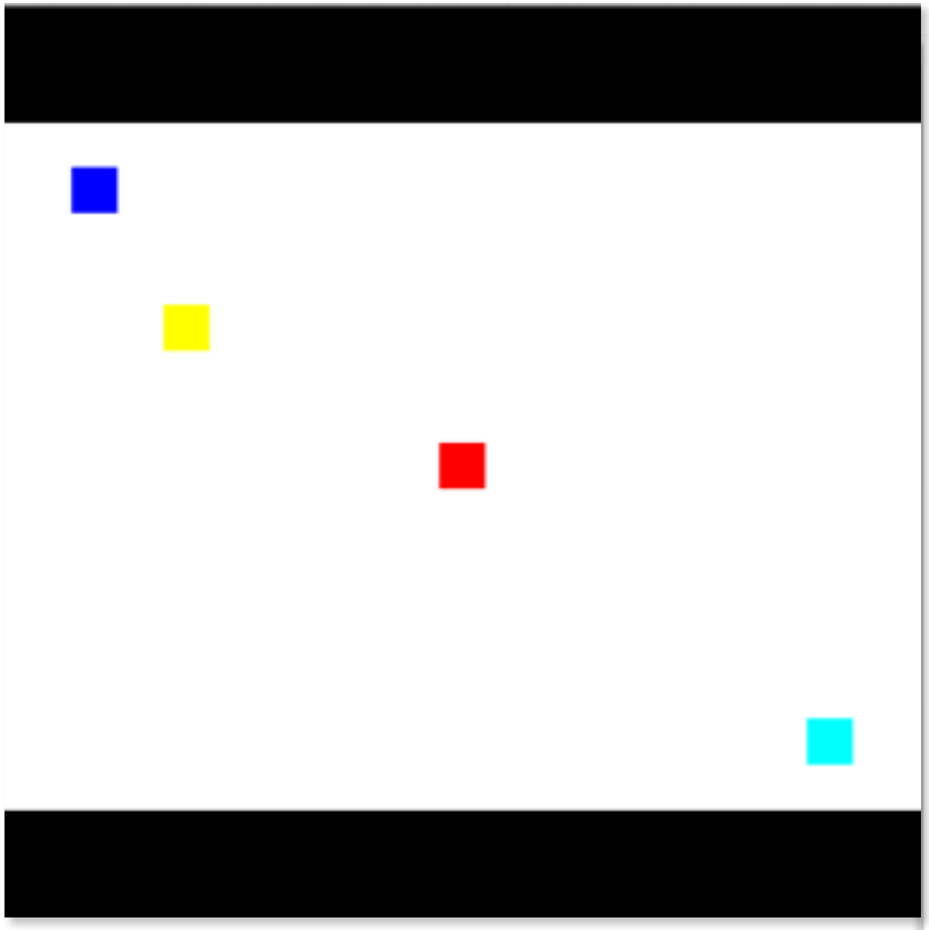
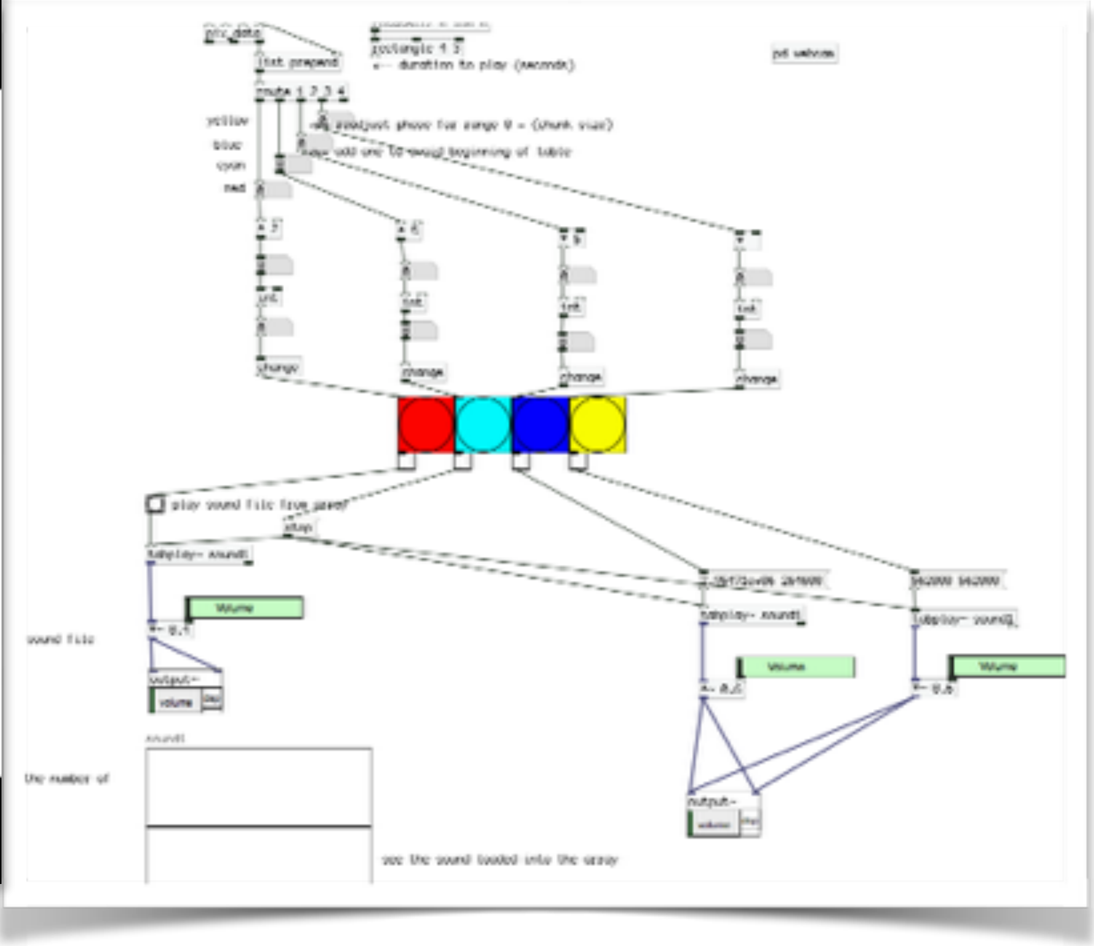
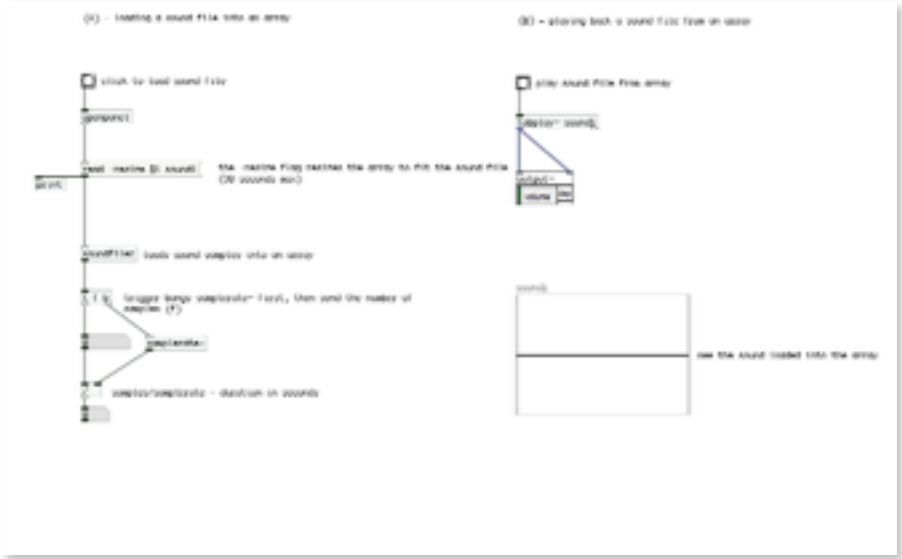
Musical Human-Computer Interaction
Motion Sensing - Pure Data Patch




pix_data
list prepend
route 1 2 3 4
← duration to play (seconds)
yellow
blue
cyan
red
adjust phase for range 0 - (chunk size)
add one to avoid beginning of table
int
change
pd webcam
play sound file from start
stop
tabplay~ sound1
Volume
sound file
output~
volume
sound1
the number of
see the sound loaded into the array

Max Neuperts patch shown Here, is what i consider to be the building blocks of my pure data patch. This patch will also be included the Patch Development folder.



(A) - loading a sound file into an array

(B) - playing back a sound file from an array

click to load sound file

openpanel

read -resize $1 sound1

the -resize flag resizes the array to fit the sound file (30 seconds max)

print

soundfiler loads sound samples into an array

t f b trigger bangs samplerate~ first, then send the number of samples (f)

samplerate~

samples/samplerate - duration in seconds

play sound file from array

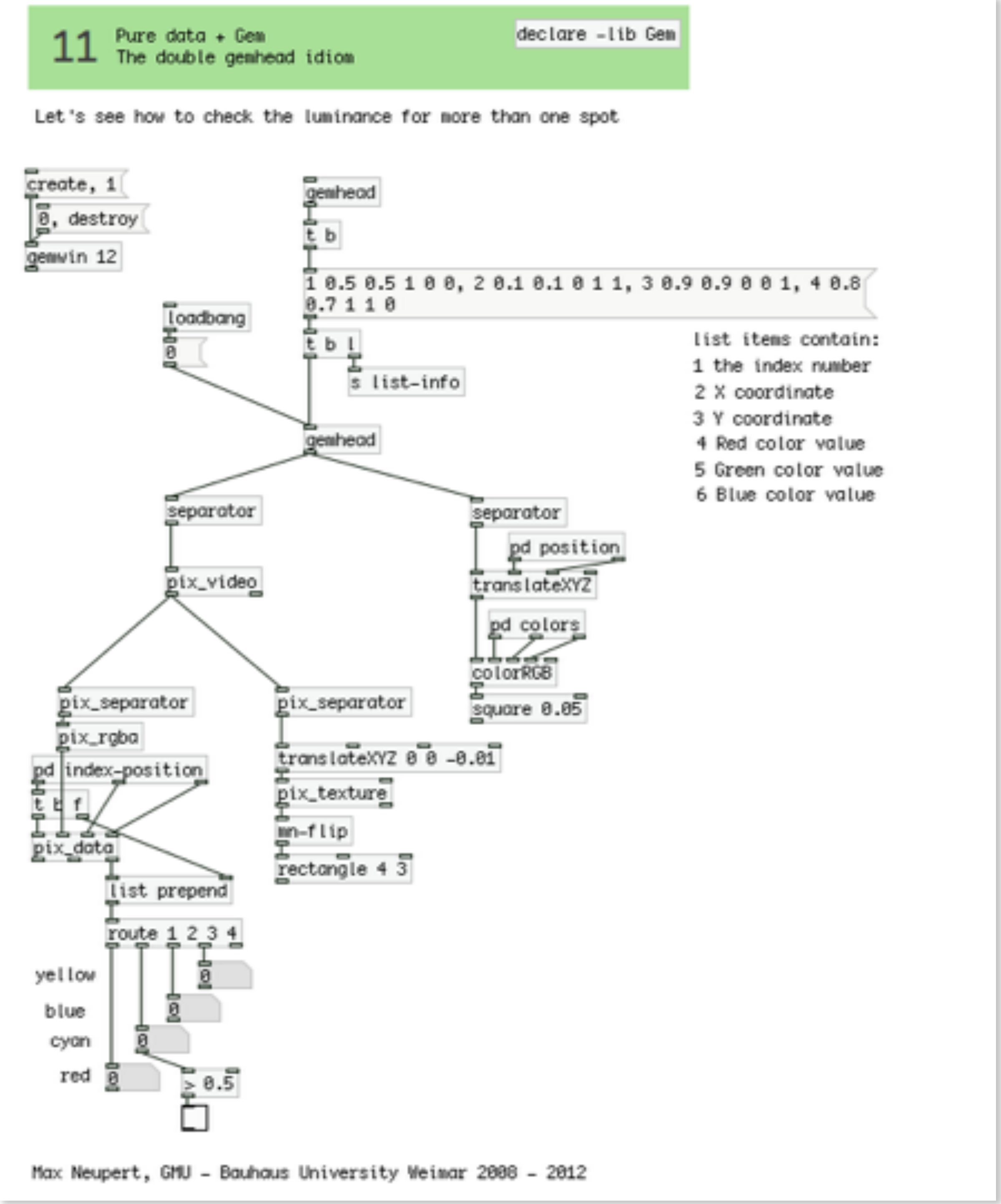
tabplay~ sound1

output~

volume

sound1

see the sound loaded into the array



11 Pure data + Gem
The double gemhead idiom

declare -lib Gem

Let's see how to check the luminance for more than one spot

create, 1

0, destroy

gemwin 12

gemhead

t b

1 0.5 0.5 1 0 0, 2 0.1 0.1 0 1 1, 3 0.9 0.9 0 0 1, 4 0.8 0.7 1 1 0

loadbang

0

t b l

s list-info

gemhead

list items contain:
1 the index number
2 X coordinate
3 Y coordinate
4 Red color value
5 Green color value
6 Blue color value

separator

pix_video

separator

pd position

translateXYZ

pd colors

colorRGB

square 0.05

pix_separator

pix_rgba

pd index-position

t b f

pix_data

list prepend

route 1 2 3 4

pix_separator

translateXYZ 0 0 -0.01

pix_texture

mn-flip

rectangle 4 3

yellow

blue

cyan

red

0

0

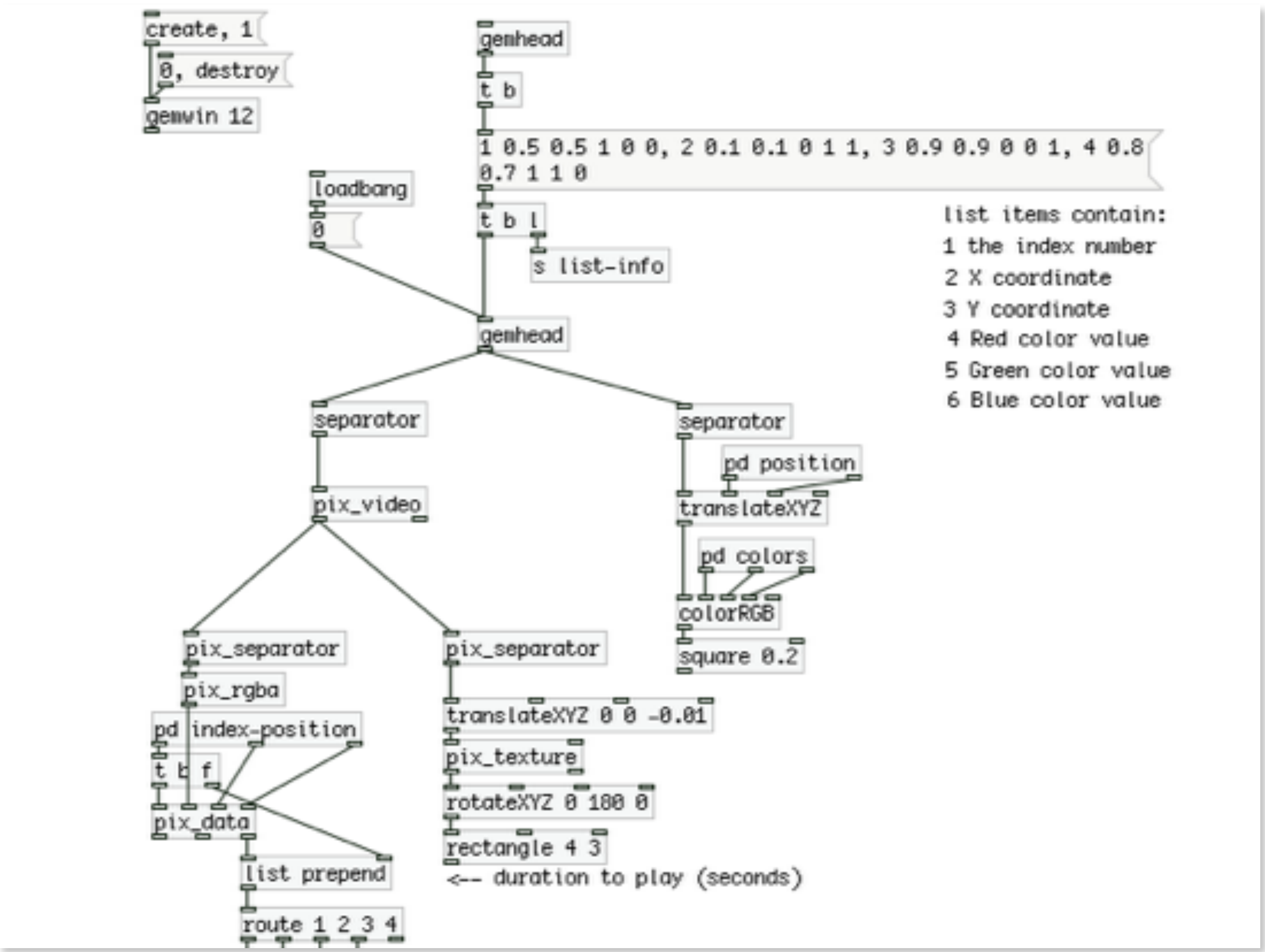
0

0

> 0.5

Max Neupert, GMU - Bauhaus University Weimar 2008 - 2012

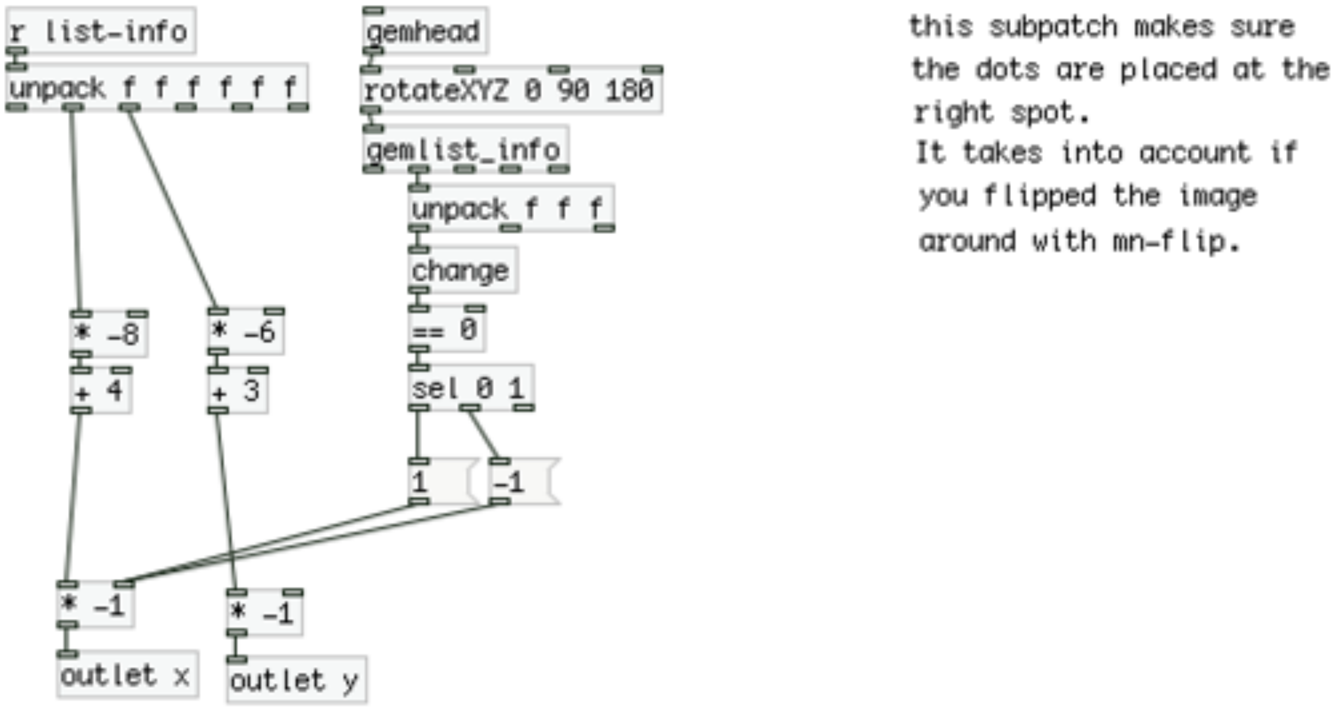
List Information sent through “s list-info” to
Subpatch [pd shape].We can create a number
of Trigger Points as long as they are listed
within.
This is added onto gem head in the second
part
of the patch which is then divided into two
parts
by the separator modules.
The left side dedicated to what happens
inside the video feed and the right hand
side dedicated to the coloured squares
overlaying on the video feed.



list items contain:
1 the index number
2 X coordinate
3 Y coordinate
4 Red color value
5 Green color value
6 Blue color value

[Separator Module Right Hand Side]

As it is displayed in side the patch this is what makes sure the dots are in the right location within the video allowing for a trigger to be made via motion.

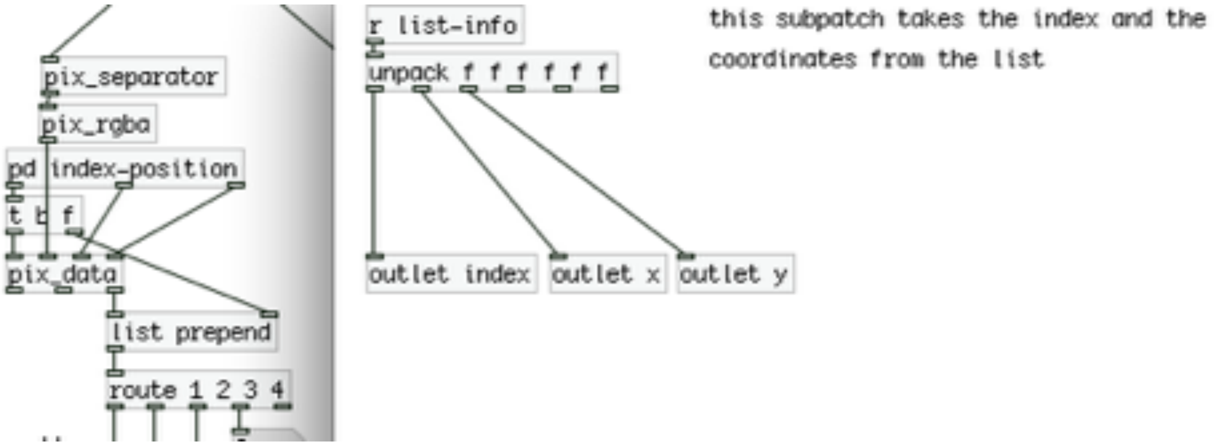


this subpatch makes sure the dots are placed at the right spot.
It takes into account if you flipped the image around with mn-flip.

[Separator Module Left Hand Side]

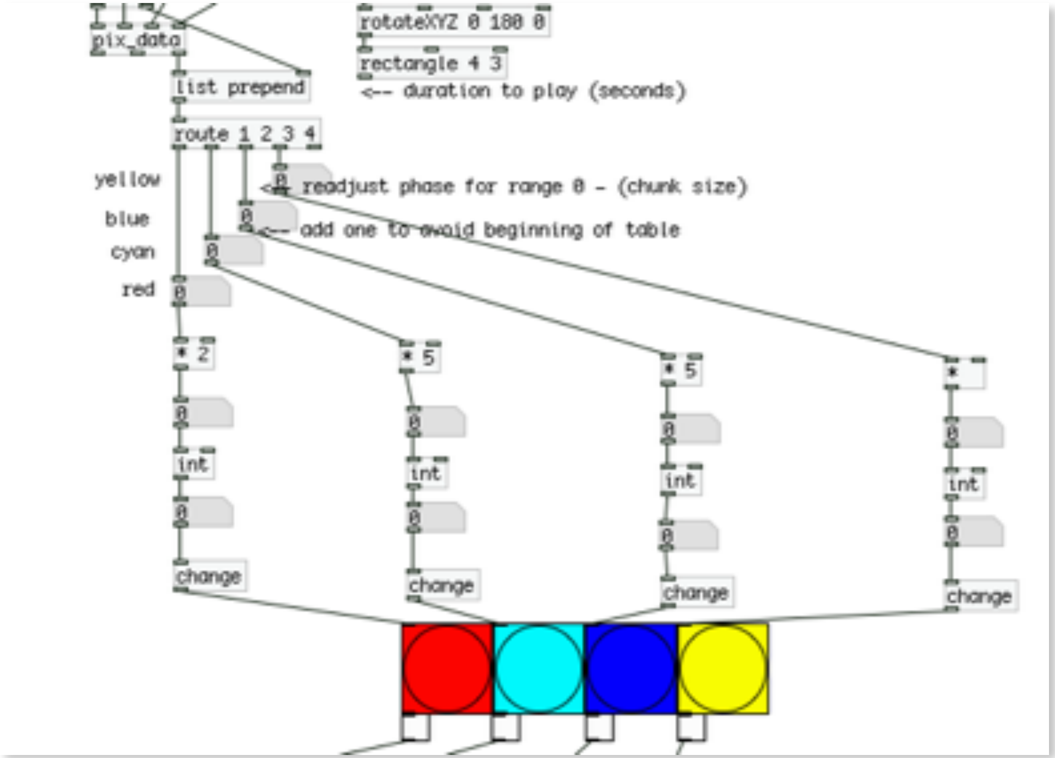
Here we have a module called pd index-position this module takes the information present from s list spoken about earlier to be able to create the actual trigger points from the coloured squares.

or when a threshold has been met at X,Y and Z coordinates send a bang/trigger something.

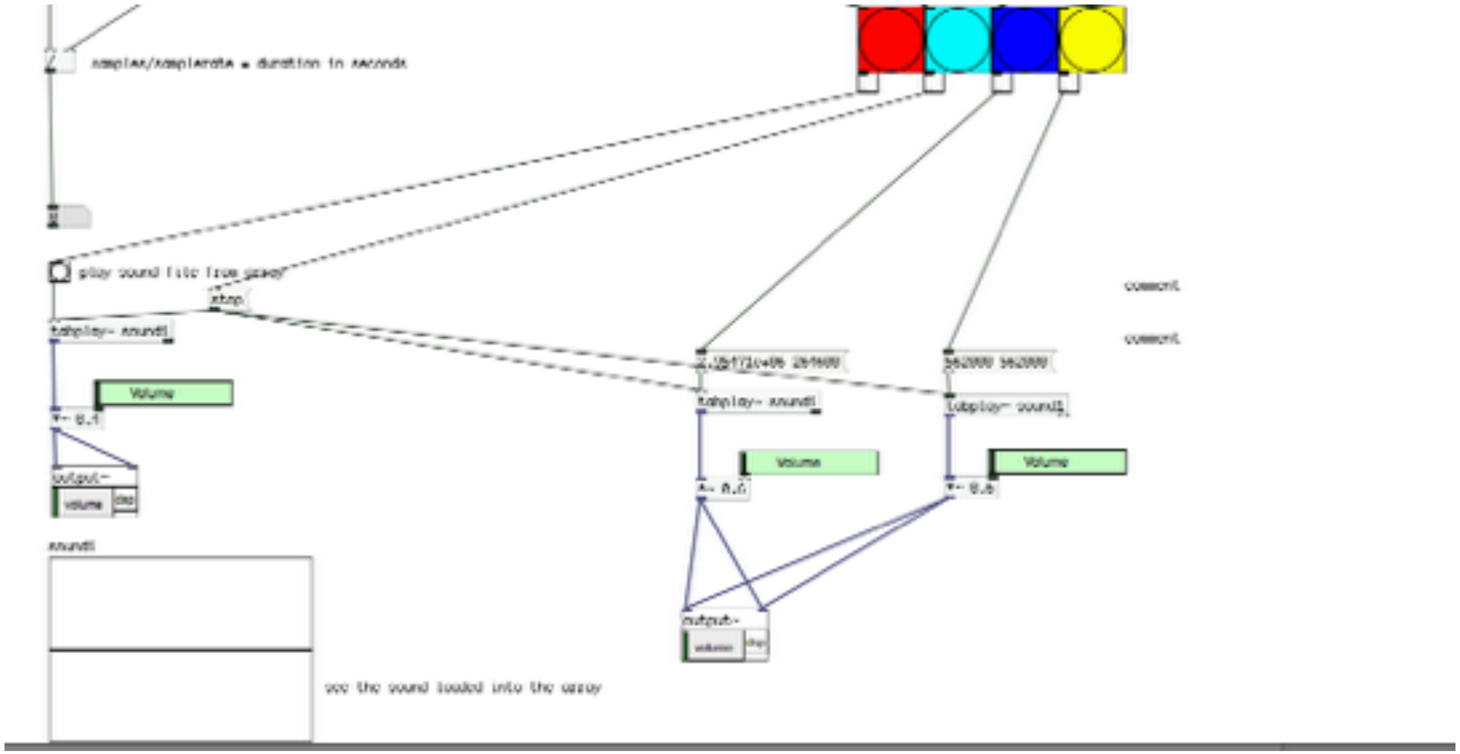


this subpatch takes the index and the coordinates from the list

After all the information is collected it is then routed via which ever channel you
select here i just have 4 channels to choose from.
The Red Square in the Feed being 1 and the last square being yellow which is 4.
After a threshold is met inside the video a data is fed via the routing module to
the appropriate coordinates or squares this number is then fed into a multiplier which then causes a trigger, the triggers are each set so that 0 is off and 1 is on, when 1 is actuated a trigger or bang is set off causing a audio sample from the channel that caused it.



Once the appropriate channel is triggered an appropriate action follows, so as we can see here in the diagram it is colour coded. When red has been triggered it plays the audio, when cyan is triggered it stops all audio, blue and yellow where used to demonstrate that the same sample could be used but to play at varying starting points. However using the tabplay~ function i could change arrays to load in different samples and play more than one audio sample at any given time.


samples/samplerate = duration in seconds
play sound file from array
stop
tabplay~ sound1
Volume
*~ 0.1
output~
volume
dsp
sound1
see the sound loaded into the array
562000 562000
tabplay~ sound1
Volume
*~ 0.6
tabplay~ sound1
Volume
*~ 0.6
output~
volume
dsp
comment
comment