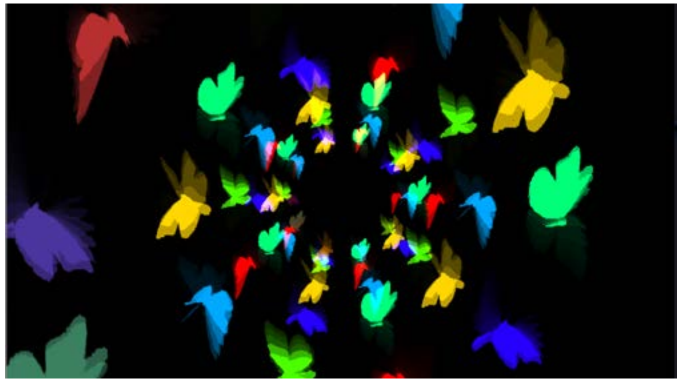
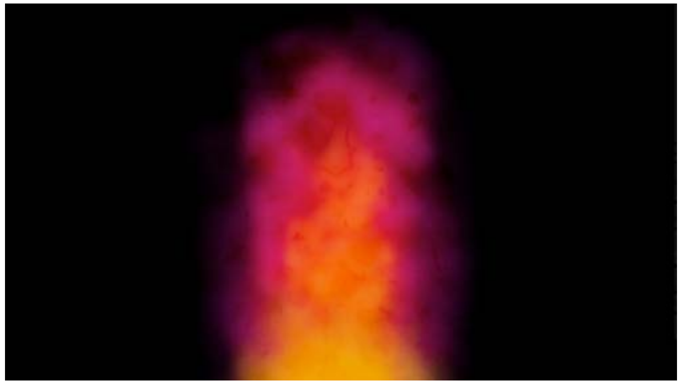
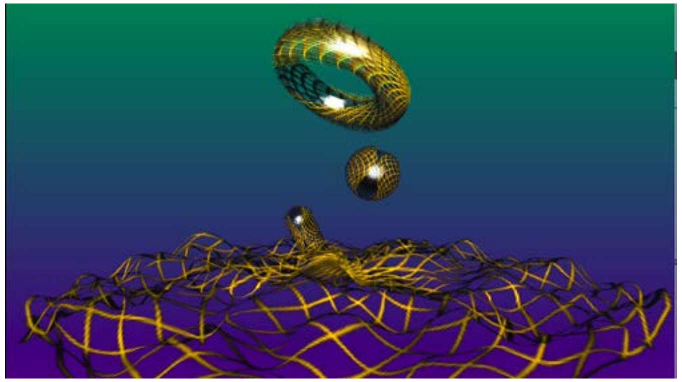
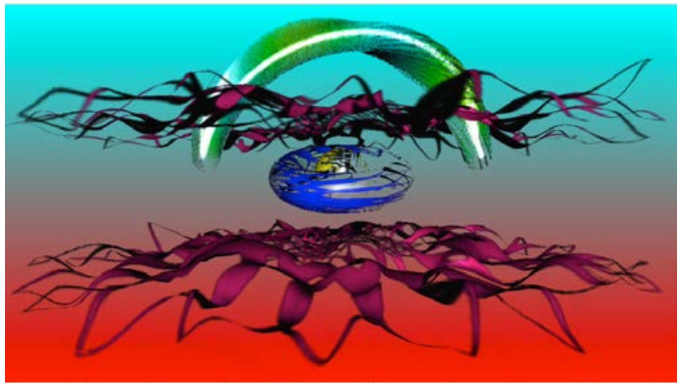
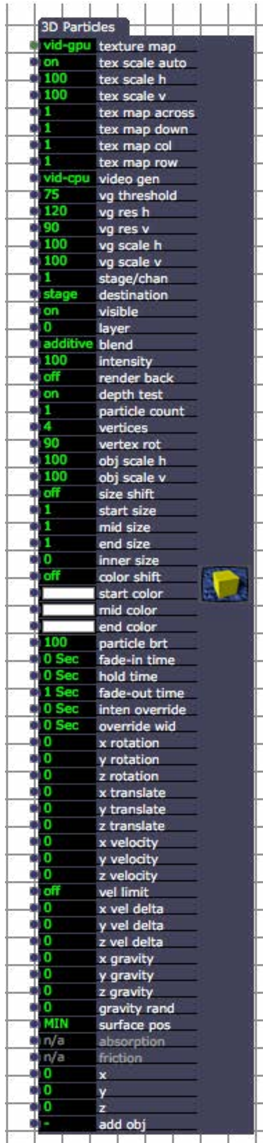
Extending Isadora's 3D Particles and 3D Model Particles

By Bonemap © 2018

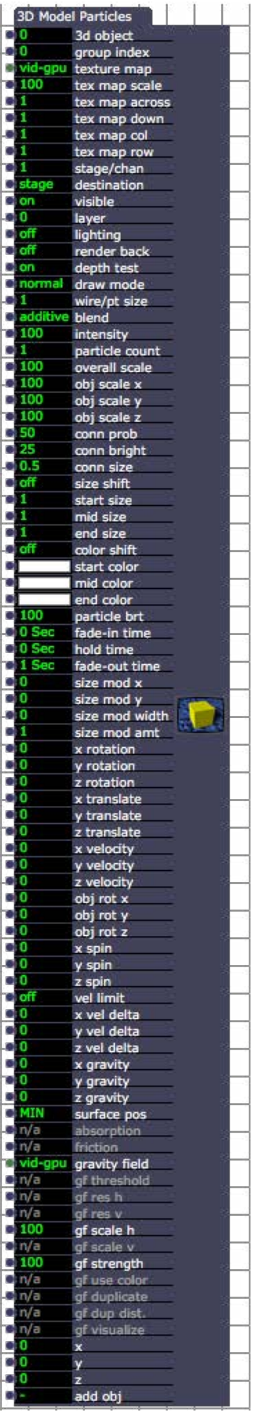





3D Particles

vid-gpu	texture map
on	tex scale auto
100	tex scale h
100	tex scale v
1	tex map across
1	tex map down
1	tex map col
1	tex map row
vid-cpu	video gen
75	vg threshold
120	vg res h
90	vg res v
100	vg scale h
100	vg scale v
1	stage/chan
stage	destination
on	visible
0	layer
additive	blend
100	intensity
off	render back
on	depth test
1	particle count
4	vertices
90	vertex rot
100	obj scale h
100	obj scale v
off	size shift
1	start size
1	mid size
1	end size
0	inner size
off	color shift
⬜	start color
⬜	mid color
⬜	end color
100	particle brt
0 Sec	fade-in time
0 Sec	hold time
1 Sec	fade-out time
0 Sec	inten override
0 Sec	override wid
0	x rotation
0	y rotation
0	z rotation
0	x translate
0	y translate
0	z translate
0	x velocity
0	y velocity
0	z velocity
off	vel limit
0	x vel delta
0	y vel delta
0	z vel delta
0	x gravity
0	y gravity
0	z gravity
0	gravity rand
MIN	surface pos
n/a	absorption
n/a	friction
0	x
0	y
0	z
-	add obj



3D Model Particles

0	3d object
0	group index
vid-gpu	texture map
100	tex map scale
1	tex map across
1	tex map down
1	tex map col
1	tex map row
1	stage/chan
stage	destination
on	visible
0	layer
off	lighting
off	render back
on	depth test
normal	draw mode
1	wire/pt size
additive	blend
100	intensity
1	particle count
100	overall scale
100	obj scale x
100	obj scale y
100	obj scale z
50	conn prob
25	conn bright
0.5	conn size
off	size shift
1	start size
1	mid size
1	end size
off	color shift
⬜	start color
⬜	mid color
⬜	end color
100	particle brt
0 Sec	fade-in time
0 Sec	hold time
1 Sec	fade-out time
0	size mod x
0	size mod y
0	size mod width
1	size mod amt
0	x rotation
0	y rotation
0	z rotation
0	x translate
0	y translate
0	z translate
0	x velocity
0	y velocity
0	z velocity
0	obj rot x
0	obj rot y
0	obj rot z
0	x spin
0	y spin
0	z spin
off	vel limit
0	x vel delta
0	y vel delta
0	z vel delta
0	x gravity
0	y gravity
0	z gravity
MIN	surface pos
n/a	absorption
n/a	friction
vid-gpu	gravity field
n/a	gf threshold
n/a	gf res h
n/a	gf res v
100	gf scale h
n/a	gf scale v
100	gf strength
n/a	gf use color
n/a	gf duplicate
n/a	gf dup dist.
n/a	gf visualize
0	x
0	y
0	z
-	add obj

v.1

Extending Isadora's 3D Particles and 3D Model Particles with media instances

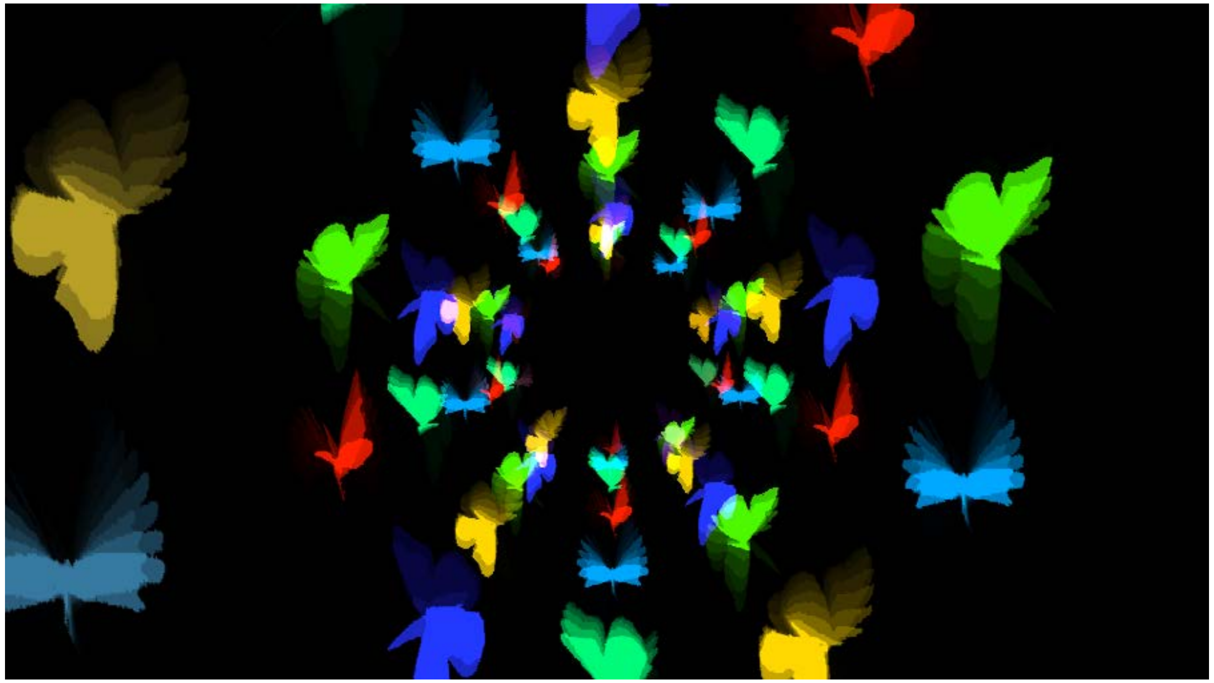


Fig 1. Isadora 3D Particles with custom particle animation. There are no limits to the variations possible.

One of the more difficult tasks for a visually oriented person, like myself, is to write descriptions of the processes that occur at a visual level. In light of this I hope that the following reference for extending 3D Particles will be useful to the Isadora user community.

I engage with Isadora because it provides a visual programming environment at its heart. For the most part, all of the code fragments are neatly packaged into visual nodes with only the variable parameters available for input or modification. These nodes represent the functionality of programming as visual snippets with inputs and outputs that are 'wired' together in a process called 'patching'. In Isadora the 'nodes' are called 'actors' and connection 'wires' are used to link actors within a work area before their visual effects are routed to a 'stage' or video output for display. Isadora patches can do more than just output to a visual screen. The potential for interactivity and connectivity with other devices and physical computing that links Isadora to the analogue world is another compelling function of the software.

Over the last couple of years I have turned to Isadora's particle actors to achieve interactive effects that are responsive to real time input. There are two particle actors that are fundamentally different, but have very similar characteristics and parameters. One is the '3D Particles' and the other is '3D Model Particles'. The simple difference is that 3D Particles use a 2D image source and 3D Model Particles use a 3D model, specifically a .3DS file, as its source. However, both the 3D Particles and the 3D Model Particles have options to assign different source instances to sequential particles making it possible to produce animation effects within the emitted particles themselves. The functionality of animating individual particles makes Isadora's native particle actors very powerful and effective. While these appear as the most complex actors native to Isadora, and therefore the most challenging to learn and master, they are also possibly the most playful and rewarding in terms of the breadth of responsive visual effects that they can achieve.

You can jump in and generate 3D particles with a 'Shapes' actor connected. However, for this tutorial I have pre-prepared the media in other image editing software. There are three types of files that will demonstrate how the 3D particle actors can be extended to achieve a greater range of effects.

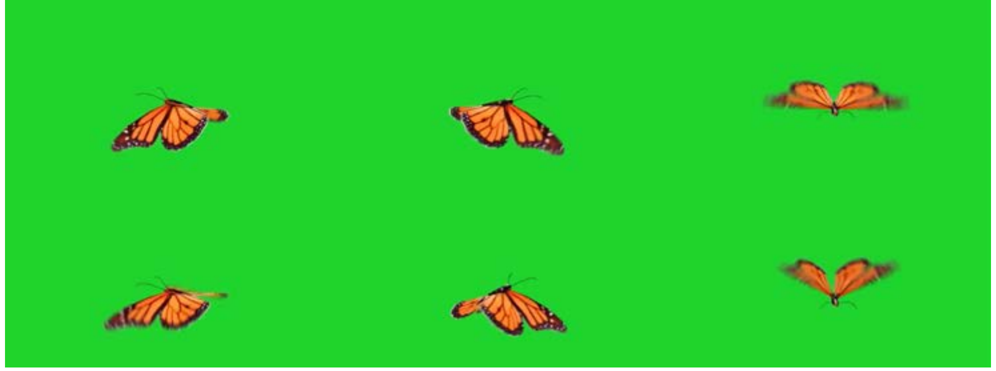


Fig 2, Source One represents a single movie file that has been composited in a grid of six animated butterflies. Each animation has a different flight path orientation that represents the butterfly. However each instance is derived from a single video source and re-composited into continuously looping motion graphics.

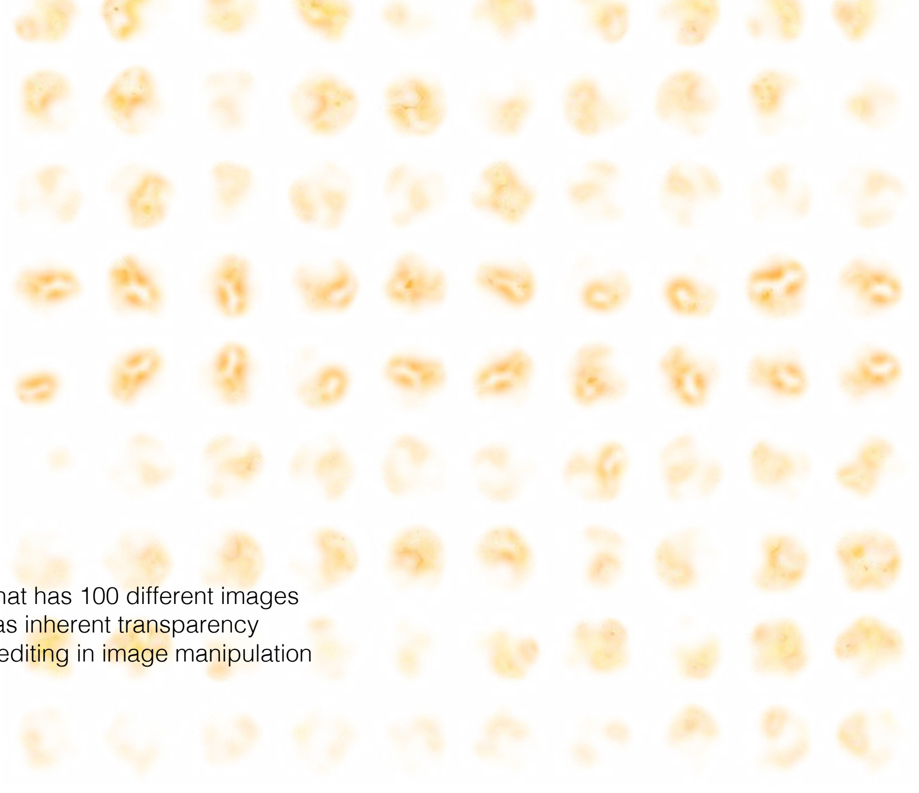


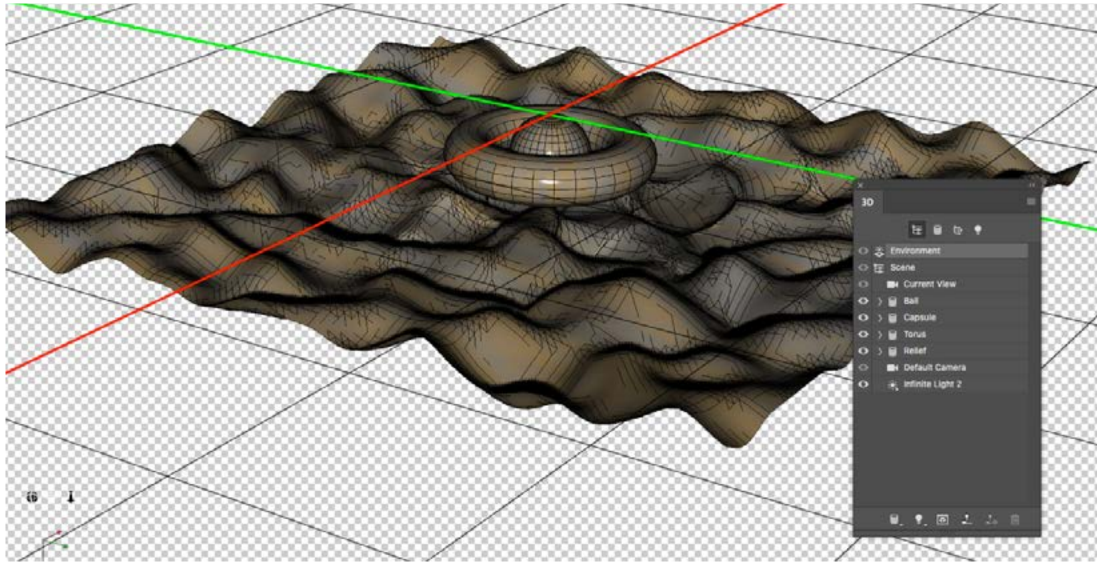
Fig 3. Source Two, a PNG file that has 100 different images within a 10 x 10 grid. The file has inherent transparency incorporated into the file when editing in image manipulation software.

Source One (figure 2) represents a single movie file that has been composited in a grid of six animated sections. The first tutorial (page 4) describes the process of manipulating this file into a particle system using the 3D Particles actor in Isadora.

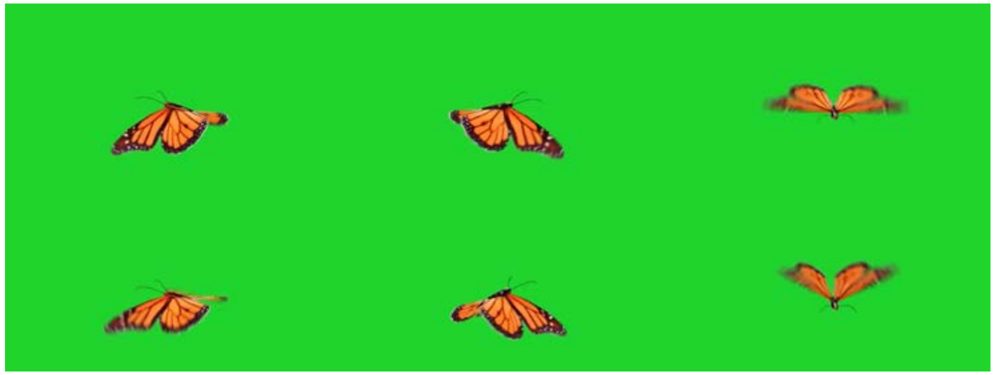
Source Two (figure 3) is a 10 x 10 grid of 2d raster image with an inherent transparency channel. Isadora will recognise the transparency (alpha channel) of the PSD as well as PNG file formats. The Diffusion Cloud tutorial that begins on page 7 outlines how the file is used to create a particle system with the 3D Particles actor.

Source Three (figure 4) represents a 3D file with grouped geometry instances for use with the 3D Model Particles actor. The tutorial for working with the 3D Model Particles actor starts on page 11. Each of these ‘types’ of media sources linked to Isadora’s particle actors are able to be segmented and assigned to individual particle instances within a patch.

Fig 4, Source Three is a 3DS file (below) that has four separate 3D geometry components. The composition and points of origin of the elements in the source file are retained and Isadora will read the geometry and display the mesh as individual components that can be manipulated and modified, in terms of their independent motion within the patch



EXAMPLE ONE: BUTTERFLIES



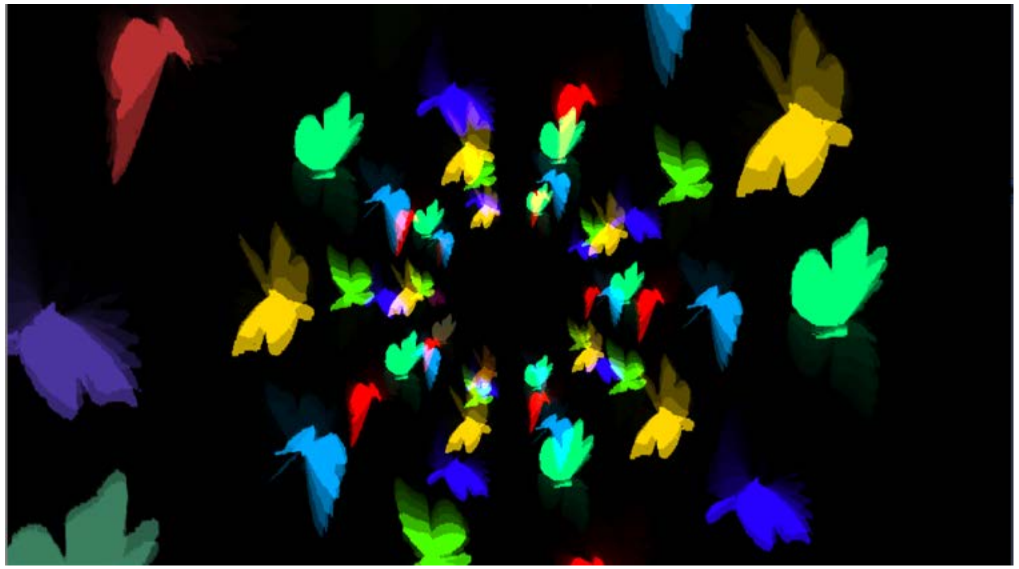


Fig 5 & 6. Before and After :Source One (left): represents a file providing six different temporal iterations of the same animation sequence composited into a single looping movie file using video editing software. This will allow variation in the individual particles emitted by Isadora's 3D Particle actor as demonstrated in the screen capture of the 'Stage' output on the right.

Let's start with Source One, the animated butterflies. First import the video source file into the Media Bin. To construct the particles we will begin by patching in the Scene Editor (Isadora's workspace) using the following actors:

1 x Movie Player,
2 x Counters,
1 x Pulse Generator, and
1 x 3D Particles.

Working from the top of the 3D Particles actor the following connection and parameters are going to be adjusted (figure 7):

Movie Player ‘video out’ -> texture map
‘3’ -> tex map across
‘2’ -> tex map down
Counter (1) ‘output’ -> tex map col
Counter (2) ‘output’ -> tex map row
Pulse Generator ‘trigger 1’ -> ‘add’ Counter (1 & 2)
‘-2.5’ -> ‘z’ (at the bottom of the 3DParticles actor)
Pulse Generator ‘trigger 1’ -> add obj (at the bottom of the 3DParticles actor)

In figure 7, the following parameters of the Counter actors are set to match the number of grid cells associated with the movie file. Note the parameters of the Counters match the ‘tex map across’ and ‘tex map down’ of the 3D Particles actor. The grid of the movie is 3 x 2, therefore the counters are set to ‘wrap’ at 3 and 2 and then linked to the ‘tex map column’ and ‘tex map row’ inputs of the 3D Particles. The Pulse Generator is also linked to the ‘add’ of each Counter and to the ‘add obj’ input at the very bottom of the 3D Particles actor. The ‘z’ depth parameter will require a negative number before the particles will appear on the stage in this case -2.5.

The effects generated by the patch require some refinement and additional actors can be added (figure 8). The green background of the video can be removed using a ‘Chroma Key’ actor. The color calibration can be shifted to the 3D Particles ‘color shift’, ‘start’, ‘mid’ and ‘end’ color parameters but first desaturate the movie with a ‘HSL Adjust’ actor.

The Counter system can be enhanced with a ‘Comparator’ actor to create a switch that more accurately moves along the columns and rows of

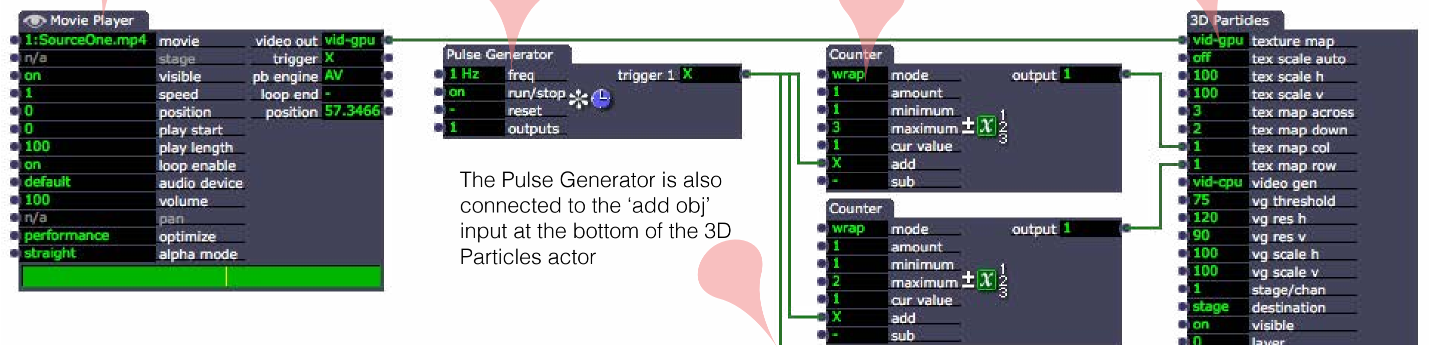
The Movie Player handles the linked source file

The Pulse Generator provides a trigger in hertz (triggers per second). Its purpose in the patch is to initiate each new particle in the system

The Counters are cycling through the coordinates of the grid areas (columns and rows)

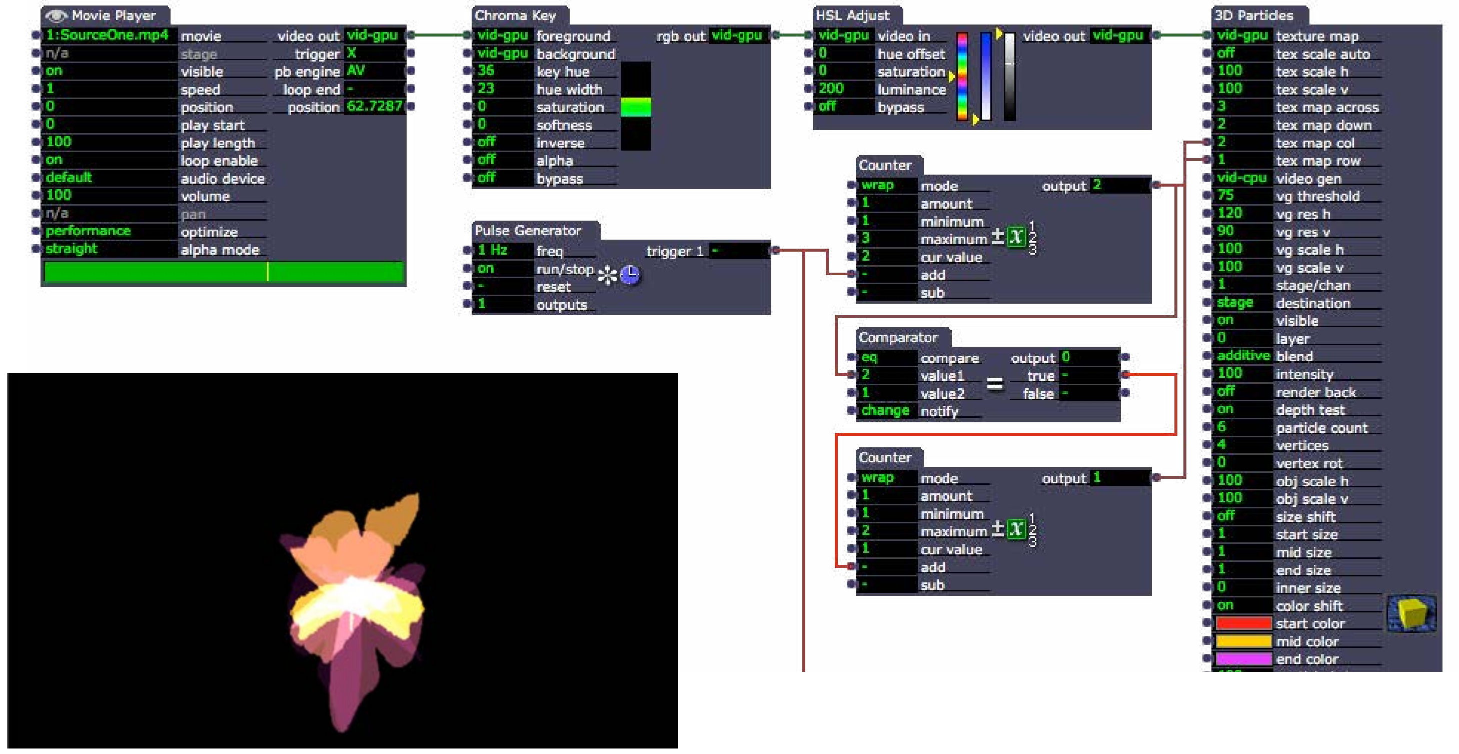
The first 8 parameters of the 3D Particles actor calibrates a gridded image input

The Pulse Generator is also connected to the ‘add obj’ input at the bottom of the 3D Particles actor



```
Movie Player
1:SourceOne.mp4  movie          video out  vid-gpu
n/a              stage          trigger    X
on               visible        pb engine  AV
1                speed          loop end   -
0                position       position   57.3466
0                play start
100              play length
on               loop enable
default          audio device
100              volume
n/a              pan
performance      optimize
straight         alpha mode

Pulse Generator
1 Hz   freq         trigger 1  X
on     run/stop
-      reset
1      outputs

Counter
wrap   mode         output 1
1      amount
1      minimum
3      maximum
1      cur value
X      add
-      sub

Counter
wrap   mode         output 1
1      amount
1      minimum
2      maximum
1      cur value
X      add
-      sub

3D Particles
vid-gpu  texture map
off      tex scale auto
100      tex scale h
100      tex scale v
3        tex map across
2        tex map down
1        tex map col
1        tex map row
vid-cpu  video gen
75       vg threshold
120      vg res h
90       vg res v
100      vg scale h
100      vg scale v
1        stage/chan
stage    destination
on       visible
0        layer
```

Fig 7. The first patching task sets up the counters that will assign a specific part of the movie file area each time the Pulse Generator triggers.



```
Movie Player
1:SourceOne.mp4  movie          video out  vid-gpu
n/a              stage          trigger    X
on               visible        pb engine  AV
1                speed          loop end   -
0                position       position   62.7287
0                play start
100              play length
on               loop enable
default          audio device
100              volume
n/a              pan
performance      optimize
straight         alpha mode

Chroma Key
vid-gpu  foreground     rgb out vid-gpu
vid-gpu  background
36       key hue
23       hue width
0        saturation
0        softness
off      inverse
off      alpha
off      bypass

Pulse Generator
1 Hz   freq         trigger 1  -
on     run/stop
-      reset
1      outputs

HSL Adjust
vid-gpu  video in       video out vid-gpu
0        hue offset
0        saturation
200      luminance
off      bypass

Counter
wrap   mode         output 2
1      amount
1      minimum
3      maximum
2      cur value
-      add
-      sub

Comparator
eq       compare    output 0
2        value1     true   -
1        value2     false  -
change   notify

Counter
wrap   mode         output 1
1      amount
1      minimum
2      maximum
1      cur value
-      add
-      sub

3D Particles
vid-gpu   texture map
off       tex scale auto
100       tex scale h
100       tex scale v
3         tex map across
2         tex map down
2         tex map col
1         tex map row
vid-cpu   video gen
75        vg threshold
120       vg res h
90        vg res v
100       vg scale h
100       vg scale v
1         stage/chan
stage     destination
on        visible
0         layer
additive  blend
100       intensity
off       render back
on        depth test
6         particle count
4         vertices
0         vertex rot
100       obj scale h
100       obj scale v
off       size shift
1         start size
1         mid size
1         end size
0         inner size
on        color shift
🟥        start color
🟨        mid color
🟪        end color
```

Fig 8. Additional actors refine the video input and the way the movie image grid cells are being distributed to the 3D Particles.

the movie's image grid cells.

Now that the foundation of the particle system is in place the spatial qualities of the particle motion can be established by adding inputs to the various x, y and z parameters in the lower half of the 3D Particles actor. And using a combination of Pulse Generator (Hz) values, 'particle count', 'hold time' (life span) and parameters such as 'z gravity' and 'z velocity', different animated effects can be achieved.

Additional effects and modifications can continue to be added to various areas of the patch, for example 'Color Maker HSBA' and 'Motion Blur'. In Figure 9, the Color Maker HSBA is represented with a series of 'Trigger Value' actors and a 'Sequential Trigger' actor all connected to the same Pulse Generator as the rest of the patch. Synchronising all of the changing attributes, to match the input at the 'add obj' trigger, allows individual parameters to be delivered to the particle as it is being generated by the 3D Particles actor. Using actors that perform a cycle such as Wave Generators and Sequential Trigger, allows some control over how different parameters are delivered to individual particles in turn.

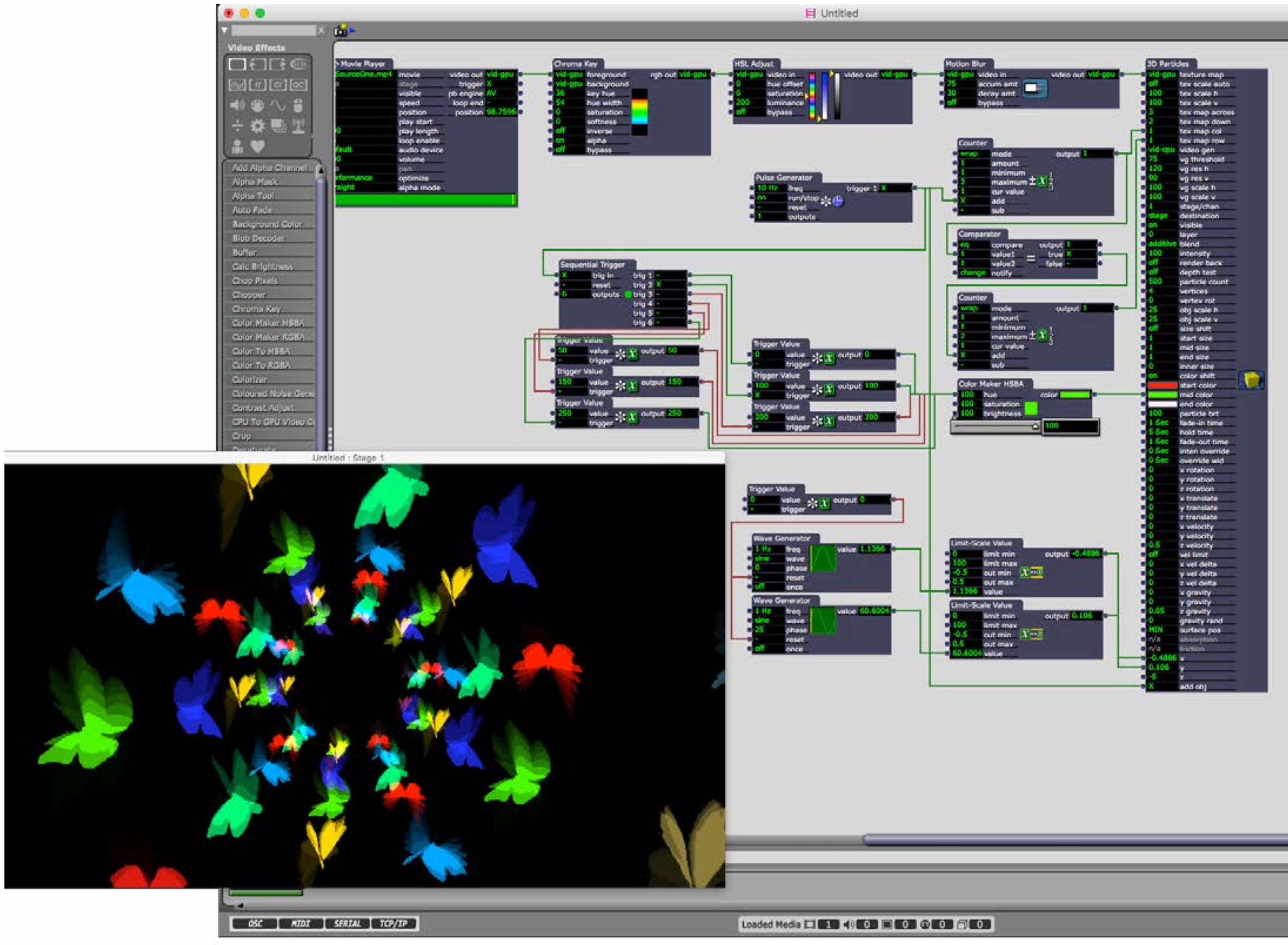


Fig 9. The direction and 'path' that the particles travel can be controlled in many different ways. Here, two Wave Generator actors provide a circular path for the particles by changing the 'x' and 'y' parameters. At the same time 'z gravity and 'z velocity' move the particles along the depth (z) axis giving the effect of the butterflies flying towards the viewer and out of the screen..

EXAMPLE TWO: DIFFUSION CLOUD

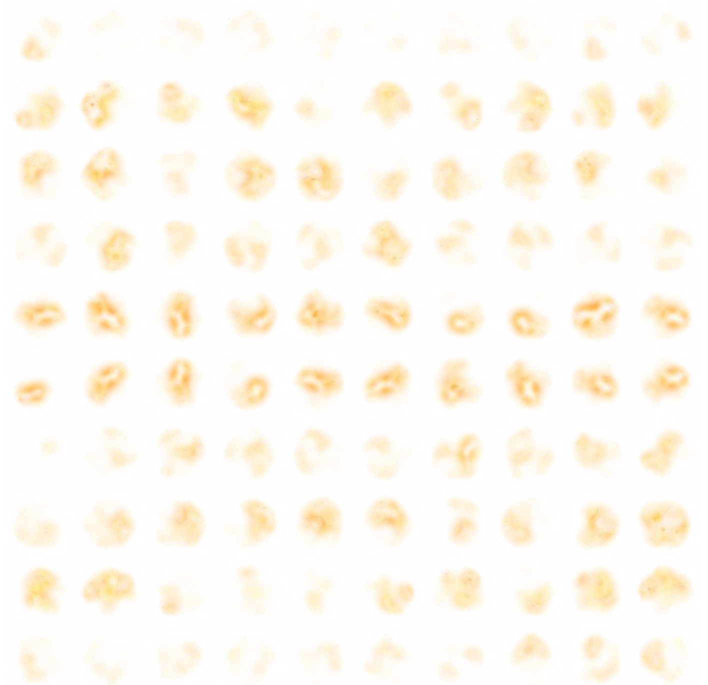
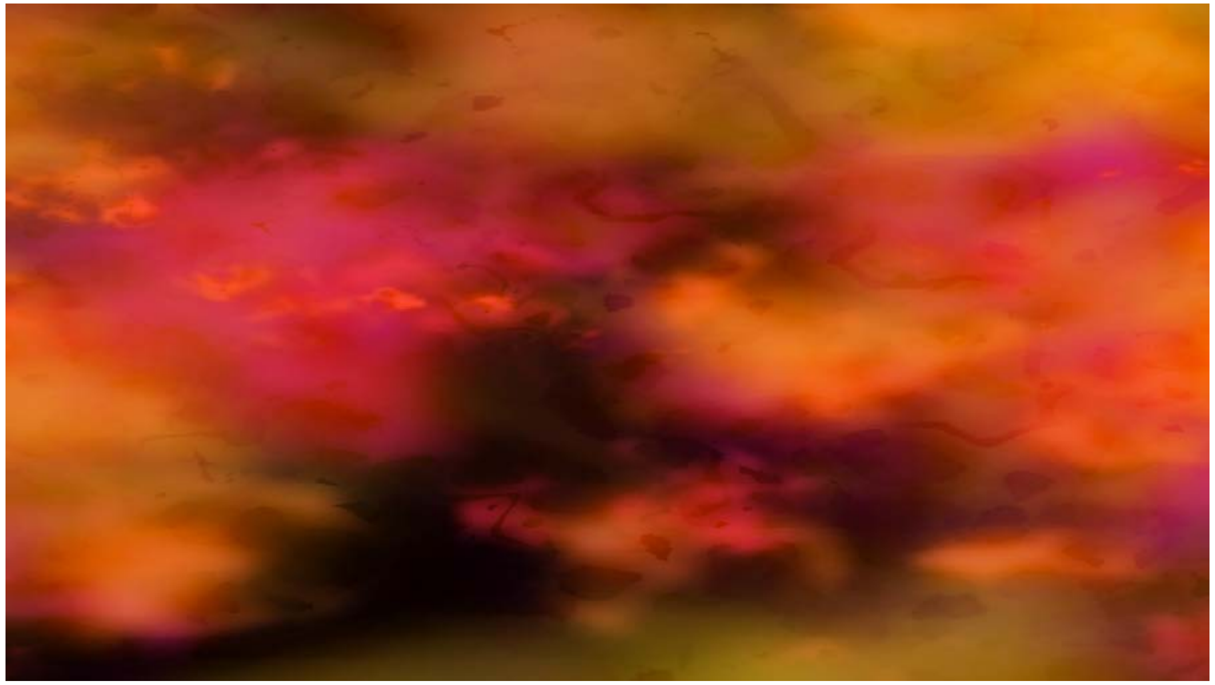



Fig 10 & 11. Before and After. Source Two, a PNG file that has 100 different images within a 10 x 10 grid. The file has inherent transparency around each shape and the overall negative spaces. The size of the grid and the number of unique 2d images that can be incorporated into a particle system can be much larger than this. I have used a sprite sheet with more than 2000 unique icons without any performance issues.

This example, Source Two (figure 10) replicates some aspects of the previous demonstration, but instead uses a PNG file prepared in image editing software. The file consists of a 10 x 10 grid of shapes with varying degrees of transparency and levels of opacity. The intention, when preparing this file, was to create a particle effect with a high percentage of diffusion as the particles overlap creating a smoke or cloud-like simulation. Unlike the movie file, the PNG file does not have inherent animation and the individual image elements are reliant on the parameters of the 3D Particles actor for all motion and movement effects.

The development of this particle system begins by importing the PNG source image and a source music/audio (MP4) file into the Media Bin. To construct the particles we will begin by patching:

1 x Picture Player,
2 x Counters,
1 x Comparator,
1 x Movie Player,
1 x Pulse Generator and
1 x 3D Particles actor

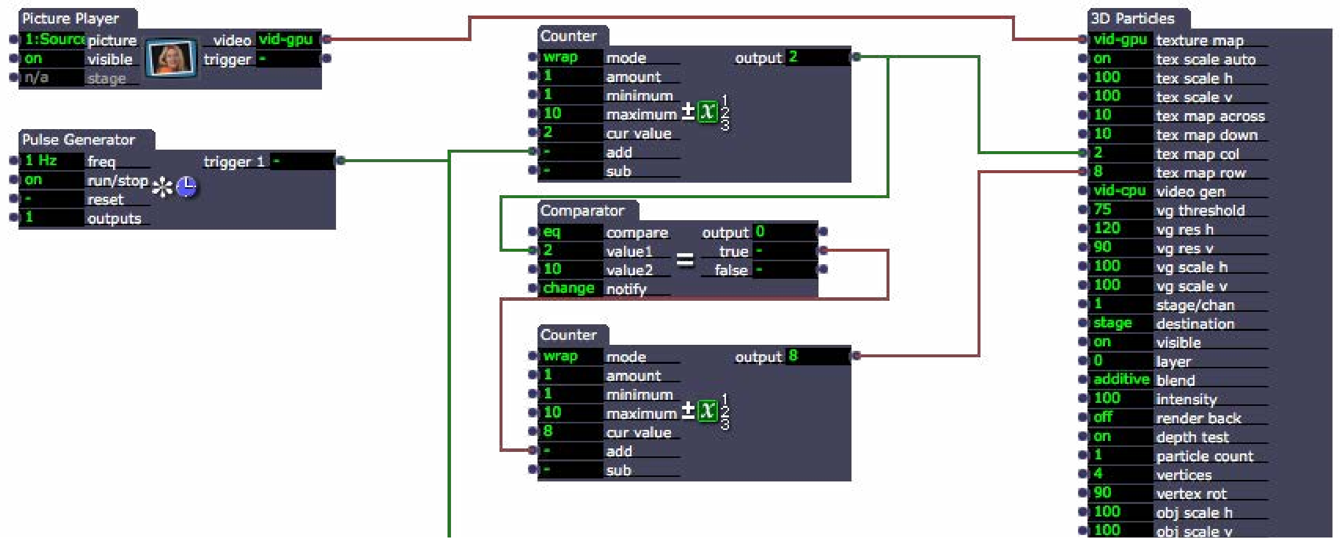
Picture Player ‘video’ -> ‘texture map’ 3D Particles
‘10’ -> ‘tex map across’ 3D Particles
‘10’ -> ‘tex map down’ 3D Particles
Counter (1) output -> ‘tex map col’ 3D Particles
Counter (1) output -> ‘value1’ Comparator
Comparator ‘true’ -> ‘add’ Counter (2)
Counter (2) ‘output’ -> ‘tex map row’ 3D Particles
Pulse Generator ‘trigger1’ -> ‘add’ Counter (1)
‘-2.5’ -> ‘z’ (at the bottom of the 3DParticles actor)
Pulse Generator ‘trigger1’-> ‘add obj’ (at the bottom of the 3DParticles actor)

The schematic is represented in figure 12 on page 8.

There are some standard concepts to consider when entering the parameters for your patch. For example, knowing how many columns and rows there are in your source image media may involve counting the horizontal and vertical grid axis in image preview software. The Pulse Generator triggers individual particles to be emitted from the 3D Particles actor. Higher frequencies produce more particles per second.

This patch uses audio frequency analysis from the Movie Player to drive a series of generative cycles within the composition. This requires preparing the Movie Player actor (figure 13 & 14) to accomplish a ‘freq’ output that can be integrated into the patch

The Movie Player audio track is used in ‘interaction’ mode and the ‘freq bands’ enabled by clicking through the Movie Player tab at the top of the actor. Place a ‘1’ in the freq bands channel of the Movie Player (figure 14). We will use only one audio frequency output to drive the parameters of the particle system.



Picture Player

1:Source	picture	video	vid-gpu
on	visible	trigger	-
n/a	stage		

Pulse Generator

1 Hz	freq	trigger 1	-
on	run/stop		
-	reset		
1	outputs		

Counter

wrap	mode	output	2
1	amount		
1	minimum		
10	maximum		
2	cur value		
-	add		
-	sub		

Comparator

eq	compare	output	0
2	value1	true	-
10	value2	false	-
change	notify		

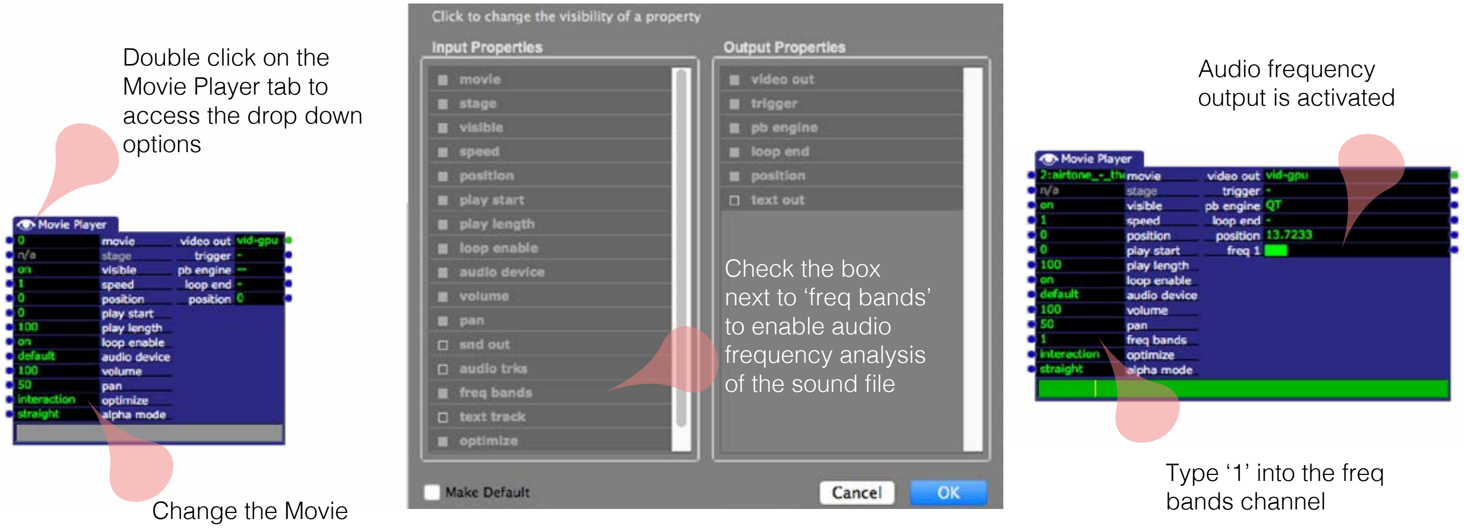
Counter

wrap	mode	output	8
1	amount		
1	minimum		
10	maximum		
8	cur value		
-	add		
-	sub		

3D Particles

vid-gpu	texture map
on	tex scale auto
100	tex scale h
100	tex scale v
10	tex map across
10	tex map down
2	tex map col
8	tex map row
vid-cpu	video gen
75	vg threshold
120	vg res h
90	vg res v
100	vg scale h
100	vg scale v
1	stage/chan
stage	destination
on	visible
0	layer
additive	blend
100	intensity
off	render back
on	depth test
1	particle count
4	vertices
90	vertex rot
100	obj scale h
100	obj scale v

Figure 12. The schematic for connecting the actors to the top half of the 3D Particles actor



Double click on the Movie Player tab to access the drop down options

Change the Movie Player mode to ‘interaction’

Movie Player

0	movie	video out	vid-gpu
n/a	stage	trigger	-
on	visible	pb engine	--
1	speed	loop end	-
0	position	position	0
0	play start		
100	play length		
on	loop enable		
default	audio device		
100	volume		
50	pan		
interaction	optimize		
straight	alpha mode		

Click to change the visibility of a property

Input Properties	Output Properties
■ movie	■ video out
■ stage	■ trigger
■ visible	■ pb engine
■ speed	■ loop end
■ position	■ position
■ play start	☐ text out
■ play length	
■ loop enable	
■ audio device	
■ volume	
■ pan	
☐ snd out	
☐ audio trks	
■ freq bands	
☐ text track	
■ optimize	

☐ Make Default

Cancel OK

Check the box next to ‘freq bands’ to enable audio frequency analysis of the sound file

Audio frequency output is activated

Movie Player

2:airtone_-_th	movie	video out	vid-gpu
n/a	stage	trigger	-
on	visible	pb engine	QT
1	speed	loop end	-
0	position	position	13.7233
0	play start	freq 1	🟩
100	play length		
on	loop enable		
default	audio device		
100	volume		
50	pan		
1	freq bands		
interaction	optimize		
straight	alpha mode		

Type ‘1’ into the freq bands channel

Fig 13 & 14. The Movie Player has built in audio frequency analysis, but it must be ‘turned on’ by placing the actor in ‘interaction’ mode and then accessing the in/out drop-down options by double clicking the actor name tab ‘eye’ symbol

Now we will construct a 'Value Delay Line' chain. It will take the frequency response from the Movie Player audio and spread the output across 10 x instances that will then be assigned to a cycle of particles in turn. To build the chain place 10 x Value Delay Line actors in a column within the work area of your Isadora patch. Link the 'value out' to the 'value' of the actor immediately below until they are all connected. A single output of a 'Absolute Value' actor linked to all of the 'size' inputs creates a value delay chain that can be calibrated easily. The Absolute Value actor can then be renamed to a more descriptive title such as 'delay chain amount' (Figure 15).

The use of 10 actors in the column is a number derived from the Source Two image grid 10 x 10. The intention is to spread out the repetition within the cycle so that there is less chance of visible patterns of repetition appearing in the final video output.

The column of Value Delay Line actors are connected to a 'Selector' actor with 10 inputs. The 'select' input has a connection from the Counter (1) actor that is driving the grid selection at the top of the 3D Particles actor. Counter (1) is already programmed to wrap in a series of 10 linked to the Pulse Generator, therefore its value output is perfect for also linking to the 'select' input of the Selector actor. In turn the Selector 'output' is connected to a Limit-Scale Value actor and then to 'y' input at the bottom of the 3D Particles actor.

The input value going to the first Value Delay Line actor in the stack comes from the Movie Player 'freq 1' output

The input 'select' value is linked from the Counter(1) output that is already cycling through 1 - 10 and synchronised with the Pulse Generator.

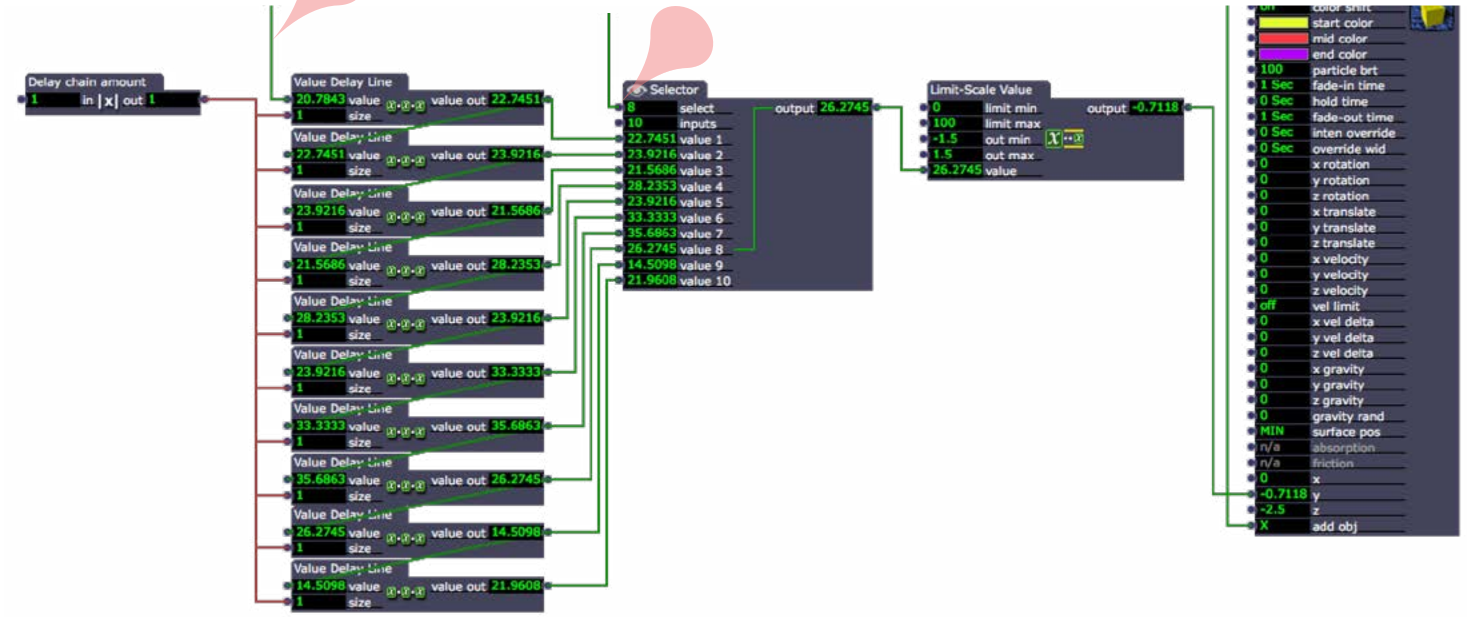
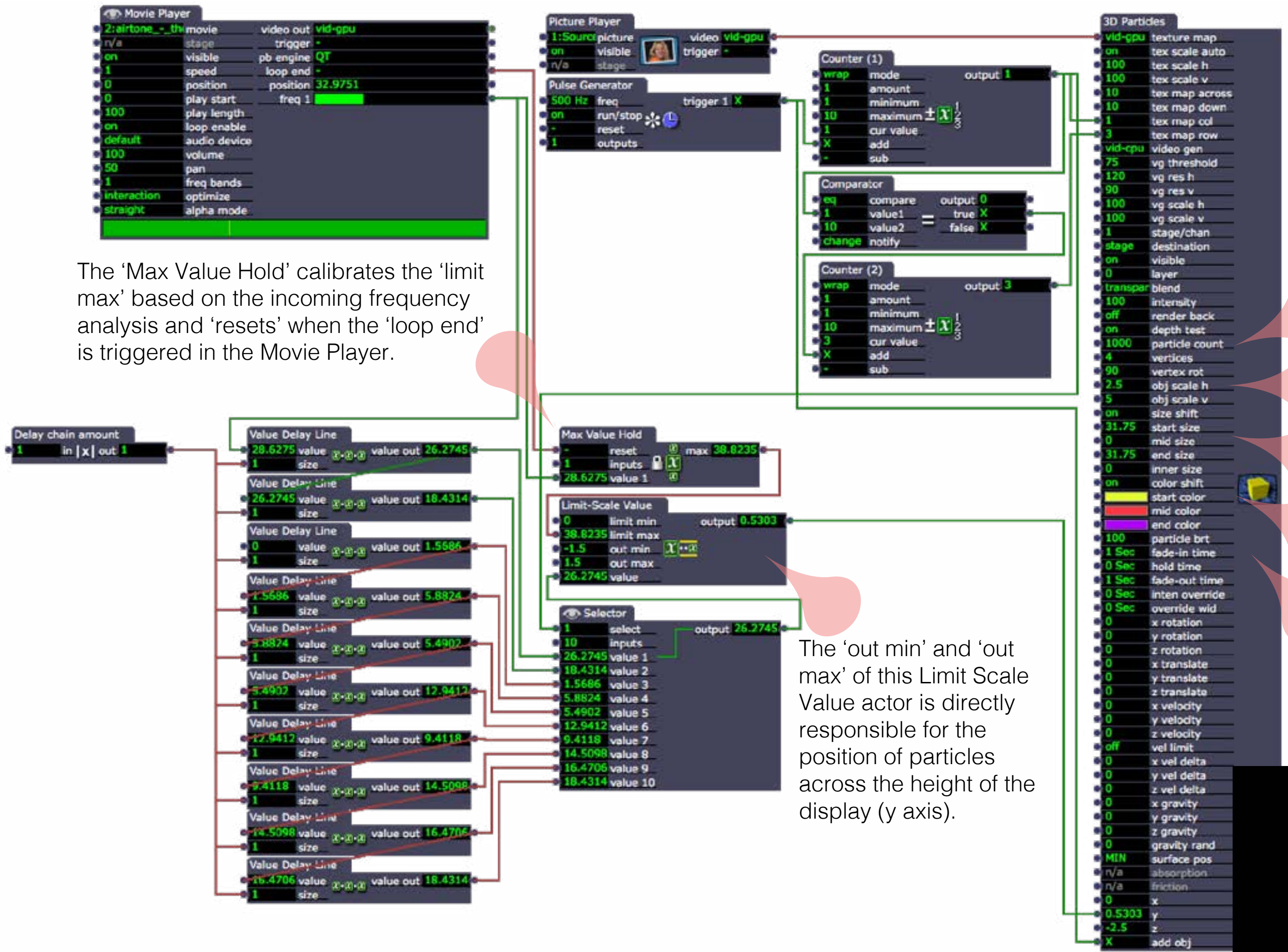


Fig15. The Value Delay Line chain delivers 10 unique numerical representations of the audio frequency analysis that can be used to distribute the particles across the visual display. This kind of complexity in an Isadora patch can be encapsulated in a user actor to reduce the amount of patch clutter.

Note that the calibration of the Limit-Scale Value actor is intended to spread the appearance of the particles along the vertical axis of the stage, in this case the range is '-1.5 , 1.5' and this is dependent on what has been set in the 'z' (depth) input. Generally, a patch like this will require a bit of fine tuning or tweaking in a number of different places to find the balance of variables to create an acceptable particle display. Try changing the z value in the 3D Particles to -5, for example.

There are infinite variations and nuance possible through small incremental changes to the variable parameters of the 3D Particles parameters.



```
Movie Player
2:airtone_-_th  movie          video out  vid-gpu
n/a             stage          trigger    -
on              visible        pb engine  QT
1               speed          loop end   -
0               position       position   32.9751
0               play start     freq 1
100             play length
on              loop enable
default         audio device
100             volume
50              pan
1               freq bands
interaction     optimize
straight        alpha mode

Picture Player
1:Source        picture        video    vid-gpu
on              visible        trigger  -
n/a             stage

Pulse Generator
500 Hz          freq           trigger 1  X
on              run/stop
-               reset
1               outputs

Counter (1)
wrap            mode           output  1
1               amount
1               minimum
10              maximum
1               cur value
X               add
-               sub

Comparator
eq              compare        output  0
1               value1         true    X
10              value2         false   X
change          notify

Counter (2)
wrap            mode           output  3
1               amount
1               minimum
10              maximum
3               cur value
X               add
-               sub

Delay chain amount
1   in |x| out   1

Value Delay Line
28.6275   value   value out  26.2745
1         size
Value Delay Line
26.2745   value   value out  18.4314
1         size
Value Delay Line
0         value   value out  1.5686
1         size
Value Delay Line
1.5686    value   value out  5.8824
1         size
Value Delay Line
5.8824    value   value out  5.4902
1         size
Value Delay Line
5.4902    value   value out  12.9412
1         size
Value Delay Line
12.9412   value   value out  9.4118
1         size
Value Delay Line
9.4118    value   value out  14.5098
1         size
Value Delay Line
14.5098   value   value out  16.4706
1         size
Value Delay Line
16.4706   value   value out  18.4314
1         size

Max Value Hold
-               reset          max  38.8235
1               inputs
28.6275         value 1

Limit-Scale Value
0               limit min      output  0.5303
38.8235         limit max
-1.5            out min
1.5             out max
26.2745         value

Selector
1               select         output  26.2745
10              inputs
26.2745         value 1
18.4314         value 2
1.5686          value 3
5.8824          value 4
5.4902          value 5
12.9412         value 6
9.4118          value 7
14.5098         value 8
16.4706         value 9
18.4314         value 10

3D Particles
vid-gpu         texture map
on              tex scale auto
100             tex scale h
100             tex scale v
10              tex map across
10              tex map down
1               tex map col
3               tex map row
vid-cpu         video gen
75              vg threshold
120             vg res h
90              vg res v
100             vg scale h
100             vg scale v
1               stage/chan
stage           destination
on              visible
0               layer
transpar        blend
100             intensity
off             render back
on              depth test
1000            particle count
4               vertices
90              vertex rot
2.5             obj scale h
5               obj scale v
on              size shift
31.75           start size
0               mid size
31.75           end size
0               inner size
on              color shift
🟨              start color
🟥              mid color
🟪              end color
100             particle brt
1 Sec           fade-in time
0 Sec           hold time
1 Sec           fade-out time
0 Sec           inten override
0 Sec           override wid
0               x rotation
0               y rotation
0               z rotation
0               x translate
0               y translate
0               z translate
0               x velocity
0               y velocity
0               z velocity
off             vel limit
0               x vel delta
0               y vel delta
0               z vel delta
0               x gravity
0               y gravity
0               z gravity
0               gravity rand
MIN             surface pos
n/a             absorption
n/a             friction
0               x
0.5303          y
-2.5            z
X               add obj
```

The 'Max Value Hold' calibrates the 'limit max' based on the incoming frequency analysis and 'resets' when the 'loop end' is triggered in the Movie Player.

The 'out min' and 'out max' of this Limit Scale Value actor is directly responsible for the position of particles across the height of the display (y axis).

The 'particle count' determines how many particles can be active at one time. Once the limit is reached older particles will disappear.

The 'obj scale h' and 'obj scale v' are determining the overall scale and aspect ratio of the particles as they are being emitted.

The 'size shift' parameters determine the changes of a particles size over its lifespan.

The color of each particle can be determined using the 'color shift' parameters. In this case the colors are unchanging.

The 'fade-in time', 'hold-time' and 'fade-out time' determine the lifespan of each particle and dependant on if there are enough in the 'particle count' to allow a complete lifespan to occur.

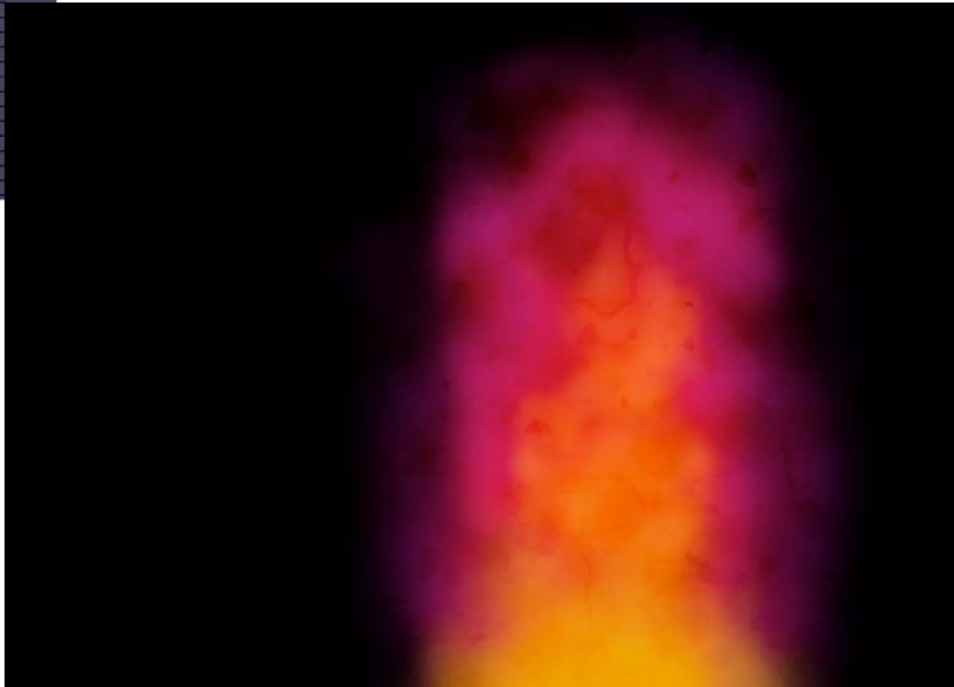
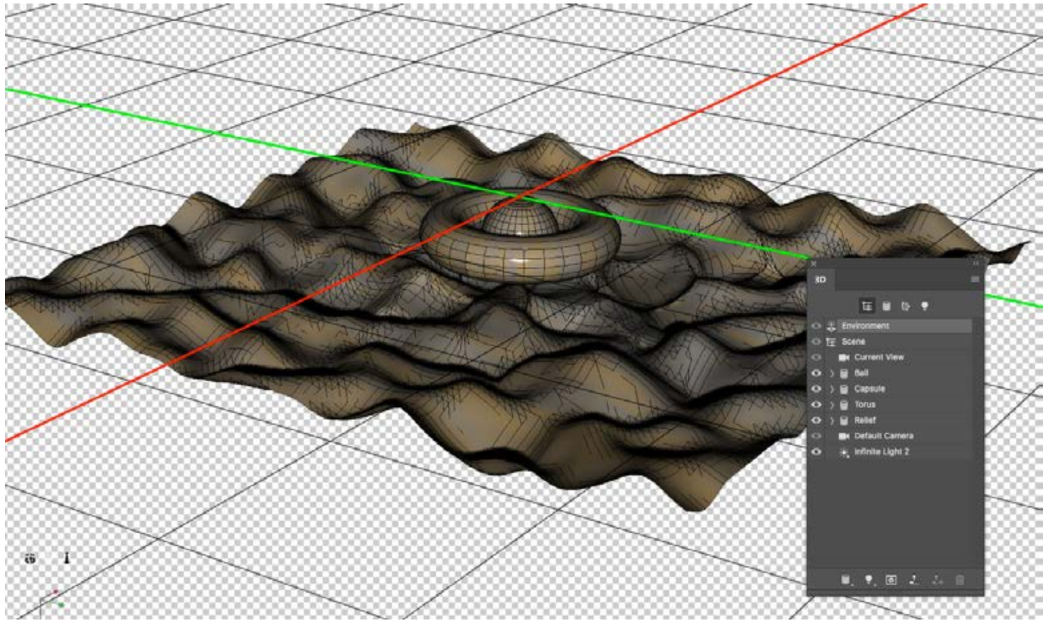


Fig 16. The final inclusions to the particle system are the attenuation of the audio frequency output linked to a 'Max Value Hold' progressively updating the Limit Scale Value and hence the distribution of particles along the 'y' axis through 3D Particles actor.

EXAMPLE THREE: GEOMETRY GROUPS & 3D MODEL PARTICLES



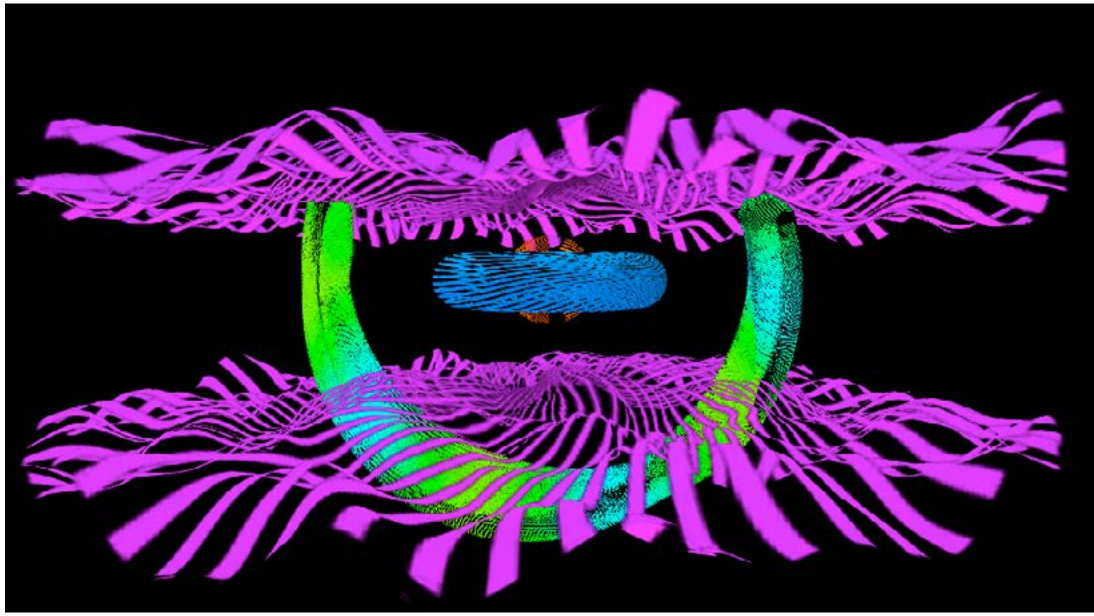


Fig 17 & 18. Before & After: Source Three represents a 3DS file (left) that has four separate 3D geometry components. In the image render on the left, one of the components is completely hidden inside another (there is a capsule shape inside the sphere). Using the 3D Model Particles actor in the Isadora software (right) with similar data delivery techniques used in the previous demonstrations the individual geometry encapsulated in a 3DS file can appear to respond independently.

We take a step away from the 3D Particles actor to explore how the concepts used in the previous demonstrations might work to produce idiosyncratic effects in the 3D Model Particles actor.

There are three media files that have been prepared for use with the 3D Model Particles actor. The first, figure 19, is an image file (.JPG) that is linked to the diffusion texture material applied to the 3D model before the file is exported from the 3D modelling software. This file must be present in the folder of linked media along with the .3DS file or the Isadora software will not function as expected with this demonstration.

Fig 19. A jpeg file used to apply a diffusion material texture to the 3D model and saved along with the .3DS file

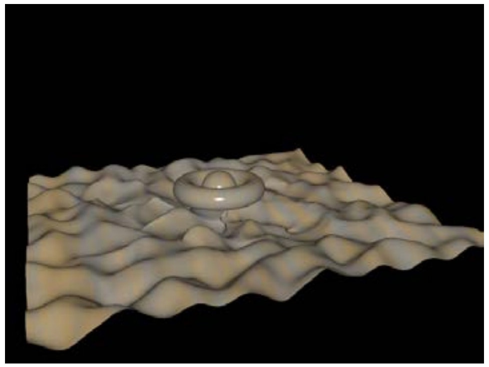


Fig 20. A 3D scene with numerous individual geometry models, in this case four, exported and saved in a single .3DS formatted file

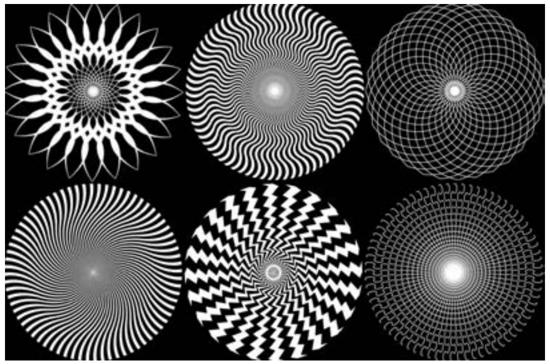


Fig 21. An MP4 movie file with six animated zones, this will be used to replace the 3D models material texture through the 3D Model Particles actor inside the Isadora patch

Persistence of Vision

It appears to me that this patch works because of human 'persistence of vision' and that we are programming the patch to refresh its image much faster than our eyes can perceive, therefore, the elements appear to coalesce into a continuous image.

A Pulse Generator linked to a Counter and the 'group index' of the 3D Model Particles actor is the foundation of this approach to working with 3D models encapsulated in a single 3DS file.

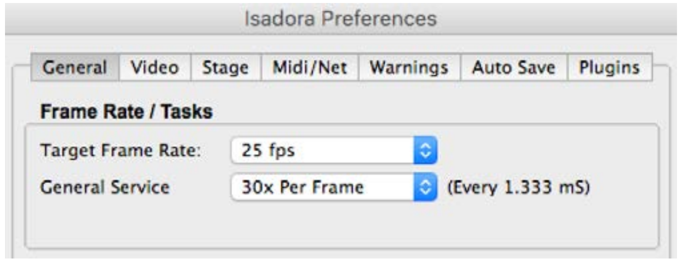
A 'group' in the 3DS file can be an amalgamation of several meshes together and the 'grouping' that is referenced as the 'group index' of 3D Model Particles is determined by third-party 3D modelling software where the 3DS file originated.

The basis for the patch to function is to calculate the number of 3D mesh groups in the 3DS file and multiply by the target frames per second (fps), this will return the nominal refresh rate frequency required for the Pulse Generator.

To test that the patch will display correctly set the Isadora preferences to a moderate frame rate. We will use 25 frames per second for this demonstration. And the task rate to 18 - 25 per frame or higher.

Geometry Doubling

The last part of the demonstration will look at a technique for multiplying the appearance of


Isadora Preferences

General | Video | Stage | Midi/Net | Warnings | Auto Save | Plugins

Frame Rate / Tasks

Target Frame Rate: 25 fps

General Service: 30x Per Frame (Every 1.333 mS)

Fig 22. Isadora Preferences can be changed to the nominal frame rate for the patch. It is recommended to select a moderate fps

the mesh 'groups'. Secondary mesh positions can be programmed into the patch by implementing a 'doubling' of the refresh rate.

To do this, it is first required to double the frequency rate of the Pulse Generator. This will generate the image refresh rate to allow the display of two instances of the model within a single image frame (page 16).

The 3D file associated with this patch has 4 mesh geometries that are in separate groups. The basic formula for this file is:
4 mesh x 25 fps = 100 Hz Frequency refresh rate, therefore, 100 would be entered into the Pulse Generator to run the patch.

However, to create the refresh rate that will allow secondary instances of the 4 meshes the frequency (Hz) will need to be doubled. In theory to have secondary instances of 1 mesh or secondary instances of all 4 meshes it would require the Hz to be doubled. If there was an attempt to have a third instance of a mesh or a third instance of all 4 meshes the basic refresh rate would need to be multiplied by 3 etc.

Groups

This technique involves considering the 3D mesh models within a single file as geometry groups. The models can be complex but the 3DS file format is limited to a mesh constructed of under 65,000 vertices. (I use the open source software 'Meshlab' to effectively simplify complex geometry to fit this limitation: www.meshlab.net). In 3D modelling software shapes can be grouped together to form more complex forms. The 3D Model Particles actor in Isadora segments the different groups of geometry within a single 3DS file.

In this demonstration we are using very simple geometry (Figure 17): a ball, torus, capsule and a 'relief' or deformed plane. However, this technique will work with many more geometry groups within a single 3DS file.

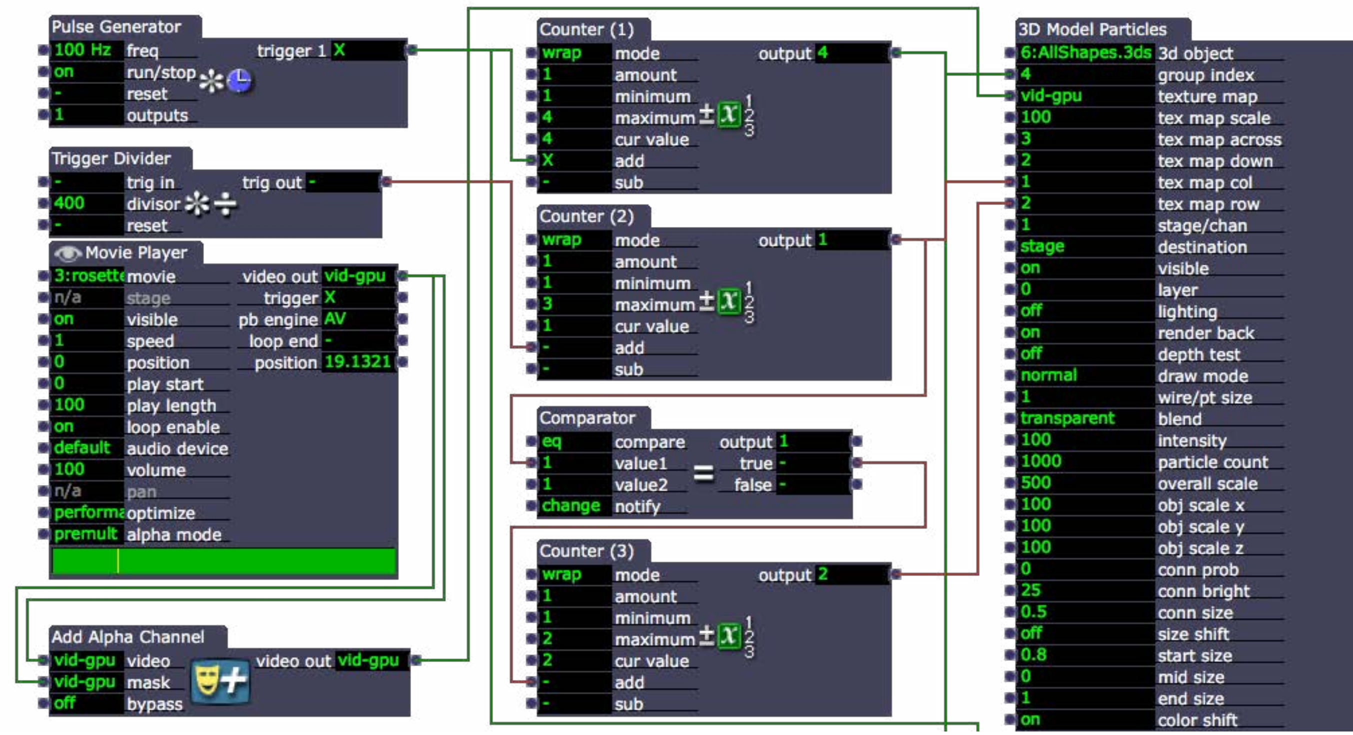
After the media components are prepared, saved and imported into Isadora, the following actors can be placed in the work area (Scene Editor):

1 x 3D Model Particles
1 x Pulse Generator
1 x Movie Player
1 x Add Alpha Channel
3 x Counter
1 x comparator
1 x Trigger Divider

This patch will rapidly cycle through the geometry groups in the.3DS file. And, in addition, cycle through the animated zones associated with the movie file. Therefore, x 3 Counter actors are required to be setup.

The Pulse Generator will drive the first Counter (1) and a ‘Trigger Divider’ will drive the Counter (2). Patch the actor set as indicated in figure 23.

Pulse Generator ‘trigger1’ -> ‘add’ Counter (1)
Counter (1) ‘output’ -> ‘group index’ 3D Model Particles
Pulse Generator ‘trigger1’ -> ‘trig in’ Trigger Divider
Trigger Divider ‘ trig out’ -> ‘add’ Counter (2)
Counter(2) ‘output’ -> ‘value1’ Comparator
Comparator ‘true’ -> ‘add’ Counter (3)
Movie Player ‘video out’ -> ‘video’ Add Alpha Channel
Movie Player ‘video out’ -> ‘mask’ Add Alpha Channel
Add Alpha Channel ‘video out’ ->‘texture map’ 3D Model Particles
Pulse Generator ‘trigger1’ -> ‘add obj’ 3D Model Particles
‘-75’ -> ‘z’ 3D Model Particles



```
Pulse Generator
100 Hz  freq          trigger 1 X
on      run/stop
-       reset
1       outputs

Trigger Divider
-       trig in       trig out -
400     divisor
-       reset

Movie Player
3:rosett movie        video out vid-gpu
n/a     stage         trigger X
on      visible       pb engine AV
1       speed         loop end -
0       position      position 19.1321
0       play start
100     play length
on      loop enable
default audio device
100     volume
n/a     pan
performa optimize
premult alpha mode

Add Alpha Channel
vid-gpu video         video out vid-gpu
vid-gpu mask
off     bypass

Counter (1)
wrap    mode          output 4
1       amount
1       minimum
4       maximum
4       cur value
X       add
-       sub

Counter (2)
wrap    mode          output 1
1       amount
1       minimum
3       maximum
1       cur value
-       add
-       sub

Comparator
eq      compare       output 1
1       value1        true -
1       value2        false -
change  notify

Counter (3)
wrap    mode          output 2
1       amount
1       minimum
2       maximum
2       cur value
-       add
-       sub

3D Model Particles
6:AllShapes.3ds 3d object
4       group index
vid-gpu texture map
100     tex map scale
3       tex map across
2       tex map down
1       tex map col
2       tex map row
1       stage/chan
stage   destination
on      visible
0       layer
off     lighting
on      render back
off     depth test
normal  draw mode
1       wire/pt size
transparent blend
100     intensity
1000    particle count
500     overall scale
100     obj scale x
100     obj scale y
100     obj scale z
0       conn prob
25      conn bright
0.5     conn size
off     size shift
0.8     start size
0       mid size
1       end size
on      color shift
```

Fig 23. The 3DS file is assigned to the 3D Model Particles in the first parameter channel at the very top of the actor. Follow this schematic to patch the top half of the 3D Model Particles to use geometry groups embedded in the 3DS file..

We can begin to calibrate the parameters based on the number of geometry groups and the layout of texture map grid segments. In the example (figure 22), there are four geometry groups that the 3D Model Particles actor can access for display and manipulation. However, to calibrate an acceptable display of the individual geometry groups a number of variables are required to work together so that the frequency of the model switching at the ‘group index’ input is aligned with the particle ‘hold time’ (lifespan) set in the 3D Model Particles actor. In addition ensure there are enough assigned in the ‘particle count’.

Let’s make some calculations based on an intended minimum of 25 frames per second (fps). The first calculation will return the frequency rate of the Pulse Generator, it is the number of geometry groups in the 3DS file multiplied by 25 (the target fps): 4 x 25 = 100. Setting the Pulse Generator at a frequency of 100Hz will return an adequate particle refresh and persistence for each of the geometry groups encapsulated in the Source Three 3DS file used in the patch. Adjusting the parameter at the ‘hold time’ input will provide an indication of how this is going to work. For example ‘0.01’ may produce flashing but ‘0.08’ might present a constant image of the geometry. We will revisit the ‘hold time’ again as it can produce some desirable special effects.

Next, the Counter (1) is driving the display of the geometry through the ‘group index’ input of the 3D Model Particles actor. Therefore Counter (1) will ‘wrap’ with a ‘minimum of 1 and a ‘maximum’ of 4. This creates a particle instance of the four geometry meshes, each in turn.

The Movie Player is outputting a 3 x 2 grid of animated material. Counter (2) and (3) wrap on a ‘minimum’ of 1 and a ‘maximum’ of 3 for Counter (2) and a ‘minimum’ of 1 and a ‘maximum’ of 2 for Counter (3). The output values will update the ‘tex map col’ and ‘tex map row’ with individual sections of the movie image. This connection to the 3d Model Particles actor will replace the jpeg texture associated with the 3D meshes when the file was modelled. However, the image file is required to be in the same directory folder as the 3DS file for Isadora to perform this function.

The ‘Trigger Divider’ determines the rate at which the movie sections appear on the geometry. With the Pulse Generator set at 100Hz then a value of 300 ‘divisor’ in the Trigger Divider will produce a 3 second increment of change to the texture map grid.

Change the parameters of the 3D Model Particles as follows:
‘3’ -> ‘tex map across’ and ‘2’ -> ‘tex map down’. Refer to figure 30 to check that all connections and parameters match.

Now that the top section of the 3D Model Particles system has been set-up we can start to look at the spatial manipulation of the geometry groups.

We will construct a way to deliver a unique coordinate variable each time the particle emitter is displaying one of the 4 group geometries. A nested patch inside a ‘user actor’ will be needed to effectively manage connections in the Scene Editor. (link to User Actor tutorial here)

Place a new ‘User Actor’ in the work area and double-click to open it’s editor window. Place the following into the User Actor (figure 23):

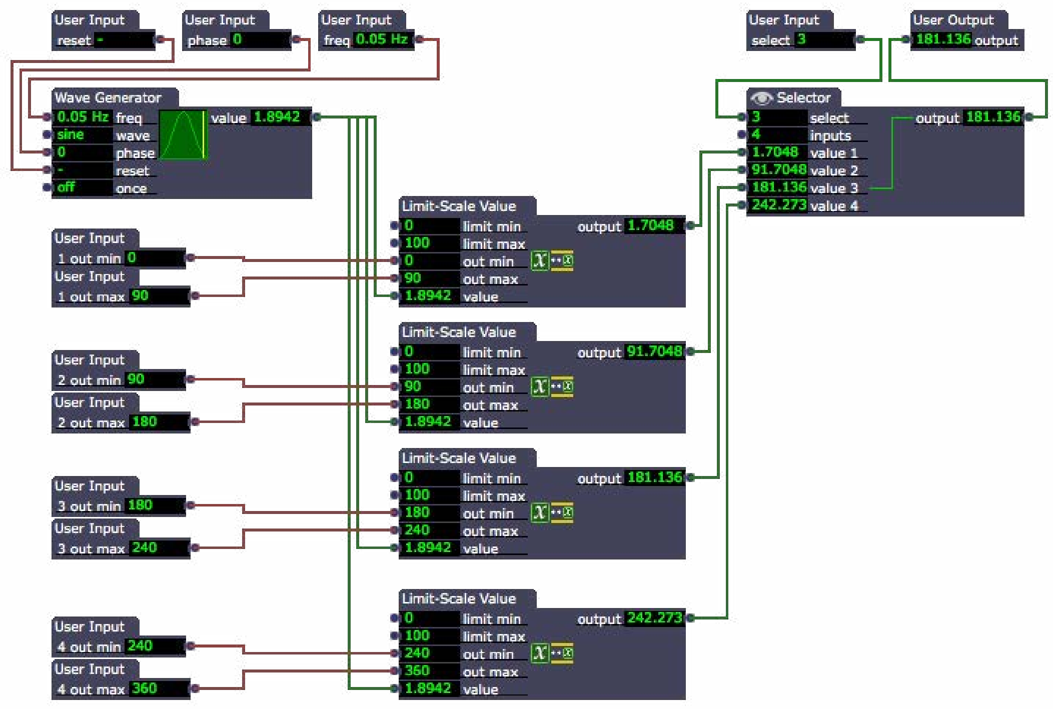
1 x Wave Generator
4 x Limit -Scale Value
1 x Selector
12 x User Input
1 x User Output

Save the User Actor as a ‘Macro’ then rename it, for example: Range Selector (figure24).

Duplicate (copy/paste) the custom user generated actor and position 2 in the Scene Editor next to the 3d Model Particles and under the Counter actors.

Create a reset actor by renaming a ‘Trigger Value’ then connect its ‘output’ to all the ‘reset’ inputs of the 2 x user actors. It is now time to start providing the parameters for each numbered range as a minimum and maximum and connect it to the variables of the 3D Model Particles.

The sets of ranges associated with the user actor, 1 - 4, (figure 25) will correspond to the same number and sequence of geometry groups being assigned to the particle system by the Counter (1) actor. You should expect the geometry order that appeared as a list in the 3D modelling software to be the


User Input: reset -
User Input: phase 0
User Input: freq 0.05 Hz
User Input: select 3
User Output: 181.136 output

Wave Generator: 0.05 Hz freq, sine wave, 0 phase, - reset, off once; value 1.8942

Selector: 3 select, 4 inputs, 1.7048 value 1, 91.7048 value 2, 181.136 value 3, 242.273 value 4; output 181.136

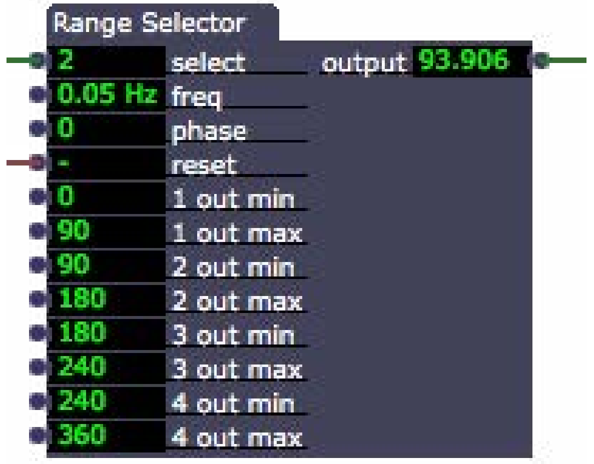
User Input: 1 out min 0
User Input: 1 out max 90
Limit-Scale Value: 0 limit min, 100 limit max, 0 out min, 90 out max, 1.8942 value; output 1.7048

User Input: 2 out min 90
User Input: 2 out max 180
Limit-Scale Value: 0 limit min, 100 limit max, 90 out min, 180 out max, 1.8942 value; output 91.7048

User Input: 3 out min 180
User Input: 3 out max 240
Limit-Scale Value: 0 limit min, 100 limit max, 180 out min, 240 out max, 1.8942 value; output 181.136

User Input: 4 out min 240
User Input: 4 out max 360
Limit-Scale Value: 0 limit min, 100 limit max, 240 out min, 360 out max, 1.8942 value; output 242.273

Fig 24, A user actor is a nested scene editor within a node in Isadora. They assist with the management of complex networks within a patch


Range Selector

2	select
0.05 Hz	freq
0	phase
-	reset
0	1 out min
90	1 out max
90	2 out min
180	2 out max
180	3 out min
240	3 out max
240	4 out min
360	4 out max

output 93.906

Fig 25, The User Actor is transformed into a ‘Macro’ that can be duplicated to apply to different variable parameters in the patch same order in Isadora. Connect the output of Counter (1) to the 'select' input of the two Range Selector user actors.

Place a 'Color Maker HSBA' near to the 'output' of the user actor closest to the 'color shift' parameters of the 3D Model Particles (figure 26). Link the 'output' to the 'mid color' with a connection wire. Try the following settings in the user actor:

'0' -> '1 out min'
'90' -> '1 out max'

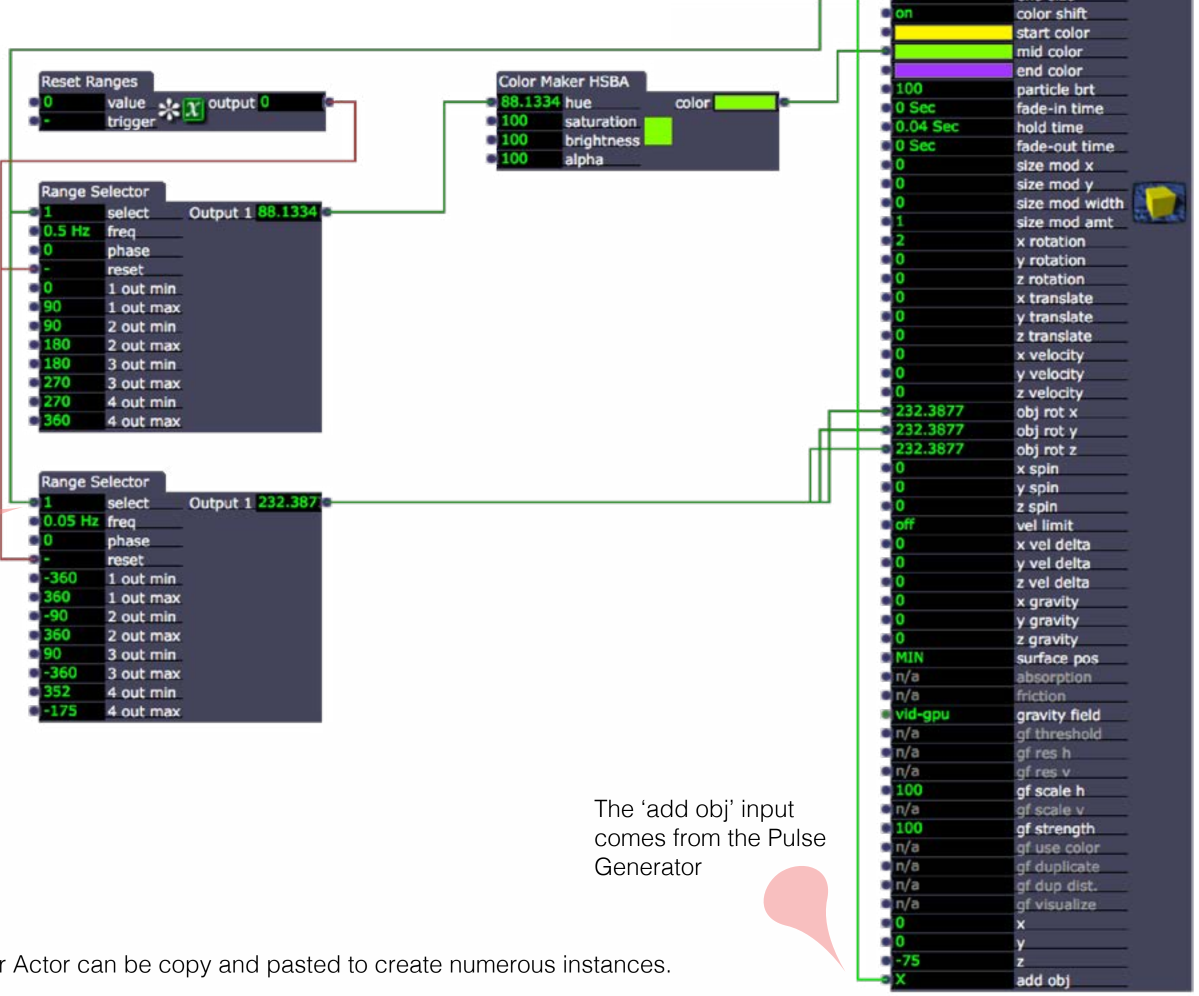
'90' -> '2 out min'
'180' -> '2 out max'

'180' -> '3 out min'
'240' -> '3 out max'

'240' -> '4 out min'
'360' -> '4 out max'

If the patch is operating correctly the geometry will each have its own colour. If the colour is not independent or there is flashing try adjusting the parameter settings based on the indicated settings in figure 26.

The movie with the six spinning rosettes is intended to display with transparency. Although transparency is not inherently encoded in the file it is possible to use the image itself to effect an appropriate transparency channel (this works because the content of the video file is black and white).



```
Reset Ranges
0   value     output 0
-   trigger

Color Maker HSBA
88.1334  hue          color
100      saturation
100      brightness
100      alpha

Range Selector
1        select   Output 1 88.1334
0.5 Hz   freq
0        phase
-        reset
0        1 out min
90       1 out max
90       2 out min
180      2 out max
180      3 out min
270      3 out max
270      4 out min
360      4 out max

Range Selector
1        select   Output 1 232.387
0.05 Hz  freq
0        phase
-        reset
-360     1 out min
360      1 out max
-90      2 out min
360      2 out max
90       3 out min
-360     3 out max
352      4 out min
-175     4 out max

on        color shift
          start color
          mid color
          end color
100       particle brt
0 Sec     fade-in time
0.04 Sec  hold time
0 Sec     fade-out time
0         size mod x
0         size mod y
0         size mod width
1         size mod amt
2         x rotation
0         y rotation
0         z rotation
0         x translate
0         y translate
0         z translate
0         x velocity
0         y velocity
0         z velocity
232.3877  obj rot x
232.3877  obj rot y
232.3877  obj rot z
0         x spin
0         y spin
0         z spin
off       vel limit
0         x vel delta
0         y vel delta
0         z vel delta
0         x gravity
0         y gravity
0         z gravity
MIN       surface pos
n/a       absorption
n/a       friction
vid-gpu   gravity field
n/a       gf threshold
n/a       gf res h
n/a       gf res v
100       gf scale h
n/a       gf scale v
100       gf strength
n/a       gf use color
n/a       gf duplicate
n/a       gf dup dist.
n/a       gf visualize
0         x
0         y
-75       z
X         add obj
```

The 'select' inputs come from the Counter (1)

The 'add obj' input comes from the Pulse Generator

Fig 26, The User Actor can be copy and pasted to create numerous instances.

The 'Add Alpha Channel' actor in the Scene Editor takes a connection from the Movie Player 'video output' to both the 'video' and 'mask' inputs. Make sure the 'video out' of the Add Alpha Channel is connected to the 'texture map' of the 3D Model Particles. Transparency will now replace the darker areas of the video.

The next Range Selector 'output1' will connect to the 'obj rot x', and 'obj rot y' of the 3D Model Particles to spin each of the mesh groups individually. Use the following settings in the second user actor:

'-360' -> '1 out min'
'360' -> '1 out max'

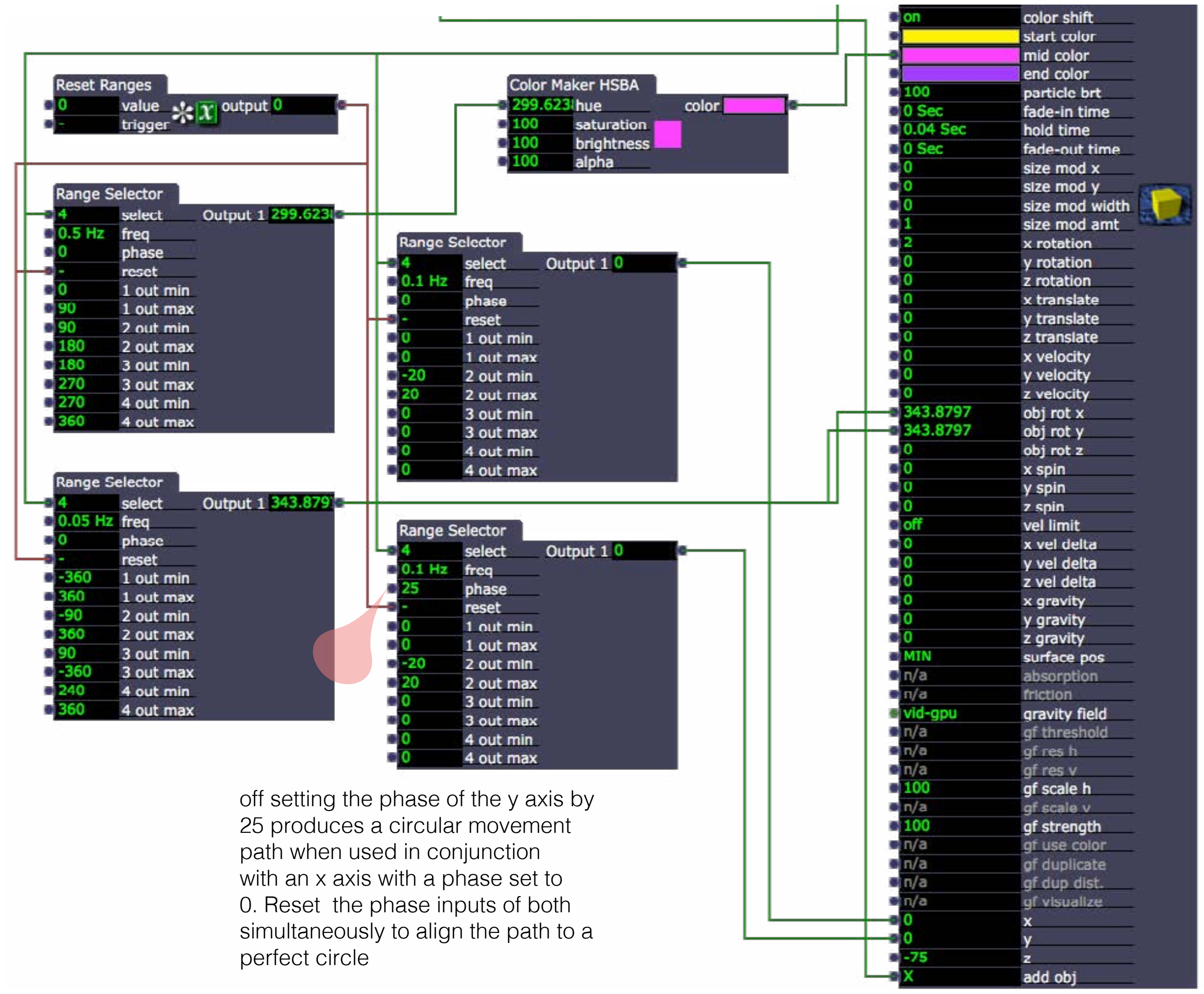
'-90' -> '2 out min'
'360' -> '2 out max'

'90' -> '3 out min'
'-360' -> '3 out max'

'360' -> '4 out min'
'-180' -> '4 out max'

Preview the results in the Isadora stage.

More Range Selector user actors can be added to the patch to create additional effects and movement. Figure 27 indicates an addition of x 2 user actors that provide a circular path for one of the mesh groups. One Range Selector is connected to the ‘x’ input and the other to the ‘y’ input of the 3D Model Particles actor. The second user actor, connected to the y input has a phase offset of 25.


Reset Ranges
0 value
- trigger
output 0

Range Selector
4 select
0.5 Hz freq
0 phase
- reset
0 1 out min
90 1 out max
90 2 out min
180 2 out max
180 3 out min
270 3 out max
270 4 out min
360 4 out max
Output 1 299.623

Range Selector
4 select
0.05 Hz freq
0 phase
- reset
-360 1 out min
360 1 out max
-90 2 out min
360 2 out max
90 3 out min
-360 3 out max
240 4 out min
360 4 out max
Output 1 343.879

Color Maker HSBA
299.623 hue
100 saturation
100 brightness
100 alpha
color

Range Selector
4 select
0.1 Hz freq
0 phase
- reset
0 1 out min
0 1 out max
-20 2 out min
20 2 out max
0 3 out min
0 3 out max
0 4 out min
0 4 out max
Output 1 0

Range Selector
4 select
0.1 Hz freq
25 phase
- reset
0 1 out min
0 1 out max
-20 2 out min
20 2 out max
0 3 out min
0 3 out max
0 4 out min
0 4 out max
Output 1 0

on color shift
🟨 start color
🟪 mid color
🟪 end color
100 particle brt
0 Sec fade-in time
0.04 Sec hold time
0 Sec fade-out time
0 size mod x
0 size mod y
0 size mod width
1 size mod amt
2 x rotation
0 y rotation
0 z rotation
0 x translate
0 y translate
0 z translate
0 x velocity
0 y velocity
0 z velocity
343.8797 obj rot x
343.8797 obj rot y
0 obj rot z
0 x spin
0 y spin
0 z spin
off vel limit
0 x vel delta
0 y vel delta
0 z vel delta
0 x gravity
0 y gravity
0 z gravity
MIN surface pos
n/a absorption
n/a friction
vid-gpu gravity field
n/a gf threshold
n/a gf res h
n/a gf res v
100 gf scale h
n/a gf scale v
100 gf strength
n/a gf use color
n/a gf duplicate
n/a gf dup dist.
n/a gf visualize
0 x
0 y
-75 z
X add obj

off setting the phase of the y axis by 25 produces a circular movement path when used in conjunction with an x axis with a phase set to 0. Reset the phase inputs of both simultaneously to align the path to a perfect circle

Fig. 27 Additional Range Selector user actors used to create a motion path for one or more of the mesh groups.

Mesh Doubling

Multiplying the visual instance of a mesh group requires refreshing the frequency rate of the particles at a high enough speed to allow the appearance of multiple instances of the same mesh. There are x4 mesh groups encoded in the 3DS file used in this demonstration. Therefore a basic frequency of the Pulse Generator will be 100 Hz at 25 fps (25 x 4 = 100). In the case where we want to manipulate another instance of one of these four mesh groups, the frequency rate is required to increase to allow the instance to occur without disrupting the recurrence of the existing mesh.

What we will be doing is doubling the refresh frequency of all the geometry instances, since they are linked by the same Pulse Generator, so that at least one of the mesh groups can occupy two positions in the image at the required refresh rate for ‘persistence of vision’ (the perception of continuous animated movement).

Increase the Pulse Generator to 200 Hz to start this process. Next a new and different User Actor will be required to switch the parameters of the existing Range Selector user actors created previously.

The new User Actor will have the following included:

1 x Sequential Trigger
4 x Trigger Value
5 x User Input
4 x User Output

Figure 28 provides the schematic required for linking the User Actor.

Save the new User Actor as a macro and rename it to Value Selector.

We will use three of the new Value Selector user actors to complete the mesh doubling process. One for each set of parameters associated the two duplicate mesh groups and the third to combine the parameters and send them to one of the existing Range Selectors already in the patch.

We will also use a Trigger Divider actor to reduce the frequency rate of the Pulse Generator by a divisor of 2 and connect to the first two Value Selectors (1 & 2).

Figure 29 indicates the 'wiring' of the three Value Selector user actors. The outputs of the third Value Selector (3) are used to divide the position data and effectively create a second instance of a mesh group, dependent on the setting of the parameters in the Value Selector (1 & 2).

The full schematic of the patch (figure 30) demonstrates mesh doubling of the fourth group in the 3DS file (the deformed plane). The Value Selector (1 & 2) parameters allow the independent animation of two instances of the mesh.

There is an endless number of ways to modify the patch to create unique and interesting results. I am just about done here, It is over to your imagination now.

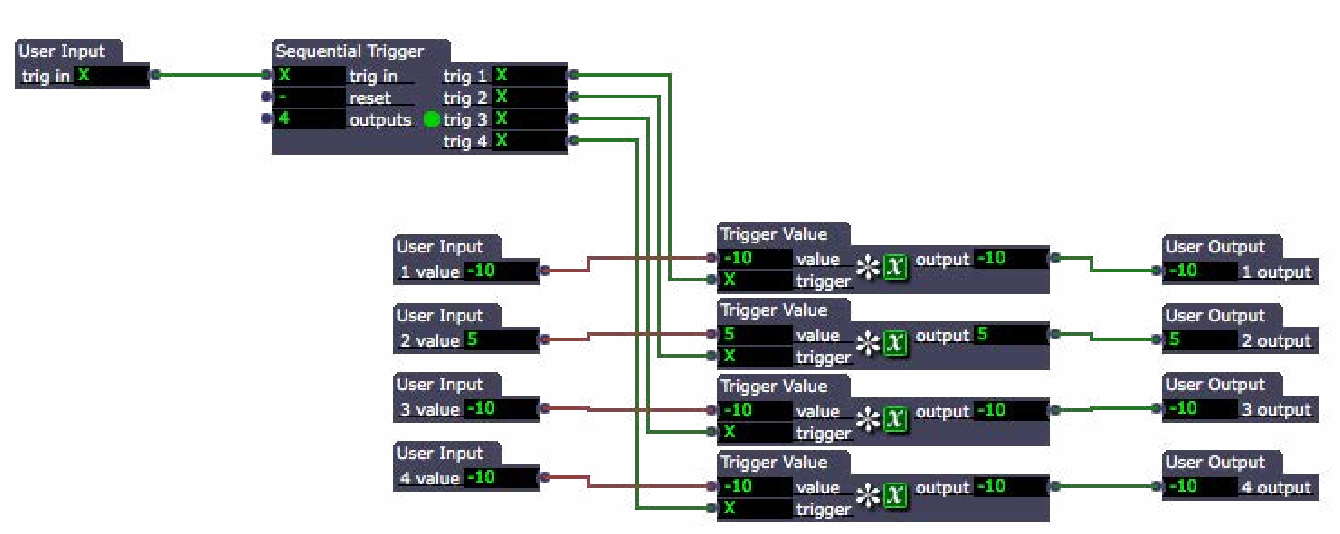
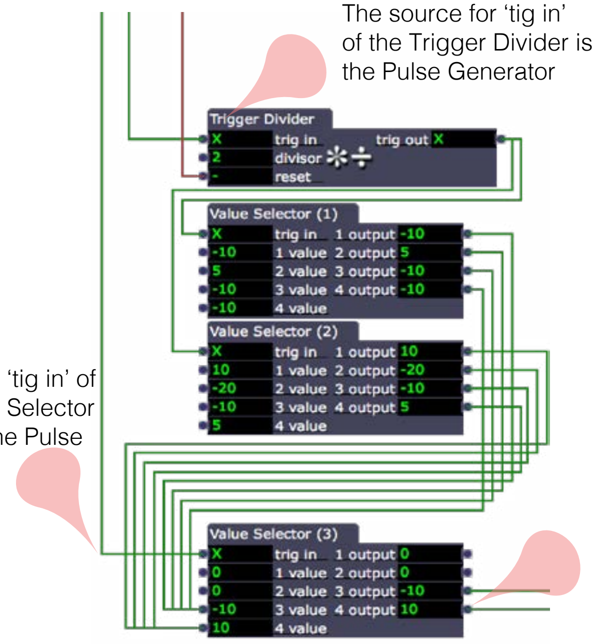


Fig. 28 The nested node structure of the second user actor used in the demonstration.

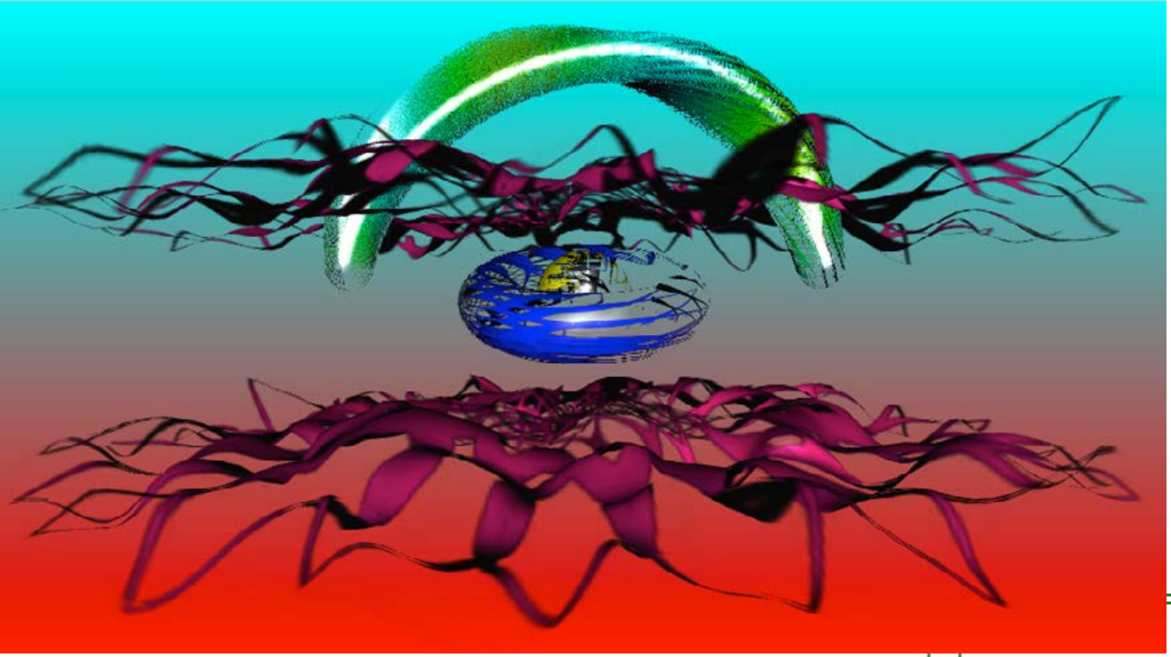
The source for 'tig in' of the Trigger Divider is the Pulse Generator

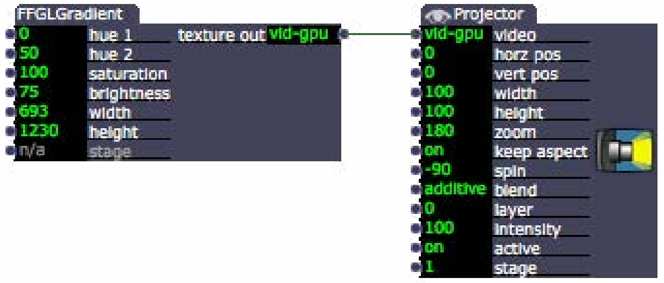


The source for 'tig in' of the third Value Selector user actor is the Pulse Generator

The active outputs are connected to a set of 'out min' & 'out max' of the Range Selector user actor constructed previously.

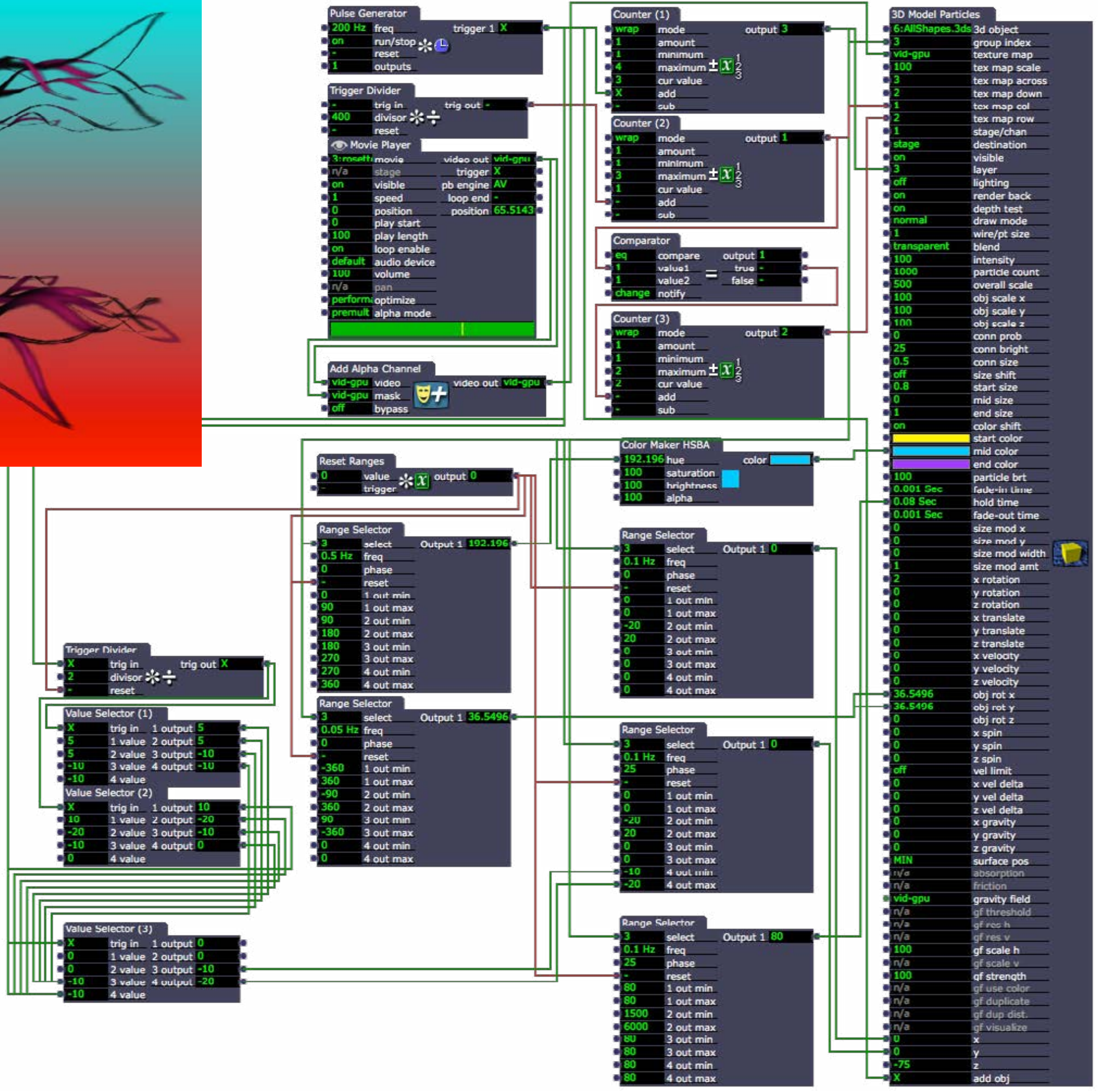
Fig. 29 The three Value Selector user actors allow for the parameters of a second instance of a mesh group to display based on allocating frequency width in the patch





```
FFGLGradient
0     hue 1        texture out vid-gpu
50    hue 2
100   saturation
75    brightness
693   width
1230  height
n/a   stage

Projector
vid-gpu   video
0         horz pos
0         vert pos
100       width
100       height
180       zoom
on        keep aspect
-90       spin
additive  blend
0         layer
100       intensity
on        active
1         stage
```

Fig. 31 The use of a ‘FFGLGradient’ actor places colour behind the particles and provides additional atmosphere within the generated image.



```
Pulse Generator
200 Hz  freq           trigger 1 X
on      run/stop
-       reset
1       outputs

Trigger Divider
-    trig in          trig out -
400  divisor
-    reset

Movie Player
3:rosetti  movie          video out vid-gpu
n/a        stage          trigger X
on         visible        pb engine AV
1          speed          loop end -
0          position       position 65.5143
0          play start
100        play length
on         loop enable
default    audio device
100        volume
n/a        pan
performa   optimize
premult    alpha mode

Add Alpha Channel
vid-gpu  video       video out vid-gpu
vid-gpu  mask
off      bypass

Counter (1)
wrap  mode          output 3
1     amount
1     minimum
4     maximum
3     cur value
X     add
-     sub

Counter (2)
wrap  mode          output 1
1     amount
1     minimum
3     maximum
1     cur value
-     add
-     sub

Comparator
eq      compare     output 1
1       value1      true -
1       value2      false -
change  notify

Counter (3)
wrap  mode          output 2
1     amount
1     minimum
2     maximum
2     cur value
-     add
-     sub

Color Maker HSBA
192.196  hue          color
100      saturation
100      brightness
100      alpha

Reset Ranges
0  value      output 0
-  trigger

Range Selector
3       select      Output 1 192.196
0.5 Hz  freq
0       phase
-       reset
0       1 out min
90      1 out max
90      2 out min
180     2 out max
180     3 out min
270     3 out max
270     4 out min
360     4 out max

Range Selector
3        select      Output 1 36.5496
0.05 Hz  freq
0        phase
-        reset
-360     1 out min
360      1 out max
-90      2 out min
360      2 out max
90       3 out min
-360     3 out max
0        4 out min
0        4 out max

Range Selector
3       select      Output 1 0
0.1 Hz  freq
0       phase
-       reset
0       1 out min
0       1 out max
-20     2 out min
20      2 out max
0       3 out min
0       3 out max
0       4 out min
0       4 out max

Range Selector
3       select      Output 1 0
0.1 Hz  freq
25      phase
-       reset
0       1 out min
0       1 out max
-20     2 out min
20      2 out max
0       3 out min
0       3 out max
-10     4 out min
-20     4 out max

Range Selector
3       select      Output 1 80
0.1 Hz  freq
25      phase
-       reset
80      1 out min
80      1 out max
1500    2 out min
6000    2 out max
80      3 out min
80      3 out max
80      4 out min
80      4 out max

Trigger Divider
X  trig in       trig out X
2  divisor
-  reset

Value Selector (1)
X    trig in    1 output 5
5    1 value    2 output 5
5    2 value    3 output -10
-10  3 value    4 output -10
-10  4 value

Value Selector (2)
X    trig in    1 output 10
10   1 value    2 output -20
-20  2 value    3 output -10
-10  3 value    4 output 0
0    4 value

Value Selector (3)
X    trig in    1 output 0
0    1 value    2 output 0
0    2 value    3 output -10
-10  3 value    4 output -20
-10  4 value

3D Model Particles
6:AllShapes.3ds  3d object
3            group index
vid-gpu      texture map
100          tex map scale
3            tex map across
2            tex map down
1            tex map col
2            tex map row
1            stage/chan
stage        destination
on           visible
3            layer
off          lighting
on           render back
on           depth test
normal       draw mode
1            wire/pt size
transparent  blend
100          intensity
1000         particle count
500          overall scale
100          obj scale x
100          obj scale y
100          obj scale z
0            conn prob
25           conn bright
0.5          conn size
off          size shift
0.8          start size
0            mid size
1            end size
on           color shift
             start color
             mid color
             end color
100          particle brt
0.001 Sec    fade-in time
0.08 Sec     hold time
0.001 Sec    fade-out time
0            size mod x
0            size mod y
0            size mod width
1            size mod amt
2            x rotation
0            y rotation
0            z rotation
0            x translate
0            y translate
0            z translate
0            x velocity
0            y velocity
0            z velocity
36.5496      obj rot x
36.5496      obj rot y
0            obj rot z
0            x spin
0            y spin
0            z spin
off          vel limit
0            x vel delta
0            y vel delta
0            z vel delta
0            x gravity
0            y gravity
0            z gravity
MIN          surface pos
n/a          absorption
n/a          friction
vid-gpu      gravity field
n/a          gf threshold
n/a          gf res h
n/a          gf res v
100          gf scale h
n/a          gf scale v
100          gf strength
n/a          gf use color
n/a          gf duplicate
n/a          gf dup dist.
n/a          gf visualize
0            x
0            y
-75          z
X            add obj
```

Isadora tutorial document prepared by Bonemap © 2018
https://community.troikatronix.com/topic/5434/3d-particles-tutorial

Fig. 30, The full patch work incorporates a number of user actors that sequence the parameters associated with the discrete mesh groups.