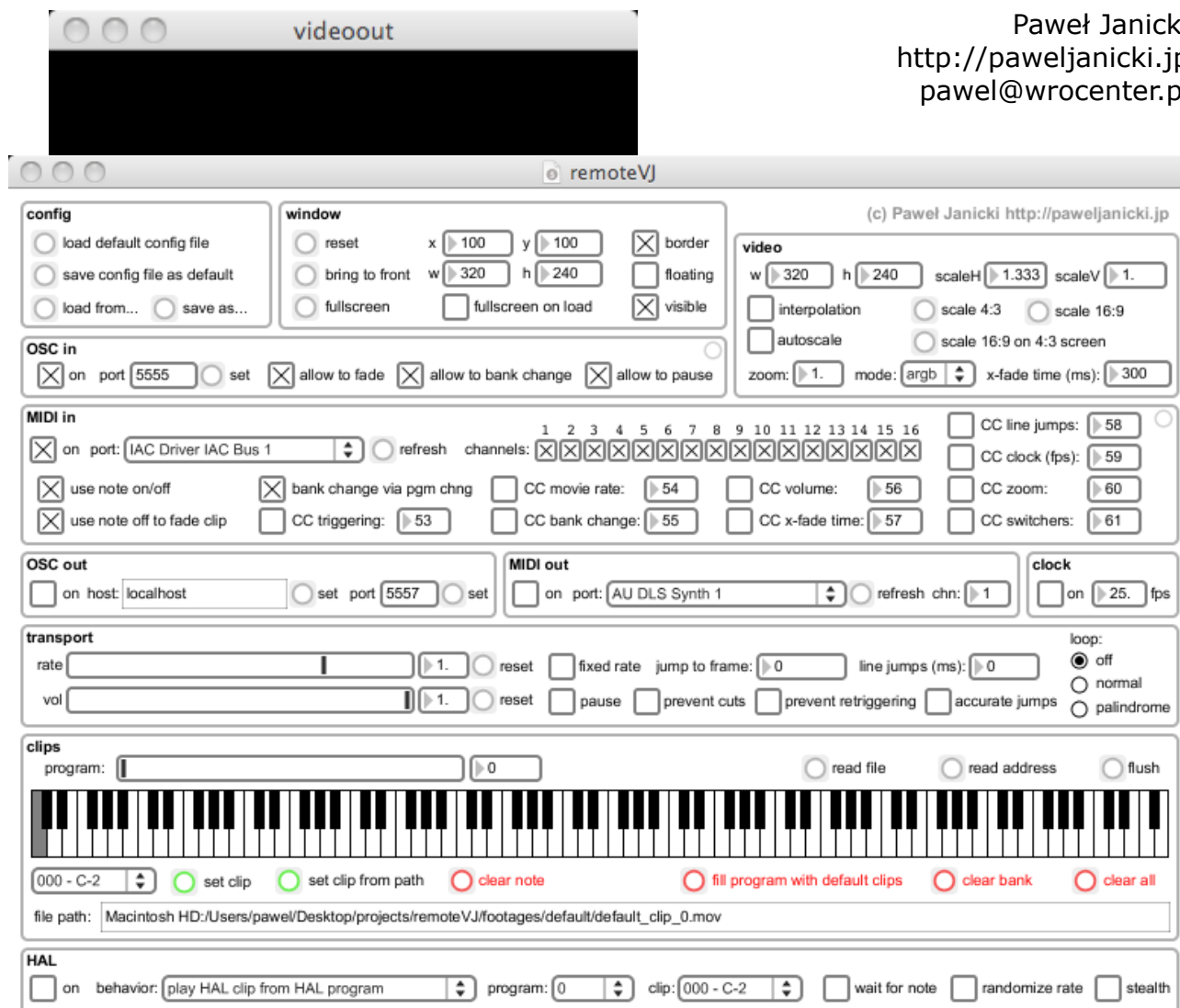


remoteVJ

wprowadzenie

Paweł Janicki
<http://paweljanicki.jp>
pawel@wrocenter.pl



remoteVJ to prosty, stworzony przede wszystkim do celów edukacyjnych, ale zarazem i funkcjonalny projekt (*patch*) napisany w środowisku MaxMSP/Jitter. Służy do kontrolowanego poprzez MIDI lub OSC (*OpenSound Control*, więcej informacji o protokole OSC: <http://opensoundcontrol.org>) odtwarzania klipów video (jakkolwiek możliwe jest też odtwarzanie za jego pomocą pojedynczych grafik lub plików audio oraz korzystanie z mediów strumieniowych).

Aplikacja obsługuje tryb pełnoekranowy (*fullscreen*) i zawiera narzędzia umożliwiające wykorzystanie wyświetlaczy o nietypowych proporcjach lub rozdzielczościach.

remoteVJ umożliwia stworzenie 128 programów grupujących po 128 klipów video i zdalne (za pomocą MIDI lub OSC) przełączenie się pomiędzy programami oraz – oczywiście – wyzwalamie klipów oraz kontrolę tempa i kierunku wyzwalamia, trybu zapętlania, itp. **remoteVJ** zawiera też kilka prostych mechanizmów ułatwiających jego zastosowanie w projektach bardziej skomplikowanych, niż typowe wizualizacje, m.in. blokowanie odtwarzania kolejnych klipów, jeśli poprzedni nie został ukończony oraz funkcję wysyłania przez MIDI i OSC informacji o statusie odtwarzacza.

Licencja

GNU General Public License 3.0: <http://www.gnu.org>

Instalacja i wymagania systemowe

remoteVJ działa zarówno na Mac OSX jak i MS Windows – wymaga jednak zainstalowanego w systemie środowiska MaxMSP/Jitter (5.x) w wersji pełnej, demonstracyjnej lub *runtime* (wersja demonstracyjna działa bez żadnych ograniczeń przez 30 dni, *runtime*, czyli „player” dla projektów stworzonych w MaxMSP/Jitter dystrybuowany jest nieodpłatnie i bez ograniczeń czasowych), które można pobrać ze strony producenta: <http://cycling74.com> lub zakupić (w przypadku pełnej wersji) u dystrybutora.

Dla systemów z rodziny Windows potrzebny będzie jeszcze Apple QuickTime, dostępny nieodpłatnie na stronach producenta, firmy Apple: <http://apple.com>. MaxMSP/Jitter, a co za tym idzie i **remoteVJ**, pozwala na korzystanie z – również nieodpłatnego – QuickTime Alternative, zamiast Apple QuickTime. Więcej informacji o QuickTime Alternative: http://en.wikipedia.org/wiki/QuickTime_Alternative.

Poza wymienionymi komponentami **remoteVJ** nie wymaga żadnych specjalnych zabiegów – wystarczy skopiować folder z aplikacją w wybrane miejsce na dysku. Dwukrotne kliknięcie na ikonie pliku „remoteVJ.maxpat” uruchomi projekt.

Formaty mediów

remoteVJ odczytuje klipy video za pomocą QuickTime, rozpoznaje więc formaty typowe dla tego ostatniego. Natywnym formatem (w zasadzie kontenerem) dla QuickTime jest „.mov”, ale pracując z **remoteVJ** można korzystać też z klipów video w formacie „.avi”, plików audio „.wav” i „.aiff” oraz – w zależności od systemu operacyjnego i zainstalowanych w nim pluginów – np. plików „.mpeg”, grafik w formatach „.png”, czy „.jpg”, itd. Oczywiście w ramach obsługiwanych kontenerów (np. „.mov”) istotne są wykorzystane kodeki, szereg parametrów kompresji, rozdzielczość i klatkaż. Ponadto aplikacja potrafi obsłużyć media strumieniowe w formatach zgodnych z QuickTime.

Optymalna struktura projektu

W przeciwieństwie do poprzednich wersji aplikacji **remoteVJ** posiada teraz mechanizmy umożliwiające tworzenie projektów, które łatwo archiwizować i przenosić pomiędzy np. różnymi komputerami (do tej pory mogło być to kłopotliwe ze względu na zapamiętywane jako „absolutne” ścieżki dostępu do plików).

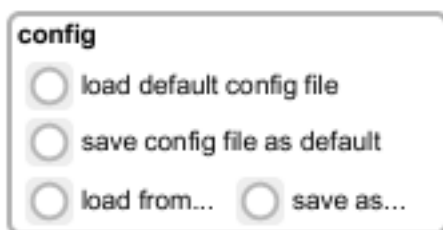
W folderze z aplikacją znajduje się teraz folder „footages” przeznaczony do przechowywania plików mediów, z których korzysta **remoteVJ** (oczywiście ciągle można dodawać do projektu pliki z dowolnych lokalizacji, jednak stosowanie folderu „footages” daje pewne dodatkowe możliwości opisane poniżej). Standardowo w folderze tym znajdują się dwa podfoldery „default” i „custom”. Podfolder „default” zawiera zestaw 128 plików służących do celów testowych, natomiast podfolder „custom” jest pusty – proponuję umieszczać pliki mediów, które zamierzamy dołączyć do projektu właśnie w podfolderze „custom” (własne pliki możemy organizować w kolejne podfoldery, również zagnieżdżone – pliki wchodzące w skład projektu można więc dość swobodnie porządkować).

remoteVJ traktuje pliki umieszczone w tym podfolderze w specjalny sposób: jeśli do któregoś z klawiszy wirtualnej klawiatury przypiszemy plik znajdujący się w podfolderze „custom”, aplikacja zapamięta ścieżkę dostępu do pliku jako relatywną w stosunku do położenia aplikacji na dysku (innymi słowy: jeśli skopiujemy pliki z podfolderu „footages/custom” folderu z **remoteVJ** zainstalowanego na naszym komputerze do analogicznego podfolderu folderu z **remoteVJ** na jakiejś innej maszynie, to **remoteVJ** uruchomiony na tej drugiej maszynie odwoła się do „własnego” folderu „footages/custom” i tam „sięgnie” po pliki przypisane do poszczególnych klawiszy). Brzmi to może nieco zawile, jednak – jeśli tylko poświęcimy chwilę na jego zrozumienie – mechanizm ten okazuje się w rzeczywistości bardzo prosty i użyteczny.

Interfejs

Po otwarciu **remoteVJ** wyświetla dwa okna: główne, zawierające kontrolki interfejsu użytkownika i okno służące do wyświetlania klipów video (okno „videoout”). Interfejs głównego okna projektu podzielony jest na sekcje grupujące tematycznie poszczególne parametry. Interfejs programu jest w zasadzie jednoekranowy (jeśli pominiemy występujące w kilku miejscach rozwijalne menu). Opisy poszczególnych sekcji znajdują się poniżej.

config



Przyciski dostępne w tej sekcji umożliwiają zapisanie lub załadowanie zapisanej konfiguracji. Konfiguracja obejmuje takie parametry jak częstotliwość odświeżania i stan zegara, parametry okna video (pozycja, wielkość, itd.), zapętlenie, odtwarzanie audio, port, kanały i inne parametry MIDI, porty OSC – krótko mówiąc: wszystkie parametry ustawiane za

pomocą elementów interfejsu użytkownika znajdujących się w głównym oknie **remoteVJ**. Przede wszystkim jednak zapisywane i ładowane są informacje o klipach video przypisanych do poszczególnych klawiszy wirtualnej klawiatury (więcej informacji o tym znajduje się w sekcji **clips**).

Pliki konfiguracyjne mają postać zwykłych plików tekstowych (łącznie z rozszerzeniem „.txt”) – w razie potrzeby można je edytować „ręcznie” za pomocą dowolnego edytora tekstu (może to mieć sens, jeśli np. chcemy masowo zmienić ścieżki dostępu do wszystkich plików projektu). W razie uszkodzenia domyślnego pliku konfiguracyjnego (co może się wydarzyć w przypadku samodzielnej edycji pliku za pomocą edytora tekstu, czyli z pominięciem interfejsu programu) można zastąpić go plikiem „config_arch.txt” zmieniając temu ostatniemu nazwę na „config.txt” – plik „config_arch.txt” warto jednak skopiować i zachować na przyszłość.

Przycisk *load default config file* umożliwia załadowanie domyślnego pliku konfiguracyjnego (tym samym aktualne parametry projektu zostaną utracone – oczywiście możemy wcześniej zapisać aktualny stan aplikacji).

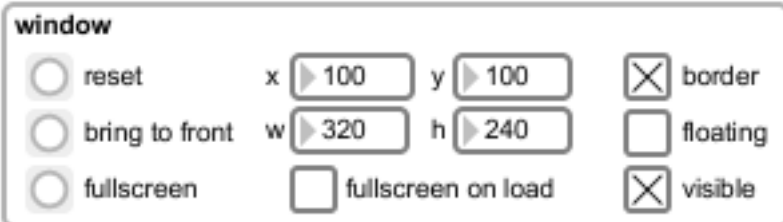
Przycisk *save config file as default* służy do zapisania aktualnego stanu aplikacji do domyślnego pliku konfiguracyjnego (podobnie jak w przypadku poprzedniej kontrolki, warto przemyśleć jego zastosowanie, zanim nadpiszemy wartościowe dla nas ustawienia). Ostatnio zapisana za pomocą przycisku *save config file as default* konfiguracja jest ładowana podczas startu **remoteVJ**.

Przycisk *load from...* umożliwia wczytanie pliku konfiguracyjnego z dowolnej lokalizacji (po wciśnięciu przycisku zostanie wyświetlone okno umożliwiające wybór interesującego nas pliku).

Przycisk *save as...* pozwala nam na zapisanie aktualnych ustawień programu do pliku konfiguracyjnego o dowolnej nazwie i w dowolnym miejscu (po wciśnięciu przycisku zostanie wyświetlone okno umożliwiające wybór interesującego nas pliku).

UWAGA: pełne obłożenie klipami wszystkich 128 programów (banków) może zauważalnie (ale nie drastycznie) wydłużyć czas ładowania pliku konfiguracyjnego.

window



Sekcja umożliwiająca ustawienie parametrów okna wyświetlającego pliki video („videoout”).

Przycisk *reset* umożliwia powrót do standardowych

ustawień wielkości i pozycji okna wyświetlającego klipy video (np. jeśli eksperymentując z pozycją i wielkością okna „videoout” wyprowadzimy je poza powierzchnię wyświetlacza).

Przycisk *bring to front* powoduje przemieszczenie okna video ponad pozostałe otwarte okna.

Kliknięcie przycisku *fullscreen* powoduje przejście do trybu pełnoekranowego (wciśnięcie klawisza „Esc” spowoduje powrót do podstawowego trybu wyświetlania). Z przyciskiem *fullscreen* skojarzone jest pole wyboru *fullscreen on load* – jeśli je zaznaczymy i zapiszemy konfigurację aplikacji jako domyślną (przyciskiem *save config file as default*), to po ponownym otwarciu aplikacji **remoteVJ** powinien automatycznie przejść do trybu pełnoekranowego.

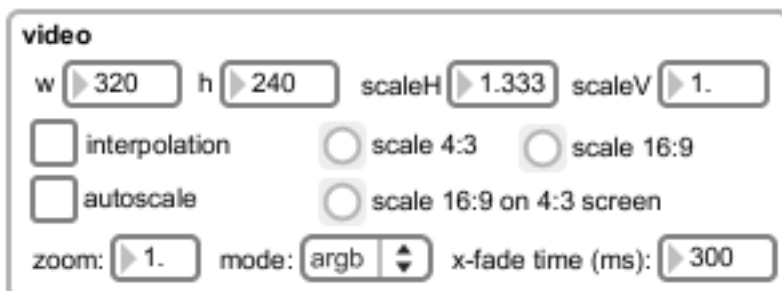
Grupa czterech pól numerycznych opisanych *x*, *y*, *w*, *h* umożliwia kontrolę położenia i wielkości okna video (oczywiście okno to możemy przemieszczać również za pomocą myszy) – w połączeniu z możliwością ukrycia ramki wokół okna video (pole wyboru *border*) daje to możliwość np. zasymulowania trybu pełnoekranowego w przypadku wykorzystania nietypowych wyświetlaczy.

Pole wyboru *border* pozwala na włączenie lub wyłączenie ramki wokół okna video.

Pole wyboru *float* włącza lub wyłącza tryb *floating*.

Pole wyboru *visible* dopowiada za ukrywanie lub wyświetlanie okna „videoout”.

video



Sekcja odpowiedzialna za parametry odtwarzania plików video.

W przeciwieństwie do poprzednich wersji programu **remoteVJ** nie dostosowuje sztywno proporcji wyświetlanego obrazu do

proporcji okna „videoout”: teraz obraz w oknie może mieć dowolną proporcję wysokości do szerokości (ponieważ operacja przeskalowywania odbywa się po stronie karty graficznej, aplikacja niejako „przy okazji” zyskała nieco na sprawności).

Dwa umieszczone w tej sekcji pola numeryczne opisane *w* i *h* umożliwiają zdefiniowanie (poziomej [*w*] i pionowej [*h*]) rozdzielczości odtwarzanych klipów video (jeśli klip zapisany jest w innej rozdzielczości, zostanie przeskalowany do wielkości ustalonej wartościami wpisanymi w polach numerycznych).

Ponieważ rozdzielczość silnie wpływa na obciążenie procesora warto w miarę możliwości trzymać się jak najniższych wartości parametrów *w* i *h*. Większość osób pracujących z video w czasie rzeczywistym korzysta z rozdzielczości 640 x 480 pixeli lub nawet o połowę mniejszej (320 x 240 pixeli) – dotyczy to przypadków, kiedy nie stosuje się wyświetlaczy (szczególnie projektorów) HD. Z drugiej strony: przeskalowanie video również jest procesem w jakimś stopniu obciążającym procesor, więc w niektórych przypadkach może się okazać, że lepiej trzymać się rozdzielczości oryginalnych klipów. Optymalny zestaw ustawień najlepiej dobrać eksperymentalnie.

Kolejne dwa pola numeryczne *scaleH* i *scaleV* służą do ustawienia wielkości i proporcji obrazu wyświetlanego w oknie „videoout” (w celu odwrócenia obrazu w pionie lub poziomie można stosować ujemne wartości parametrów *scaleH* i *scaleV*). Pola te można edytować manualnie lub skorzystać z jednego z trzech umieszczonych poniżej przycisków: *scale 4:3*, *scale 16:9* bądź *scale 16:9 on 4:3 screen* – przyciski te umożliwiają ustawienie skali dla typowych plików video i wyświetlaczy.

Pole wyboru *interpolation* pozwala włączyć lub wyłączyć tryb interpolacji podczas nakładania klatek odtwarzanego klipu na panel video.

Pole wyboru *autoscale* pozwala włączyć lub wyłączyć automatyczne dostosowywanie rozmiaru panelu video do rozmiaru okna „videoout”.

Kontrolka *scale 4:3* pozwala ustawić proporcję szerokości do wysokości panelu video na odpowiadającą obrazowi w standardzie 4:3.

Kontrolka *scale 16:9* pozwala ustawić proporcję szerokości do wysokości panelu video na odpowiadającą obrazowi w standardzie 16:9.

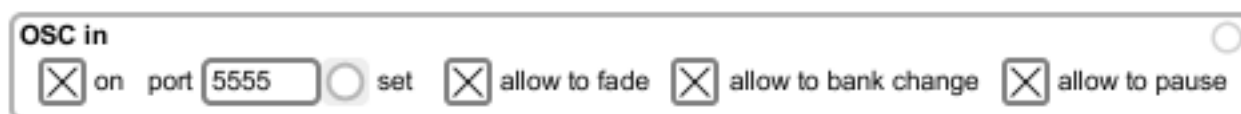
Kontrolka *scale 16:9 on 4:3 screen* pozwala ustawić proporcję szerokości do wysokości panelu video na odpowiadającą obrazowi w standardzie 16:9, ale jednocześnie wielkość panelu jest dostosowana do wyświetlacza o proporcji 4:3 (dzięki temu na ekranie mieści się cały obraz o proporcji 16:9).

Pole numeryczne *zoom* umożliwia zmianę wielkości wyświetlanego obrazu bez zmiany proporcji i rozdzielczości (także i tutaj można stosować ujemne wartości parametru, celem jednoczesnego obrócenia obrazu w pionie i poziomie).

Menu rozwijalne *mode* służy do wyboru sposobu dekodowania kolorów pikseli strumienia wideo. Do wyboru mamy tryby *argb* i *uyvy*.

Pole *x-fade time (ms)* odpowiada za czas przenikania (podawany w milisekundach) pomiędzy klipami video: jeśli **remoteVJ** odbierze komunikat MIDI lub OSC powodujący odtworzenie nowego klipu video, to przenikanie od klipu aktualnie odtwarzanego do nowego nastąpi w czasie ustalonym w tym właśnie polu. Ustawienie czasu przenikania na „0” (zero) – będzie powodowało natychmiastowe przejście do nowego klipu.

OSC in



Sekcja analogiczna do MIDI in, ale związana z protokołem OSC (*OpenSound Control*).

Jeśli zamierzamy skorzystać z możliwości sterowania aplikacją *via* OSC wystarczy zaznaczyć pole wyboru *on* i wybrać numer portu (klikając w przycisk *set* – pojawi się okno, w którym możemy wpisać numer portu i potwierdzić lub anulować wybór).

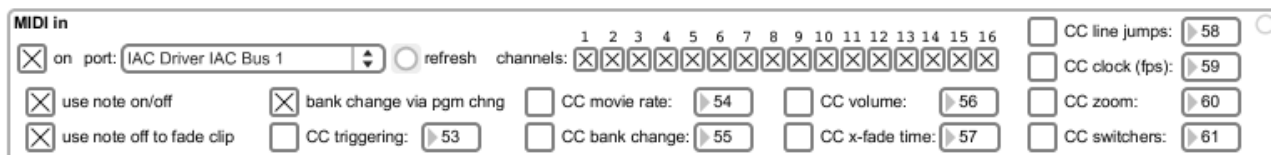
Zaznaczenie pola *allow to fade* umożliwia wyłączanie odtwarzania klipów (w sposób analogiczny do tego, jak mogą zostać wykorzystane komunikaty MIDI *note off*) za pomocą komunikatów OSC. Zasadniczo pole to, jeśli nie jest zaznaczone, powoduje blokowanie komunikatów OSC odpowiedzialnych za wyłączanie odtwarzania.

Pole *allow to bank change* odpowiada za możliwość zmiany aktualnego programu, a *allow to pause* za wstrzymywanie i przywracanie odtwarzania za pomocą OSC (pauza).

Ponadto w prawym górnym rogu sekcji OSC in znajduje się kontrolka „błyszcząca” w momencie odebrania komunikatu.

Pełna lista komunikatów OSC odczytywanych przez **remoteVJ** znajduje się w dokumencie *OSC_and_MIDI_specs_pl* dołączonym do archiwum z wersją dystrybucyjną programu.

MIDI in



Sekcja grupująca parametry związane z odbiorem komunikatów MIDI.

Sterowanie **remoteVJ** poprzez MIDI można zablokować (lub włączyć) polem wyboru *on* – oczywiście, jeśli wyłączymy **MIDI in** (kontrolką *on*) mamy jeszcze możliwość kontrolowania aplikacji przez OSC (zasadniczo OSC oferuje więcej możliwości, więc –

jeśli mamy taką możliwość warto skorzystać z tego protokołu zamiast MIDI) i graficznego interfejsu użytkownika (czyli za pomocą myszy).

Za pomocą (umieszczonego w lewej górnej części sekcji) menu *port* można wybrać port MIDI *in*, z którego będzie korzystał **remoteVJ**, a za pomocą umieszczonych obok pól wyboru opisanych *channels* można wybrać interesujący/e nas kanał/y wejścia MIDI. Przycisk *refresh* odświeża listę dostępnych portów MIDI.

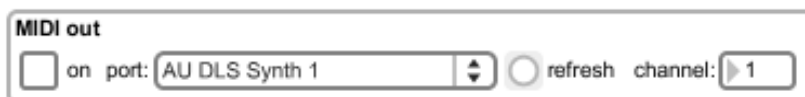
remoteVJ nasłuchuje również komunikatów *MIDI Time Code (MTC)*. Reaguje na komunikaty *start*, *stop* oraz *continue* (komunikat *continue* traktowany jest przez aplikację tak samo jak *start*) – jeśli do wybranego (menu *port*) portu MIDI kierowane są komunikaty *MTC*, możemy skorzystać z nich do zatrzymywania i wznowiania odtwarzania klipów video.

Pozostałe kontrolki w sekcji **MIDI in** umożliwiają wykorzystanie komunikatów *control change (CC)* do sterowania różnymi funkcjami aplikacji. Aby skorzystać z tej możliwości należy zaznaczyć pole wyboru i wybrać numer komunikatu (wpisując odpowiedni parametr w pole numeryczne) *CC*, z którego chcemy skorzystać.

Ponadto w prawym górnym rogu sekcji MIDI in znajduje się kontrolka „błyszcząca” w momencie odebrania komunikatu.

Pełna lista komunikatów MIDI odczytywanych przez **remoteVJ** znajduje się w dokumencie *OSC_and_MIDI_specs_pl* dołączonym do archiwum z wersją dystrybucyjną programu.

MIDI out



Sekcja grupująca parametry związane z możliwością wysyłania przez **remoteVJ**

komunikatów MIDI informujących o stanie aplikacji i odtwarzanych aktualnie mediach.

Transmisję komunikatów można zablokować (lub włączyć) polem wyboru *on* – oczywiście, nawet jeśli wyłączymy *MIDI out* możemy ciągle wysyłać komunikaty OSC (zasadniczo OSC oferuje więcej możliwości, więc – jeśli mamy taką możliwość – warto skorzystać z tego protokołu zamiast MIDI).

Za pomocą (umieszczonego w lewej części sekcji) menu *port* można tu wybrać port MIDI *out*, z którego będzie korzystał **remoteVJ**, a za pomocą pola numerycznego *channel* można wybrać kanał wyjściowy MIDI. Przycisk *refresh* odświeża listę dostępnych portów MIDI.

Pełna lista komunikatów MIDI transmitowanych przez **remoteVJ** znajduje się w dokumencie *OSC_and_MIDI_specs_pl* dołączonym do archiwum z wersją dystrybucyjną programu.

OSC out



Sekcja grupująca parametry związane z możliwością wysyłania

przez **remoteVJ** komunikatów OSC (analogicznie jak w przypadku sekcji **MIDI out**).

Transmisję komunikatów można zablokować (lub włączyć) polem wyboru *on* – oczywiście, nawet jeśli wyłączymy *OSC out* możemy ciągle wysyłać komunikaty MIDI (patrz sekcja **MIDI out**).

Komunikaty OSC wysyłamy do adresata identyfikowanego przez dwa parametry: *host* (zazwyczaj podawany w formie numeru IP) i *port* – oba te parametry możemy ustawić klikając odpowiednie przyciski *set*.

Pełna lista komunikatów OSC transmitowanych przez **remoteVJ** znajduje się w dokumencie *OSC_and_MIDI_specs_pl* dołączonym do archiwum z wersją dystrybucyjną programu.

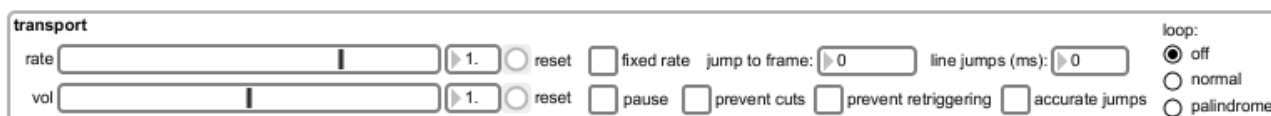
clock



Ta sekcja pozwala na ustawienie częstotliwości odświeżania ekranu w oknie video oraz włączenie lub wyłączenie zegara. Częstotliwość odświeżania nie wpływa na prędkość odtwarzania klipów video.

Typowe częstotliwości odświeżania mieszczą się pomiędzy 12 a 30*fps*, ale nic nie stoi na przeszkodzie aby – jeśli dysponujemy materiałem video zarejestrowanym z większą ilością klatek na sekundę i odpowiednio wydajnym komputerem – wykorzystać większe wartości. W przeciwieństwie do poprzednich wersji aplikacji parametr *fps* może przyjmować wartości zmiennoprzecinkowe.

transport



Sekcja grupuje parametry związane z odtwarzaniem klipów video (lub innych mediów). Ze względu na specyfikę **remoteVJ** sekcja **transport** wygląda i działa nieco inaczej, niż typowe funkcje transportu.

Suwak i pole numeryczne opisane *rate* służą do ustawienia szybkości i kierunku odtwarzania. Dostępne wartości parametru *rate* mieszczą się w przedziale od „-2” do „2” – wartości mniejsze od „0” (zero) oznaczają, że klip odtwarzany będzie od końca do początku (jakkolwiek w połączeniu z trybem *loop pendulum* zależności pomiędzy kierunkiem odtwarzania i parametrem *rate* mogą być nieco bardziej zawiłe).

Jeśli pole wyboru *fixed rate* nie jest zaznaczone wybranie nowego klipu do odtwarzania resetuje wartość parametru *rate* do „1” (bazowa prędkość i kierunek). Resetu możemy dokonać również manualnie, przyciskiem *reset* znajdującym się po prawej stronie suwaka i pola numerycznego *rate*, albo wysyłając do **remoteVJ** odpowiedni komunikat MIDI lub OSC.

Pole numeryczne *jump to frame* umożliwia natychmiastowy przeskok do wybranej klatki aktualnie odtwarzanego klipu. Umieszczone poniżej pole wyboru *accurate jumps* włącza tryb, w którym przejście do wybranej klatki odbywa się wolniej, ale w bardziej precyzyjny sposób. Przeskok do wybranej klatki klipu możemy również nakazać aplikacji zdalnie, za pomocą odpowiedniego komunikatu OSC.

Pole numeryczne *line jumps* umożliwia zdefiniowanie czasu płynnego przejścia pomiędzy aktualną klatką video, a klatką wskazaną w polu *jump to frame* (lub odpowiednim komunikatem OSC). Wartość „0” (zero) powoduje oczywiście natychmiastowy przeskok do wybranej klatki. Zazwyczaj *line jumps* wykorzystać można najlepiej w sytuacji, kiedy włączone jest pole *pause* (w przeciwnym razie jednocześnie działanie standardowej funkcji i *line jumps* nałożą się).

Po prawej stronie sekcji znajdziemy kontrolkę, dzięki której możemy wybrać tryb zapętlenia klipów video (*loop*). Do wyboru mamy wyłączenie zapętlenia (*off*), odtwarzanie w tradycyjnej pętli (*normal*) i tryb *palindrome*, w którym klip jest odtwarzany od początku do końca, następnie od końca do początku, itd (często dla tego trybu odtwarzania stosuje się nazwę „ping-pong”).

Pole wyboru *pause* wstrzymanie lub wznowienie odtwarzania (pauza).

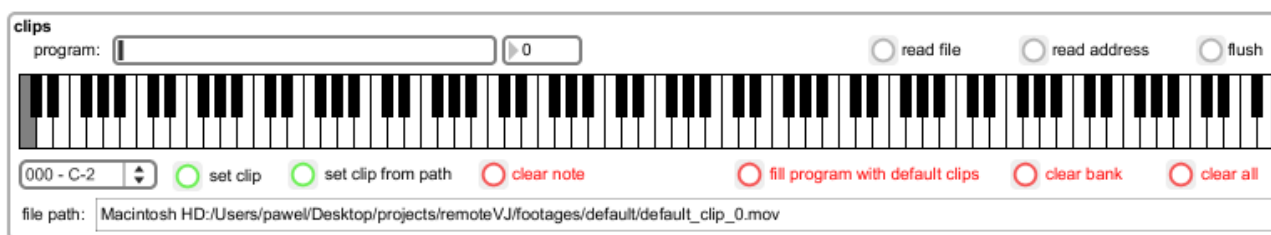
Suwak i pole numeryczne opisane *vol* umożliwia kontrolę poziomu dźwięku w odtwarzanym klipie (jeśli klip posiada ścieżkę dźwiękową). Przyciskiem *reset* znajdującym się po prawej stronie suwaka (albo wysyłając do **remoteVJ** odpowiedni komunikat MIDI lub OSC) możemy powrócić do bazowego poziomu głośności.

Pole wyboru *prevent retriggering* umożliwia zablokowanie wielokrotnego wyzwania tego samego klipu (włączona opcja *prevent retriggering* oznacza, że jeśli np. wielokrotnie naciśniemy ten sam klawisz klawiatury sterującej **remoteVJ** zareaguje tylko na pierwsze naciśnięcie klawisza i nie będzie „przeskakiwał” do początku odtwarzanego już klipu).

Pole wyboru *prevent cuts* umożliwia zablokowanie wyzwania jakiegokolwiek klipu do czasu ukończenia odtwarzania aktualnego.

Pole wyboru *accurate jumps* umożliwia włączenie lub wyłączenie trybu precyzyjnego (ale wymagającego nieco więcej czasu i mocy obliczeniowej) wyszukiwania klatki video, do której zamierzamy przejść w wyniku działania funkcji *jump to frame*.

clips



Ostatnia sekcja grupuje kontrolki związane z przypisaniami klipów video w ramach poszczególnych programów.

remoteVJ umożliwia stworzenie 128 programów grupujących po 128 klipów video i zdalne (za pomocą MIDI lub OSC) lub lokalne (suwak i pole numeryczne *program*) przełączenie się pomiędzy programami oraz – oczywiście – wyzwianie klipów i innych mediów. Wyzwalanie może zachodzić również bez dokonywania przypisań – adres medium można przekazać poprzez OSC lub za pomocą przycisków *read file* i *read address*.

Sprawność przełączania odtwarzanych klipów zależy zarówno od ogólnej wydajności (prędkości transferu z dysku, procesora, itp.) komputera, na którym uruchomiony jest **remoteVJ**, jak i formatu zapisu i sposobu kompresji klipów – zasadniczo współczesne komputery pozwalają wykorzystać aplikację w charakterze audiowizualnego instrumentu i przełączać się pomiędzy klipami bez uciążliwych „zawiesznień” i opóźnień.

Umieszczone w lewej górnej części sekcji suwak i pole numeryczne *program* umożliwiają wybór aktywnego programu (banku) przypisać klipów do klawiszy wirtualnej klawiatury.

Przyciski *read file* i *read address* umożliwiają bezpośrednie wyzwalanie mediów za pomocą okna wyboru (*read file*) lub poprzez wpisanie adresu medium (*read address*) – ta druga technika umożliwia m.in. dostęp do mediów strumieniowych rozpoznawalnych przez QuickTime.

Przycisk *flush* wyłącza odtwarzanie klipu (jeśli jakiś klip jest w danym momencie odtwarzany).

Widoczna poniżej klawiatura umożliwia wyzwalanie klipów przypisanych do konkretnych klawiszy lub wybranie klawisza do edycji – w jednym i drugim przypadku wystarczy po prostu kliknąć w interesujący nas klawisz – bądź też użyć komunikatów MIDI lub OSC. Wyboru klawisza do edycji możemy dokonać za pomocą rozwijalnego menu poniżej klawiatury (po lewej stronie, pomiędzy przyciskami *set clip* i *clear note*). 128 klawiszy klawiatury to program – za pomocą suwaka *program*, komunikatów MIDI lub OSC możemy aktywizować, a następnie edytować każdy ze 128 programów.

Przypisanie klipu video do wybranego uprzednio klawisza (w ramach aktywnego programu) nastąpi po kliknięciu w zielony przycisk *set clip* (umieszczony pod klawiaturą, po lewej stronie okna) – pojawi się wówczas okno umożliwiające wybór pliku. Po wskazaniu klipu i potwierdzeniu przypisanie zostanie dokonane, a w znajdującym się poniżej polu tekstowym *file path* pojawi się ścieżka dostępu do pliku. Jeśli teraz klikniemy w klawisz klawiatury ekranowej (albo wykorzystamy MIDI lub OSC) klip zostanie odtworzony w oknie „videoout” (jeśli to nie nastąpiło, najbardziej prawdopodobnym powodem może być wyłączony zegar w sekcji **clock**, włączona pauza, lub wybór pliku nierozpoznawalnego przez QuickTime).

Przypisanie można także dokonać przyciskiem *set clip from path* – wówczas do ostatnio wyselekcjonowanego klawisza zostanie przyporządkowane medium, którego adres widnieje aktualnie w polu *file path*.

Przypisanie możemy zresetować klikając przycisk *clear note*.

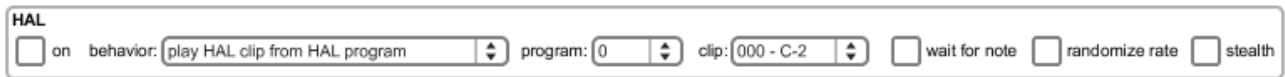
Kontrolka *fill program with default clips* umożliwia wypełnienie aktualnego programu dostarczonymi wraz z aplikacją klipami testowymi (naturalnie użycie tej funkcji jest równoznaczne ze skasowaniem aktualnych ustawień aktywnego programu).

Możemy też resetować (czyli pozbyć się wszystkich przypisań) aktualny program lub wszystkie 128 programów jednocześnie za pomocą klawiszy *clear bank* lub *clear all* (czerwone klawisze w prawej dolnej części sekcji) – z klawiszy tych, z oczywistych powodów, należy korzystać ostrożnie.

Zestaw przypisań jest zachowywany po kliknięciu *save* w sekcji **config** i przywoływany przyciskiem *load* lub w chwili startu programu. Ponieważ **remoteVJ** nie zapisuje przypisań automatycznie dobrze pamiętać o zachowaniu efektów swojej pracy przez

kliknięcie *save* przed zamknięciem aplikacji.

HAL



HAL był – o ile można tak powiedzieć o maszynie – jednym z bohaterów nakręconego w 1968 roku filmu Stanley'a Cubrica 2001: Space Odyssey (2001: Odyseja Kosmiczna). Obdarzony świadomością komputer HAL (nazwa pochodzi od liter poprzedzających w alfabecie litery w skrócie IBM) zajmować się miał kontrolą funkcjonowania statku kosmicznego – oczywiście faktyczna aktywność HALa przeszła najśmielsze oczekiwania konstruktorów.

HAL w **remoteVJ** jest modułem służącym do automatycznego wyzwalania klipów. Możemy go potraktować jako wewnętrzną pętlę sprzężenia zwrotnego – kiedy kończy się aktualnie odtwarzany klip, a włączyliśmy funkcję **HAL**, **remoteVJ** włączy odtwarzanie następnego klipu (jaki to będzie klip, zależy od ustawień dokonanych za pomocą kontrolki sekcji).

Kontrolka *on* (po lewej stronie sekcji) służy oczywiście włączeniu lub wyłączeniu działania **HAL**.

Aby **HAL** zadziałał, należy pamiętać o wyłączeniu zapętlenia – kontrolka *loop* w sekcji **transport**. W przeciwnym razie – czyli, gdy klip będzie odtwarzany potencjalnie w nieskończoność – **HAL** nie będzie otrzymywał komunikatów o zakończeniu odtwarzania klipu i nie będzie miał okazji do działania (wyjątkiem od tej reguły jest użycie **HAL** po wymuszeniu zakończenia odtwarzania za pomocą kontrolki *flush* w sekcji **clips**, lub uruchomienie **HAL** automatycznie po starcie **remoteVJ**).

Menu rozwijalne *behavior* pozwala ustawić tryb pracy (metodę wyboru kolejnych klipów odtwarzanych automatycznie przez **HAL**). Do wyboru mamy następujące algorytmy:

- *play HAL clip from HAL program* – odtwarza klip określony za pomocą kontrolki *program* i *clip* (obie w sekcji **HAL**);
- *play HAL clip ++ from HAL program* – odtwarza kolejne klipy (w porządku rosnącym, po dojściu do ostatniego klipu procedura powtarza się od klipu o najniższym indeksie w ramach wyselekcjonowanego kontrolką *program* w sekcji **HAL** programu) z programu określonego z pomocą kontrolki *program* (w sekcji **HAL**), rozpoczynając od klipu określonego za pomocą kontrolki *clip* (w sekcji **HAL**);
- *play HAL clip -- from HAL program* – odtwarza kolejne klipy (w porządku malejącym, po dojściu do ostatniego klipu procedura powtarza się od klipu o najwyższym indeksie w ramach wyselekcjonowanego kontrolką *program* w sekcji **HAL** programu) z programu określonego z pomocą kontrolki *program* (w sekcji **HAL**), rozpoczynając od klipu określonego za pomocą kontrolki *clip* (w sekcji **HAL**);
- *play random clip from HAL program* – odtwarza losowe klipy z programu

określonego z pomocą kontrolki *program* (w sekcji **HAL**);

- *repeat last clip* – powtarza ostatnio odtworzony klip;
- *play clip ++ from selected program* – odtwarza kolejne klipy (w porządku rosnącym, po dojściu do ostatniego klipu procedura powtarza się od klipu o najniższym indeksie w ramach wyselekcjonowanego kontrolką *program* w sekcji **clips** programu) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od klipu określonego za pomocą kontrolki *clip* (w sekcji **HAL**);
- *play clip -- from selected program* – odtwarza kolejne klipy (w porządku malejącym, po dojściu do ostatniego klipu procedura powtarza się od klipu o najwyższym indeksie w ramach wyselekcjonowanego kontrolką *program* w sekcji **clips** programu) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od klipu określonego za pomocą kontrolki *clip* (w sekcji **HAL**);
- *play clip ++ from selected program ++* – odtwarza kolejne klipy (w porządku rosnącym, po dojściu do ostatniego klipu procedura przechodzi do następnego w kolejności programu [jeśli program jest pusty – nie zawiera żadnych klipów – jest pomijany] i powtarza się od klipu o najniższym indeksie w ramach programu, po przejściu przez ostatni program zawierający przypisania do klipów procedura rozpoczyna się na nowo od klipu o najniższym indeksie w ramach zawierającego przypisania programu o najniższym indeksie) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od ostatnio odtwarzanego klipu;
- *play clip -- from selected program ++* – odtwarza kolejne klipy (w porządku malejącym, po dojściu do ostatniego klipu procedura przechodzi do następnego w kolejności programu [jeśli program jest pusty – nie zawiera żadnych klipów – jest pomijany] i powtarza się od klipu o najwyższym indeksie w ramach programu, po przejściu przez ostatni program zawierający przypisania do klipów procedura rozpoczyna się na nowo od klipu o najwyższym indeksie w ramach zawierającego przypisania programu o najniższym indeksie) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od ostatnio odtwarzanego klipu;
- *play clip ++ from selected program* – odtwarza kolejne klipy (w porządku rosnącym, po dojściu do ostatniego klipu procedura przechodzi do poprzedniego w kolejności programu [jeśli program jest pusty – nie zawiera żadnych klipów – jest pomijany] i powtarza się od klipu o najniższym indeksie w ramach programu, po przejściu przez ostatni program zawierający przypisania do klipów procedura rozpoczyna się na nowo od klipu o najniższym indeksie w ramach zawierającego przypisania programu o najwyższym indeksie) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od ostatnio odtwarzanego klipu;
- *play clip -- from selected program* – odtwarza kolejne klipy (w porządku malejącym, po dojściu do ostatniego klipu procedura przechodzi do poprzedniego w kolejności programu [jeśli program jest pusty – nie zawiera żadnych klipów – jest pomijany] i powtarza się od klipu o najwyższym indeksie w ramach programu, po przejściu przez ostatni program zawierający

przypisania do klipów procedura rozpoczyna się na nowo od klipu o najwyższym indeksie w ramach zawierającego przypisania programu o najwyższym indeksie) z programu określonego z pomocą kontrolki *program* (w sekcji **clips**), rozpoczynając od ostatnio odtwarzanego klipu;

- *play random clip from selected program* – odtwarza losowy klip z aktualnie wybranego (za pomocą kontrolki *program* w sekcji **clips**) programu;
- *play random clip from random program* – odtwarza losowy klip z losowo wybranego programu.

Kontrolki *program* i *clip* umożliwiają wybór klipu i programu, od których rozpoczynają działanie algorytmy wybrane kontrolką *behavior* (jakkolwiek nie wszystkie algorytmy używają jedno lub obu parametrów).

Kontrolka *wait for note* wpływa na sposób rozpoczynania pracy przez **HAL**, po włączeniu kontrolki *on* (lub uruchomieniu **remoteVJ**). Jeśli jest włączona, **HAL** będzie czekał na zakończenie odtwarzania klipu, zanim rozpocznie swą aktywność. Jeśli kontrolka jest wyłączona, **HAL** natychmiast po włączeniu (załadowaniu **remoteVJ** lub włączeniu kontrolki *on*) przejdzie do zdefiniowanej kontrolką *behavior* akcji – jeśli włączyliśmy kontrolkę *on* w trakcie odtwarzania klipu **HAL** zaczeka na zakończenie odtwarzania.

Kontrolka *randomize rate* pozwala włączyć tryb losowej zmiany parametru *rate* w sekcji **transport** – każdy klip odtwarzany przez algorytmy HAL będzie miał wówczas przypisane losowo prędkość i kierunek odtwarzania.

Kontrolka *stealth* służy do włączenia lub wyłączenia trybu, w jaki **HAL** przechodzi do odtwarzania kolejnych klipów. Jeśli kontrolka jest włączona, **HAL** przekaże ewentualne zmiany, jakich dokonał w ustawieniach **remoteVJ** do graficznego interfejsu użytkownika – innymi słowy, jeśli HAL odtworzy klip, zostanie to zaznaczone na wirtualnej klawiaturze, a jeśli zmieni aktywny program zostanie zmieniony parametr *program* w sekcji **clips**.