelerates the body downwards, so whenever it is jumping it is either slowing down on its way up or speeding up on its way down. The current animation rotation values and physics values are kept track of in “body.c”.



Functionality

The user interacts with the program by using the arrow keys to adjust the height and weight of the model and using the 1 and 2 number keys to adjust the weight spread for the model. The file “window.c” keeps track of the current user settings and updates them whenever one of these keys are pressed. It also calls the update method within “body.c” which then goes and appropriately updates each individual body part according to the rules for that body part. The user may then run the model through the jumping-jacks animation by pressing ‘j’ on the keyboard, and can press it again to reset and stop the animation. The parameters may be modified at any time, including during the animations.

The user can also rotate the camera around the scene and zoom in and out in order to view the model and animation from different perspectives. The file “window.c” keeps track of the current rotation and zoom for the scene and calls the draw function for the body using the current values, which then rotates and translates the scene accordingly. The user can press ‘w’, ‘a’, ‘s’ , and ‘d’ to rotate and zoom the camera.