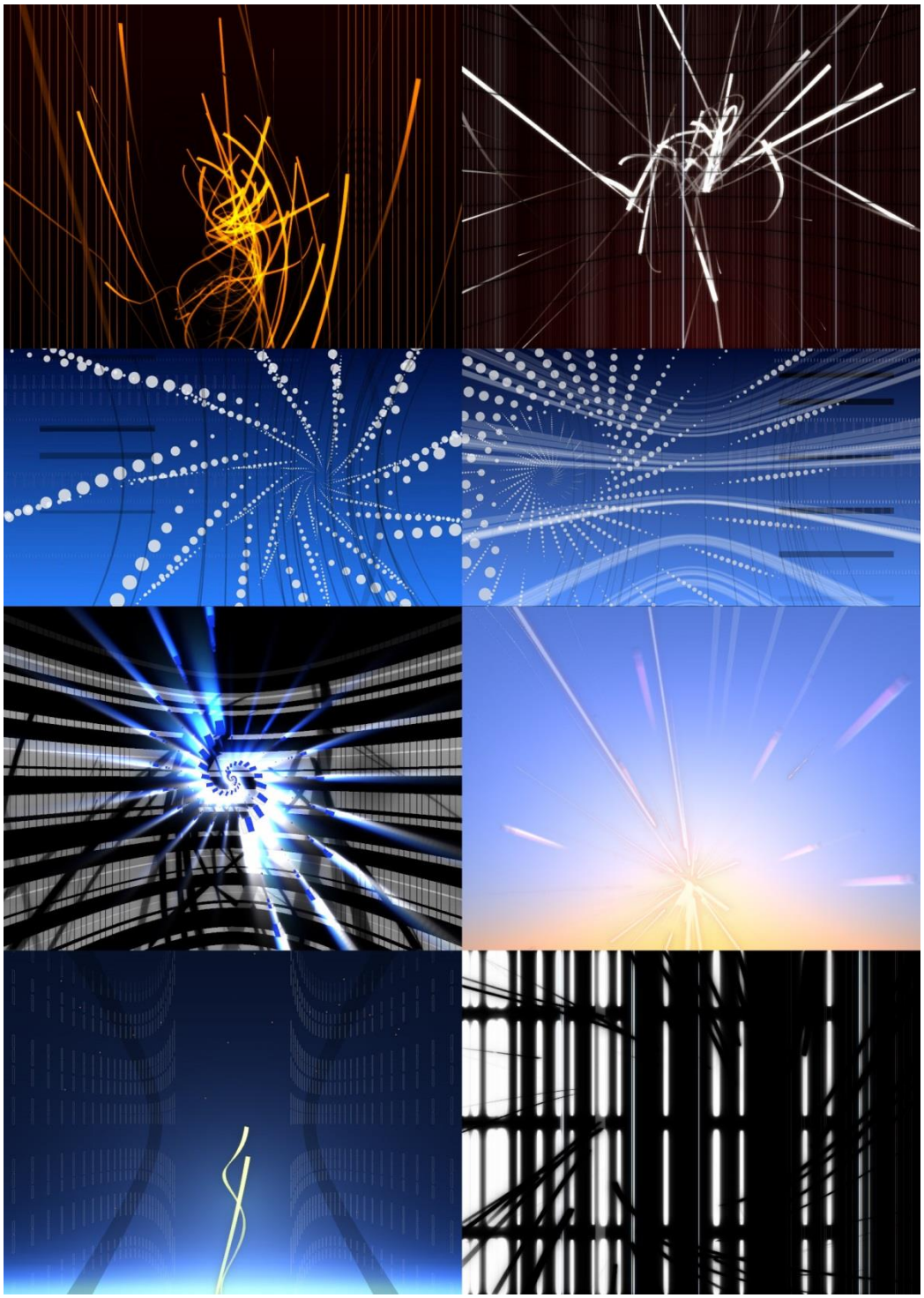
MOTHER 1.0



NEW FEATURES IN THIS VERSION:

- Processing 2.0 compatibility
- Mother now works entirely from within the Processing IDE, and is thus no longer an external program which makes use of Processing, but a Processing library. This also means that you no longer need to run Mother from Eclipse to get access to console Error messages, as these are displayed in the Processing IDE's console.
- A new controller example is included which too works entirely from within Processing, using the controlP5 and oscP5 libraries.
- An Eclipse project is included with the distribution so that more advanced users can get started more quickly with developing Synths in Eclipse.
- The synth folder can now either be in the Mother sketch's data folder, or a custom location.

GETTING STARTED:

To get the included examples running, follow the following steps:

1. Download and install the oscP5 and controlP5 libraries, needed by Mother and by the example sketch respectively.
2. Run the example controller (MotherControllerExample.pde in Contributed Libraries -> Mother folder)
3. Run MotherDelivery.pde, found in the same folder.
4. If everything goes well, Mother Delivery will display a black screen. To get things going, press the "Add Gradient", and "Add Rotating Arcs" buttons in the controller, which should result in these two synths being loaded.
5. Use the on screen controls to change the colours of the gradient, and the transparency and scaling of the rotating arcs.
6. Read the included documentation, documentation related to using OSC (http://opensoundcontrol.org/), and get started making your own synths and user interfaces for controlling them!

If it doesn't work right away:

- Make sure you graphics card and the installed drivers support OpenGL 2.0
- Make sure the communication between the controller and Mother is not blocked by a firewall, and that the network ports specified in Mother.ini are available. If the ports cannot be opened, the console in the Processing IDE's for the two sketches will contain messages alerting to the fact. If this is the case, change the ports in the

MotherControllerExample sketch, and read the documentation below on how to edit the mother.ini file towards setting an unused port for Mother.

- Post a question on the "Contributed Library Questions" section of the Processing forum :) Make sure to also send me a link to your thread at onar3d, at hotmail.com or at gmail.com, since I don't always check the Processing forum.

MOTHER 1.0 USAGE INSTRUCTIONS

Mother is a standalone host application for running multiple Processing sketches in parallel, and mixing their output, in a manner analogous to VJing. Version 1.0 has been tested with processing version 2.0.1, and might not work with sketches exported with earlier/later versions of Processing.

For a Processing sketch to be compatible, it needs to have been built using the Foetus library provided.

Diagrammatically (and this will be better explained further in this text), to create a synth for use with Mother, use the Foetus library, export your synth from the processing environment (File->Export Application), and place the “yourSketch.jar” file found in the folder “application.windowsXX/lib/”, and the .jar files of any libraries used, in the synth folder specified in the mother.ini file. The default folders for this use are ./data/Synths/, and ./data/Synths/libraries/ respectively, within the data folder of the MotherDelivery sketch. Further instructions on how to create a synth for use with Mother follow in a separate section.

START-UP SETTINGS FOR MOTHER:

In the mother.ini file, located in the same folder as the Mother executable, the following start-up parameters can be specified. # denotes a comment. Make sure to write all lines without any spaces!

```
# This is where the host looks for the synth jar files.
# the folder you point to needs to also include two subfolders, "data" and
# "libraries".
# So if you keep your synths in "X:\\Synths", you also need to create folders
# "X:\\Synths\\data", and "X:\\Synths\\libraries".
UseCustomSynthFolder=0;
SynthFolder=X:\\PortableApps\\Lumia_Synths_P2.0;

# Here the IP address of the OSC controller is specified.
IP=127.0.0.1;

# The port on which the host receives OSC messages.
osc_receive_port=7000;

# The port to which the host sends OSC messages.
osc_send_port=5432;

# If the host should run in fullscreen, set this parameter to 1.
```

```
# For windowed mode set the parameter to 0.
FullScreen=0;

# Setup secondary display output to use when in fullscreen mode.
# 0: primary display,
# 1: secondary display.
# n: n-th display (untested).
outputScreen=0;

# screenSize=3360,1050;
screenSize=800,600;

# Framerate:
frameRate=24;

# Path for storing screen captures:
imagePath=D:\\ML_Grabs\;

# Fraction of running speed. So if framerate is 30, and fraction is 5, the
# actual framerate will be 6 fps.
# This is useful when running mother in non-realtime, and recording the output
# to image files.
# The fraction value is then used to adjust timing calculations so that the
```

```
# rendered sequence corresponds perfectly
# to what it would have looked like in real time.
speedFraction=1;
```

OSC MESSAGES FOR MOTHER:

/MOTHER/GET_SYNTH_NAMES

Causes a message to be returned containing the names of all the synth types that the host has access to, in the following format:

/Synth_names synth1 synth2 synth3 (...)

/MOTHER/ADD_SYNTH SYNTH1 SYNTH1_01

To add a synth at the topmost layer, the /Mother/Add_synth message needs to be sent with two arguments, the name of the desired synth type, and a unique name to identify the particular synth instance. If an instance already exists with that name, no new synth is added to the stack.

/MOTHER/REMOVE_SYNTH SYNTH1_01

Removes the synth with the ID specified in the argument. If no synth is found with that ID then no synth is removed.

/MOTHER/MOVE_SYNTH SYNTH1_01 0

The synth specified in the first argument is moved to the stack location specified in the second argument. The synth that previously held that location is displaced.

/MOTHER/RECORD 1

If 1 is specified, Mother begins recording .png images to the path specified in the "imagePath" field of mother.ini. A message with value 0 stops the recording. Only for use in non real-time mode, as it is normally too demanding to run in real-time.

(/MOTHER/SET_SYNTH_BLENDING SYNTH1_01 SOURCE DESTINATION)

Deprecated! The way this was implemented was too costly in terms of performance for it to be worth keeping. If different blending modes are desired, you can easily implement that within your synths.

(/MOTHER/SET_SYNTH_COLOR SYNTH1_01 R G B A)

Deprecated! The way this was implemented was too costly in terms of performance for it to be worth keeping.

MESSAGES FOR FOETUS:

Each synth has to be built using the Foetus library. Besides enabling a Processing sketch to work with the host, the library also enables the synth to communicate over OSC. The foetus library itself only responds to one message, which in turn returns a message with all the additional OSC messages the synth supports, along with their typetags:

/Mother/Child/synth1_01/Get_Supported_Messages

Where synth1_01 is the ID of a synth of the desired type.

The returned message is then of the following format:

/Synth_supported_messages/synth1_01 /Param_3 ii /Param_2 ii /Param_1 ii

Where the address of each parameter followed by its typetag are listed in turn.

The parameters of each synth are then addressed using messages using the following format:

/Mother/Child/ synth1_01 /Param_Name

FOETUS METHODS:

Well, it only has one as of yet:

millis() is the same as the Processing *millis()* function, with the additional feature that it takes the specified speed fraction into account. This is useful when running in non real-time mode, as the *f.millis()* call returns the time value at a given frame number that it would have if it were running in real-time.

BUILDING A SYNTH USING PROCESSING:

1. Install the Mother library into your processing environment. This also include the Foetus library.
2. If you do not already have it, also install the OSCP5 library, as the Foetus library depends on it. Foetus also depends on the Shapetween library, but because Shapetween doesn't yet have an official Processing 2.0 port, I include it in the distribution of Mother.

   > *Note: If you also have an installation of the Shapetween library in your library folder, delete the shapetween.jar file that currently comes in the library folder of the Mother distribution, or the Processing IDE will complain that the library is installed twice, creating a conflict.*

3. Make sure you use the OPENGL-renderer (in P2.0 interchangeable with P3D), as the P2D renderer is unsupported.

4. Declare a public foetus object named f: *public Foetus f*;

5. Create a method *void initializeFoetus()*. This method will hold all your initialization code. Setup() is not called when a sketch is used as a synth in Mother, and so any initialization you would normally do in Setup(), should instead be performed here.

6. Call *initializeFoetus()* from within setup() so that you can run the sketch outside of Mother – if you don't the sketch cannot run by itself directly from the Processing IDE, only from within Mother.

7. In *initializeFoetus(),* instantiate the foetus object: *f = new Foetus(this);*

8. To create a synth for use with Mother, use the Foetus library, export your synth from the processing environment (File->Export Application), and place the "yourSketch.jar" file found in the newly created folder "application.windowsXX/lib/", and the .jar files of any libraries used, in the synth folder specified in the mother.ini file. The default folders for this use are ./data/Synths/, and ./data/Synths/libraries/ respectively, within the data folder of the MotherDelivery sketch.

If you want the synth to respond to its own set of OSC messages, you also need to do the following:

1. register what these messages are in the *initializeFoetus()* method:

   f.registerMethod("/TopColor ", "iii");

   f.registerMethod("/BottomColor ", "iii");

   etc…

2. and add a *void oscEvent(oscMessage theOscMessage)* method:

```
void oscEvent(OscMessage theOscMessage)
{
 if (theOscMessage.checkAddrPattern("/TopColor") == true)
 {
  /* check if the typetag is the right one. */
  if (theOscMessage.checkTypetag("iii"))
  {
   m_TopR = theOscMessage.get(0).intValue();
   m_TopG = theOscMessage.get(1).intValue();
   m_TopB = theOscMessage.get(2).intValue();
   return;
  }
 }
 else if (theOscMessage.checkAddrPattern("/BottomColor") == true)
 {
     etc…
```

}

SPLINE INTERPOLATION:

With Mother 0.2 came the facility of allowing spline interpolation for individual floating point synth parameters. Instead of declaring each parameter as float, you declare it as FoetusParameter, and instantiate it in the initializeFoetus() method, eg:

Declaration:

Foetus parameter m_Red;

Instantiation:

```
initializeFoetus()
{
	…

	// Instantiate foetus object here
	f = new Foetus(this);

	m_Red = new FoetusParameter(f, 0, "/Red","f");

	…
}
```

Where f is a reference to the Synth's Foetus (make sure to first instantiate it before using it here or the synth will crash!), 0 is the initial value, "/Red" is its OSC address, and "f" is the OSC typetag. With the 0.2 release the only supprted typetag is "f", but I've nonetheless added the parameter for compatibility with future releases.

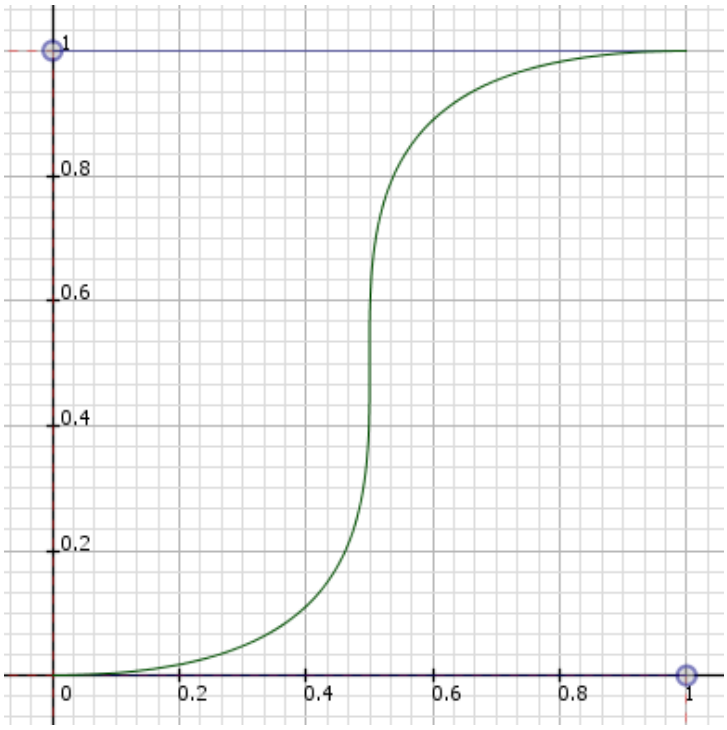
If you use FoetusParameter, you do not need to use *f.registerMethod()* for it, as this is taken care of from inside the constructor.

You then get the interpolated value using the getValue() method, and set their value using setValue().

At the moment, the speed of the interpolation is the time that passed between the two previous setValue() calls, or three seconds, whichever is the shortest.

If messages are received with intervals shorted than 100 milliseconds, no interpolation is done between them.

Also note that currently no possibility to control the spline interpolation has been added. The default setting used is that of an 'S' curve, which gives the most natural-looking type of animation:



Keep in mind though that spline interpolation is relatively expensive computationally, if you use it on a very large number of parameters it may end up affecting the frame-rate of the sketch.

Finally note that all included examples have been updated to use spline interpolation, so have a look to see how it is done, there is very little to it!

NOTES ON USING MULTIPLE SCREENS:

While running Mother on a computer with more than one display device attached may be desirable, it does also mean that certain complications need to be taken into consideration. Not all graphics cards are good at supporting OpenGL acceleration on the secondary display, and even those at are, often are by default set up to do so at the cost of reduced performance. So although Mother supports outputting to the secondary display, I recommend setting your projector to be your primary display, and disabling OpenGL acceleration on the secondary, for maximized OpenGL performance.

The only way to currently output to the secondary display, is to use the MotherDeliveryEcplipse project to run Mother, and from within there, to set the outputScreen parameter to a value greater than 0.

SUGGESTION FOR ADVANCED USERS:

Mother is most intuitively used from within the Eclipse IDE, as you can then directly reference your sketch projects from within the MotherDeliveryEclipse project. Go to Project Properties -> Java Build Path ->Projects->Add…, and from there add the synth

projects that reside in the workspace. This way, you do not need to export the .jar file for your synth every time you've made a change to it, instead Mother will use the code for the synth as is referenced from within Eclipse. For this reason, and because Eclipse also provides many important features for more advanced development, I therefore recommend using the MotherDeliveryEcplipse project, if you plan on developing complex visual synths.

MOTHER DEVELOPMENT SUPPORTERS:

Mother's development has been partially funded by the Agalma foundation, www.agalma.ch

And partially by the EU FP7 Presenccia project, from which I received funding towards carrying out my PhD (2011), thesis available for download here:
http://discovery.ucl.ac.uk/1310143/

REVISION HISTORY:

MOTHER 0.6:

- Small mistakes fixed from previous 0.5 release
    - Waltz synth example was not working.
    - I failed to mention in the documentation that you also need the Shapetween library to develop new synths.
- More comprehensive secondary screen support introduced.
- pre(), draw(), post() and dispose() methods are now called properly in synth libraries.
- It is no longer necessary to put an Init() method in sketches intended as Synths for Mother (Finally!).

MOTHER 0.5:

- Updated Mother to use Processing 1.0.7
- Performance improved
- Stability improved (bugs fixed)
- OSX-port implemented (Many thanks to Krzysztof Goliński and Splatgirl for their work on the port!)

MOTHER 0.4:

- Updated Mother to use Processing 1.0.5.

- Updated the distribution to conform to the Processing Library guidelines.
- Added Pure Data controller example.

MOTHER 0.3:

- Anti-aliased rendering, a very important feature which has improved image quality considerably.
- Support for Processing version>1.x
- A more straightforward way of handling sketches which use external libraries. In previous versions libraries had to be packed in the jar file of the sketch. Now libraries are loaded dynamically when needed from their original jar files, so all a user needs to do is place all library jar files in a designated folder.
- Non real-time mode, useful for rendering high resolution video to disk.
- Considerable speed increases through various optimizations.

MOTHER 0.2:

- Spline interpolation
- Bug fixes