

## Animation: Master ~ Victor Navone

My inspiration for purchasing Animation: Master originated out of my pursuit of a good organic modeler for the Mac. I heard some people on an e-mail list mention the program, and I thought I'd give it a try. I bought the Animation: Master Handbook by Jeff Paries, which included a demo of version 5.0 on a CD (they're now up to 7.0) and started going through the tutorials. It was then that I began to see the enormous potential of this product, which spans well beyond its excellent organic modeler.



For me, learning Animation: Master, or “Hash” (after it’s creator, Martin Hash) was a painless process. Having approached and/or mastered other 3D aps such as Electric Image, FormZ, Cinema 4DXL and Maya, I consider the learning curve for Hash to be relatively shallow. The interface is deceptively clean and simple, which is a welcome sight for anyone who has stared at the controls of 3D Studio Max and felt the sudden twinge of vertigo. If you’re still intimidated, there is a wealth of Hash tutorials on the web, as well as a friendly e-mail list, a few good books and great tech support. All of these contribute greatly to the usability of the product.

You will not find a lot of complex “power tools” in Hash, such as deformation lattices or surface blends, but that doesn’t mean you can’t create just about anything you can imagine. Working in Hash is more like working with clay; the raw material is all there, you just have to go in and push things around with your hands. This gives the software a very visceral and utilitarian feel and also makes it incredibly versatile.

You will find excellent character animation tools, including robust in-

verse kinematics and “layerable” constraints, which rival many higher-end character animation aps. Hash’s constraints are very versatile and intuitive and you can easily simulate many “expressions” without having to learn how to script or write code, as I will discuss later.

I had been interested in trying my hand at character animation for sometime before I picked up Hash. I had attempted a few things in other programs, but was never able to overcome the hurdles of the software. Hash, on the other hand, let me get animating right away. It’s very easy to set up characters for animation in Hash, though finding the ideal constraint setup for you will take plenty of time and experimentation.

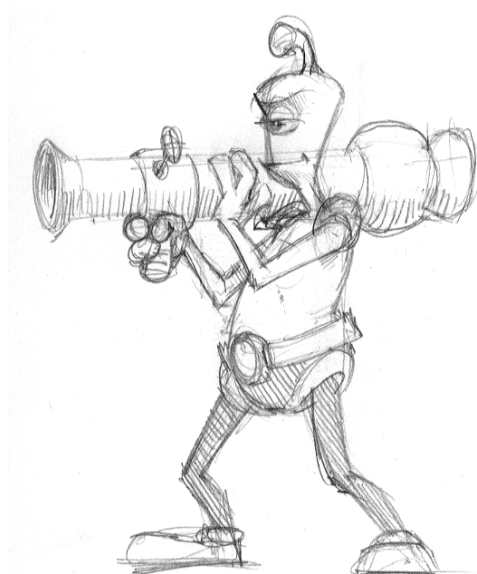


Figure 1

The alien character (yet to be named), whom I will be using in my examples, is the second 3D character that I created in Hash (or at all, for that matter). The design intent was to create something simple and easy to articulate, yet still emotive. I sketched a few ideas out on paper and soon came up with an aesthetic I liked (Figure 1). Then I jumped right into Hash and started modeling the character.

## Modeling

Modeling in Hash is achieved with “Hashsplines” (what else?), which are unique to the software. They behave similarly to beziers, in that each Control Point on a spline has a bias handle that can be rotated or lengthened to influence the shape of a curve. Splines are linked together at CPs to form “patches”, which are surfaces that the application can render. There are various rules that determine whether or not a patch is “legal” and whether or not it will crease or be smooth. Mastering these rules will probably be the most daunting task for new users, but it is far from insurmountable.

Most of the alien was created with lathes and extrusions, which I then tweaked to taste by pushing and pulling individual CPs around. I built the arms and legs separately and, because his design didn’t demand that he be constructed of a seamless, unibody mesh, I simply shoved the limbs into their respective spots on the torso and hid the seams behind costume elements (Figure 2). There are some great tools for connecting higher resolution areas of a model to lower resolution areas, such as “hooks” and 5-point patches. I made use of these mostly in the head and hands (Figures 3 and 4).

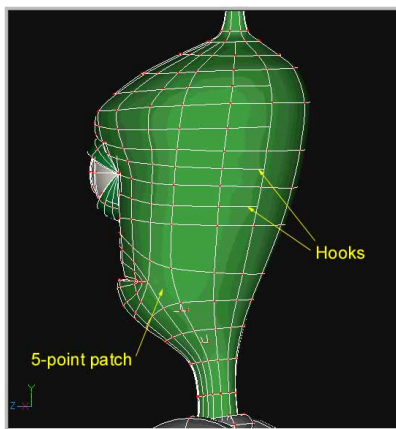


Figure 3

## Surfacing

Again, I was mostly interested in getting into animation quickly, so I didn’t spend a lot of time on surfacing my alien. I selected and saved different groupings of CPs and assigned them some basic colors and material attributes (Figure 5). The only custom image map I created initially was the happy-face logo for his chest. Since the ex-

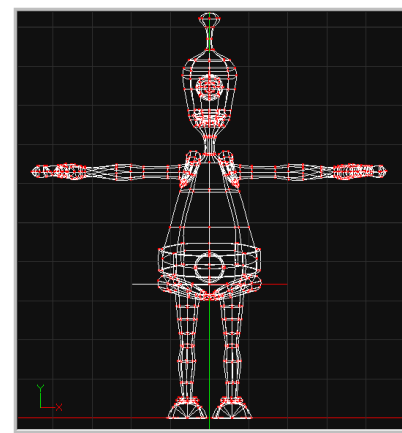


Figure 2

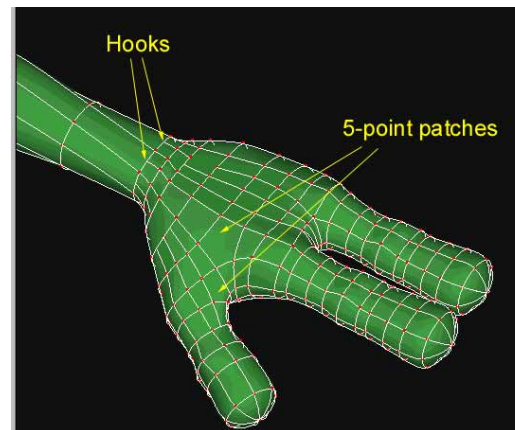


Figure 4

ample animation, I have upgraded his surfacing considerably (Figure 6). Hash has all the basic texture mapping channels that you would expect, such as diffuse, specular, reflectivity, etc. with the option for raytraced properties,

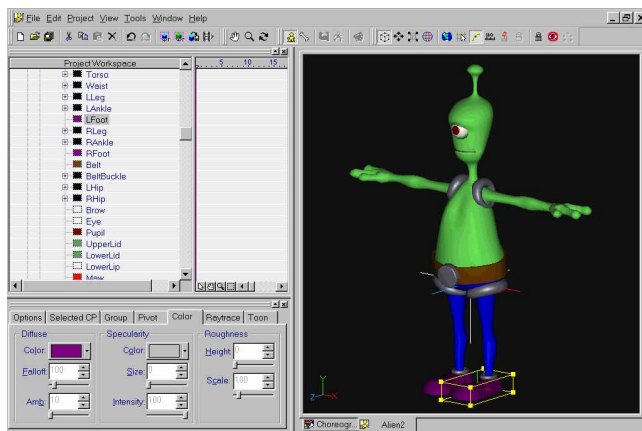


Figure 5



Figure 6

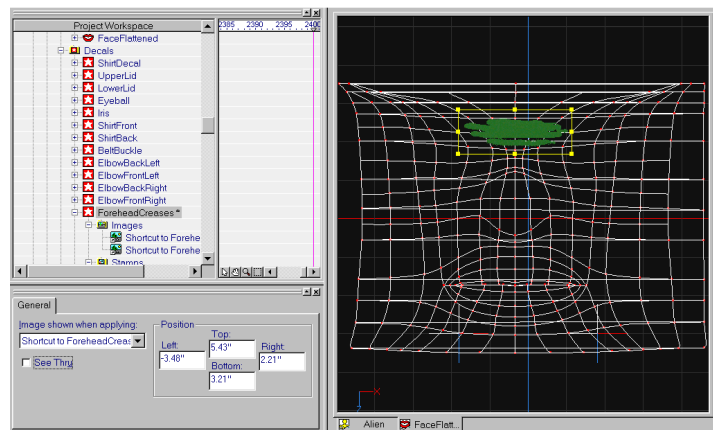


Figure 7

such as refraction and reflection. There are basic shaders available, which can be layered and mixed, along with material effectors. Image maps can be applied as decals or projection maps to selected groups of geometry. While

Hash does not support any kind of UV mapping, it does allow you to non-destructively flatten or “unwrap” a model, so that you can apply a flat map and have it conform predictably to a complex surface (Figure 7).

## Setup

After establishing the basic look of the character, I went about setting him up for animation. This included adding bones to the character, assigning CPs to the bones and assigning constraints to bones. Note the amount of bones, as shown in Figure 8. Some of them

are used to control the character’s skin and skeletal positions, while others are used as part of the constraint “rig”, much in the same way that you would attach rods to a puppet. I have turned on all of the bones for this example but during animation, most of the bones are hidden, leaving only those that are necessary for controlling the character.

There are about a dozen different constraints that you can apply to any bone in a character, such as “Aim At”,

“Orient Like”, “Translate To” and constraints that control rotational limits. You can layer multiple constraints onto one bone to achieve different results and, you can even have two of the same constraint with different tar-

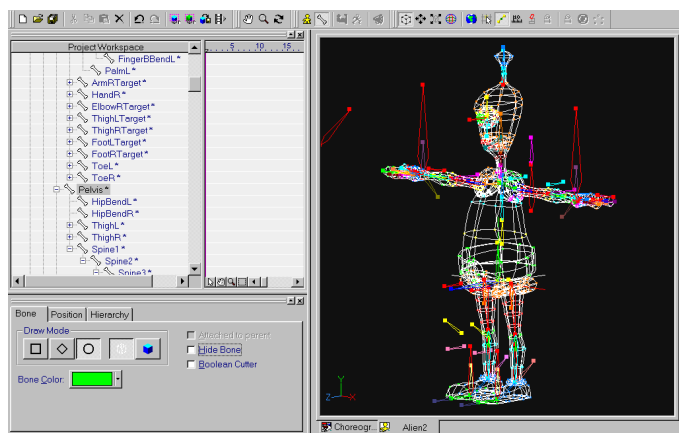


Figure 8

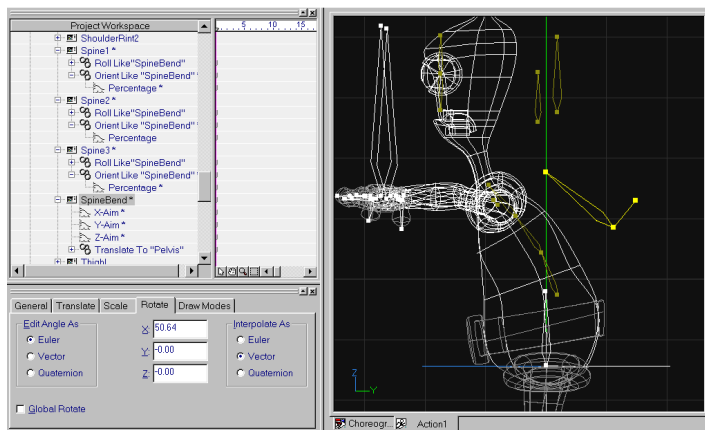


Figure 9

gets. For instance, if you put two “Translate To” constraints on Bone C, one targeted at Bone A and one at Bone B, then Bone C will always assume a position that is midway between Bones A and B. In Figure 9 the bone “Spine Bend” is used to drive three other spine bones. The spine bones are set to “Orient Like” the Spine Bend bone, each at about 30%, so cumulatively they approximate the angle of Spine Bend, while forming a controllable arc. This kind of constraint blending alleviates a lot of the need for scripting and expressions that you find in other packages. And, of course, all the constraints are animatable, so you can do things like switch from Inverse Kinematics to Forward Kinematics in mid-animation.

## Animation

The included animation is actually my second attempt with this character. The first attempt was a simple piece about the alien’s reaction to suddenly realizing that he is being watched by the viewer. I was relatively satisfied with the outcome, but more importantly, I learned a lot about working the kinks out of the IK setup, and figuring out my own personal workflow. I now had more of an idea of what kind of controls I wanted on the character and I was ready to try something a little more ambitious. This time the character would have a prop, and there would be more dynamic, anime-like motion, as well as some 70’s-style action camera animation.

I got many great ideas for constraint setups, both from the Animation Master e-mail list and from the book **Character Animation In Depth** by Doug Kelly. My new setup used less bones to control the alien's poses, while affording more control and consistency in his movements. Because I was doing this as a personal project and therefore had no deadlines, I would often continue tweaking the character's setup well into the animation stage. Sometimes this was the result of Hash adding new features that I couldn't wait to take advantage of! This resulted in a lot of back and forth and occasionally some redoing of animation, but luckily Hash is very forgiving of this in that it has a non-linear work environment. I can change textures, modify geometry and adjust constraints in mid-stream and, as long as I don't change the names of bones, the animation will still work.

I created an "Action" in which I would animate the character, and dropped in the gun (a separate model) as an "Action Object", with which he could interact (Figure 10). I assigned a "Kinematic" constraint to his right hand so that it would attach itself to the bone in the handle of the gun. The left hand also had a Kinematic constraint, but I animated it to be dormant until the left hand needed to grasp the gun, near the middle of the animation. The gun itself was completely outside of the alien's skeletal hierarchy and animating it on its own path allowed me to give it a feeling of weight and inertia.

I created a "Choreography" in which I would introduce the lights, the camera, a simple backdrop and of course, the alien (Figure 11). I began animating the character by first hiding his arms and legs and focusing on the motion

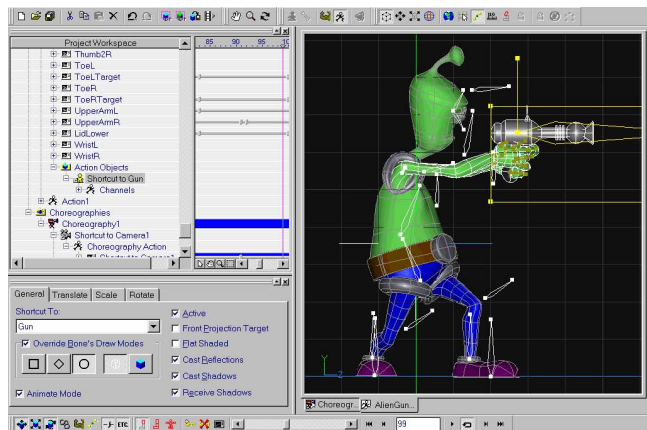


Figure 10

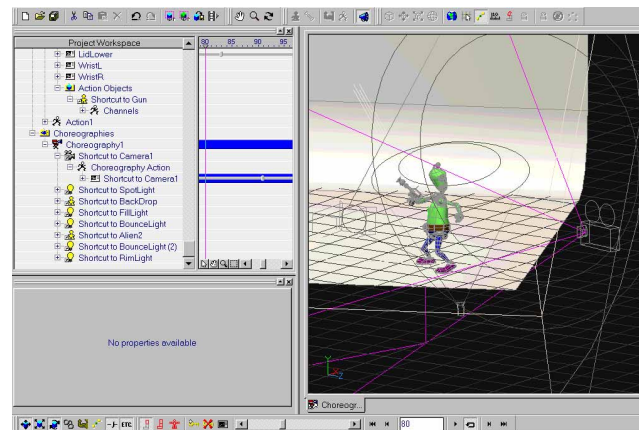


Figure 11



of the torso. This allowed me to block out the overall timing of the animation without having to worry about things like foot placement and overlapping action. Once I had the timing worked out I then animated the limbs and the secondary motions, such as the antenna. The most technically challenging aspect of this animation was the end-over-end flip. I initially ran into problems with gimbal lock, in that when I inverted the character, the bones lost track of which way was up! I was able to solve this by going back to his setup and parenting all his bones to a master bone, which was oriented sideways. I could then animate his flip without any problems by rolling this master bone on the Z axis as I translated it through the air to his final destination. The gun, on the other hand, still remained outside of the hierarchy and I ended up having to tweak it frame by frame (Figure 12). I animated his facial expressions using “Poses”, which are stored sets of animation data, similar to morph targets. Poses can contain anything from bone positions to CP or “muscle” motion, to entire constraint setups.

The animation itself took me about a month and a half, though I was only working on it on my free time,

say about 8 hours a week. Overall, I am pleased with the outcome, though in retrospect, I would have animated the left arm with Forward Kinematics in order to give it more overlap. I have since refined the alien’s appearance as well as his setup, which now includes IK/FK switches in the arms and I am currently working on a new animation, this time involving lip synch.

In conclusion, Hash is one of my favorite 3D software packages, if not my one of my favorite software packages period. It is an unbeatable value for the money and a good place to start for those interested in learning character animation, or 3D in general.

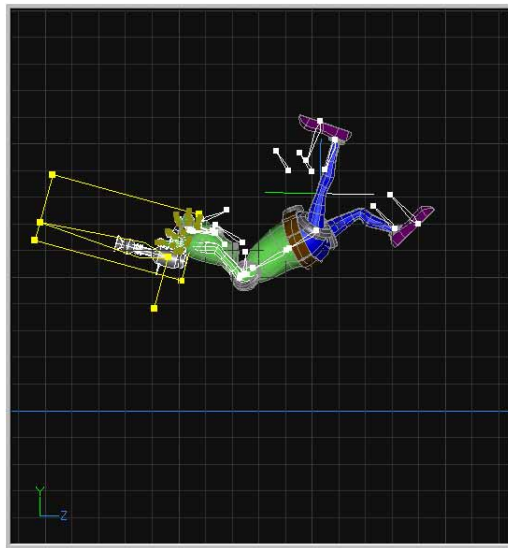


Figure 12

Of course, no software is perfect and Hash has it’s caveats.

The raytrace renderer can be tediously slow, though it’s quality has improved dramatically in version 7.0. The motion-blur leaves something to be desired, as you can see in the rendered movie; it currently only blurs within the silhouette of the object. Keep in mind also that I’m used to Electric Image’s renderer, so I have very high expectations.

The software can be very finicky if your computer isn't optimized just right and is sometimes downright crashy. Luckily, their tech support is very responsive.

Hash doesn't play well with others. The only things you can import or export to other 3D aps (other than images) are DXF files. The software environment does not support polygons, nor any other form of geometry besides Hash splines. You can import DXF geometry, which Hash will convert to splines, but this should only be done on lower res models that are not triangulated. You can also export Hash geometry as an Autocad DXF file (usually with good results) and the patch-to-polygon conversion is user-adjustable. You may run into trouble if you want to export separate groups, though. If you're really desperate to export your animation to another program, you can output any animation as a series of DXF files (yes, one per frame) and use them for morph targets or children cycles. I've actually done this and have had mixed results. Of course, if you're content to do everything within Hash, none of this is an issue.

I have yet to be able to preview an animation in realtime at a reasonable frame rate, even in wire frame mode. In fairness, I am using a 200mhz Pentium with no hardware acceleration at home and a Power Macintosh G3 333mhz at work. As of version 7.1 (beta) both the PC and Mac versions support OpenGL in software and hard-

ware, which does provide nice shaded previews with user-definable tessellation.

I don't recommend Hash for architectural modeling. That's not to say that you can't do it and indeed, you will find nice examples on the web from those who have done it, however, as it is not a polygon modeler, Hash lacks certain basic tools which make architectural modeling much simpler, such as edge-beveling and true booleans. You can replicate these effects, but it will take a lot more planning. Then again, I doubt anyone buys Hash with the intention of building bridges.

Conversely, I must mention that Hash is always adding new features and new beta versions are frequently posted for public download, simultaneously for Mac and PC. This is another thing I love: it's cross platform, so I can work on a file at home on my PC, then take it into the office and continue to work on it on my Mac. Keep in mind that the Mac version is a port, so it's not quite as speedy as it is on the PC, but otherwise they're virtually identical.

Also: Hash is cheap. Only \$199. Yet, it has features and functionality that can compare with some of the most high-end animation packages out there. It's not even uncommon to hear someone from ILM say he uses Softimage for character animation at work, but prefers to use Hash at home. While the small price tag may keep



it from being taken seriously by industry pros and large animation studios, it only further endears it to the artists and hobbyists who use it to produce CG animation on a budget. Hash can keep the price low because they don't advertise; instead they rely on word of mouth and trade show demos for sales.

And no, they're not paying me to say this.

## Additional Links:

Hash Web Page: <http://www.hash.com>

## Tutorials

### Sherwood's Forest

<http://www.hash.com/users/jsherwood/>

### The Splindicate

<http://3dark.com/hashfaqs/index.html>

### Hash FAQs

<http://www.hash.com/hashfaqs/>

### Mailing List

<http://3dark.com/hashfaqs/resources/animaster.html>

## Books

**Animation: Master Handbook** (Second Edition) by Jeff Paries

<http://www.charlesriver.com/frSearch.html>

**Character Animation In Depth** by Doug Kelly

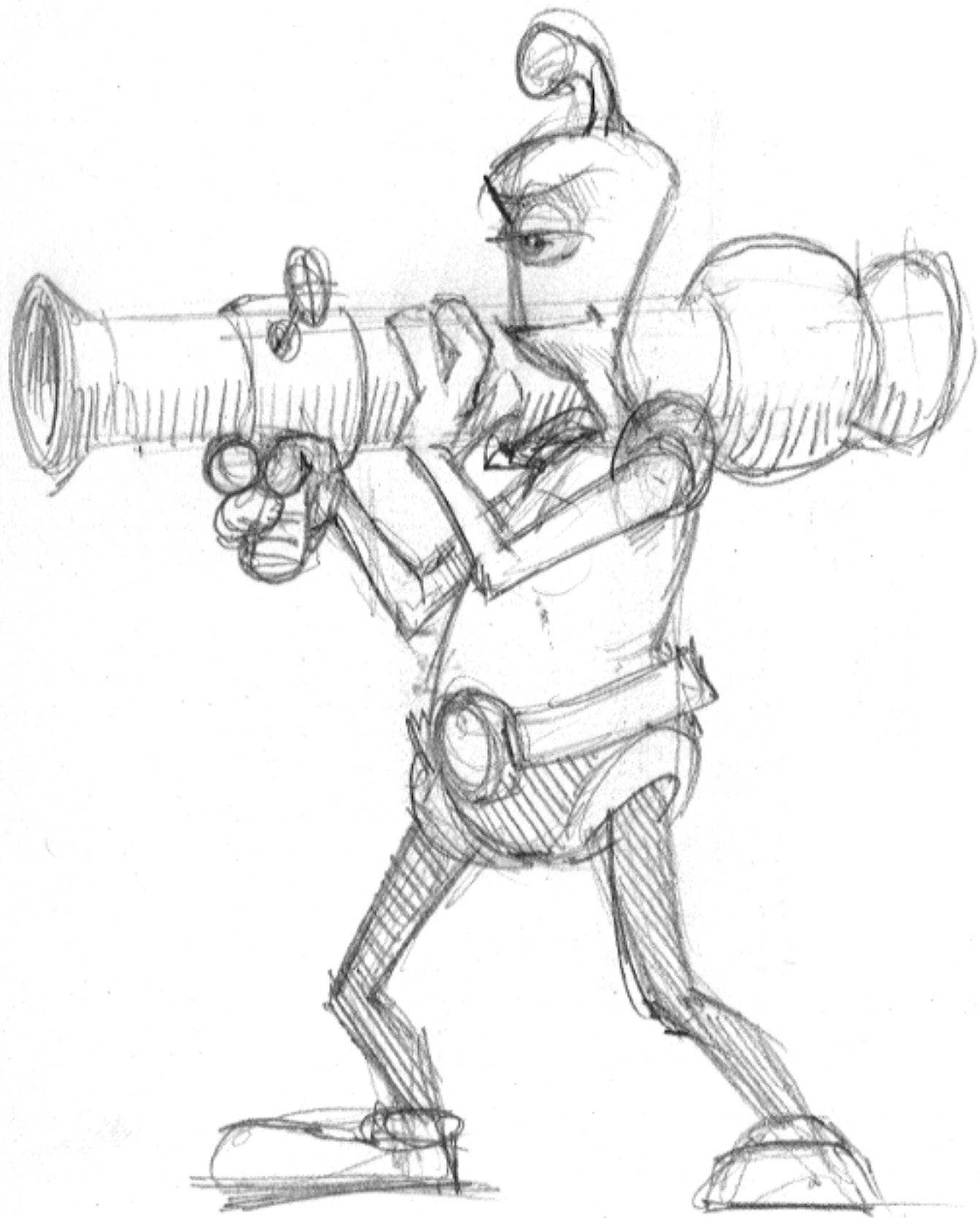
<http://www.amazon.com/exec/obidos/ASIN/1566047714/qid=926528639/sr=1-1/002-7620676-1635664>

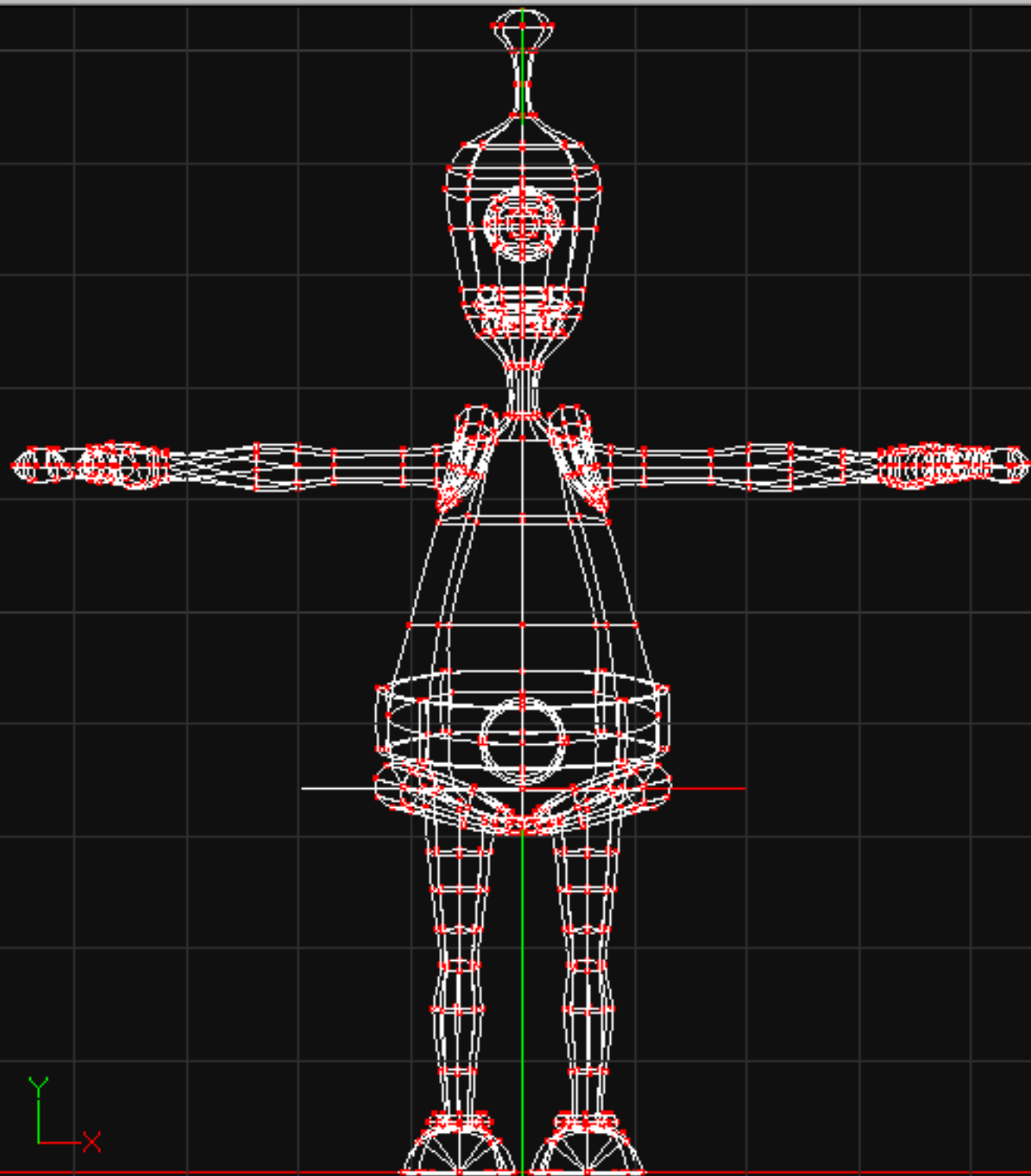
## About the author

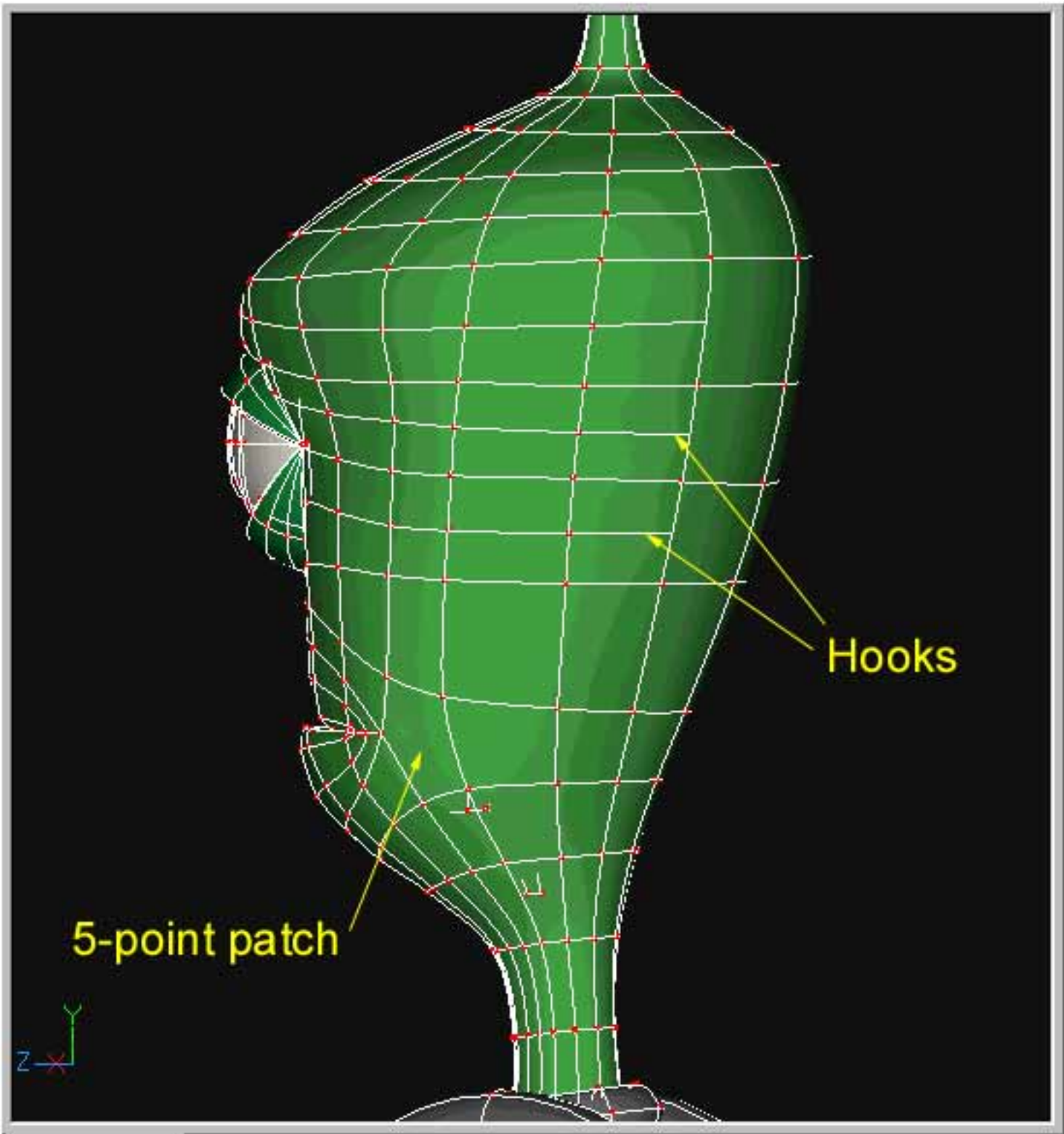
Victor Navone is currently employed as a designer and 3D artist at Presto Studios, a computer game developer in San Diego, California. He has worked on such titles as "Journeyman Project 3: Legacy of Time" and "Gundam 0079". He has a background in traditional art media, and has a degree in Fine Art from the University of California, Irvine. You can see more of his work at <http://www.presto.com/staff/victor.html> and <http://www.serious3d.com/gallery/victor.navone/index.html>.

You can contact him at: [victor@presto.com](mailto:victor@presto.com)



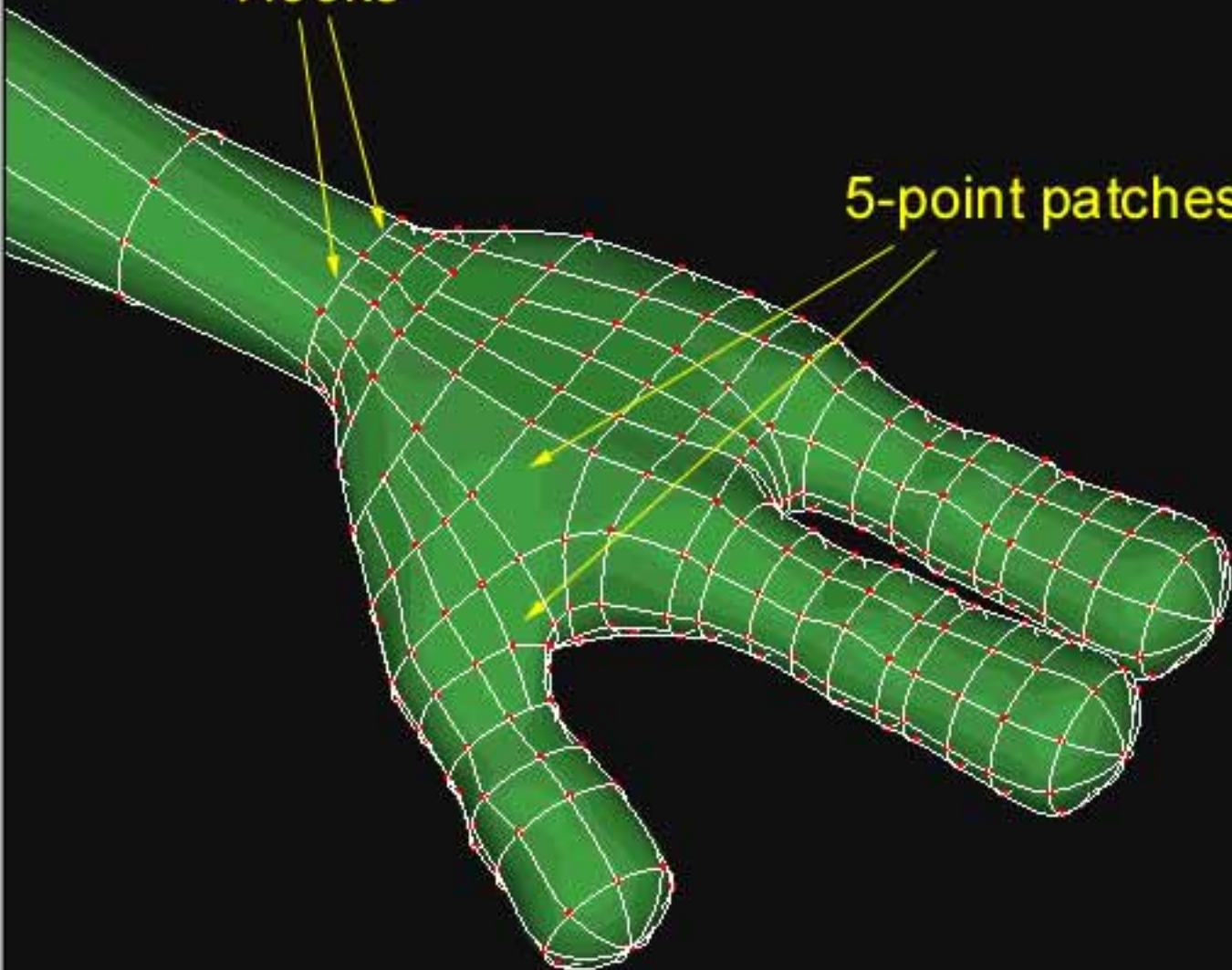




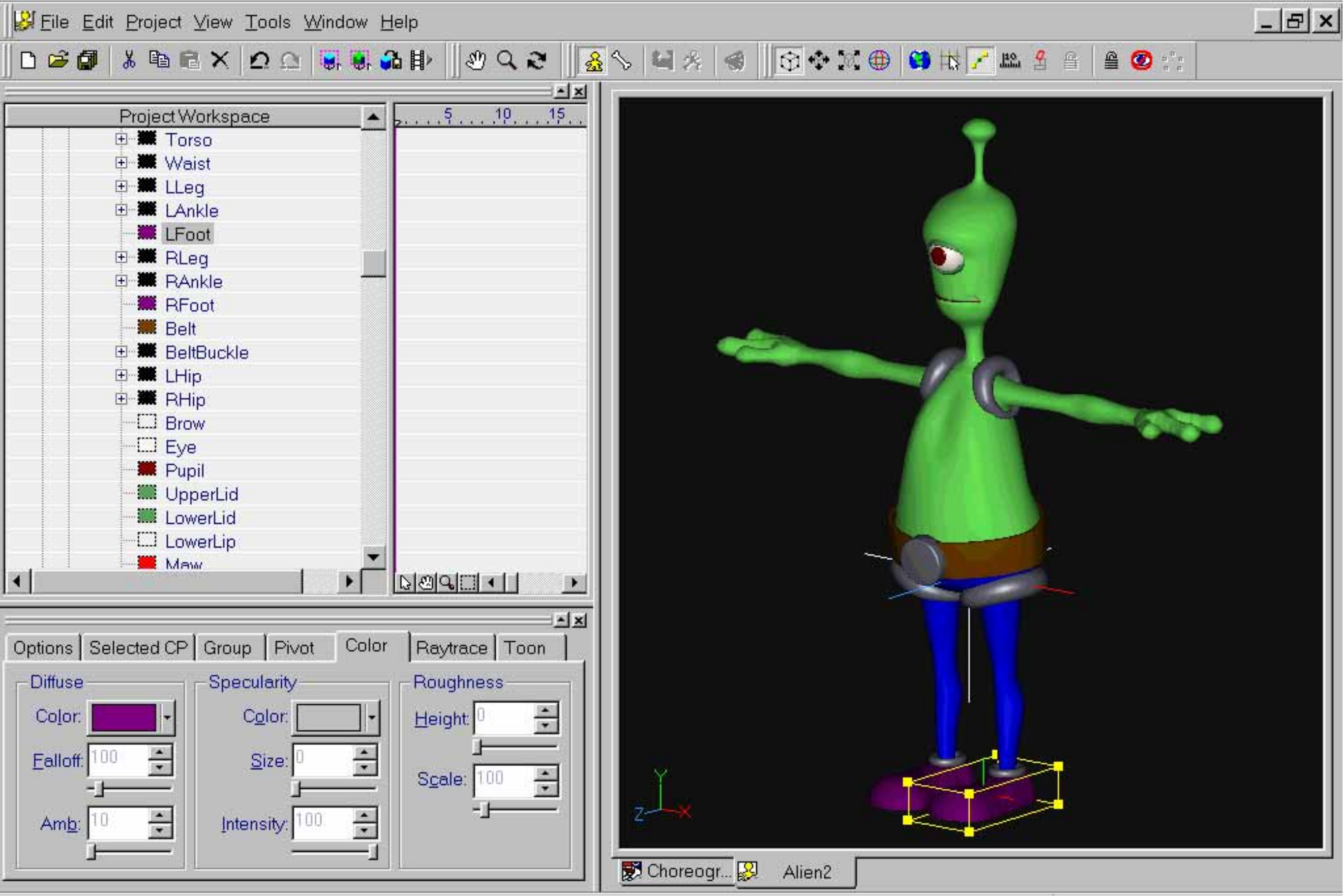


Hooks

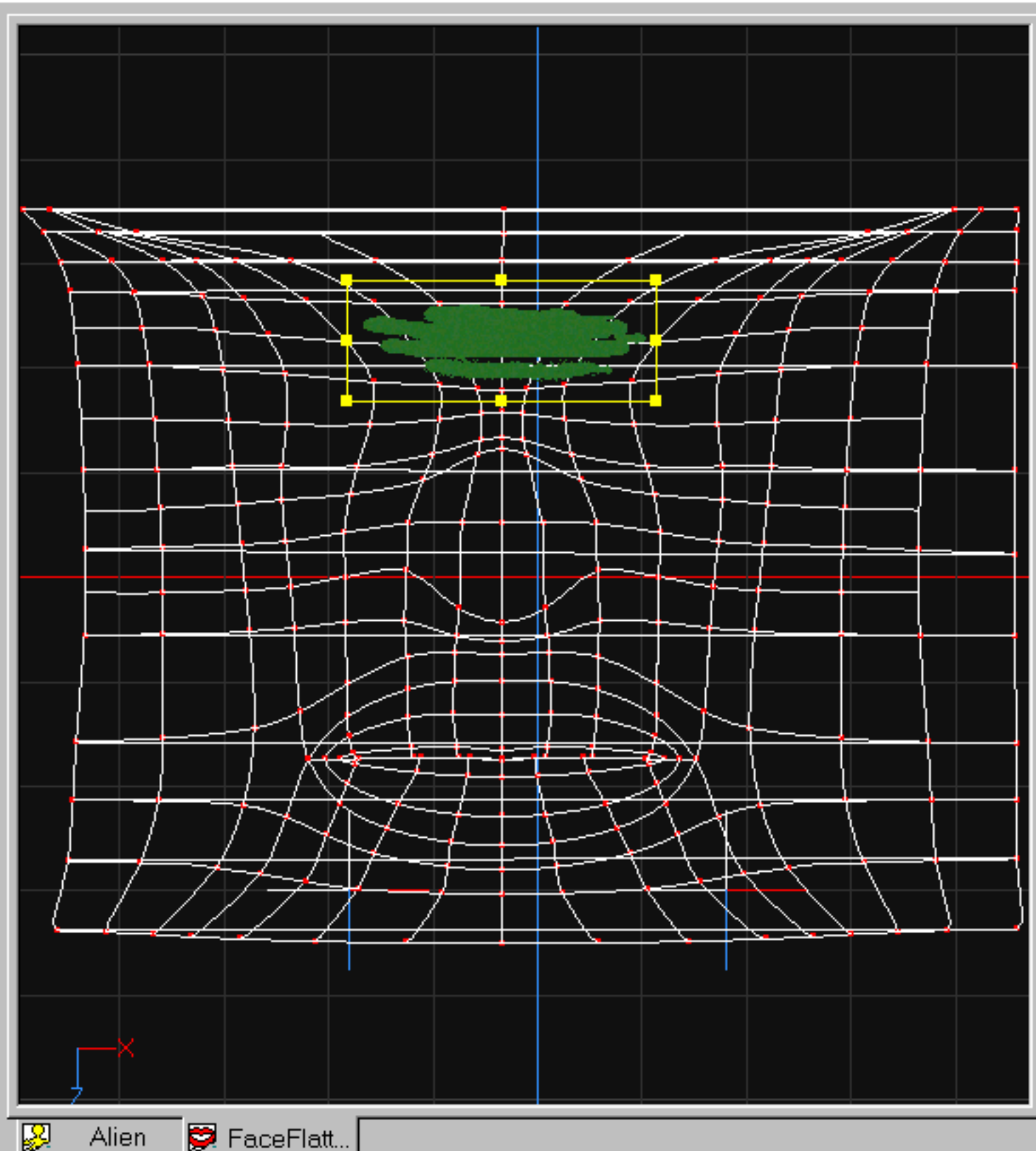
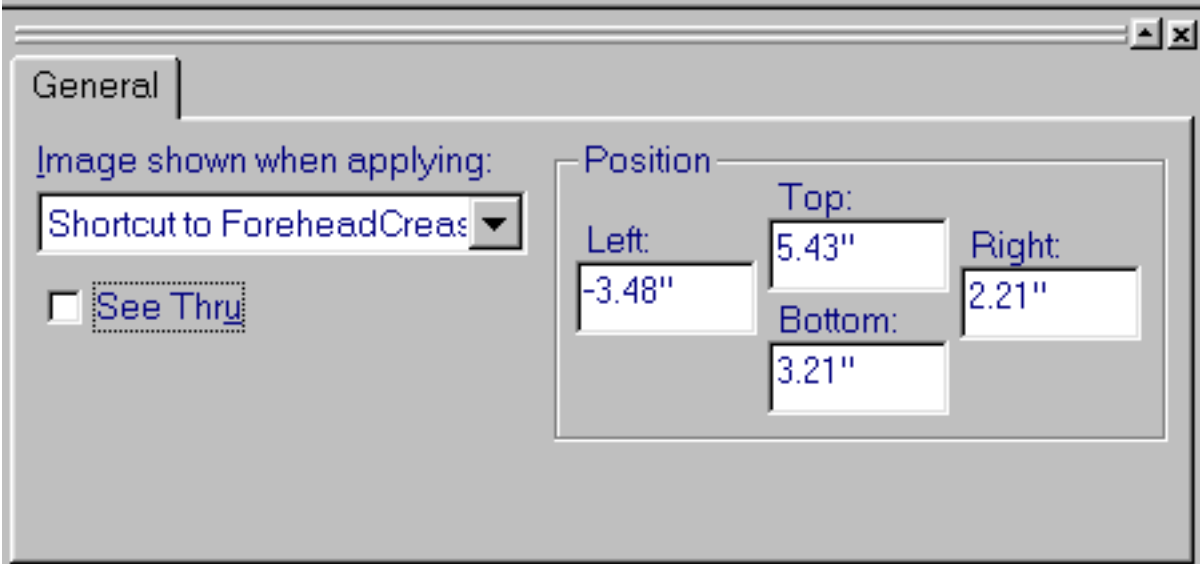
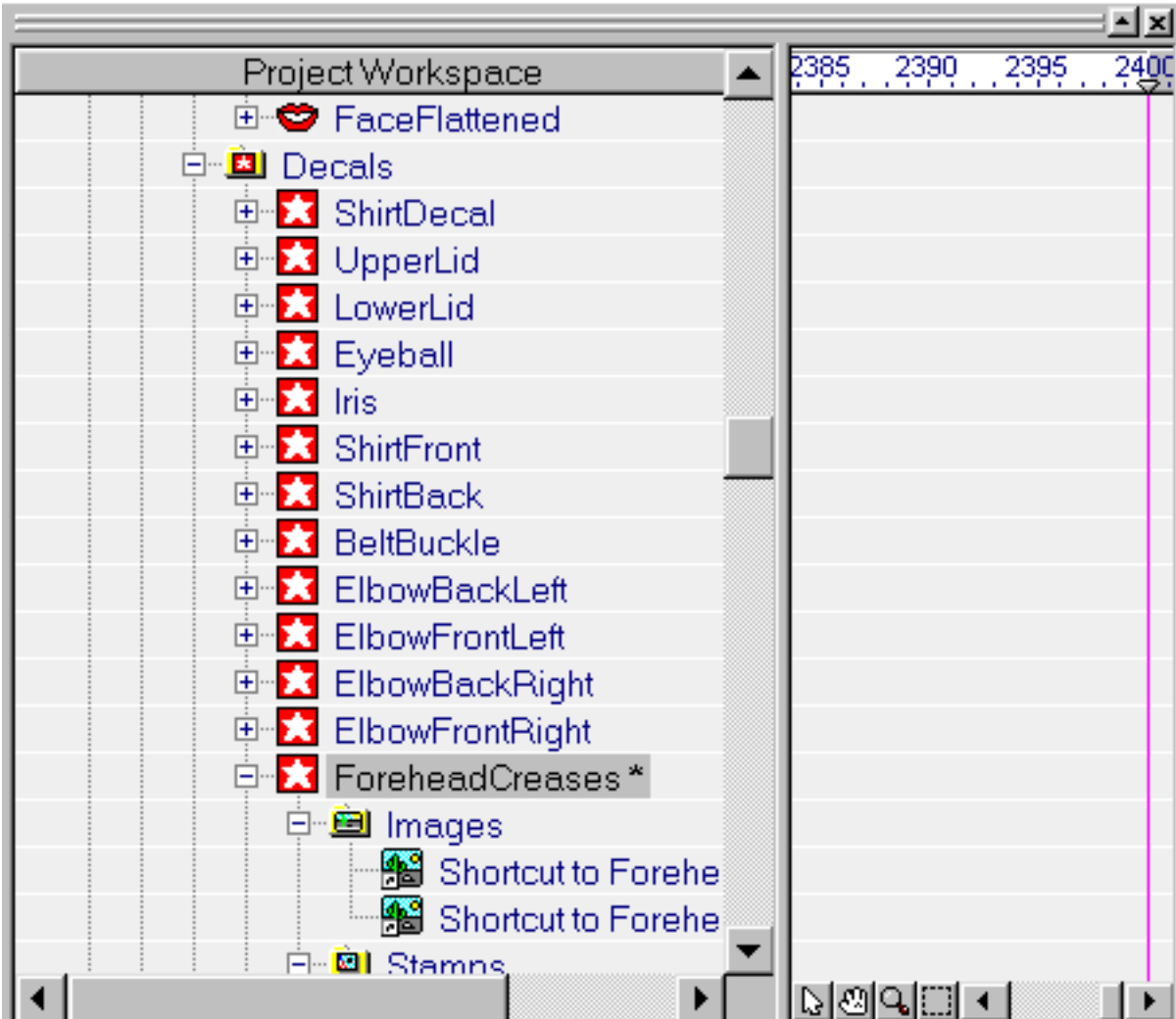
5-point patches













Project Workspace

- FingerBBendL \*
- PalmL \*
- ArmRTarget \*
- HandR \*
- ElbowRTarget \*
- ThighLTarget \*
- ThighRTarget \*
- FootLTarget \*
- FootRTarget \*
- ToeL \*
- ToeR \*
- Pelvis \*
- HipBendL \*
- HipBendR \*
- ThighL \*
- ThighR \*
- Spine1 \*
- Spine2 \*
- Spine3 \*

2... 5 ... 10 ... 15 ...

Bone | Position | Hierarchy

Draw Mode

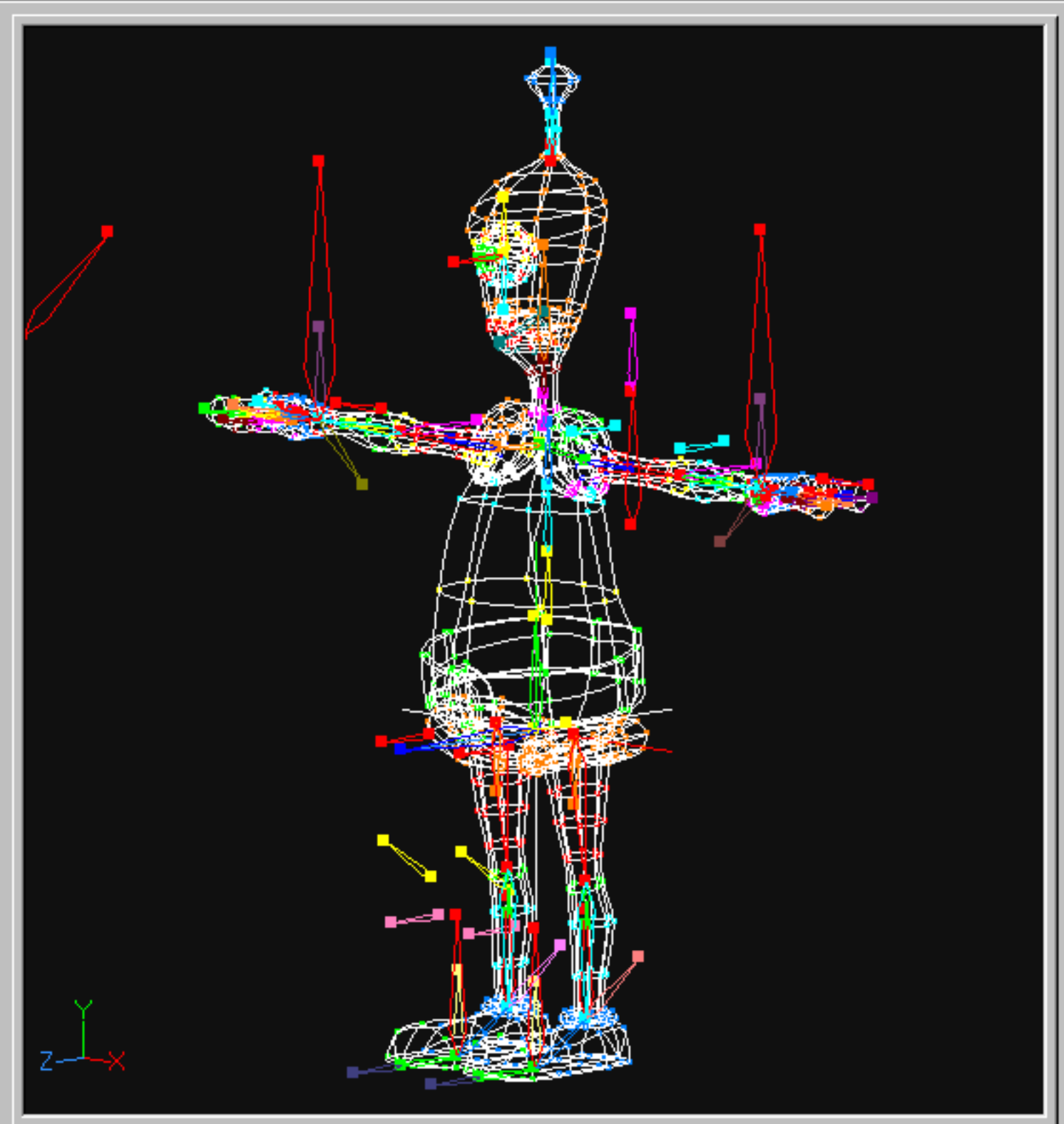
☐ ☐ ☐ ☐ ☒

Bone Color:   

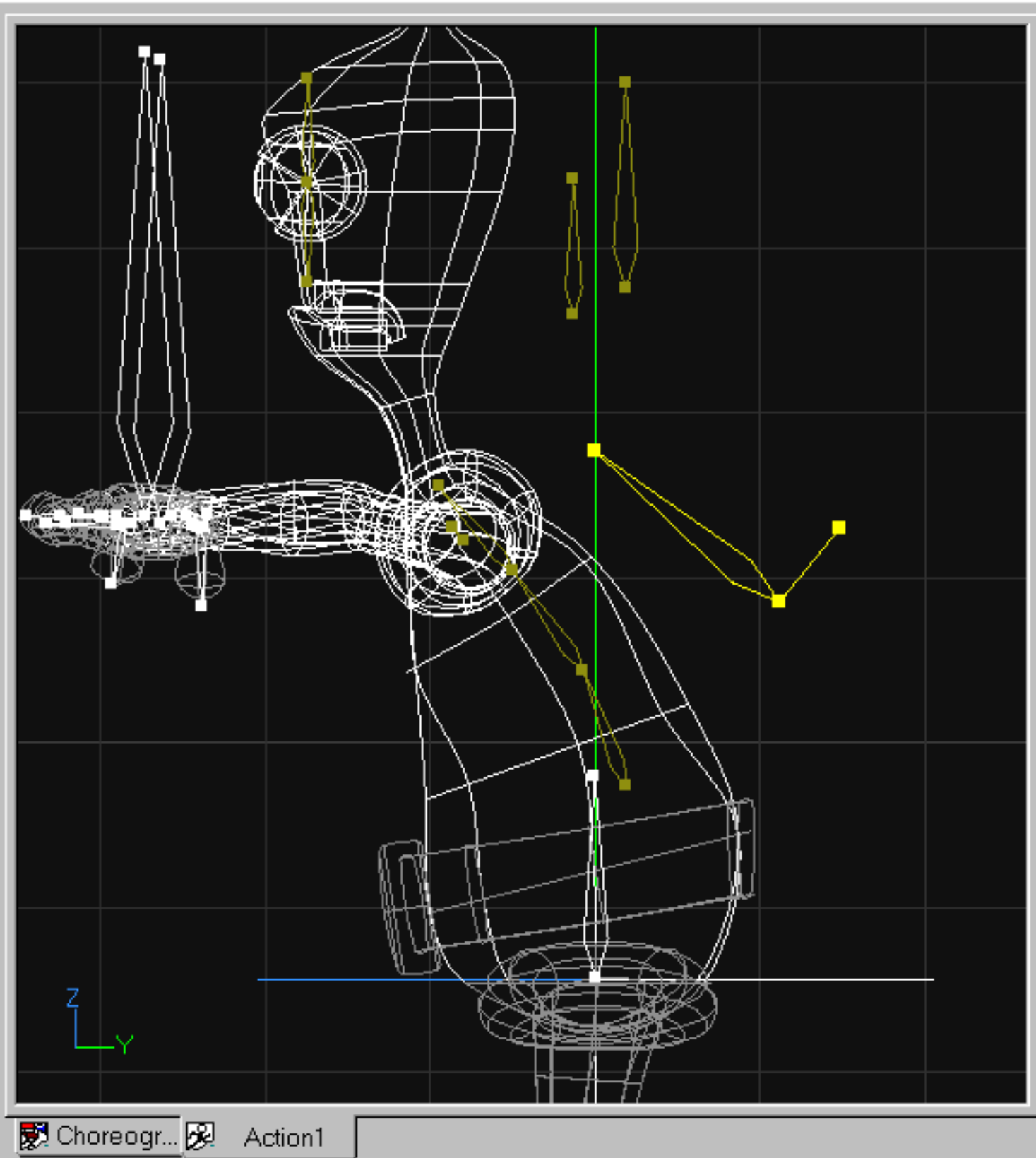
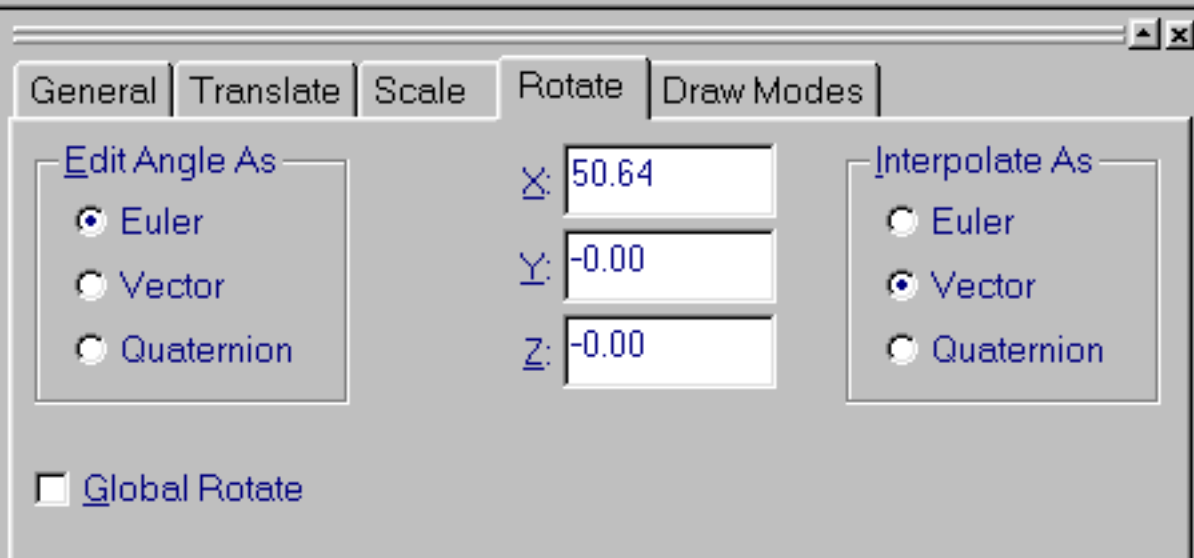
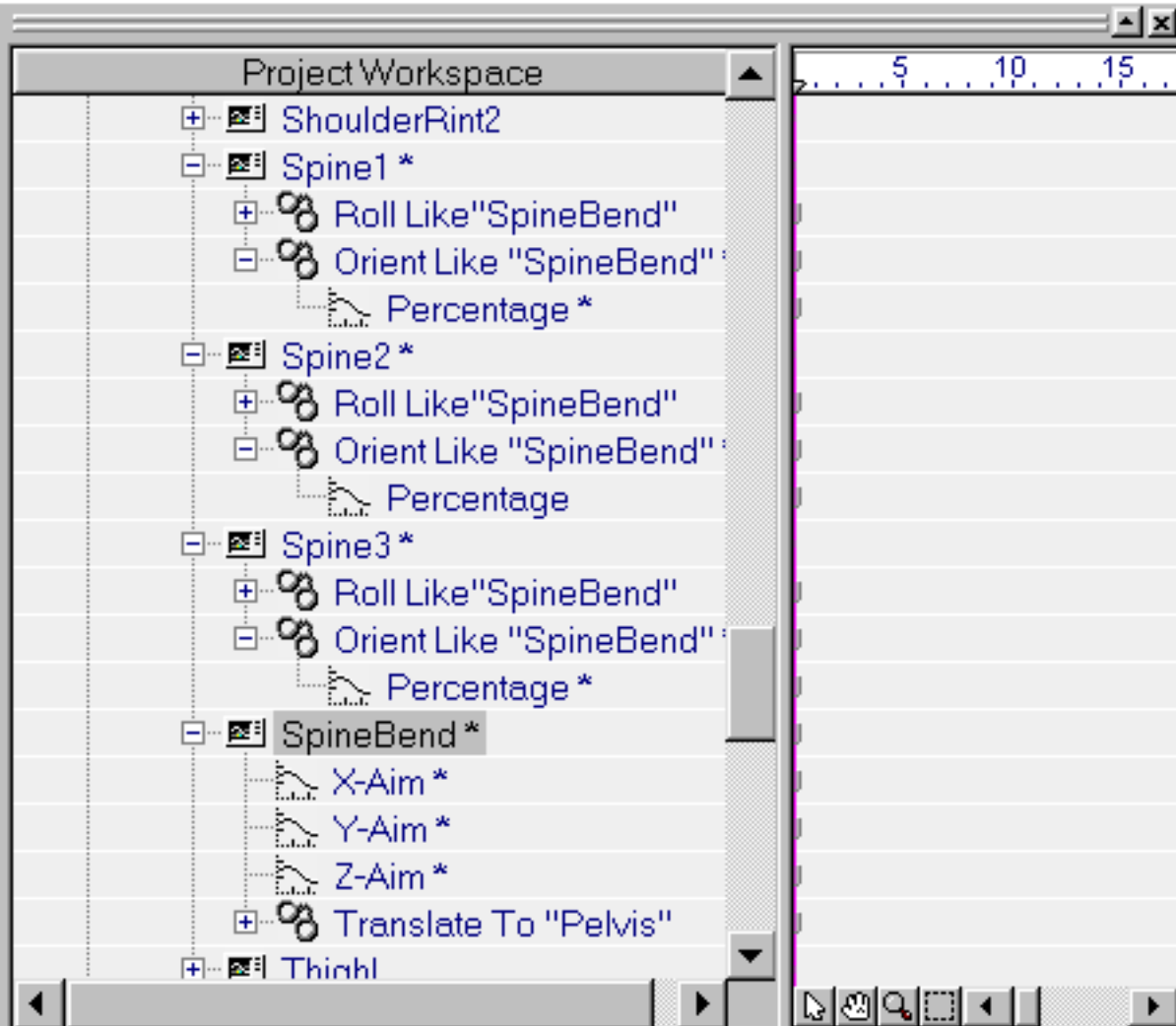
☐ Attached to parent

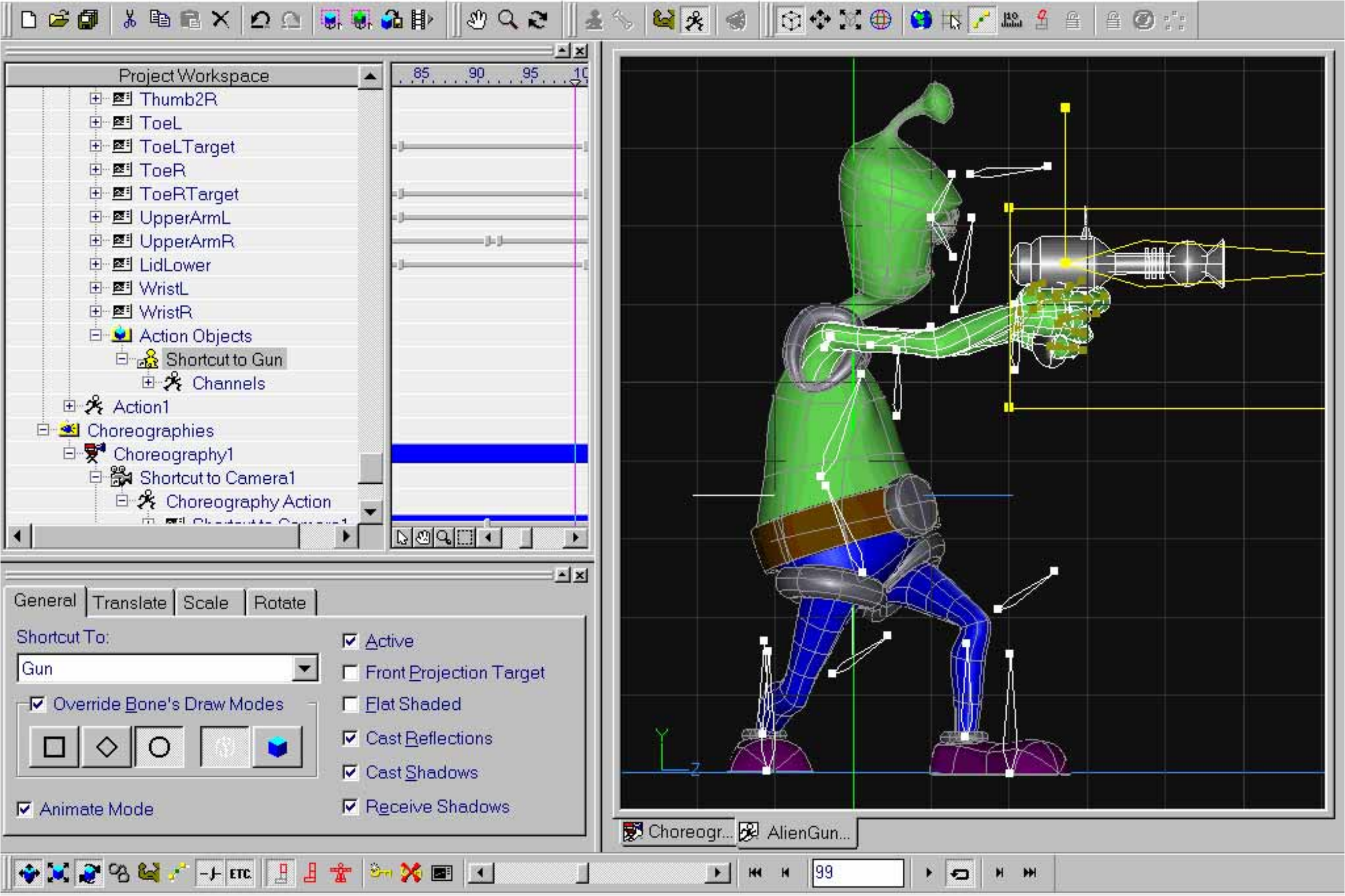
☐ Hide Bone

☐ Boolean Cutter











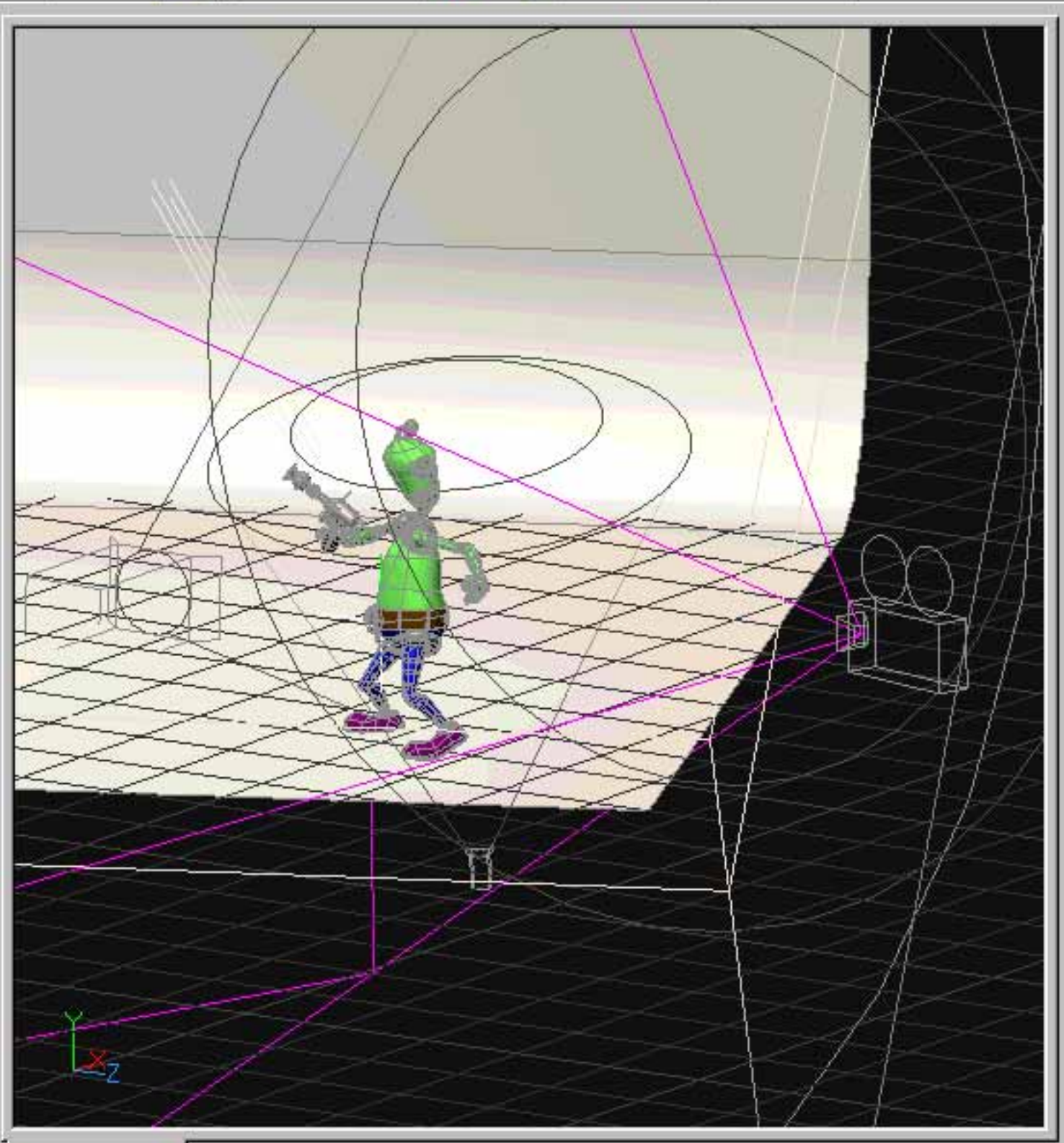


Project Workspace

- + LidLower
- + WristL
- + WristR
- Action Objects
  - Shortcut to Gun
    - + Channels
- + Action1
- Choreographies
  - Choreography1
    - Shortcut to Camera1
      - Choreography Action
        - + Shortcut to Camera1
    - + Shortcut to SpotLight
    - + Shortcut to BackDrop
    - + Shortcut to FillLight
    - + Shortcut to BounceLight
    - + Shortcut to Alien2
    - + Shortcut to BounceLight (2)
    - + Shortcut to RimLight

Timeline: 80 85 90 95

No properties available



Choreogr...



