Joe Bjorkman
KatanaForging.cpp Documentation

Project Description / Marketing Description:

For my project, I set out first to recreate a photo-realistic katana without using a texture map. However, as I researched katanas, I found myself much more fascinated with the process of making the blade of the katana. I knew a little about them from before, but the intricacies of the process of giving the katana its signature curve, was still somewhat of a mystery. It turned out that the blade of a katana is forged straight and then some mixture of clay and charcoal -- the exact ratios depended on the particular smith who was forging the katana, and are a closely guarded secret, even today -- was applied in varying thicknesses along the blade, thick on the back and thin or non-existent on the front of the blade. This coating of clay allows the blade to cool at different rates, quickly on the front, and slower on the back. The clay also gives a katana its *Hamon* (see **Figure 1**); which supposedly shows the craftsmanship of the smith. The difference in cooling has a number of desirable effects on the properties of the steel as well, including making the edge hard, but the back softer. This means the katana will not shatter when used, and it can have incredible cutting power. The curve is also conducive for cutting, whereas a European Broadsword is meant more for piercing armor and brute forcing its way through things.

I changed the focus of my project to focus on the curve that the blade gets from the cooling process. While, in its current state, it simply shows a blade that starts straight and then jumps to curved, with no regard for either the arrangement of the different types of steel, the properties of the steel, the temperatures, or the clay; it is a proof of concept that it is possible to take a straight 3D object and curve it, while keeping its cross-section uniform. **Figure 2** depicts nine different styles of layering different types of steel through the katana's cross-section. I do plan on revisiting this code one day, and adding in all the variables mentioned above. It is my hope that one day I will be able to figure out exactly how to make a katana as well as the master smiths of old, and to rediscover secrets that have been lost to time. **Figures 3** and **4** show the two states of my project.

Failures / successes:

There were quite a few failures during this project. Well, more than quite a few. My issues started with defining the model itself, before I even touched the curviness of it. The biggest issue I had was the 6 triangular Bezier patches on the tip of the blade. I started by trying to define every point, but then, after a suggestion from Professor Bloomenthal, I just decided to use the tip of the point as two corners in the same function that made the rectangular sections of the blade. I eventually got the corner points squared away and ironed out a few of the bugs; then it was on to the curve.

I initially struggled with how I was going to do the curve; I first started with trying to make it so a user could drag around points of the blade and have other points move as well; but this led to much frustration and some very weird shapes, but no curve. My first breakthrough came when I was analyzing the code in Widget.cpp; and I focused on the buttons. I then decided that I would simply make a button that made the curve happen. But a new problem presented itself: how was I supposed to make the blade curve, not just move or tilt? I decided that the answer to this was to add in resistances to every point in the x and y directions -- I ignored the z direction as that was not going to change at all, because the sword was not going to get either thicker or thinner -- and then moved all the points by an initially fixed value multiplied times the resistance of the point. To do this, I modified the provided Patch.h and Patch.cpp so that the pts array was a struct with four elements: the current point location; the starting location of the point; and a resistance for both x and y. I spent a good deal of time getting the resistances done correctly so that the sword finally had a curve. The final task was to make the movement of the point not static, which is when I decided to add the slider. The slider was by far the most straightforward, easiest part of this assignment.

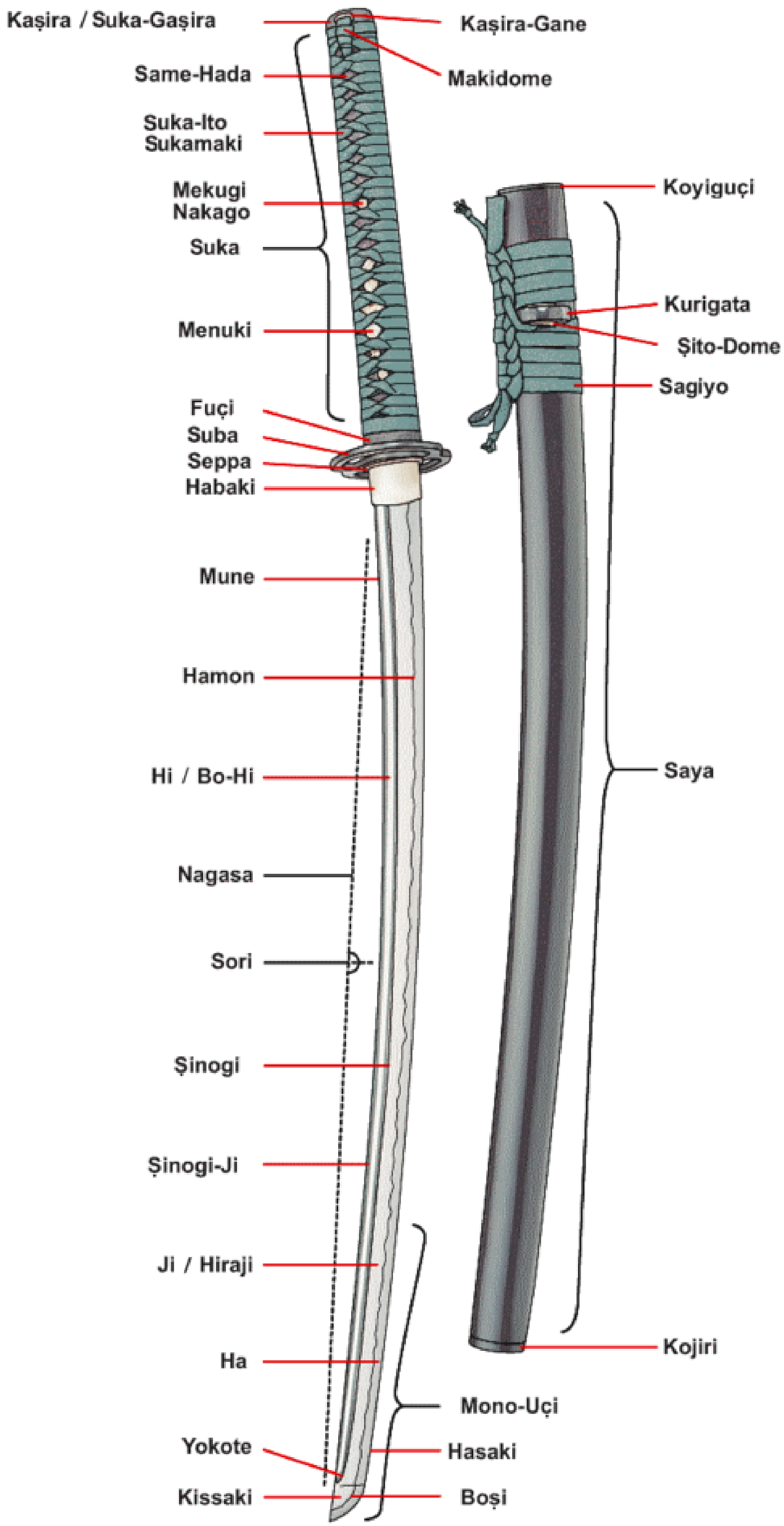

Figure 1: Visual glossary of Japanese Sword Terms

http://en.wikipedia.org/wiki/Japanese_swordsmithing#mediaviewer/File:Katana_anatomi.png

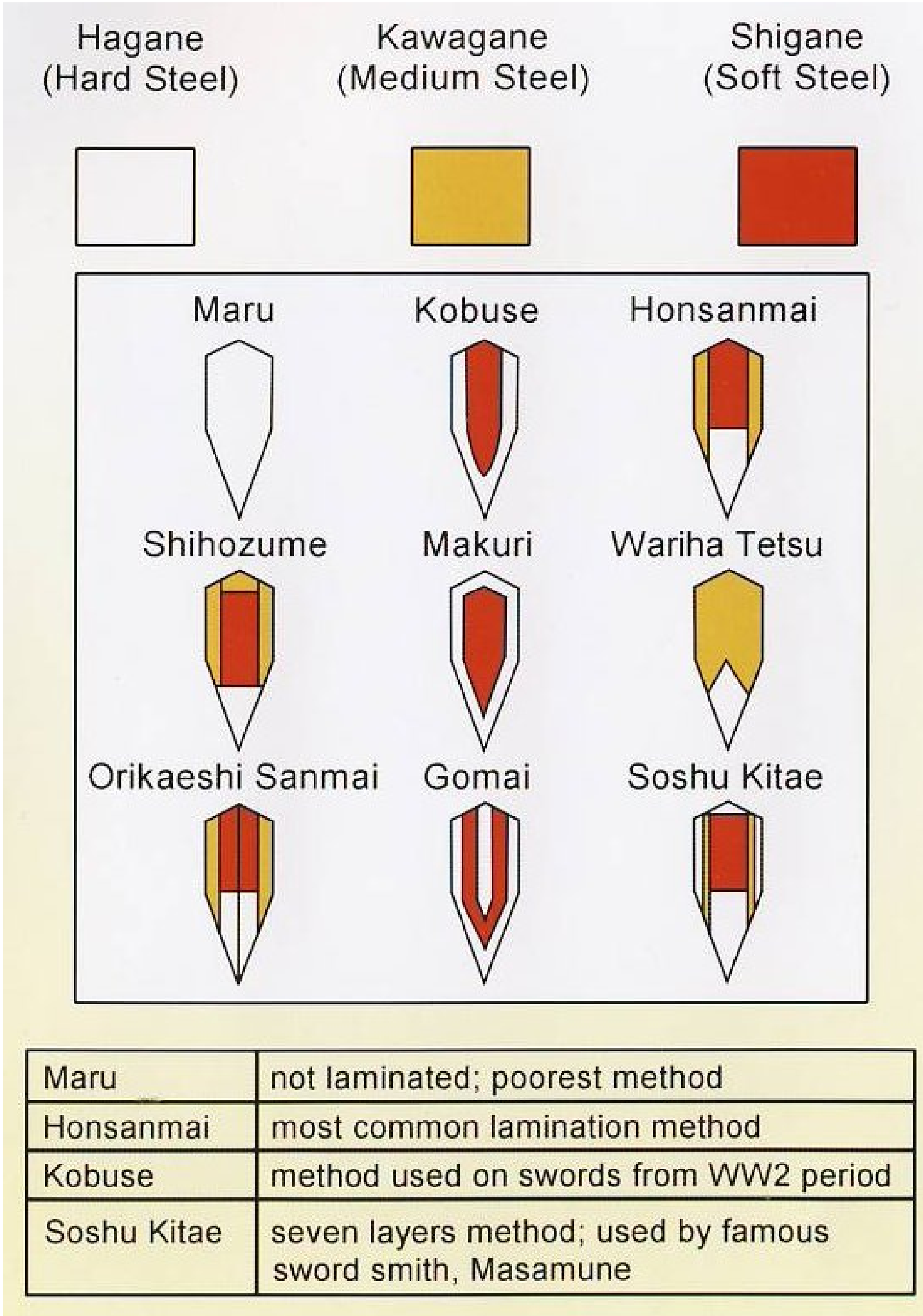

Figure 2: Nine different katana cross sections, detailing the types of steel used. Technically, my model is of the Maru type.

http://upload.wikimedia.org/wikipedia/commons/c/c4/Katana_brique.png

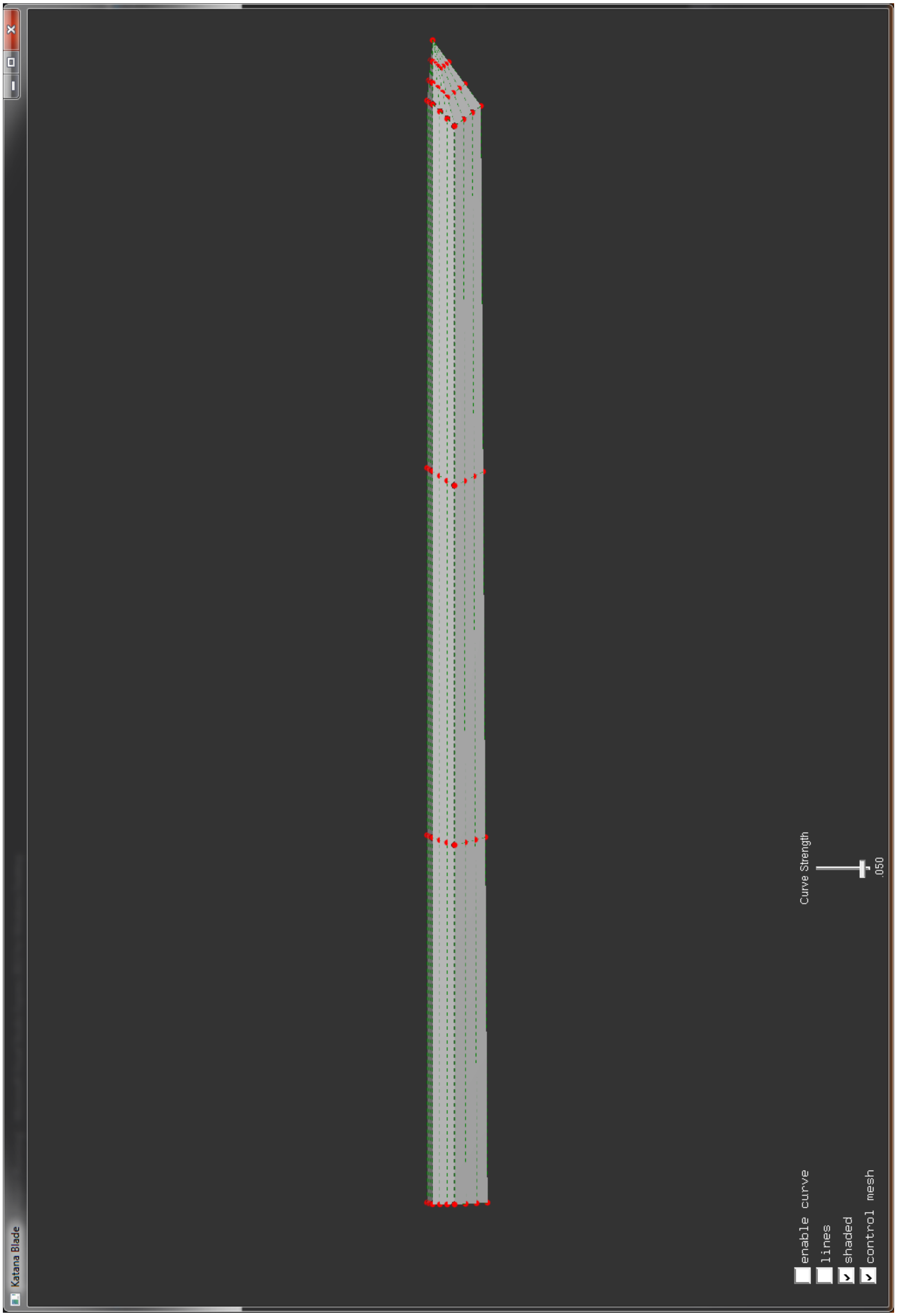

Figure 3: Blade in pre-curve mode, with the control mesh visable

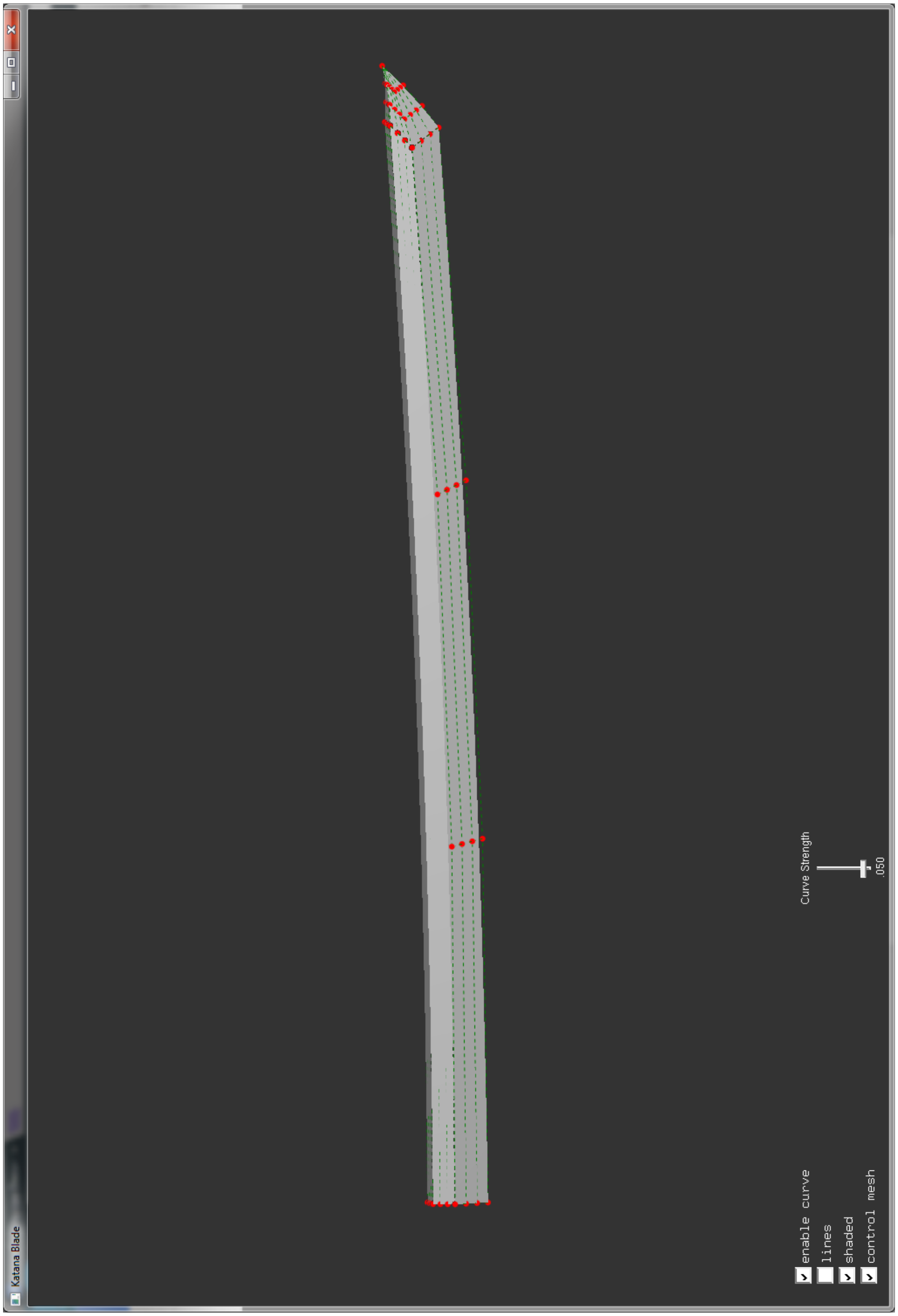

Figure 4: Blade after applying the curve with the default value. The slider seen can change how much the blade curves.