We started out by developing what later became ANIMAC. At first we called our machine "The Bone Generator" because it made sections of straight lines that could be hooked together and could be individually animated or moved in three dimensional space and to determine what a bone was, you had to determine where it was to start in X, Y, Z space and in which direction it went from there and for how long in order to determine the length of this bone. The parameters that determined which direction it was going in, also determined the actual length projected onto the face of the tube. If you saw a bone from the side you saw its full length but if it were pointing toward you, you saw only a portion of it. A bone was composed of a bi-stable multi-vibrator or a flip flop which the first bone in a series would get a pulse that started it and what we meant by starting it, it essentially put a signal on a line that governed the opening of a lot of gates, sampling gates and the inputs to the gates were the parameters that governed the position and some of the qualities and characteristics of that bone. To program it we had a patch panel that determined by going from one bone to the next bone to the next bone we determined the pattern or order in which the bones were drawn and we had a thing called the fly-back bone so that if we wanted to we could just go back to the mere starting place and we always our figures, had a navel point, we'd always flip back to the navel point, we'd go up and go out an arm and go back to the navel point, go up and go out another arm and back to the navel, go up and go out to the head. Those were all fly-back bones and we would fly back by just collapsing the information that was contained on a capacitor and I'll tell you about that as we get there. In order to determine the length of a bone we used time as the basis. We'd start drawing in a certain direction determined by the parameters of direction and we'd go in that direction until we'd turned that bone off and then essentially we'd wait there until we drew another bone from that point on or flew back depending on how we wanted to go or actually turn around and go back the other way on that same bone. We called those turn-around bones. The length was determined by plugging into a place where you could plug a timing circuit or a counter which was reset after each bone so when you started a bone you also started that counter and that flip flop was plugged into the counter and when it got up to the counter it would turn that bone off so it was pretty much all digital. That determined the length of the bone. The next bone would be plugged into another count and so forth and you varied the counts depending. A count represented some number of high frequency units that was part of the clock network of the whole machine. Anyway, while the bone was turned on we had all these gates that were open and the gates would have such values as . . . to say DC values determining the starting point, X Y Z of that bone if it were a beginning bone or a base bone or a navel point because if you moved that point you move every bone in the whole thing, so everything hung together because one bone took up after the next bone.

The patch panel was color-coded and it was a big patch panel we got out of the junk yard someplace 'cause nobody used that stuff anymore. If you understood the code you could actually see the bones on this patch panel and there was some overhead on each bone, I suppose you'd call it overhead but there were places where you'd say, okay, here's an input to the bone and that would be a certain color like a green or something and the output might be a blue. Well, if you were going to bone number one, you brought a start pulse that was located somewhere on that panel and you'd plug from the start pulse to the first bone and then you plug from the output of the first bone into the second bone and so forth. The inputs to the parameter gates were not located on that panel. They were located down a little lower on the face of the Animac and there were hundreds of those suckers, just hundreds. Because if you had thirty bones, we had thirty bones on that first Animac or the first solid state Animac, there must have been ten or fifteen parameters for each one of those so it seemed like an overpowering kind of thing but there were a lot of things you wanted to do with each bone. Remember now, we've determined how the parameters are organized, sort of a control structure if you will . . .

We had a parameter called Theta which controlled the angle of the projection of the bone on the X Z plane and the X axis, that was theta and the angle of the bone between the Y axis and the bone was called phi. So with theta and ft we could point the bone in any direction. Those inputs, theta and ft, start out being potentiometer inputs, DC essentially, that would change slowly with time, meaning there might be a slight change every thirtieth of a second or every 24th of a second or whatever depending on the frame rate. Theta was some value of voltage that represented some angle between 0 and 360 degrees and it was a DC value that existed over the same period of time that existed the length of the bone. This DC value went down a wire to a sine/cosine generator and there we took the sine of the angle and the cosine of the angle by a keen little trick. We had this all coordinated with a high frequency sine wave and cosine wave that were located in the proper phase angle to one another so that one is the sine and one is the cosine. We had a sample and hold circuit that would be controlled, where it sampled the sine wave and the cosine wave was where theta was. So it would sample that and hold that and then sample it again so over a period of time we might sample 20 or 30 or 40 times per bone and put that in a big capacitor and make that appear as a DC value that represented the sine or the cosine of the angle theta. That value then was presented to an integrator. Now the slope of that coming out of there was proportional to the sine or cosine of the angle and by applying that to the X or Y axis of the oscilloscope you got the thing going at the appropriate angle on the oscilloscope. Now actually it was a little more complex than that because when you're going in three dimensions you have products of the sine of alpha and the cosine of beta times each other. And because the multipliers weren't very good we used . . . you may recall from your trigonometry, the sine of alpha plus beta equals sine alpha/cosine beta plus or minus the cosine beta/sine alpha and so forth. You had components that were the results of multiplication and we'd take and add these together and have two times the value we wanted and divide that by two and now we had, without any multiplication, we had actually multiplied the sine of theta and the cosine of ft, the sine of ft and the cosine of theta and so forth and all the combinations just with addition which was easy to add and hard to multiply. So we got very accurate and nice things that gave us this 3 dimensional capability. Well, you can see as this DC value that represented the right value such that if it went into an integrator it came out here as a ramp and the X integrator and there was a Y integrator . . . actually there was Z integrator too but let's not get too complex here. This came out with a ramp and on the oscilloscope what you saw was something that went at the angle theta as desired when these were applied to this. Actually we went from here into . . . we had a Z integrator. These were applied to great big sine/cosine potentiometers (i.e., "pots")--again a method of multiplication without multiplying and so there were some sums (?) and additions and what this was . . . big horizontal and big vertical and these now went to the horizontal and vertical plates of the oscilloscope axis. So when we turned this knob, this was like a camera angle network, that's what we called it. It was like changing the camera relative to all these things. These individual inputs out here in the gates, these gates were all attached, they were all driven by the same flip flop that was at the head of each bone. The gates would open and we'd put theta and ft and R, what's that, well, we didn't only just make a bone and go from here to here, we added a high frequency component that was spinning around like that and R was this distance, the radius of the spin and that gave you volume. So we called that the volumetric network that would multiply this end and the patents show all these components. If you understand this much then you have a formula for X, a formula for Y, a formula for Z and a formula for horizontal and for vertical. This has some K T, where this is the ramp part of it and plus there's a sine omega ti and a cosine omega ti component and that's this (sketching), the spinning vector. Okay, now let's take a little object. Suppose I wanted to draw a vase, immediately it comes to mind that a bone goes there and that this R now is a function of time that looks like this. This is R, this is time. So the vertical axis has the length of this changing shaped vase but now if at the same time I put a video camera here like this and put a piece of paper under here like this and let's suppose I paint a flower like that and this is scanning like this or whatever and while this is drawing like this, suppose I use this information to modify the intensity and then I can get a flower that's put on there like that and this we call our surface characteristics and the idea was to keep going until we might be able to animate with the quality of a Rembrandt. Still goin' to do that.

Oh, I know what I didn't talk about. You have all these hundreds of inputs required to make the thing happen and to change it over time was the thrust . . . after this, the main thrust of our development was to make things change over time which eventually culminated in what we called key frame programming where we would turn knobs until we got a character of whatever in the position we wanted and the color we wanted and everything we wanted and then we recorded instances of all these parameters and then we'd say at some later time the character's going to be in this position, another key frame, key frame number two and we recorded the instances that put it in that position and then we could automatically go from the first set of instances to the second set of instances taking as many frames as we wished. Not only that, we could shape the velocity curves by, for example, if you had some value to start with and some value to end with, we could go straight, called straight line programming or we could go slow out, slow in or we could go slow out and fast in. We could go anyway we wanted so that the parameter changes were divided up on frames so there'd be accelerations and decelerations and things like that and sometimes we had a thing to make it as smooth as possible and sometimes we could use sines or splines or any kind of a mathematical function that you wanted. That made for some very nice animation but we didn't change this a whole lot. Now this has an equivalence, all this kind of thinking and when I say all this kind of thinking I mean, Look, I can point here and I can say oh yeah, if I put something there this is going to happen, if I put something there this is going to happen to this bone or this is going to happen to that bone. Well, if you take all this and put it together you get into a mathematics of vectors and matrices. Well, that's fine for the mathematician but you get . . . in a matrix you get down here and you have a whole bunch of equations and relationships and you get something down here and it's a combination, it's a small part of this input and a small part of that input. It's combined mathematically so you have fewer components in your machine that have to worry about that. You don't have to make multiplications twice or three times but it destroys something from the artist's standpoint because he can't, in his mind, say, oh if change that a little bit, unless he becomes a mathematician but then he doesn't know when he's changing this parameter if he's simultaneously changing this one or this one. So you lose the direct one on one kind of pointing that I think an artist thinks. An artist doesn't think in terms of matrices.

Excerpts from an interview with David Dunn and Woody Vasulka.

Lee Harrison

March 2, 1992

Denver, Colorado