

First Person Shooter Game

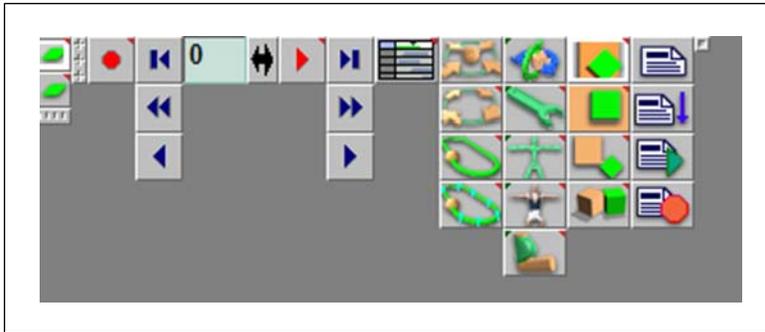
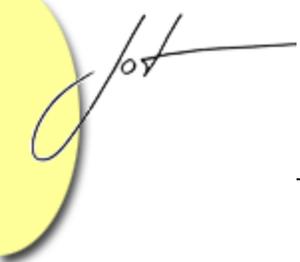
Rex Cason II
Erik Larson II
Jonathan Robertson
Jonathan Frisch
George Trice III
Dr. Lakshmi Prayaga

Abstract

3D game development is an exciting activity for many students. But getting a handle on 3D game development for novices may be a daunting task. We take this opportunity to present a quick introduction to 3D game development through a few tutorials. For the next few columns a set of tutorials for a 3D first person shooter game developed by graduate and undergraduate students under the guidance of a faculty member from the University of West Florida will be presented. These tutorials were developed with **3D game Studio** by *Conitec*. To follow along, download the software from www.conitec.com. These tutorials include all elements of game development such as modeling and animation, lighting, collision detection, sound and scripting. Each tutorial will focus on one or more of these aspects. This week we start out with creating a room and adding some objects to the room. The instructions for this are presented below.

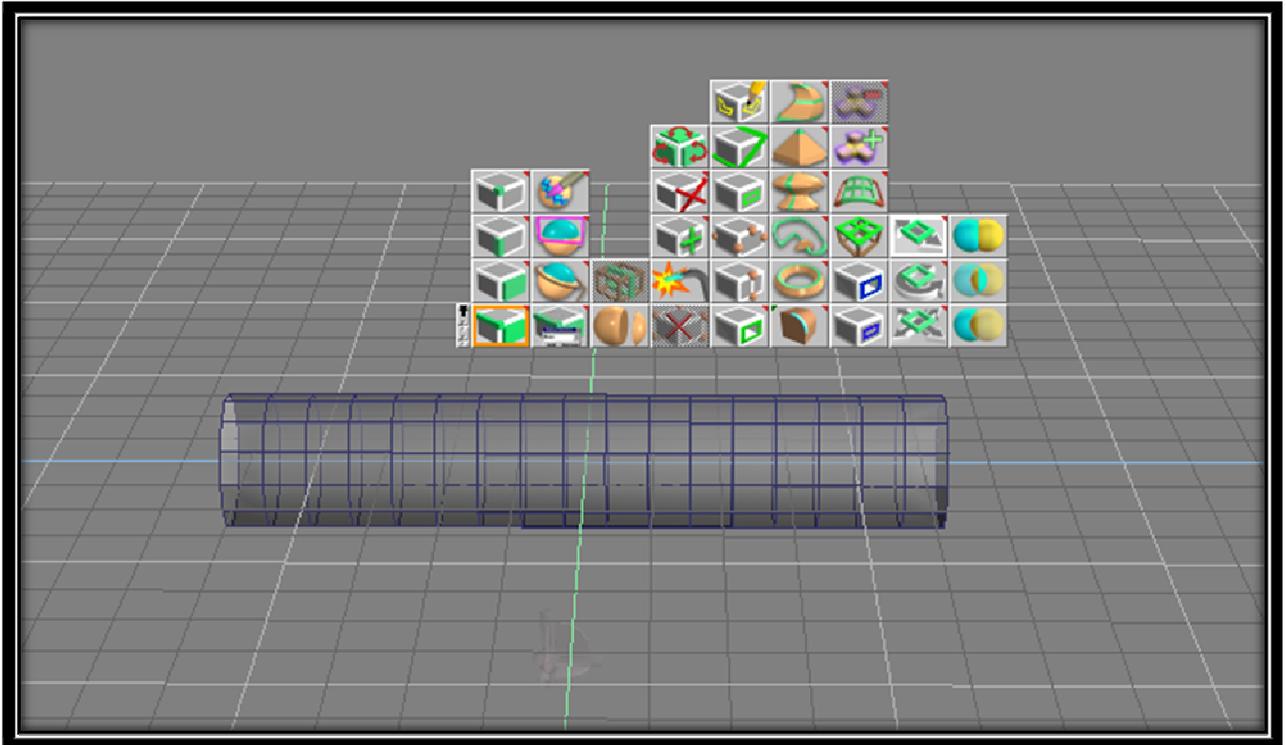
ANIMATING A MODEL

gameSpace is a 3D-modeling software package designed for game modelers. It has a good measure of tools for designing characters and other models. It is also fairly inexpensive for the full version and the "light" version is free



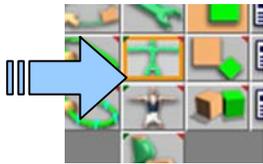
With gameSpace open, let's look at the tools we will need to use. In the upper left corner is the "Bones and Animations" toolbar

I will use something simple to demonstrate how these tools are used to animate an object. A tube should do

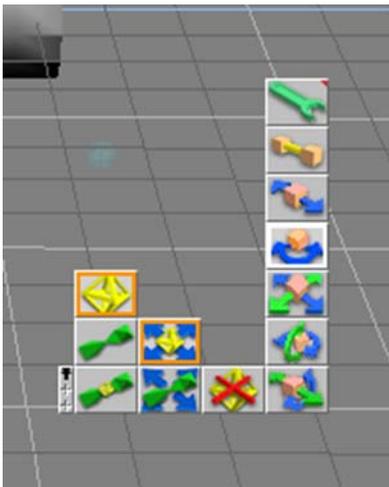




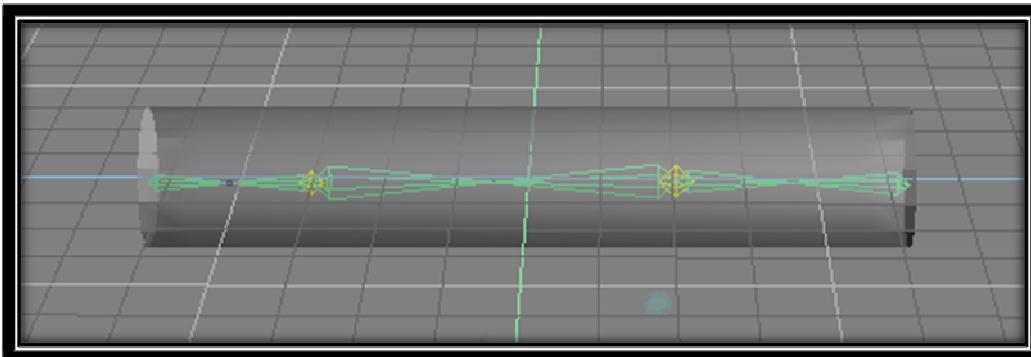
Now click on the **“Build Skeleton Inside Skin”** tool



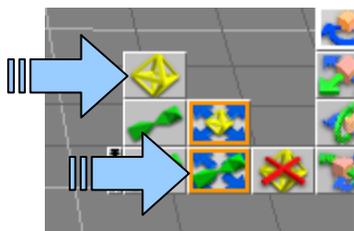
This should come on screen

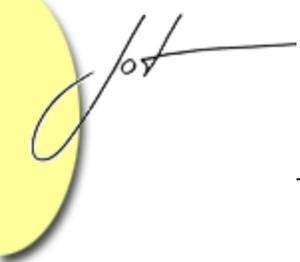


Click on a couple places in the tube to add some bones

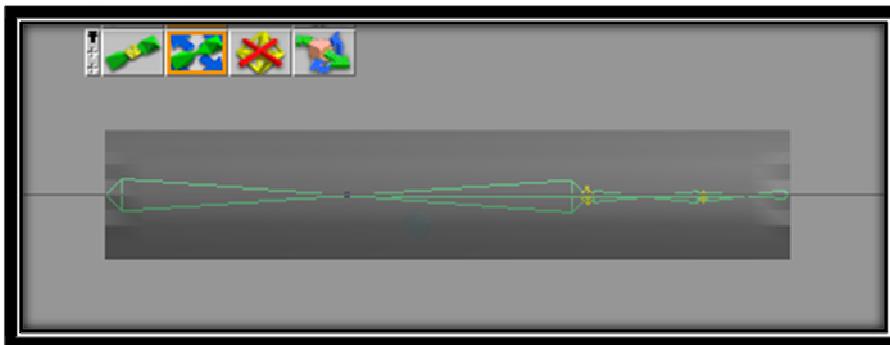


Then click the **“Add Joints”** to deselect it and click the **“Move Branch”** tool

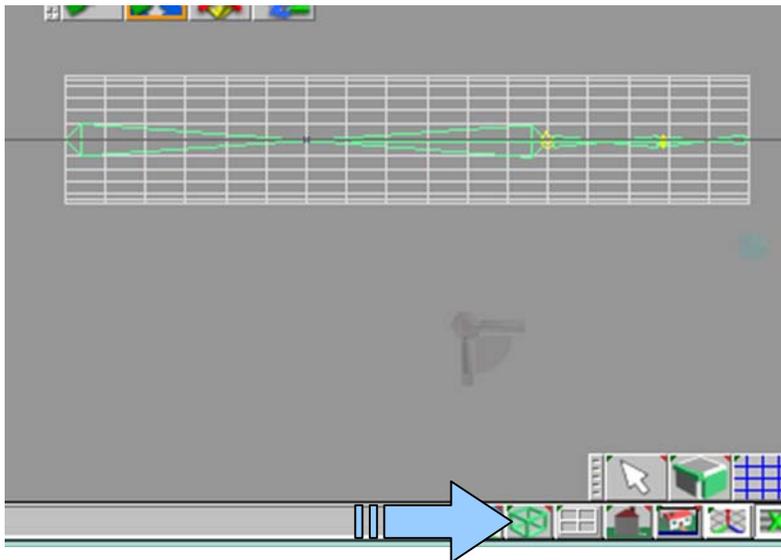




Now adjust the joints and branches if/as desired

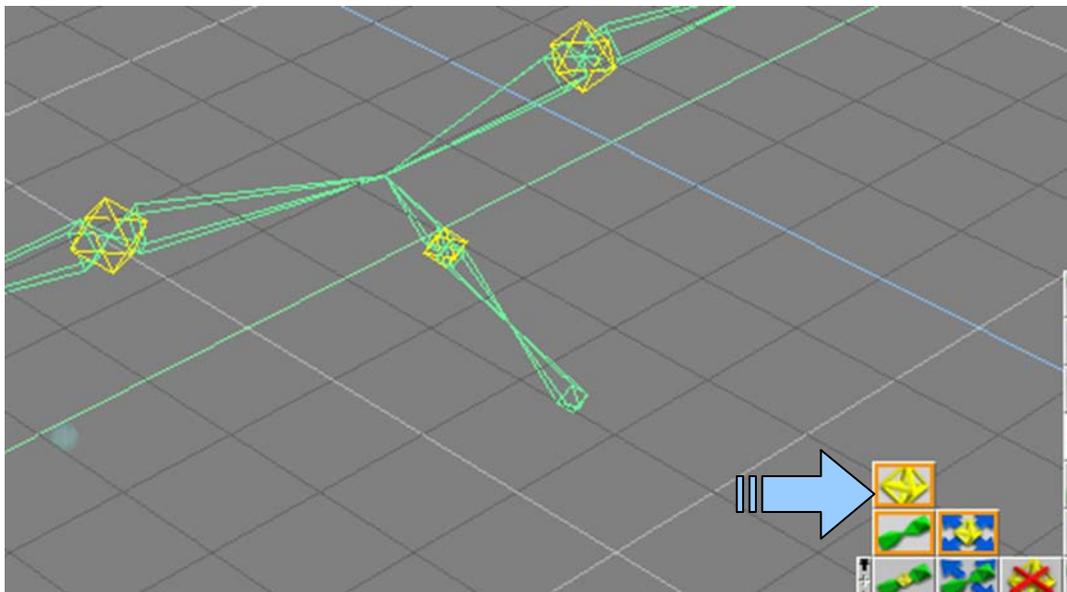


It can be helpful to change the view to wire frame while editing the bones

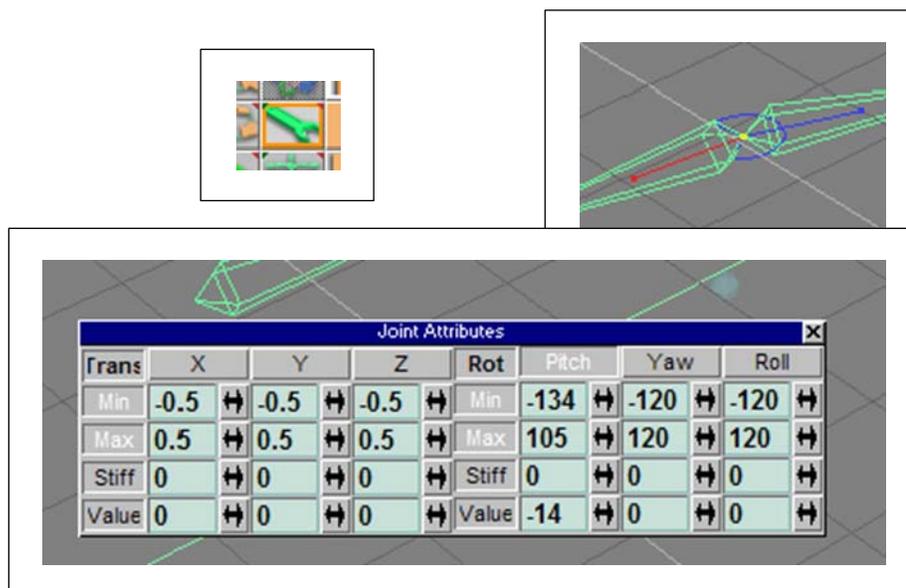




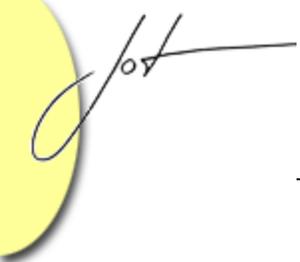
By clicking on the “**Add Branch**” tool and then clicking near a bone you can a new branch to the skeleton



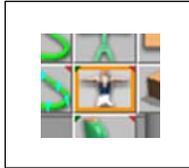
By Right-Clicking on the skeleton in **Build Skeleton Mode** or clicking the **Edit Joints Directly Tool**, you can adjust the movement of the joints. Right-Clicking on the tool will bring up a window



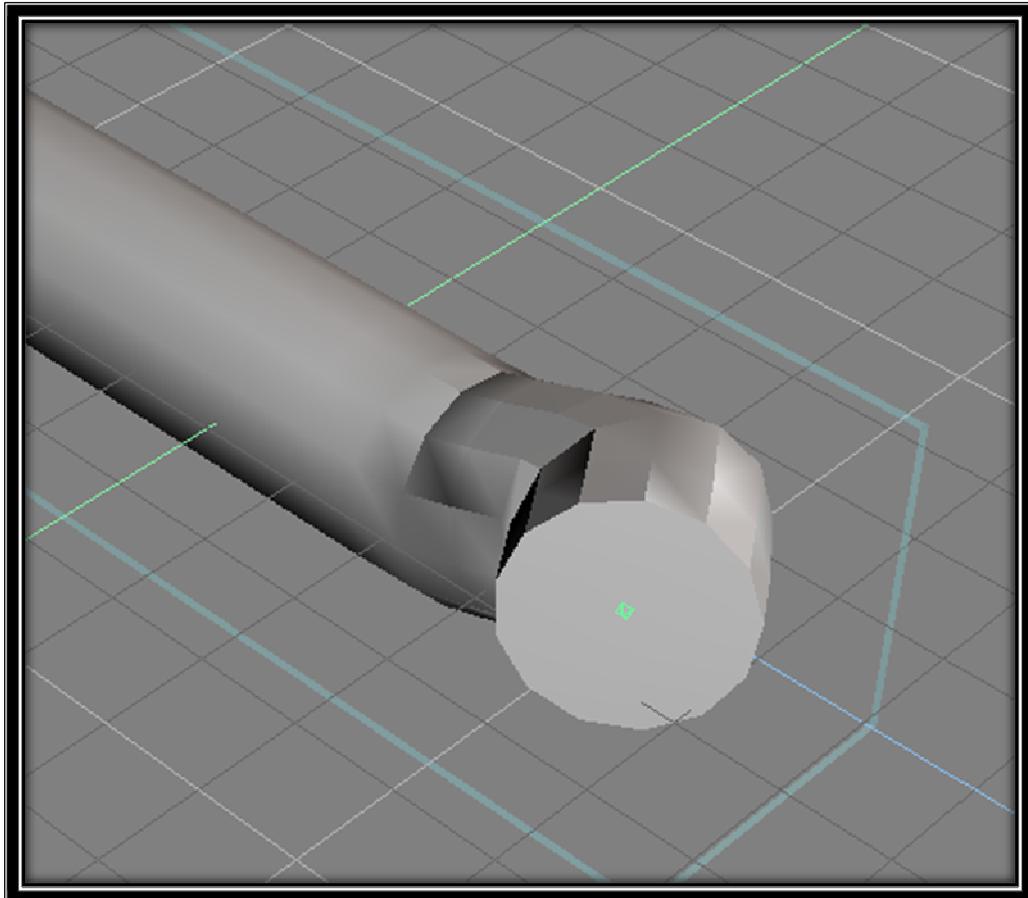
that will allow you to have precise control over the movement



Next, we need to add the skin to the bones. Click on the “**Attach Skin to Skeleton**” tool

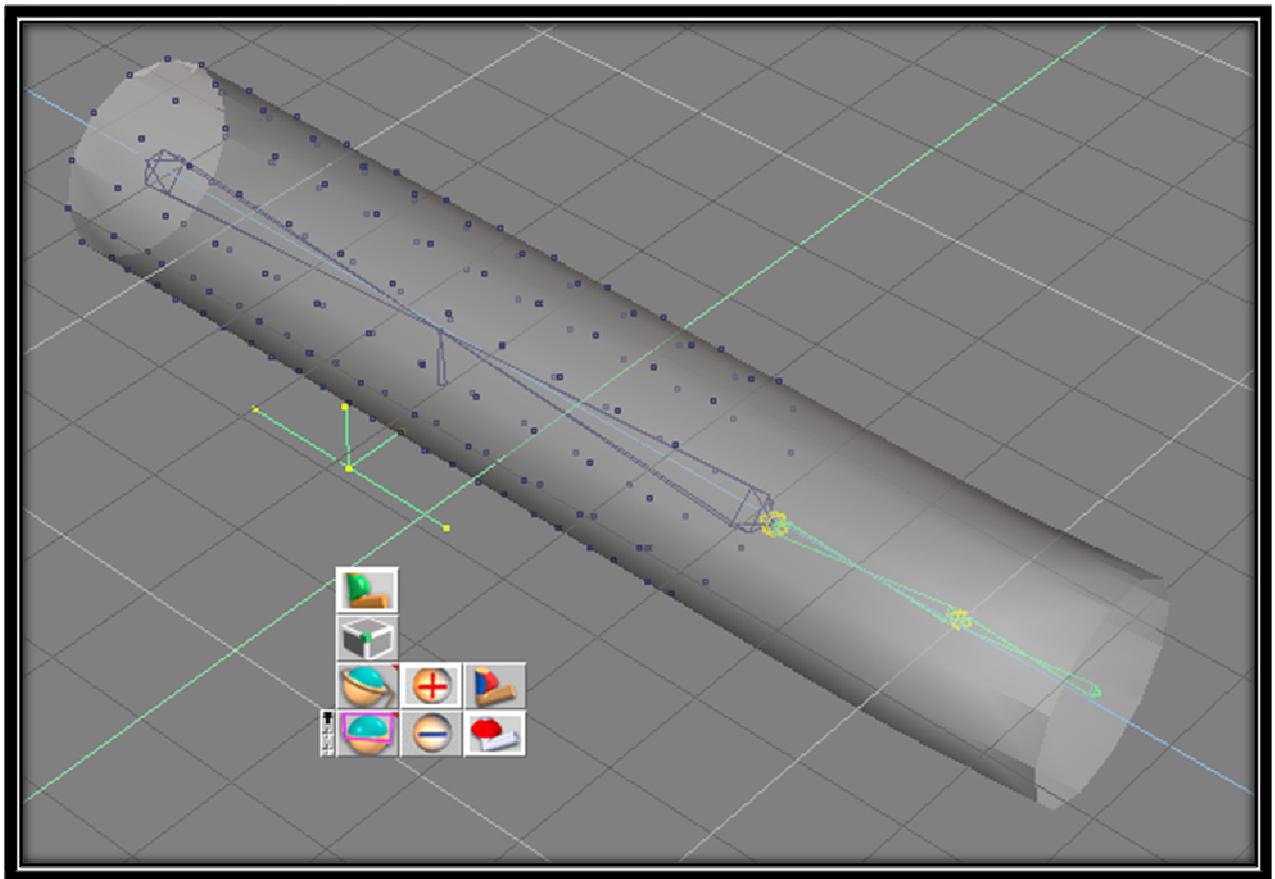


Now click on the tube. That’s it! Your tube is skinned and will deform with the bones

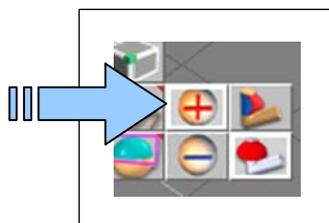




Right-Click on the tube and go into the **Edit Muscles Tool**



In this tool, you can edit which vertices are affected by which bone or joint. This can be accomplished by selecting the bone or joint and then select the **Add Vertices Tool** followed by clicking on the desired vertices

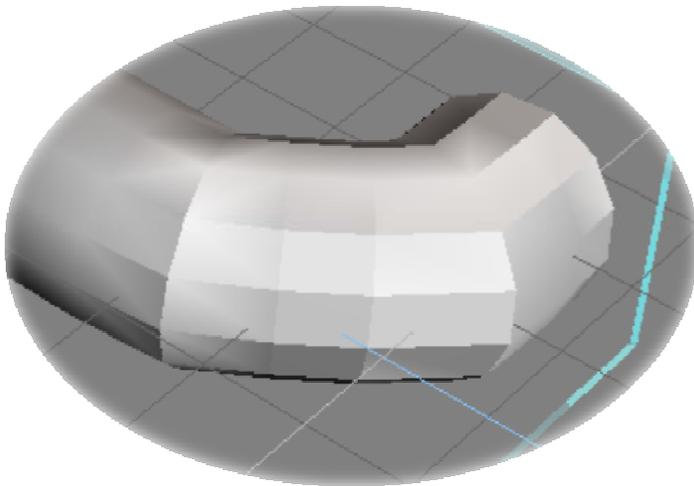


Animating an Object

Start with a base position and record this position at frame zero and at say frame 16 by clicking the “record” button at each frame. Something short just for this tutorial, but you may want or need more or less frames depending on other factors beyond the scope of this tutorial. By recording the same position at the beginning and end of an animation we can get a loop affect



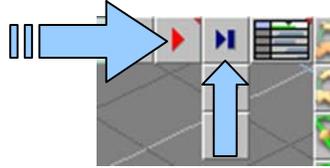
Now change the frame number to somewhere in between and then drag the bones and record it to create a movement. It is important that you change the frame number first and then move the bones or it won't play the animation correctly



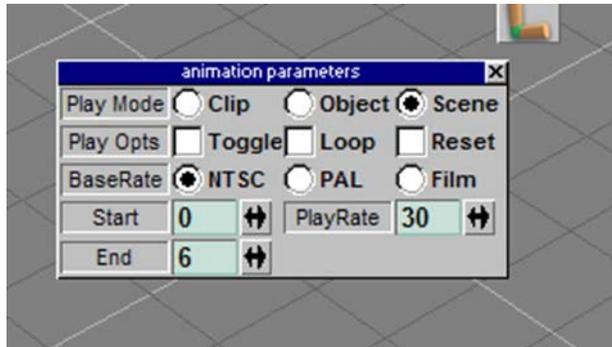
The animation sequences cannot overlap each other; they cannot share frame numbers. The animations may be adjacent to one another though. This is because each frame can only hold one model position, no more. Which frames are used in a particular animation is handled completely in code, whether fixed in the code or imported from a completely separate file



Now click play or change frames to see your animation



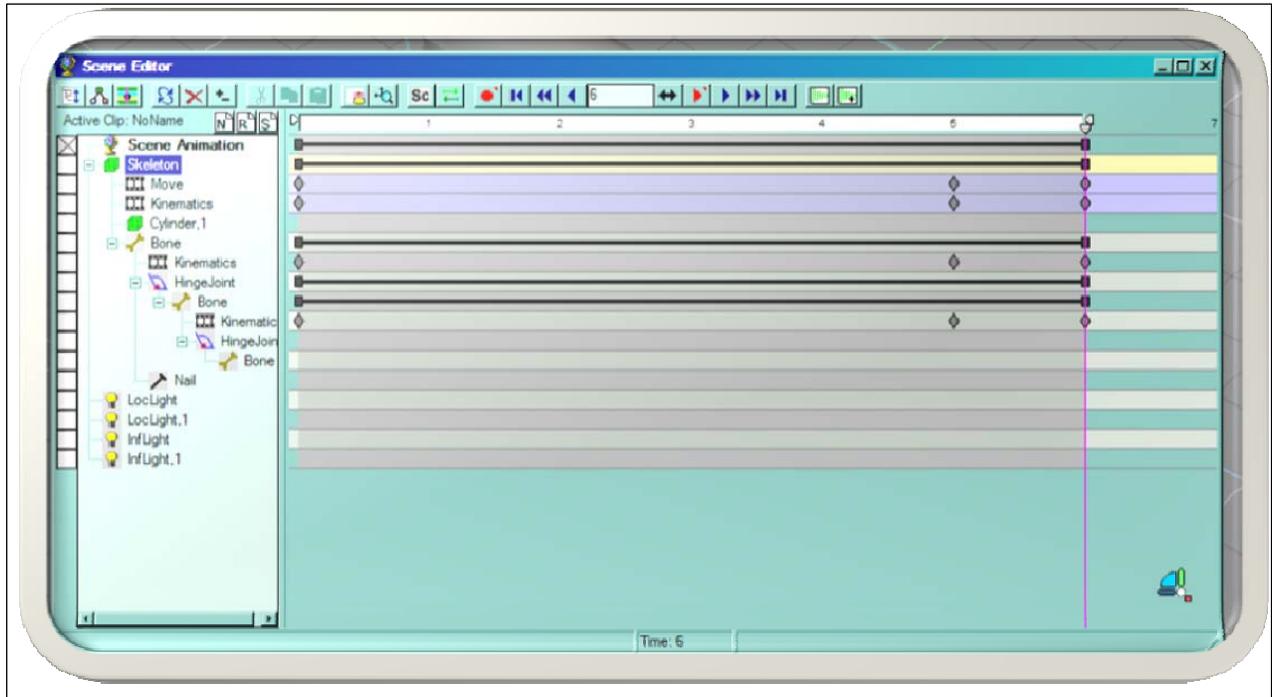
By right-clicking on the play button you can change the play rate and other play options



The scene edit tool can help in animations



The left panel of the window shows the bone hierarchy which can be expanded to show the individual animations for each individual bone or joint. These animations can be copied, pasted and cut which can make animating easier. Animations can also be edited and created in here as well as in the model editing area



Remember, this is a game so you don't need the animations to be perfect and you don't need all the animations in the world, just what will get the job done and will make you satisfied

Have fun!



CREATING A CUSTOM GUN

If you do not want to use any of the pre-made weapons, you can create your own customized weapon by assigning the model the generic "gun" action, then entering numbers into certain skill fields.

Import a gun model into your level

Right-Click on your model and select **Properties**

Click on the **Behavior** Tab

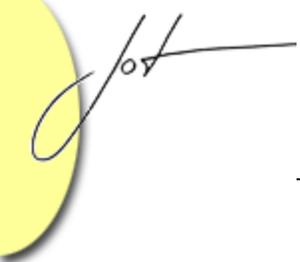
Click the **Choose Action** button

Assign the model the "Gun" Action and Click OK



Now you can modify different flags and skills to personalize your gun! The easiest way to do this is to Right-Click your gun model and select **Behavior**

An alternative would be accessing the **Behavior** Tab under **Model Properties**; but this way, you only see the relevant variable fields



Use the flag and skill descriptions below to determine what type of gun to make

Flag 1: Rotate Gun

If this flag is set, the gun model rotates before being picked up

Flag 2: Silent

If this flag is set, the gun pickup message will be suppressed

Flag 5: Bob

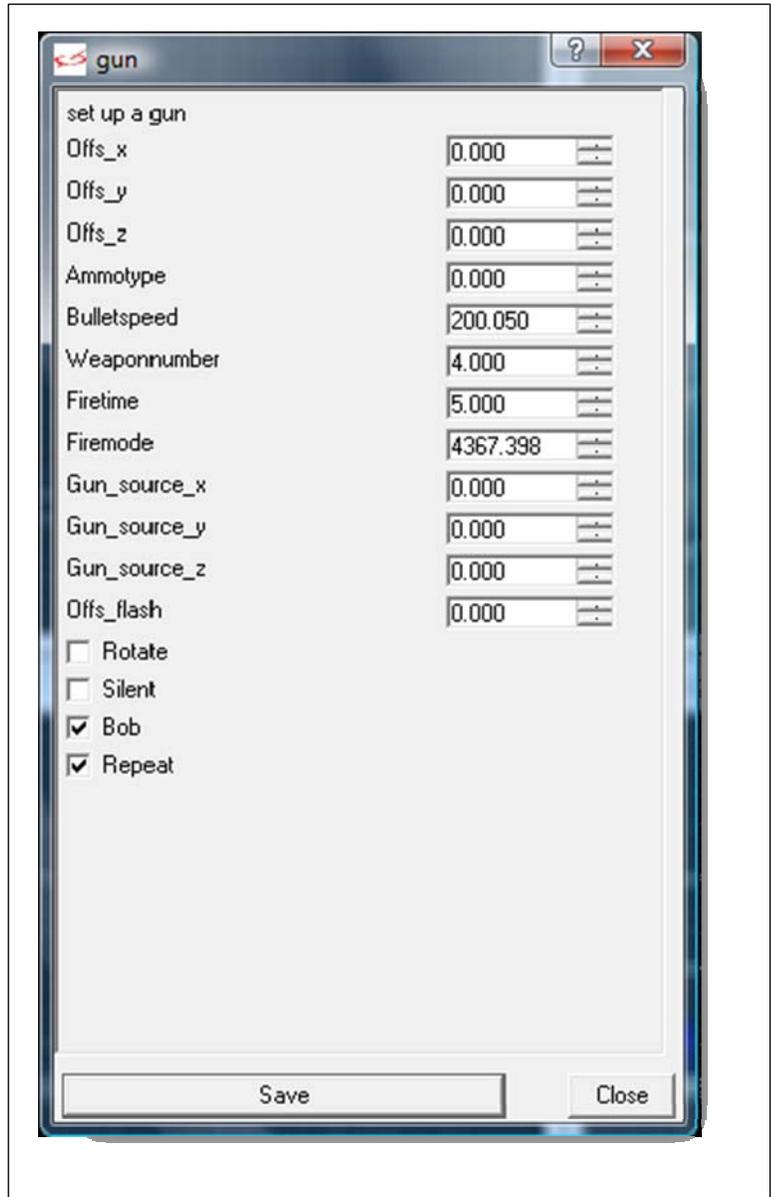
If this flag is set, the gun sways a little when you walk with it

If the flag is not set, the gun stands still

Flag 7: Repeat

If this flag is set, the gun will fire continuously as you hold down the mouse

Ex: Machine Gun, Flame Thrower, etc.





Skill 1 – Skill 3: Gun Position

Ahead, right and down position of the gun, while carried, in quants relative to the camera

Skill 4: Ammo Type

If the ammo type is 0, the gun has infinite ammunition. Otherwise, ammo type is between 1 and 7. The number after the decimal, multiplied by 100, gives the amount of ammo to be added when the gun is picked up

Ex: 2.30 = Ammo type 2, and 30 rounds are already in the gun

Skill 5: Bullet Speed

Speed of the bullet (default = 200 quants / tick). If the gun emits particles or no bullets at all, then its range in quants is twice the bullet speed. The part after the decimal, multiplied by 100, gives the strength of the recoil. If 0, then there is no recoil. If above 0, the recoil is done by sliding backwards by the given amount in quants. If below 0, then the recoil is done by swinging upwards by the given amount in degrees

Ex: 100.05 – bullets move slowly and gun recoils backwards slightly

Skill 6: Weapon Number

Weapon number that determines the key ([1]...[7]) to be pressed to select that gun, if it was picked up before

Skill 7: Fire Time

The time (in ticks) the gun needs to reload. Includes the time for the gun recoil and animation, if any

Skill 8: Fire Mode

Add up the numbers of all the effects you want your gun to have and place the sum into Skill 8.

Ex: Gun that shoots a Fire Ball and produces an explosion when it hits something

$$12 + 256 = 268$$

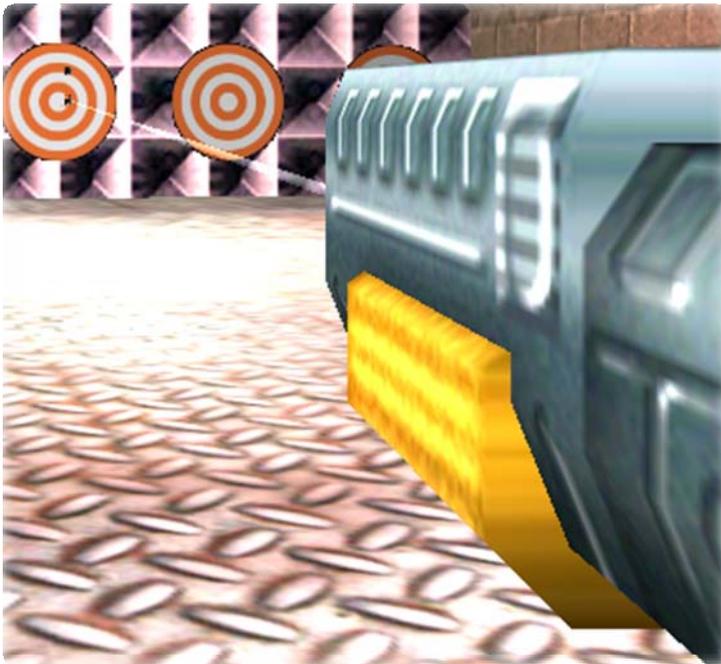
- 1 - Damage is dependent on what type of bullets you are using
- 2 - Damage is dependent on where it impacts the target
- 3 - Damage is dependent on if the target is inside the radius of the explosion (Multiple targets can be damaged at once)
- 4 - Gun produces a vapor trail to the target
- 8 - Gun fires single particles
- 12 - Gun fires orange fireballs (which radiate light)



16 - Gun fires a rocket that leaves a smoke trail



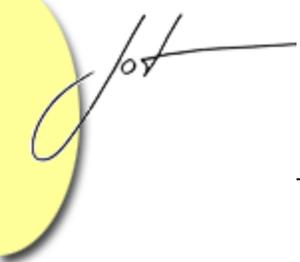
20 - Gun fires a laser beam



24 - Reserved

28 - Reserved

32 - Bullets leave smoke trails



64 – Reserved

128 - At the point of impact there will be a light flash

256 - At the point of impact, there will be an explosion



384 - At the point of impact, there will be a larger explosion





512 - At the point of impact, a cloud of smoke will ascend

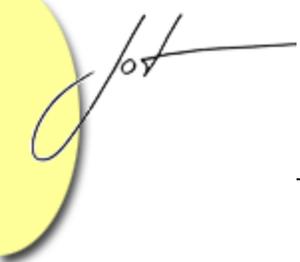
1024 - There will be multiple points of impact

Ex: a Shotgun

2048 - Gun ejects cartridge cases



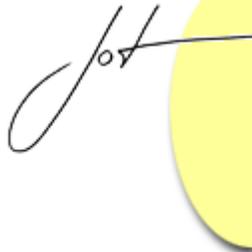
4096 - Bullets follow gravity



8192 - At the point of impact, a shower of sparks is produced



16384 - Bullet hole appears at the point of impact on map walls. The number after the decimal * 100 gives the amount of damage the bullet produces (default = 10). If the bullet explodes, the explosion radius is five times the damage value



Skill 11 – Skill 13: Gun Source

X Y Z offset position of the bullet starting point (in 1st person view). Leave all zero if you want the muzzle to be the start position for the bullets

Skill 14: Flash

Muzzle flash offset in X direction

Here is an example of one of the guns you could make. It's a flame thrower



Parameter	Value
set up a gun	
Offs_x	0.000
Offs_y	0.000
Offs_z	0.000
Ammotype	0.000
Bulletspeed	200.000
Weaponnumber	4.000
Firetime	0.000
Firemode	4367.398
Gun_source_x	0.000
Gun_source_y	0.000
Gun_source_z	0.000
Offs_flash	0.000
<input type="checkbox"/> Rotate	
<input type="checkbox"/> Silent	
<input type="checkbox"/> Bob	
<input checked="" type="checkbox"/> Repeat	

Buttons: Save, Close

About the authors

Rex Cason II has been working with Dr. Prayaga in the UWF Game Department for the past few semesters. He currently possesses a Bachelor's degree in Computer Science and is working towards a Master's degree in Software Engineering at the University of West Florida. Rex is also an active member in the Association of Information Technology Professionals (AITP). In addition to his studies, Rex works part time at the Institute for Human and Machine Cognition (IHMC), where he is currently working on developing software to coordinate the actions of semi-autonomous robotic vehicles.

Erik Larson has been working with computers since he had purchased a cheap 386 IBM Compatible in 1995. In 1999, he entered the United States Marine Corps and pursued a specialization in computers. Today he is working towards a Master's degree in Software Engineering with the University of West Florida. He currently possesses Bachelor's degrees in Information Technology and Computer Information Systems with minors in Computer Science, Internet Technologies and e-Business also from the University of West Florida. He is a member of the Phi Kappa Phi, Gamma Beta Phi, and Upsilon Pi Epsilon Honors Societies.

Jonathan Robertson currently works at the Game Design Department in the University of West Florida. He hopes to one day have a career designing entertainment software with an emphasis on the quality and involvement of the story being told through the game.

Jonathan Frisch is working for a degree in Digital Media and studying animation/modeling itself and in games and movies. He hopes to get into the animation/modeling field of game development or movie production. His ultimate future goal is to be an independent film writer/director.

George Trice III is an Honors student double-majoring in Interdisciplinary Information Technology: Digital Media and Art with a Digital Specialization. His minor is in Communication Arts. He's been a gamer since age 5. Favorite game of all time: Super Mario World

Dr. Lakshmi Prayaga has recently completed her ED.d program from the University of West Florida. She has been actively working on the influence of games in education. In partnership with Escambia County in Florida, she was awarded a \$1.5 million grant from the Florida department of education to develop serious games for 7th and 8th graders for mathematics and its relation to real life careers. These games will be implemented during this fall (2007). She is starting a gaming curriculum at the University of West Florida, and some of her students are working on the tutorials for a first person shooter game that will appear in the next few columns.