

Brackets [after John Cage] (2014)
For four or more players with computer direction
Nicolas Collins

Inspired by the late “time bracket” compositions of John Cage, *Brackets* is intended primarily for instruments whose sounds and performance gestures are ill-served by traditional Western notation: “Bent” and homemade circuits; patchcord synthesizers; samplers; computer software; DJs setups; contact-miked objects; electric instruments with effect processors; pitch-indeterminate acoustic instruments, etc. A minimum of four musicians is needed, and there is no maximum size of the ensemble.

Each musician selects a number of distinct “gamuts” (between 4-9, the quantity determined by consensus of the group) that can be produced on her or his instrument. In the spirit of Cage’s *Four*⁶, these are different sounds “with fixed characteristics”. These gamuts could be determined by: the choice of a Bent circuit with a limited but identifiable sound signature (i.e., a Speak & Spell); the patching and settings of analog synthesizer modules; a prepared sample file; a software algorithm with specific parameter settings; the choice of object to be amplified with a contact mike; settings of an effect processor on an electric guitar; a specific percussion instrument.

There should be variation within each gamut – it need not be a steady-state signal like a drone, nor need sound patterns be repeated exactly on every iteration – but it must have enough coherence to be distinguishable from the player’s other gamuts, and be identifiable as the same gamut on each repetition. For example, one gamut might be a “clean” setting on an electric guitar, while another could entail heavy distortion, and different musical passages can be played on each repetition of the gamut. Gamut choices that allow such internal variation and “playability” are preferred over drones, static sounds and pure repetition.

Each performer assigns each gamut a number. There is no coordination between players in the matching of sounds to numbers -- one musician’s “3” could be a bird sample while another’s “3” might be a burst of white noise.

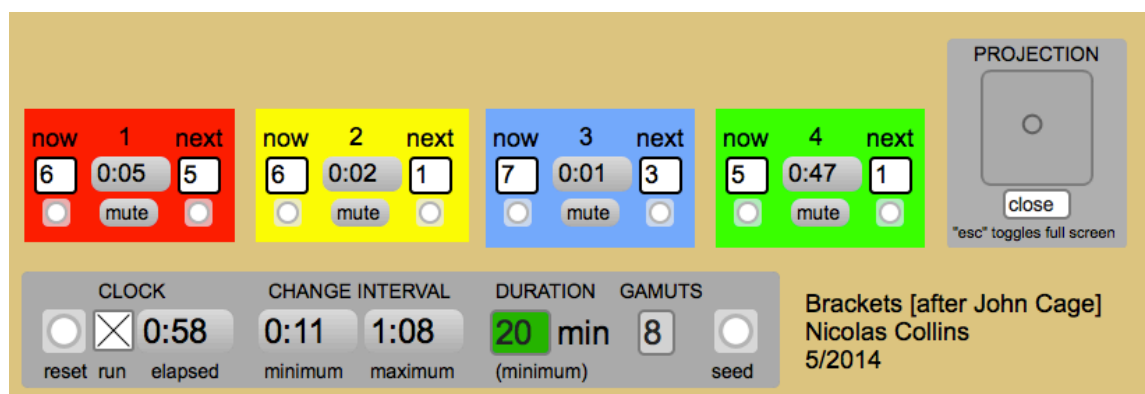
The computer program randomly selects amongst the gamut numbers and assigns each occurrence a time interval (bracket) in which it is to be played; the time counts down, and at the end of the interval a new gamut is chosen. This process is applied to four groups of players, with each gamut choice and bracket countdown displayed on a video screen in a color-coded block. To aid in preparing for each change the next gamut (“on deck”) is displayed as well as the currently active one (this is clarified in the description of software below.)

Each player is free to perform at any time within the available time bracket – filling an entire 54-second interval with continuous sound, for example, or producing a one-second spike somewhere midway through the elapsed time. The player should *perform* within the bracket, rather than merely mark time – this is an occasion for expressive improvisation within specific time constraints.

SOFTWARE SETUP

The software is delivered as a MaxPlay application. The “Brackets” folder contains both the program (with a name like “brackets080814.1.app”) and a file of settings (“bracketsettings.json”). Make sure both files remain in the same folder, which can be located anywhere on your computer. No audio connection is needed, but some form of remote video projector or large monitor is needed for the performers to follow. Use your computer’s System Preferences to insure that the remote display is not set to “mirror mode”, since different software panels appear on the computer and the remote screen.

Double-click the .app and you should see a window that looks something like this:



The four colored blocks in the upper half of the screen indicate the status of the four groups of players. On booting the boxes below **now** and **next** will be black, but as soon as the program starts to run the current gamut number will appear under **now** and the upcoming one under **next**. The numeric display between **now** and **next** counts down the remaining time until the next change. The button under each gamut box selects a new random value; **mute** sets both gamut boxes to black, which is the indication for silence.

The horizontal panel below the color blocks controls the timing of the performance. Click the **run** box under **CLOCK** at the left to start the performance; click again to pause. The **elapsed** time is indicated in the number box. **Reset** resets the counter to zero and clears all performance instructions to black (silence.)

Click and drag the numbers in the **CHANGE INTERVAL** boxes to set the **minimum** and **maximum** duration of the time between changes in performance instructions. Whenever a timer reaches zero (as displayed in the colored boxes above) the program will randomly choose a new duration within the **minimum** and **maximum** values set here, as well pass the **next** instruction to **now** and generate a new **next** instruction (like a bucket brigade.)

Click the **DURATION** indicator to select the minimum duration of the performance via a drop-down menu. When the **elapsed** time reaches this value the **DURATION** background changes from green to red and the program starts to mute the performance instructions as each one times out. The shutdown process can take a few minutes to complete, and the actual performance duration can be as long as this **DURATION** time plus the time set as **maximum** in the **CHANGE INTERVAL** control box.

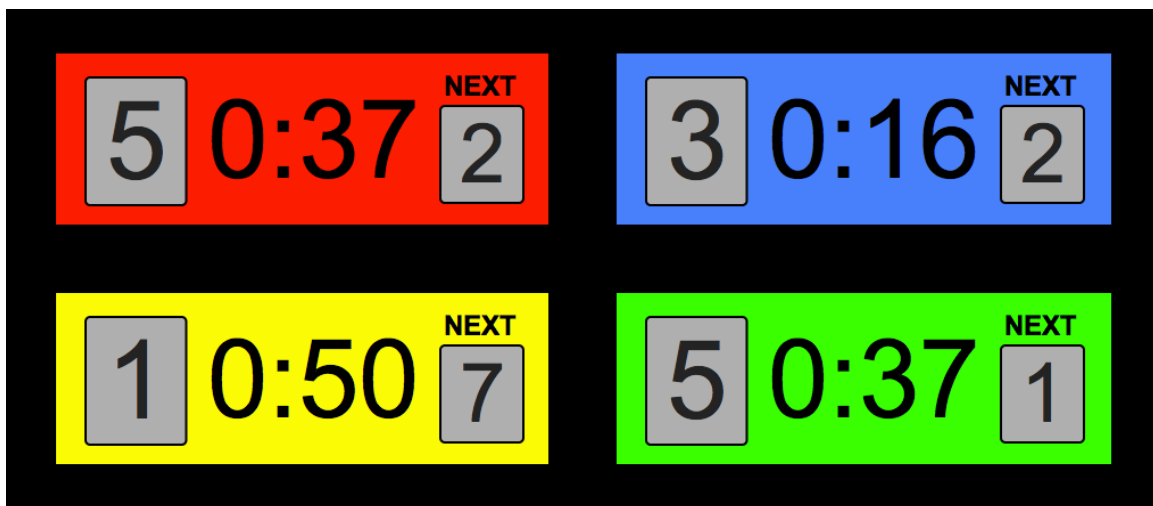
The menu below the **GAMUTS** legend sets the number of gamuts to be used in the performance (between 4 and 9). All players must have the same number of gamuts.

The **seed** button at the right end of this strip randomly assigns **now** and **next** values to all active groups, useful to start the performance.

THIS IS THE MAIN PANEL FOR SETTING THE CHARACTERISTICS OF THE PERFORMANCE, typically before starting (it should not be necessary to adjust these settings during the performance.) Any changes made to the values in this panel are saved when you exit the program and restored when you start up again.

The **PROJECTION** panel controls the **REMOTE DISPLAY** that directs the musicians. Toggle the **close/open** button to open the remote display. Drag this new window across the desktop to the projector screen. The "escape" key on your computer toggles full-screen mode on and off for the remote display. The circle in the square under the **position** legend can be clicked and moved to center the remote display on the projector screen. The usual CMD+ and CMD- key combinations change the size of the **REMOTE DISPLAY**.

The **REMOTE DISPLAY** should look something like this when the program is running:



The colored panels duplicate the information on the four colored gamut displays on the main control screen, each one referencing one group. The large square at the left of each panel indicates the current performance instruction (**now**). The smaller square at the right indicates the **next** instruction, so each performer can prepare for the upcoming change. The numbers in the center count down the time until the change. A black square indicates tacit (silence.)

The simplest solution to conveying the instructions to the players is to connect the computer's video output to a data projector, and position the screen so it is visible to all the musicians (and probably the audience as well.) Alternatively one can split the video output to multiple monitors placed strategically in front of the players.

PERFORMANCE

Prior to performing the musicians as a group set the global variables as described in the previous paragraphs: total duration, minimum and maximum values of time brackets, number of gamuts, etc.

Individually each musician selects a sound for each of the gamut choices as described at the beginning of this document. (All players must have the same number of gamuts.)

When four players are performing each selects one color. When more than four musicians are present they divide into color groups (the groups do not have to be the same size).

Confirm that the video projection is working and visible to the musicians. Toggle **run** to start the performance (**seed** the initial **now** and **next** values if necessary.)

It is not necessary or desirable to play for the entirety of a time bracket – incorporate pauses, rests, silences. Use each time interval as a “bracket” marking limits, rather than an imperative to produce continuous sound -- perform inventively within these boundaries. Remember to be silent when the now gamut display is black.

The performance ends when all **now** and **next** displays are black and the **CLOCK** shuts off (typically c. **DURATION** + **maximum**.)

BACKGROUND

In 1998 Susan Stenger asked me to participate in a performance John Cage's *Four*⁶ (1992) at the Barbican in London. This composition asks each of four players to “choose twelve different sounds with fixed characteristics” and play them within “flexible time brackets” specified in the score. These instructions are surprisingly accommodating of non-traditional electronic instruments, and I was struck by how, by the end of the 30-minute performance, twelve bizarre sounds had acquired the functional familiarity that we associate with chords or pitches in a tonal musical form.

15 years later Veniero Rizzardi suggested that I arrange a performance of the same composition by members of a workshop in hardware hacking that I was presenting at the Università Ca' Foscari in Venice. Once again I noticed how well the work functioned with the barely-controllable sounds of novice hackers.

The experience prompted me to take a closer look at Cage's and Christian Wolff's use of sound "gamuts" in their work from the early 1950s. To quote Michael Hicks and Christian Apslund from their book, *Christian Wolff*,

A 'gamut' was simply a collection of sounds, each an individual pitch or noise or clump of either. These gamuts would take the place of scales...The 'melodies' of gamuts...constituted the whole substance of new works.

As Kyle Gann explained in *No Such Thing As Silence – John Cage's 4'33"*,

The technique had its roots in [Cage's] work with percussion and the prepared piano. Unlike, say, the string orchestra, which offers a vast and continuous range of harmonic and textural possibilities, an ensemble of unpitched percussion tends to comprise a fixed group of sounds; one can strike a gong or hit one of five variously sized brake drums, but neither instrument offers a variety of harmonic progressions. The sounds available are what they are.

"Sounds [as] what they are" is a fair assessment of the output of many handmade electronic instruments, so an invitation to pursue an extended residency with TárabusT in Valparaíso (an ensemble of improvisers incorporating electronics) was the nudge I needed to write software to compose "gamut melodies" – essentially a program to generate Cagean "time-brackets" on the fly. My first attempt (*Gamut*, 2013) eventually split into two related programs: this one, *Brackets*, applies Cagean time-brackets to non-traditional instruments such as hacked circuits and patchcord synthesizers; another takes a more Wolffian approach to notation and is better suited to more traditionally "playable" instruments (*Bracken*, 2014.)

West Falmouth August 2014