[w a l l]

for WALO

Dane P. Yates | 2015

Recorded @ WAAPA.MA 18th August 2015

Premiered by John Barrett, Joseph Dignon, Brendan Polain and Dane Yates

Recording and Score Documentation is available at daneyatescomposition.weebly.com

[w a l l] is an indeterminate piece of music utilising glitch and audible manipulations of audio files. The piece is performed on four laptops although a performance of a larger ensemble can take place; the piece requires four at minimum. For the piece I have created a max/MSP software that the performers control through key clicks and subtle changes to the interface. The score for the piece is a graphic screen score, which is readable through either max/MSP or the Decibel ScorePlayer for iPad.

patch

The max/MSP patch "[wall.player].maxpat" is built around the groove~ object, all key clicks either affect the input signal or output of ~groove. The first input is sent a buffer signal; the buffer can hold ten different audio files named 0.wav, 1.wav, 2.wav etc. These files are selected and changed by pressing the number keys on the laptop keyboard. The first input of ~groove is also sent a signal from sig~, the signal controls the length of "glitching" of the audio before moving on to the next glitch. The input to sig~ is controlled by the master toggle as well as four message bangs containing preloaded figures that can be selected through the arrows of the keyboard. The messages output to a flonum controlling the length of time of a metro object which outputs an indeterminate number between zero and one thousand controlling the signal of sig~.

The second and third inputs of groove~ contains bang messages and flonums which control ten different figures ranging from 180 to 31000. The figures control minimum and maximum loop points of the buffer. The minimum start loop points are controlled through another "metro – random – flonum" setup outputting numbers zero to nine randomly – close to - every millisecond. A sel~ object outputs bang messages to the figures that control the loop points. The third input is predetermined, as it does not affect the audio unless it is small figure (<1000). The sound produced from a low maximum point was not something I

really wanted in the piece, it produces a lot of clipping and not much output other than an IO signal which I have explored in other pieces, so having a predetermined output was something that I enjoyed the idea of "stability".

groove~ is then sent both to live.gain - dac and gizmo~. gizmo~ is a subpatcher controlling pitch shift and spectral warping. The input to gizmo~ can be controlled by flonum or by the predetermined figure of -36 which reveals a low frequency sound world distant to that of the untouched sounds of the patch; gizmo~ is then also sent to the dac. The return key of the keyboard controls the volume levels of both the original sounds and the gizmo sounds which ultimately toggles between the two.

Other controls of the patch help to glitch and distort the sound; when switching between patcher and presentation mode, the patch "glitches" temporarily giving the effect of a jammed CD, by zooming in on the patch extremely, helps control length of these glitches, the more zoomed in, the longer the glitch.

score

The score for the piece is a scrolling score played in the Decibel ScorePlayer. It consist of four lines of text, giving instructions in real-time to the performer of key clicks and patterns to be performed on the laptop. The score is quite clear in performance gestures giving instructions proportionally in time. Any 'grouped' instruction (ie down.2.space.space) are to be performed as fast as possible and not proportional to the score, which adds a level of indeterminacy to the piece. In the instructions file within the ScorePlayer, performers are told they may choose to ignore the score at times if they see fit; if the performer 'finds' a sound that they see as being better as what else is instructed, they may choose to stay on that sound and then come back in once they feel the sound has 'lived its life temporarily'.

Around 3'30", performers are only given one instruction to repeat and are told they can use a different sound file if needed, to reflect upon the given instruction. At 4'10 performers only play their given instruction at whatever speed they see fit. They are then instructed that, whilst performing their continuous gesture, to perform any 'four to eight digit code' which can be anything and they can also change between sounds/gestures at their own pace.

At 5'00" they are told to pick any favoured sound with any amount or type of manipulation and leave that to play for the duration of the piece without interference. The space toggling performed at the end can be performed at any pace, and not a quick IO gesture.

performance

The performance is indeterminate as the performers are told to import any ten .wav files.

Figure I. Instruction file in the .dsz file for the Decibel ScorePlayer

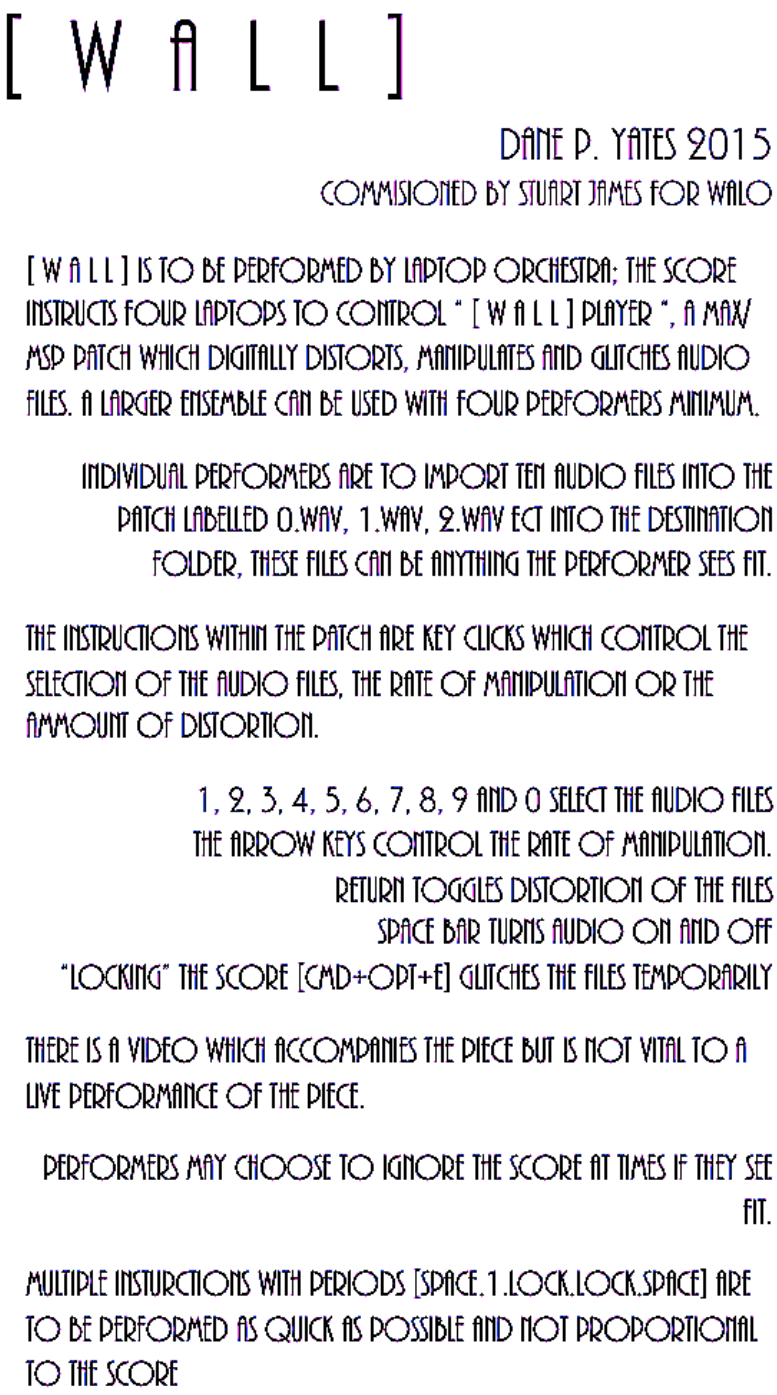

Figure II. max/MSP patch [wall.player]

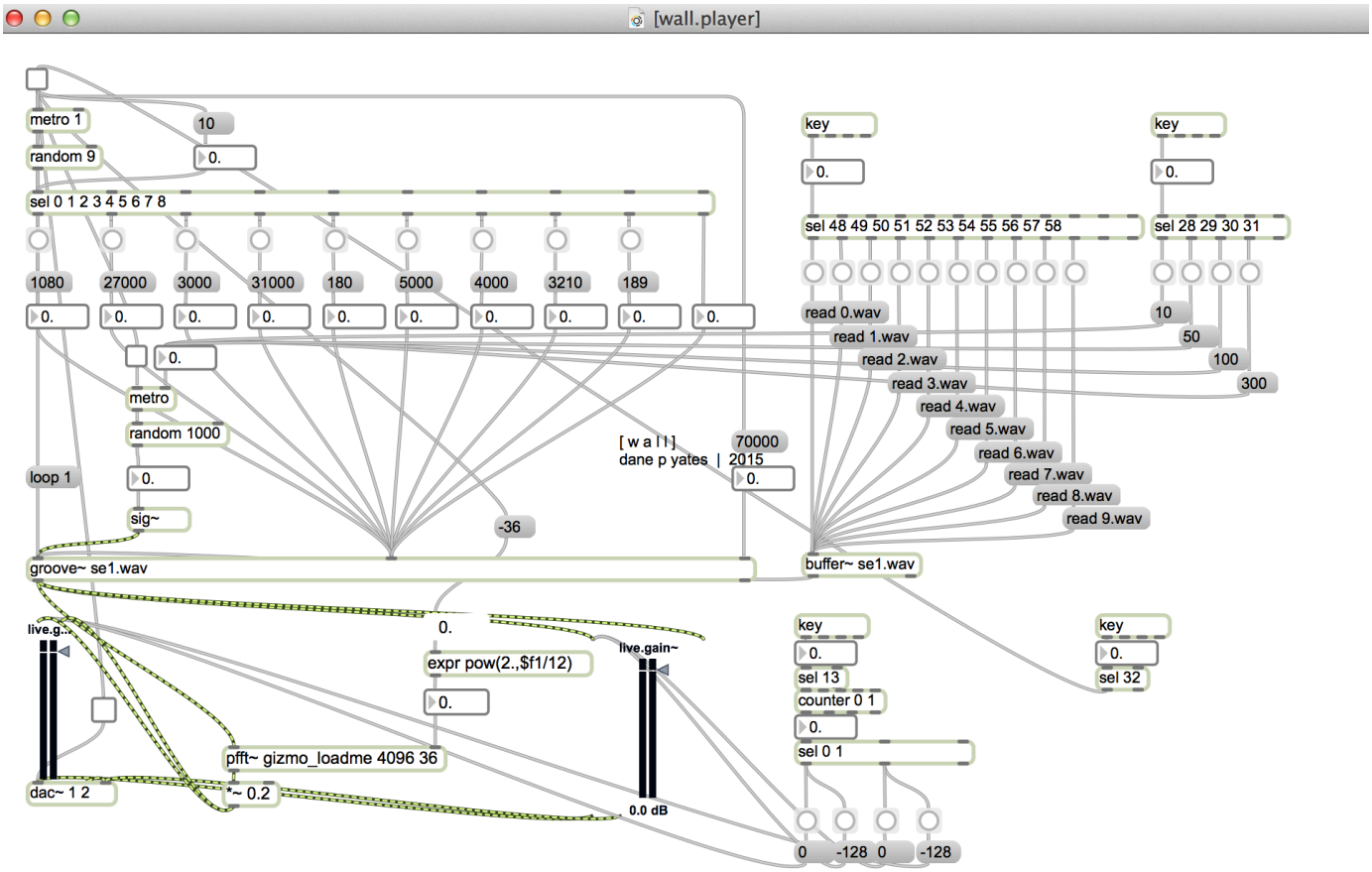

Figure III. Score Excerpt from [w a l l]