LESSON

Morph Targets, Bones, & Particles

New Skills
- Bones
- Skinning (Point Weights)
- Morph Targets
- Particles
- Gravity
- Collision

Objective
Animate the “Soda Can” seamless mesh object by manipulating a virtual armature of bones as well as morph targets to deform the mesh. Add particles to create an explosion of soda fizz when the can is opened.

- Student successfully rigged the Soda Can object with bones
- Student successfully rigged the Soda Can object with morph targets
- Student successfully animated the Soda Can object using Fwd-K/IK bones and morph targets to deform the mesh.
- Student successfully added a particle emitter
- Student successfully applied gravity and collision to the particles

Definitions:
- Morph Map (Morph Target) = vertex map that displaces points (vertices) in your object’s geometry; the position of points in a morph map are based off a neutral “base” position; because the targets only work if they have the same number of vertices as the base mesh, all morph targets should be created as clones of the original mesh that are then modified.
- Morpher = Max’s modifier allowing you to mix the % of influence of 1 or several morph targets on a default (base) geometry. Only works if the targets have the same number of vertices as the base.
- Bone = Object type used to bend, twist, and stretch and generally deform a polygonal or sub-patch mesh; a bone hierarchy can act as a virtual armature inside a seamless mesh object.
- Particles = Objects composed of single point polygons; most often used to simulate star fields, but can be used to simulate smoke, fire, and sparkles. Objects having computed mass rather than modeled mass. Used for environmental special effects like clouds, smoke, water, etc.
PART I: Rigging the Soda Can

Lecture Notes

Morph Targets
“Morph” is short for “Metamorphosis” meaning “to change forms”. In computer graphics “morphing” is an animation transforming an image or model from one thing to another thing -- a metamorphosis. In 3D animation, we use vertex maps called Morph Targets that allow us to reshape our model by changing the position of the vertices without having to build or load another model.

Morph Targets are mostly used for character animation. For example, if you wanted to have a character talk, you could use bones, but you would have to use a lot of bones in the face in order to get the expressions you wanted. Creating and mixing morph targets allows you to get a wide range of character expressions without having to set up complicated bones.

Though Morph Targets are mainly used with creating facial expressions, you can also use them to morph your objects into custom shapes. In the case of this Soda Can, we want to be able to SQUASH & STRETCH the can in a way that reflects the can’s flexibility. We could “scale” the can, but this will squash and stretch the points in the mesh uniformly. Instead, we want to create a shape that shows the can is rigid at the top and bottom, but flexible in the middle. To do this, we will use Morph Targets to squash & expand the geometry in the middle (where the can is flexible), while maintaining the shape of the geometry at the top and bottom (where the can is rigid).
**Bones**

Bones are a special object type used to bend, twist, and stretch and generally deform a polygonal or sub-patch mesh; a bone hierarchy can act as a virtual armature inside a seamless mesh object.

Knowing how you want the bones to affect your model will help you “rig” it. In the case of this Soda Can object, we want the bones to act as a spine to help us bend it around.

Though you can use Forward Kinematics on a bone chain, you can also rig it for IK. But we know the IK solution will be more reliable if A) you use a chain of just 2 bones, and B) the bones are pre-bent to tell the solver which way you prefer the bones to bend.

When rigging a character’s spine -- which is relatively straight and has more than 2 bones in it -- we are better off using Max’s “Spline IK Solver” to control the IK. “Spline IK solvers” control an IK chain by calculating the bone rotation needed to match the shape of the spline. The vertices in the spline are converted to control handles you can move to deform the spline. You usually want your spline to have the same number of control vertices as there are bones, or better yet, less. This type of IK solver is best used with bone chains that have a high number of bones (I.E. a character’s spine or tail, a rope, etc.)

**Point Weights**

After you’ve added bones to your scene (and rigged them), you then need to tell each vertex in the mesh what bone to follow, and how much. Because you have to assign weight to EVERY point in your mesh, you are better off using a low-poly control mesh, rather than a mesh that has already been subdivided.

You do assign the amount a Bone(s) influences the points in your model by creating vertex maps called “Bone Point Weights” (aka “Point Weights”). These vertex maps tell each vertex in a how much it should move with one bone versus another.

This is true no matter how many bones are influencing it. This means it is easier to adjust bone weights in areas where there are less bones influencing the geometry. In the case of a character model, the easiest place to adjust your weight is in the middle of a bone. The next level of difficulty is where two bones meet. Even harder are the areas like the crotch and the neck & shoulders.
For instance, in a knee joint the points are influenced by 2 bones – the Thigh and the Calf. Therefore, the points at the bone joint are influenced 50% by the Thigh-bone, and 50% by the Calf-bone.

In Max, you add Bone Weights by adding a modifier called “Skin” to your mesh. In the dialogue panels for “Skin”, the vertex weights are managed by the “Envelopes”. These envelopes give you more control on how the bones affect your model. The key is to overlap the envelopes so that there’s a good blend. If the bone is affecting 50% by “Spine01”, then “Spine02” should also affect it 50% if you want perfectly smooth transitions between the 2. That way the total percentage adds up to 100%. Now you don’t have to go setting weights for each point of your mesh, that’s what the envelopes do, but you can manually set a vertex point’s weight spinner if you are selecting the current bone/envelope to assign it to. But, we’ll get to that later on.

RULE #1: the total % of influence for each point must total to 100%.

RULE #2: Each point at a 2-bone joint must be weighted to follow each bone 50%.

RULE #3: In a subdivision model that you intend to rig with bones, include one Isoparm (row of control points) sitting in the middle of the joint, and at least one Isoparm on either side of that joint (3 Isoparms total per joint). [For joints were you need better control over the mesh, you can include 2 Isoparms on either side of the joint (5 Isoparms total per joint).]

**Exercise (Adding Morph Targets)**

1. Open the “SodaCan_Start.max” and immediately save as “SodaCan_01.max” in your work directory.
2. Clone the can out to the right [Shift + Drag], and name the copy “Can-Squash”.
3. In the Modify>Edit Mesh>Vertex sub-object level of the “Can-Squash” object, scale the middle points down and out until you get something like the image below.
4. Clone the original can out to the right [Shift + Drag] past the “Can-Squash”, and name the 2nd copy “Can-Stretch”.
5. In the Modify>Edit Mesh>Vertex sub-object level of the “Can-Stretch” object, scale the middle points up and in until you get something like the image above.
6. Select the original Can mesh.
7. Add the “Morpher” modifier.
8. In the “Channel List” rollout, click on the “Load Multiple Targets” button and select the “Can-Squash” and “Can-Stretch” objects.
9. You should now see both Morph Targets in the list.
You can turn them on/off by increasing their channel value.

**Reload All Morph Targets**—Reloads all the morph targets. If the targets have been edited, the channels are updated to reflect the changes. If a morph target has been deleted from the scene, then the morpher updates using the stored data in the channel, functions using the last stored morph data.

**Zero Active Channel Values**—Click to create keys with a value of 0 for all active morph channels, if the Auto Key is on. This is handy to prevent key interpolation from distorting the model. First click Zero Active Channel Values, and then set a particular channel to the value you want; only the altered channel affect the model.

**Automatically reload targets**—Turn this on to allow animated targets to be updated dynamically by the Morpher modifier. There is a performance penalty when using this option.

10 To animate the morph targets, turn on AutoKey and alter the channel values. You can also add keys and edit the Morph Target curves in the Curve Editor.

**Exercise (Adding Bones)**

11 Select Create>Systems>Bones to add bones which will act as the Can’s spine.

12 Once you've clicked the “Bones” button, you’ll notice that you can change the Bone Object Width & Height (size of your bones) in the Bone Parameters rollout.

12 Because our can is small (5” tall), you will want to set the bones to be small, too – about:

- width = [0’0.25”]; height = [0’0.25”]

13 You can also add Bone Fins (helps tailor bone’s area of influence over the points in the mesh).

13 Activate the Side Fines, Front Fin, and Back Fin option, and set the size of each fin = 1.0”
In the Front or Left view, click and drag out “Spine01”, “Spine02”, “Spine03”, “Spine04”, “Spine05”, “Spine06”. The bone joints should sit on an isoparm.

RMB to exit the bone create mode. When you do this, you will get a tiny bone at the end of your chain which we’ll name and later use as the “Spine_Puller”.

If necessary, you can modify the length of the Bones by using the Character>Bone Tools> Edit Bone Mode.

When you’re done, turn off Edit Bone Mode.

In the Modify panel, change the name of your bones to “Spine01”, “Spine02”, “Spine03”, “Spine04”, “Spine05”, “Spine06” & “Spine_Puller”.

Use the PAGE UP and PAGE DOWN keys to move up/down through the bone hierarchy.

Select your bones and make a selection set called “Bones”

Save your scene.

Exercise (Rigging with the Spline IK solver)

Under the Display tab, check “Geometry” to hide your Can mesh.

To create the spline we will use to control the Spline IK Solver, you need to activate the Create tab>Shapes button>Line tool. With the tool active, change the “Initial Type” (located in the “Creation Method” rollout) to SMOOTH.

In the Front view, click next to the Can to set down 4 points in your line – one for each bone.

You CAN add fewer points than the number of bones, but do NOT add more points. The benefit of the Spline IK Solver is that it allows you to animate a bunch of links using a limited number of controls. Remember: “LESS is best”.

Name your spline “Spine_Bender”

Select the bone you want to be the base of your IK chain (“Spine01”)

Add the Spline IK Solver by selecting: Animation menu>IK Solvers>Spline IK Solvers

Click to select the bone you want to be the end of your IK chain (“Spine_Puller”)

Click to select the spline you want to control the Spline IK solver (“Spine_Bender”)

Once you click the spline, the program moves the Can to where the spline sits and adds 1 point helper object per vertex in your spline. These point helper objects will now act as handles you can position and rotate to transform the shape of the spline (which in turn will transform the IK chain).

With the IK Chain 01 selected, you can adjust the size of the “IK Goal” display in the Motion Tab>IK Display Options rollout. I set mine to [1.0].

With any of the Point Helpers selected, you can adjust the size of the “boxes” under the Modify tab. I set mine to [1.0”]

Test the IK Chain

Select any of the Point Helpers and move/rotate them around. The IK chain should try to match the shape of the spline.

When you are done, UNDO [ctrl+Z] any move or rotate commands.
30 Under the Display tab, UNcheck “Geometry” to show your Can mesh.
31 Move the rig back to the world origin (0,0,0) & save your scene.
   ► You can move the entire rig by selecting and moving the “Point01” Point Helper.

Exercise (Assigning Point Weights)
32 Select the Can mesh.
33 In the Modify tab, select “Skin” from the “Modifier List” pulldown menu.
34 Under the Parameters rollout, activate the “Edit Envelops” rollout.
35 In the same rollout, click “Select Vertices”
36 In the same rollout, click the ADD button to add the bones you wish to deform the mesh: “Spine01”, “Spine02, “Spine03”, “Spine04”, Spine05”, and “Spine06” – not the “Spine_Puller” bone. We’re only using the “Spine_Puller” bone to give our IK chain something to reach for).
37 Select “Spine06”
   ① You should now see the vertices ranging in color from Blue (0% influence) to Red (100% influence). The vertices colors will change depending on which bone you have selecting in the Parameters rollout. Because we added the Fins, the program does a pretty good job of guessing the percentage the points should follow the bones. We can continue to work with the Envelopes (the capsule shaped gizmos) or we can work directly with the vertices, editing the % of influence by selecting individual points and typing in the percent we want. I have rigged enough models to know what % I want to use. As a beginning rigger, you may want to continue to work with the Envelopes until you have a better idea of the effect you want.
38 Test your rig.
   ► To see through the Can mesh, hit [ALT+X]
   ► Activate the AutoKey mode
   ► Scrub to about frame 20. Do not animate at frame zero!!
   ► Move your Point helpers around. You should now see the Can mesh deforming to match the spline.

Exercise (Adding the PopTop Bones)
39 Create two bones – one for the LEVER and one for the POPTOP – each will have a Puller bone at the end.
40 Name the bones.
41 Activate the Edit Bone Mode: Character>BoneTools>Edit Bones.
42 Move the bones so they intersect the middle of the PopTop, and their pivots sit where you want the PopTop to pivot from.
43 In the Skin Modifier, add these two bones (not the pullers).
Under the Skin’s Display rollout, Hide All Envelopes.
Select the LEVER bone
Select a single vertex in the Lever and click GROW until all the verts are selected.
Assign an Abs Weight of [1.0] to these points
Invert your selection and assign an Abs Weight of [0.0] to all the other points.
Repeat for the PopTop bone.

Exercise (Animating with Bones)
Extend your frames between 0 and 120 (5 seconds).
Use forward kinematics, inverse kinematics, or a combination of the two to create your own version of the following 5-second sequence:

► 5-Second Sequence
  → SodaCan enters into the camera’s view
    ➢ Try to create personality – happy/energetic = bouncy hop; sad/tired = sluggish hop
    ➢ Don’t forget you can add Squash & Stretch with your morph targets.
  → SodaCan “looks around”
  → SodaCan see’s camera and does a double-take
    ➢ You can create a double-take by making the can bend forward/backward (with a little anticipation) and maybe add a “hop”
    ➢ Again, try to create some emotion in the can. Is it Scared? Happy? Angry?
  → SodaCan anticipates the soda “pop” explosion
    ➢ Might want to start by having the can bend over & wiggle its “rear-end”
  → SodaCan top pops open and soda explodes out (soda fizz will be created with particles).
    ➢ You can open the pop top with another Morph Target.
    ➢ As the top pops open, you may want to have the can hop into the air to help accent the explosion.

Save your scene file in versions as you work (ie. “SodaCan_02.max”, “SodaCan_03.max”, “SodaCan_04.max”) so you won’t lose anything.
**PART II: Particles**

**Lecture Notes**

**Particle Systems**
Particles are a special object type made of simple shapes (usually small spheres or points) that are used in massive groups, or systems, to simulate a variety of natural materials and phenomena whose shape constantly changes over time. This includes snowstorms, clouds, flowing water, moving soil, and fire. To create natural phenomena, the particle system produces large numbers of individual particles that group together and change over time according to rules that define the behavior of that particular element (water, fire, smoke). When particles are used to recreate the motion of water, they each represent a drop of water with attributes like density, cohesion, transparency, and refraction. When particles are used to recreate the light of fireworks, they represent a point of light with a variety of attributes such as intensity, flickering, and tail-tracking values. Particles have a life span during which they are created, behave in a certain way, and then fade away or merge with other particles.

**Exercise (Particles)**

53 Create a “Super Spray” particle system.

54 In the top view, click and drag out the Super Spray particle system.

55 In the Modify panel, change the Icon Size to 1.0”

56 Use the Schematic View to link/connect the SuperSpray01 object to the Spine Puller.

57 Use the “Parent Coordinate System” and zero out the SuperSpray01’s position & Rotation.

58 In the Top & Side Views, center the SuperSpray01 object in the middle of the mouth of the can.
Enter these settings in the Basic Parameters & Particle Generation roll-outs.

- Under the Basic Parameters roll-out, the “Off Axis/Spread” and “Off Plane/Spread” values keep the particles from emitting in a straight line.
- Under Particle Generation, I used [1000] as my “Use Total”. I also used [8] as my “Emit Start” as that is the frame where I wanted the particles to start emitting. By setting the “Emit Stop” -- 10 frames later -- I will end up with 100 particles per frame (1000 total / 10 frames = 100 per frame). The life of my particles is 24 frames. After that, they will disappear.
60 Enter these settings in the Particle Type roll-out.

- The “MetaParticles” type will make the particles clump together.

- The “Tension” determines how clumpy they are.

- The “Get Material From: Icon” means that the particles will render with whatever material you assign to the SuperSpray01 object.

61 Select the Can.
62 Open the Material Editor and change the “Can-Label” material’s Diffuse map to use the Flavor label of your choice.
63. Select the “SuperSpray01” object.
64. Back in the Material Editor, select a blank slot.
65. “Get Material” from the Material Library.
   ► Choose something that is kind of shiny and see through (I picked “Ground Water”)
66. Change the color so it is appropriate for the flavor of your soda.
67. Make the material about 75% Opacity.
68. Make the material double sided.
69. Assign it to the “SuperSpray01” object.

Exercise (Render)
70. It’s a good idea to make a preview first.
   ► Animation>Make Preview
71. When you’re ready to make a final render, select Render Scene [F10]
72. Follow your Net Rendering notes, or if you render locally, make sure that...
   → Under Time Output Active Time Segment is selected OR you can select Range and type in the start frame # and end frame #.
   → Output Size = 320 x 240
   → Render Output
      ➢ Save as type: QuickTime (*.mov) or AVI (*.avi)
      ➢ Give it a file name in the appropriate folder and click Save
      ➢ Leave Compression Setting at Best and make sure frames per second is set to 24. Click OK
73. Make sure viewport is set to Camera01 and click Render.