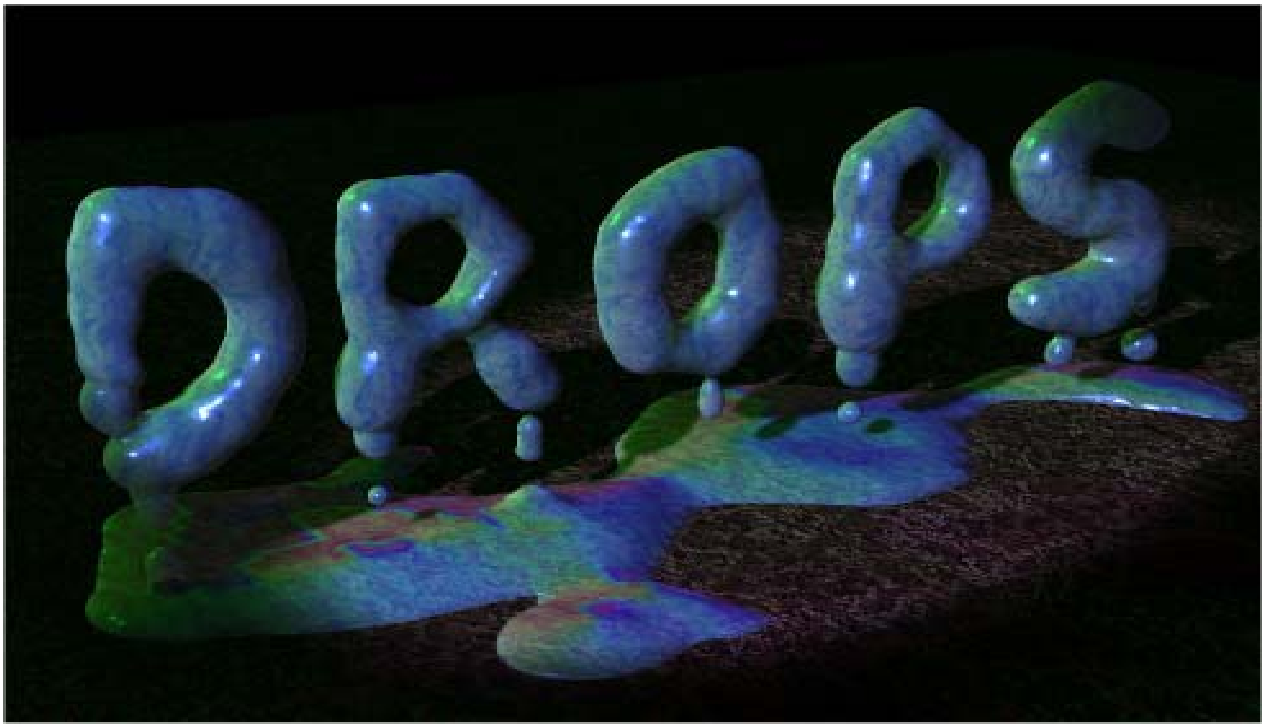
Metaballs for Poser™

User's Guide



Trademark & Copyright

Contents

Introducing – Metaballs in Poser

WeirdJuice:Drops is an implementation of Metaballs for Poser 4 ProPack and Poser 5 users. Metaballs are commonly used for modeling, especially of biological and natural shapes, and can be animated to simulate fluid flow.

For those unfamiliar with the concept from other 3D software, **Metaballs** (**Blobs**) are spheres that blend into each other when they are in close proximity.

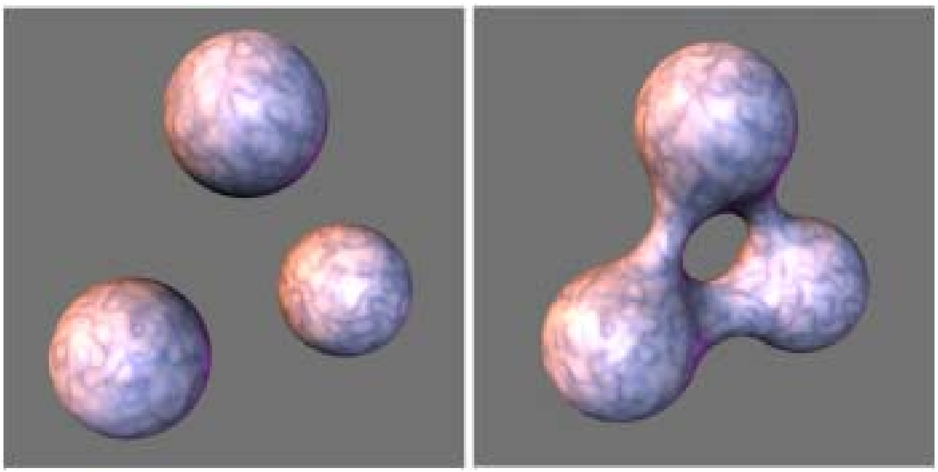


Figure 1. As Metaballs are moved together, they begin to merge.

With ***Drops***, Blobs can be created in Poser with a simple click of a button. Blobs are implemented as normal Poser props and can be positioned using the standard Poser controls. When the desired arrangement is achieved, another click of a button will display the interacting surface, which can be textured and rendered as in the images above.

For animators the position, rotation and scaling of Blob props can be key-framed just like any other Poser props, and again a single click of a button will reveal the resulting surface which will animate throughout the timeline according to the movement of the underlying Blob props.

Installation

See the ReadMe file included with the package for information on installing the plug-in.

Launching the Plug-in

Once the plug-in has been unpackaged, it can be started by launching the Python script 'Drops.py'. In the default installation, this script can be found in the `Runtime\Python\poserScripts\WeirdJuice` subdirectory of your main Poser directory.

You can launch this script directly using the **File->Run Python Script…** menu option in Poser and navigating to the location in the Open File Dialog that appears.

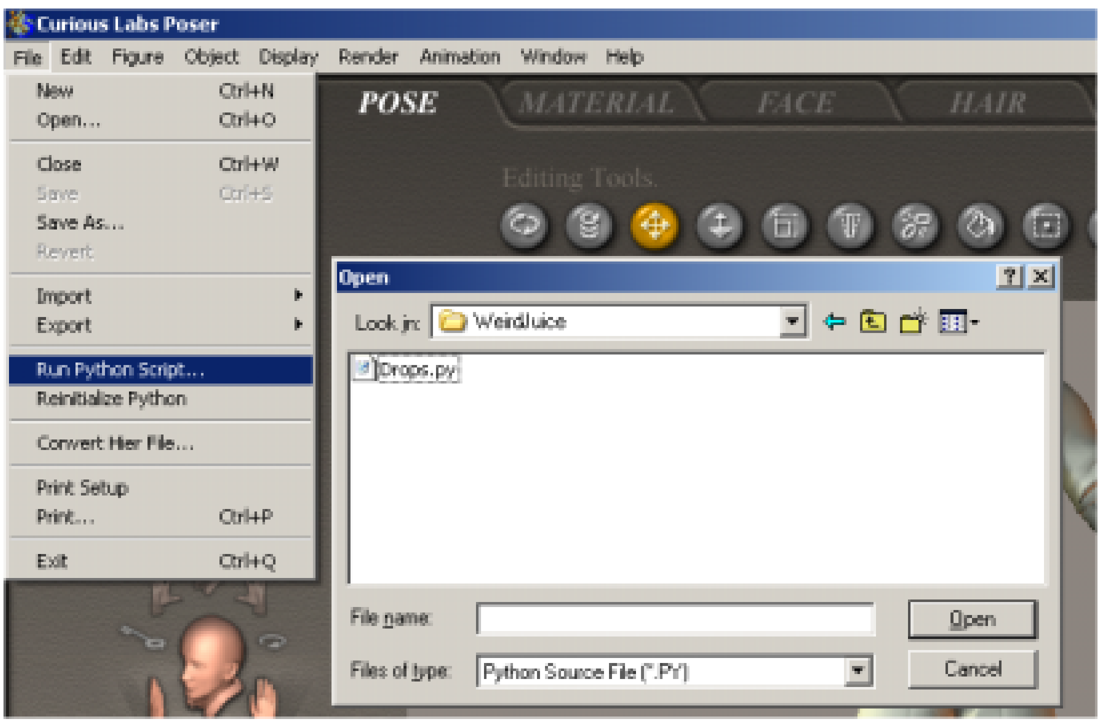


Figure 2. Launching ***WeirdJuice:Drops*** Directly

Alternatively, you can open the Python Scripts Window from Poser's **Window** menu.

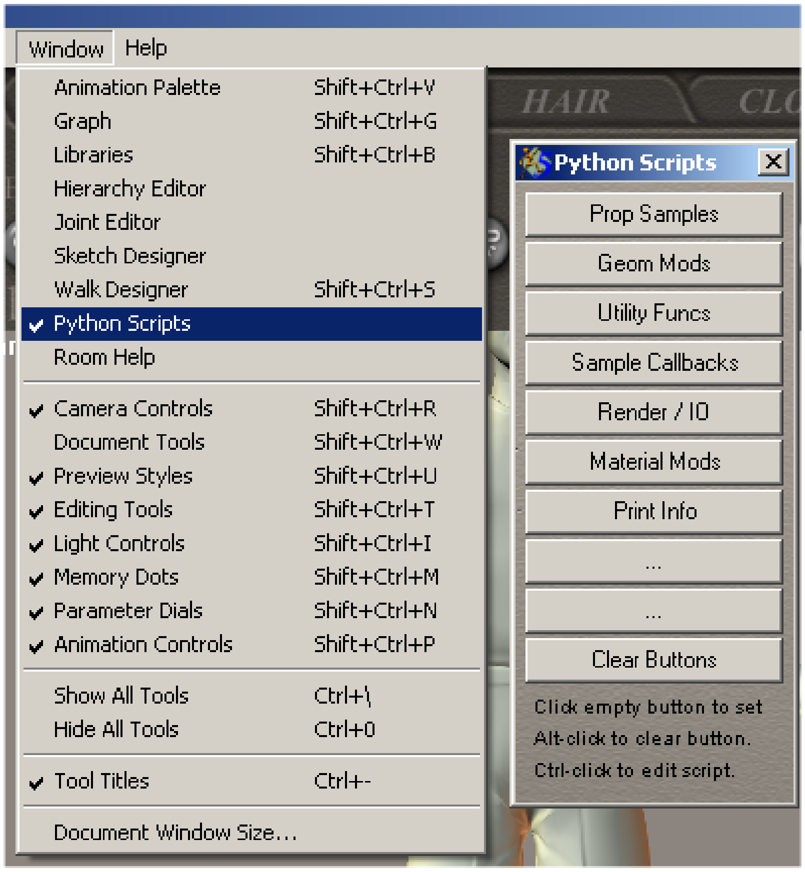


Figure 3. The Python Scripts Window

Once you have opened this window, you can temporarily install the ***Drops*** plug-in by clicking on one of the empty buttons. If there are no empty buttons, alt-click can be used to clear existing buttons. Changes made to the button list in this way are not permanent; the buttons will be reset the next time you open Poser, and you will need to repeat the process again.

In order to install the plug-in permanently to this menu, see Editing the Python Buttons Script.

Using WeirdJuice:Drops

Once you have started the plug-in, this window appears:

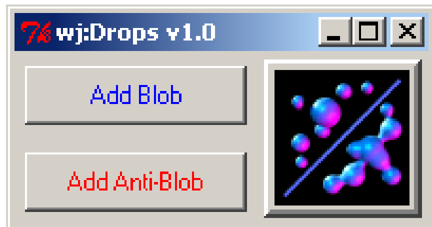


Figure 4. The ***Drops*** Interface.

This window stays on top of the Poser window. It is intended to be small and unobtrusive, but if you want to get it out of the way you move it in the normal way, use the standard minimize control on the top left to reduce it to an icon, or even close it and restart it again when you have further need for the functionality.

The ***Drops*** window contains three buttons, on the right hand side are the "Add Blob" and "Add Anti-Blob" buttons which create the Blobs and, on the left side, is the big *Surfacing* button, which makes the individual Blob props invisible and the iso-surface that surrounds their fields appear.

* ***Quick Exercise****

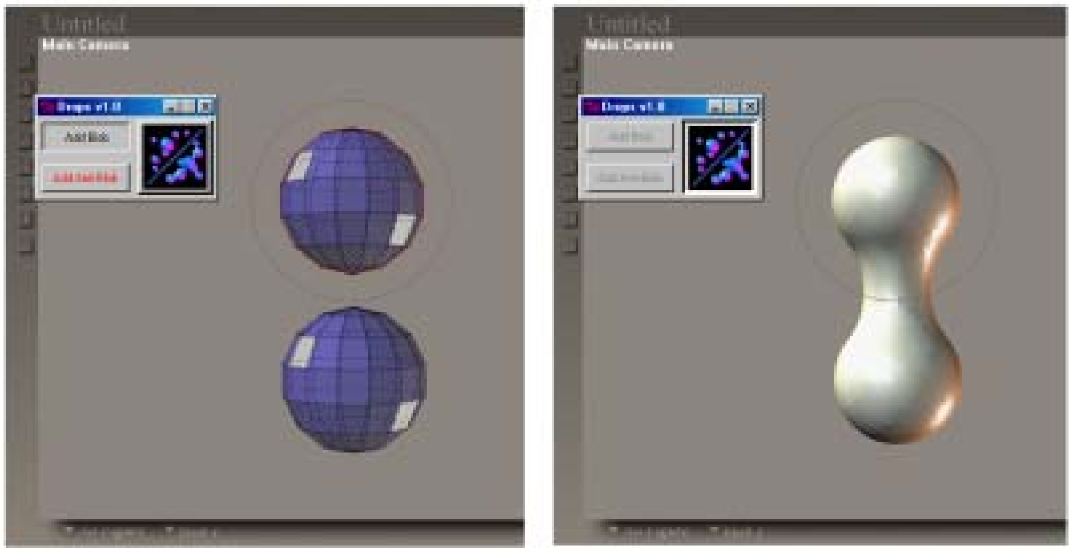


Figure 5. Two Blobs

Click on the "Add Blob" button. Near the centre of the main Poser window, a blue semi-transparent sphere prop should appear. This is your first Metaball. You can select this, move it, manipulate it and delete it using all of the standard poser tools and dials.

Now move your first Blob prop up a little and create a second by clicking on the "Add Blob" button again. Move this Blob so that it is close to or touching the first Blob. Now click on the *Surfacing* button. The blue Blobs disappear and a (default white) **Surface** prop appears in their place. Depending on how close the Blobs were, the surface might appear to be

a) Two separate spheres (the blobs are far apart).
b) Two "kissing" spheres (as in the diagram below).
c) One distorted ellipsoid (the blobs are close together).

The *Surfacing* button remains depressed until you click on it again, at which time, the surface disappears and the Blobs re-appear. Try changing the distance between the Blobs, or add more Blobs and then click on the *Surfacing* button to see the effect.

Even when the *Surfacing* button is depressed, you can still select the invisible Blobs using the props menu and manipulate them, but the effects will not be visible on the surface until you re-toggle the surfacing button.

The "Add Anti-Blob" button is identical to the "Add Blob" button, but it creates a **negative** strength Blob, colored red for reference. Negative strength Blobs can be used to create depressions or holes in the surface.

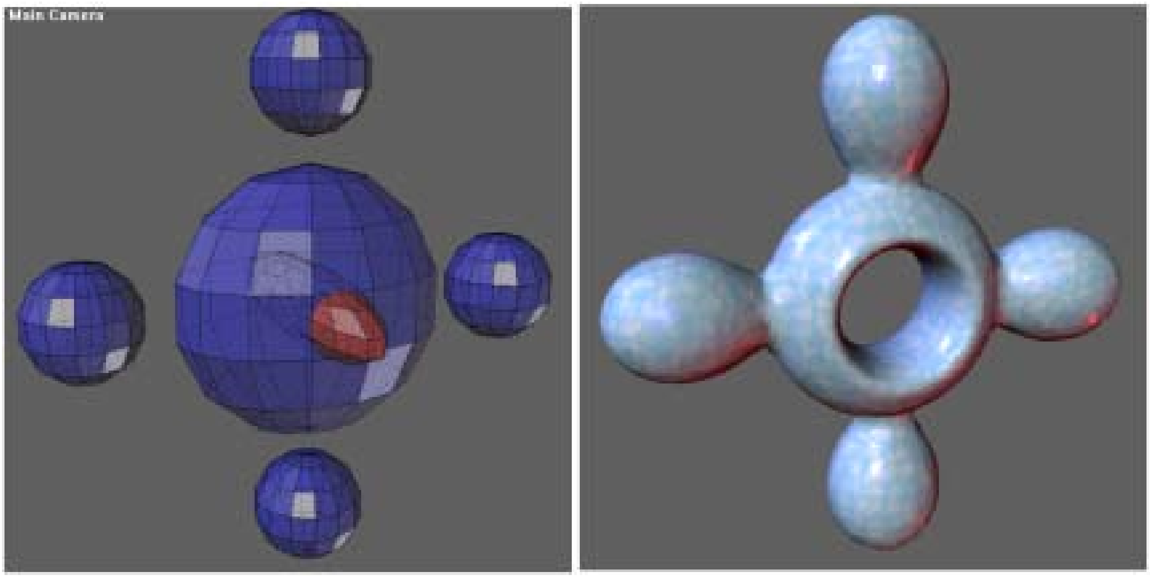


Figure 6. Positive *Blobs* and a Negative *Anti-Blob*.

You can also select the Surface prop, either through the 'props' submenu at the bottom of the Document Window, or by selecting it directly with the mouse in the document window when the *Surfacing* button is active. The Surface prop, cannot be moved, scaled or rotated. You can, however, change its material settings, and the material settings you apply to this object will be the ones visible when you render or display in full textured mode.

The Blob Props

As a convention, in this guide, the prop objects themselves are referred to as "Blobs", whereas the term "Metaball" is used for the conceptual idea of the (field generating point) objects.

One important subtlety is that the Blob props do not form the surface; they only act as guides. When *Surfacing* is activated, the plug-in, simply turns all of the Blob props invisible and then creates the geometry of the Surface prop. Therefore, changing the material setting or morphing the geometry of the props themselves with magnets or other deformers has no real effect on the surface created.

However, Blobs and Anti-Blobs can be scaled universally as well as in separate axis and rotated, just like any other Poser objects and the effects of these transformations are applied directly to the iso-surfaces they contribute to.

The blue and red, colors of the Blob props are dependant on the field strength of the Blobs. Shades of blue indicate positive strength Blobs and shades of red are negative. Blobs with zero field strength appear black.

The Blob props have field strength parameters and subdivision parameters. See Fields and Thresholds and Subdivision and Sampling respectively for the effects of altering these parameters.

The Surface Prop and Texture Mapping

What was not immediately obvious when you created your first Blob prop was that the plug-in also created an (initially) invisible Surface object prop as well. This prop is the object that becomes visible when the *Surfacing* button is depressed (Surfacing Mode) and the Blob props disappear. When a Blob is created, the plug-in checks to see if a surface exists in the current scene and if not, creates one.

You can select this Surface prop, either through the menus, or directly by clicking on the visible surface in the document window when in surfacing mode.

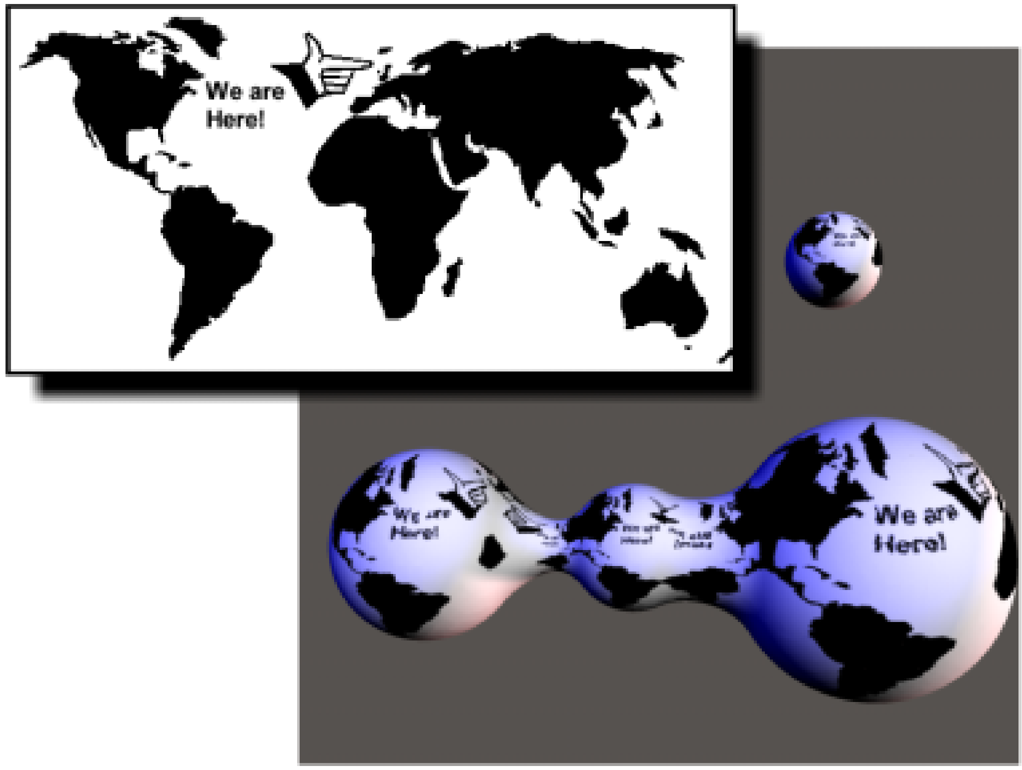


Figure 7. Route Planner.

The surface can have materials and textures applied to it, and these will give it its visible appearance when surfacing is active. Apply material changes to the surface prop material labeled ***Layer0***. The other surface materials (Layer1 to Layer31) are include for compatibility with other Weird Juice products and are not used by ***Drops***.

The mapping of the surface's UV coordinates is spherical and based on the normals of the surface. If you are using Poser 5, you can also make use of object and world space mapping in the P5 material room. Bear in mind that when object space mapping is used, the origin of the Surface prop is always the origin of the Poser universe, regardless of where in the scene the surface is visible.

Fields and Thresholds

In theoretical terms, Metaballs define surfaces by emitting a virtual field out to a set distance (the extent). Where the strength of that field matches a predefined threshold value, the surface is drawn or created. The strength of the field is at its maximum value at the centre of the Metaball, and falls to a zero value at its perimeter. In regions where the extents of more than one Metaball overlap, the field contributions are added.

This process is best described with the weather map / isobar analogy. Isobars are (one dimensional) lines on a (two-dimensional) map at the points where the atmospheric pressure is at some pre-defined value, hence they form concentric rings about areas of low and high pressure. In the same way Metaballs generate (two dimensional) spherical surfaces (also referred to as Isosurfaces) in (three dimensional) space around areas of low or high field strength.

Where pressure areas on a weather map move close together, the isobars begin to join, in the same way that the surfaces of Metaballs in close proximity also begin to deform and merge.

The Field Strength Parameter of Blob props can be used to increase the strength of the virtual field that the Metaball generates. Increasing the value for a positive field generating Metaball has the direct effect of pushing the surface that will be formed away from the centre of the Metaball. The maximum extent of the Metaball (beyond which its field is zero regardless of the Field Strength) is twice the radius of the blob prop.

Conversely, decreasing the Field Strength parameter will cause the surface to be formed closer to the centre of the Metaball. If the Field Strength is lower than the current threshold value, that Metaball in isolation will form no surface, but it may still have an effect on the surface created by neighboring Metaballs.

Metaballs with negative field strengths will, likewise, not create surfaces, but their fields will affect the shape of the surfaces formed by their positive neighbors.

The Threshold parameter of the Surface prop controls the Field Strength at which the surface is formed for all Metaballs in the scene. Therefore, changing this value has a global effect on the entire surface.

When the threshold is increased, the surface shrinks towards the centre of each Metaball, as it is in these regions that field strength is greatest. Decreasing the threshold (it can never go to zero or negative) allows the surface to be created at points of much lower field strength, causing the surface to expand outward from the centre of the Metaballs towards their spherical extents.

Adjusting the Field Strength and Threshold parameters also affects the curvature of the surface where multiple Metaballs interact. For more details, see Improving Smoothness and Surfacing.

Subdivision and Sampling

The plug-in cannot realistically calculate the field strengths at all points in the 3D universe, so instead it only tests (samples) the field strengths at certain fixed positions in space. The process of dividing up space into sampling points separated by a specified distance is referred to as subdivision and can be controlled using the Subdivision Level parameters of the Blob props.

In ***Drops***, spatial subdivision increases in steps, with each step, subdividing space into volumes with edges half the length of the previous subdivision level (each cubic volume of space is d*iced* into 8 sub-volumes). Therefore subdivision does not change smoothly. The field strength samples are made at the 8 corners of each of the cubic subdivision volumes.

Therefore, increasing the subdivision level makes the resulting surface much smoother, but at the cost of an increase in memory resources and processing time whenever the surface geometry is re-calculated as well as increased polycount of the final Surface prop.

Creating Props

A surface can be saved as a prop in the Poser library in the normal way. The most important thing to remember is that the Surface prop must be saved, not the Blob props that contribute to it.

However, when a library prop is created from a surface, its parameter dials are retained, and, if it is reloaded into a scene with ***Drops*** active, it will be treated just like a surface again, i.e. its geometry will be recalculated according to the Metaballs in the scene at the time, effectively replacing the saved (library) geometry. If another surface prop already exists in the scene, the library surface will most likely be ignored.

A better method to create props using surfaces is to export the surface prop in one of the 3D file formats supported by Poser, using Poser's **File->Export** menu option. Generally, the Wavefront OBJ format is preferable since this is closely related to Poser's own native object representation. Once the prop has been exported, third party utilities can be used to remap the textures or provide additional modeling before the object is re-imported into Poser or other 3D applications.

Regularizing Meshes

Another aspect of modeling props using ***Drops*** is the unevenness of the subdivision and hence the surface polygons created when Metaballs of different sizes are used. This is because the degree of subdivision in a particular volume is dependent on the scale and parameter values of each of the Blob props.

In order to make the scale more regular throughout the prop, you can create a single large Metaball with a high Subdivision Level parameter and zero field strength that encloses the entire prop (or the regions that you want regularized). The Metaball will not add any field contribution to the model as it has zero field strength within its volume, but it will create a regular subdivision at the desired level.

Animating Metaballs

Just like any other Poser objects, the Blob props can be key framed and animated. When in surfacing mode, the surface will also animate with the animation of the Blobs.

Animation of the surface can be generated by using all of the usual methods in Poser, including the animation VCR controls and scrubber at the bottom of the screen. Movies can also be made with "Make Movie" dialogue in either display or full rendered mode. Just remember to click on the Surfacing button before starting that all-night render.

Improving Smoothness and Surfacing

The most obvious method to improve the smoothness of surfaces is to increase the values of the Subdivision Level parameters of the Blob props. However, the Subdivision Level parameter only serves as a guide, so changing them can be a matter of trial and error.

Another factor that governs smoothness when two or more Metaballs interact is the gradient of the fields. The effects of this are most noticeable when working with Metaballs of different sizes.

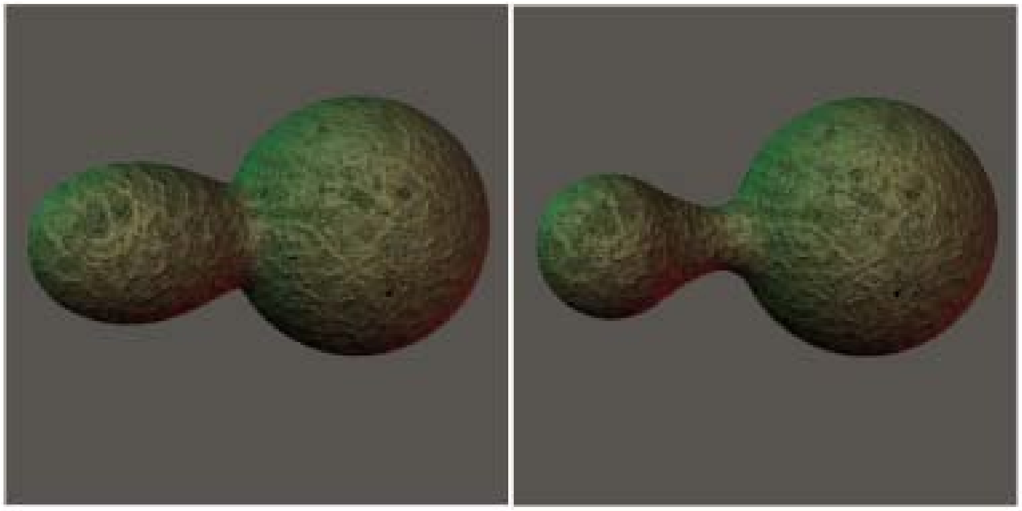


Figure 8. The Effects of Field Gradients.

In the two images above the surfaces are created by the interaction between two unequally scaled Metaballs at the same positions in each image.

In both images, the Blob prop on the right-hand side has a Field Strength of +1.0 and a scale of 100%.

In the left hand image, the Blob prop on the left has a Field Strength of +1.0 and a scale of 200%. When the surfacing is applied the effect of the right-hand Metaball in this image appears to be swollen in comparison with the same Metaball on the second image. This is because the field of the left hand Metaball overlaps and adds to the field strength.

Furthermore the *neck* between the two Metaball regions is (aesthetically) somewhat less elegantly curved than that of the second image. This is because the rate at which the field declines (over twice the radius) of the left-hand Metaball is quite low and this is the factor that governs the curvature of the surface where Metaballs interact.

In the second image, the left hand Metaball has a Field Strength of +2.0 and a scale of 150% (remember that increasing the Field Strength also has an apparent effect on the size of the surface around the Metaball).

In this case the curvature of the neck is much more pronounced, as a result of the relatively high field gradient.

When scaling Metaballs it is always a good idea to experiment with scaling the Field Strength as well as the actual scale and Field Extent dials (plus the Surface prop's Threshold setting). All of these parameters have direct effects on the curvature and smoothness of the final surface.

Editing the Python Button Script

In order to install Drops permanently in the buttons of the Python Scripts Window, it is necessary to edit the file mainButtons.py in the Runtime\Python\poserScripts of your main Poser installation directory.

Before editing this file, it is important to first make a backup copy.

Open the file with a text editor, the contents should look like this:

```
import poser

poser.DefineScriptButton(1,  ":Runtime:Python:PoserScripts:CreateProps:propButtons.py", "Prop
Samples")
poser.DefineScriptButton(2,  ":Runtime:Python:PoserScripts:GeomMods:geomModButtons.py", "Geom
Mods")
poser.DefineScriptButton(3,  ":Runtime:Python:PoserScripts:Utility:utilityButtons.py", "Utility
Funcs")
poser.DefineScriptButton(4,  ":Runtime:Python:PoserScripts:SampleCallbacks:callbackButtons.py",
"Sample Callbacks")
poser.DefineScriptButton(5,  ":Runtime:Python:PoserScripts:RenderControl:renderButtons.py",
"Render / IO")
poser.DefineScriptButton(6,  ":Runtime:Python:PoserScripts:MaterialMods:materialModButtons.py",
"Material Mods")
poser.DefineScriptButton(7,  ":Runtime:Python:Lights:LightsButtons.py", "Lights")
poser.DefineScriptButton(8,  "", "...")
poser.DefineScriptButton(9,  "", "...")
poser.DefineScriptButton(10, ":Runtime:Python:PoserScripts:clearButtons.py", "Clear Buttons")
```

The main part of the file comprises a list of button entry lines, each starting with:

`poser.DefineScriptButton`.

Following each of these entries in parenthesis is a list of three parameters separated by commas, for example:

`(1,":Runtime:Python:PoserScripts:CreateProps:propButtons.py", "Prop Samples")`

The first of these parameters ('1' in this example) is the button number. The second parameter is the location of the script that will be run when the button is selected, defined as a Macinstosh style path (even on Windows) relative to the Poser root directory . The third parameter is the title of the button that will appear in the Scripts Window.

Locate one of the empty buttons positions such as the one highlighted in red, where the second script path parameter is two adjacent quotation marks (`""`) and the third title parameter is a line of three dots inside quotation marks (`"..."`).

Replace these two parameters with the location of the Drops Python script (`":Runtime:Python:poserScripts:WeirdJuice:Drops.py"`) and whatever title you want to see on the button in the script window (`"Drops"`) respectively.

Do not forget the quotation marks around these entries, and make sure that the commas and parenthesis remain intact.

The file contents should now look like this:

```
import poser

poser.DefineScriptButton(1,  ":Runtime:Python:PoserScripts:CreateProps:propButtons.py", "Prop
Samples")
poser.DefineScriptButton(2,  ":Runtime:Python:PoserScripts:GeomMods:geomModButtons.py", "Geom
Mods")
poser.DefineScriptButton(3,  ":Runtime:Python:PoserScripts:Utility:utilityButtons.py", "Utility
Funcs")
poser.DefineScriptButton(4,  ":Runtime:Python:PoserScripts:SampleCallbacks:callbackButtons.py",
"Sample Callbacks")
poser.DefineScriptButton(5,  ":Runtime:Python:PoserScripts:RenderControl:renderButtons.py",
"Render / IO")
poser.DefineScriptButton(6,  ":Runtime:Python:PoserScripts:MaterialMods:materialModButtons.py",
"Material Mods")
poser.DefineScriptButton(7,  ":Runtime:Python:Lights:LightsButtons.py", "Lights")
poser.DefineScriptButton(8,  ":Runtime:Python:poserScripts:WeirdJuice:Drops.py", "Drops")
poser.DefineScriptButton(9,  "", "...")
poser.DefineScriptButton(10, ":Runtime:Python:PoserScripts:clearButtons.py", "Clear Buttons")
```

Save the file back to its original location and restart Poser to test if the script is OK.

End User License Agreement

WEIRD JUICE SOFTWARE CO. END-USER LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE PRODUCT, IN WHICH THIS EULA IS CONTAINED. IT CONTAINS COMPUTER SOFTWARE (BOTH BINARY AND SCRIPTS) AND MAY INCLUDE DOCUMENTATION, IMAGES, MODELS PRESETS, INITIALISATION DATA AND OTHER COPYRIGHTED ELEMENTS. THE USE OF THE PRODUCT IS LICENSED BY THE WEIRD JUICE SOFTWARE COMPANY, TO ITS CUSTOMERS FOR THEIR USE ONLY AS SET FORTH BELOW. BY INSTALLING, COPYING OR OTHERWISE USING ANY PART OF THIS PRODUCT, YOU AGREE TO BE BOUND BY THE TERMS OF THIS EULA. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE PRODUCT.

GRANT OF LICENSE:

Weird Juice Software Co. (the "Licensor") grants to you this personal, limited, non-exclusive, non-transferable, non-assignable license for the Product solely for use by a single concurrent user only, and solely provided that you adhere to all of the terms and conditions of this Agreement. The foregoing is an express limited use license and not an assignment, sale, or other transfer of the Product or any Intellectual Property Rights of Licensor.

ASSENT:

By opening the file package containing the Product, you agree that this Agreement is a legally binding and valid contract, agree to abide by the intellectual property laws and all of the terms and conditions of this Agreement, and further agree to take all necessary steps to ensure that the terms and conditions of this Agreement are not violated by any person or entity under your control or in your service.

OWNERSHIP:

The Product is owned and copyrighted by the Licensor. The Agreement confers no title or ownership of the product and is not a sale or transfer of any rights in the Product.

ALL INTELLECTUAL PROPERTY RIGHTS IN AND TO THE PRODUCT ARE AND SHALL REMAIN IN LICENSOR.

COPYRIGHT:

All title and copyrights in and to the Product are owned by the Licensor, except for images of Poser(TM), which are owned by Curious Labs, Inc. and egi.sys AG. The Product is protected by copyright laws and international treaty provisions. Therefore you must treat the Product like any other copyrighted material.

RESTRICTIONS:

(a) You are expressly prohibited from copying, modifying, merging, selling, leasing, redistributing, assigning, or transferring in any matter, the Product or any portion thereof.

(b) You may make copies of materials within the package or otherwise related to Product only as required for backup purposes.

(c) You are also expressly prohibited from reverse engineering, decompiling, translating, disassembling, deciphering, decrypting, or otherwise attempting to discover the source code of the Product as the Product contains proprietary material of Licensor. You may not otherwise modify, alter, adapt, port, or merge the Product.

(d) You may not remove, alter, deface, overprint or otherwise obscure Licensor patent, trademark, service mark or copyright notices.

(e) You may not publish or distribute in any form of electronic or printed communication the materials within or otherwise related to Product, including but not limited to the object code, documentation, help files, examples, and benchmarks.

TERM:

This Agreement is effective until terminated. You may terminate this Agreement at any time by uninstalling the Product and destroying all copies of the Product. Upon any termination, you agree to uninstall the Product and return or destroy all copies of the Product, any accompanying documentation, and all other associated materials.

WARRANTIES AND DISCLAIMER:

NO WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE LICENSOR DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT WITH REGARD TO THE PRODUCT, AND THE PROVISION OF AND FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

IN NO EVENT SHALL LICENSOR OR ITS SUPPLIERS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT LICENSOR HAD BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE PRODUCT. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

SEVERABILITY:

In the event any provision of this License Agreement is found to be invalid, illegal or unenforceable, the validity, legality and enforceability of any of the remaining provisions shall not in any way be affected or impaired and a valid, legal and enforceable provision of similar intent and economic impact shall be substituted therefore.

Weird Juice Software Company
Unit GL1A/1B, Buchan House,
Carnegie Campus,
Queensferry Road,
Dunfermline, FIFE
United Kingdom
KY11 5WW
Telephone (44)-1383-739574

19th May 2003.