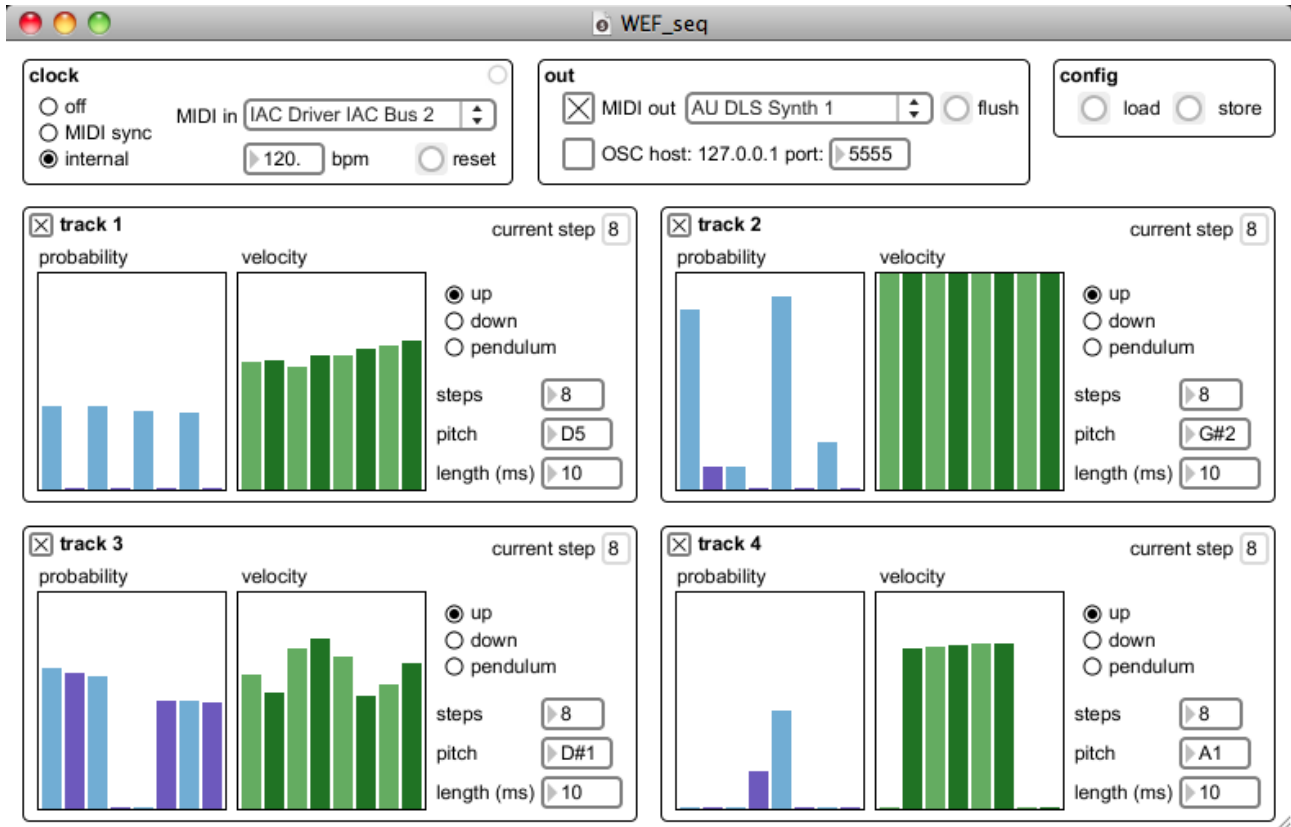


wefSeq

wprowadzenie

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



wefSeq to prosty, stworzony przede wszystkim do celów edukacyjnych, ale zarazem i funkcjonalny projekt (*patch*) napisany w środowisku MaxMSP/Jitter. Zasadniczo jest to 4-ścieżkowy sekwencer perkusyjny (jakkolwiek generowane przez **wefSeq** zdarzenia można wykorzystywać w innych celach) z możliwością programowania struktur polirytmicznych i wariacyjnych (w oparciu o przypisywane do każdego kroku na ścieżce prawdopodobieństwo wystąpienia zdarzenia). Interfejs sekwencera nawiązuje do wczesnych automatów perkusyjnych (choć z drugiej strony **wefSeq** dysponuje możliwościami niespotykanymi w tego typu konstrukcjach, np. wspomnianą możliwością tworzenia wariacji na bazie jednego wzorca rytmicznego i polirytmią).

wefSeq generuje komunikaty MIDI i OSC (*OpenSound Control*, więcej informacji o protokole OSC: <http://opensoundcontrol.org>). MIDI może posłużyć też do synchronizacji z innym oprogramowaniem lub urządzeniami.

Seqencer powstał podczas warsztatów MaxMSP/Jitter prowadzonych przeze mnie w ramach WEFCON 2010 (<http://wef.pl>). Niniejszy dokument omawia przebudowaną, ale wciąż bazującą na pierwotnym projekcie, wersję sekwencera.

Licencja

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

Instalacja i wymagania systemowe

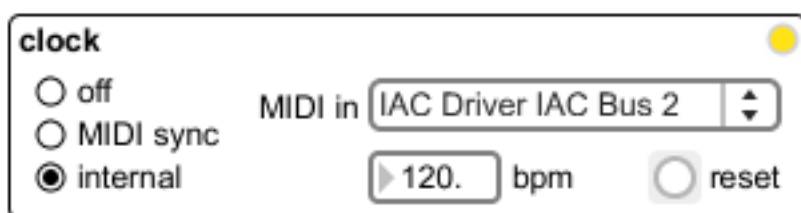
wefSeq działa zarówno na Mac OSX jak i MS Windows – wymaga jednak zainstalowanego w systemie środowiska MaxMSP/Jitter (v. 5.x) w wersji pełnej, demonstracyjnej lub *runtime* (wersja demostracyjna działa bez żadnych ograniczeń przez 30 dni, *runtime*, czyli „player” dla projektów stworzonych w MaxMSP/Jitter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

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

Interfejs

Po otwarciu **wefSeq** wyświetla okno zawierające wszystkie elementy interfejsu użytkownika. Okno jest podzielone na – grupujące tematycznie poszczególne parametry – sekcje **clock**, **out**, **config** i cztery sekcje ścieżek **track 1 ... track 4**.

clock



Sekcja umożliwia ustawienie tempa (*bpm*) i – ewentualnie – synchronizacji tempa do sygnału MTC z wybranego wejścia MIDI (w aktualnej wersji **wefSeq** może pracować w oparciu o

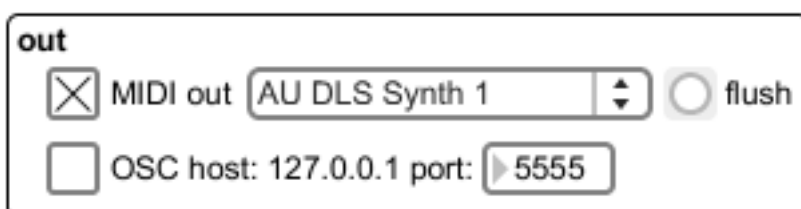
wewnętrzny zegar lub synchronizować się z innymi programami lub urządzeniami jako *slave*).

Lewą część sekcji zajmuje kontrolka umożliwiająca wybór źródła synchronizacji: wybranie pozycji *off* powoduje wstrzymanie odtwarzania, wybranie *MIDI sync* pozwala na synchronizację z sygnałem MTC (pobieranym z wejścia MIDI wybranego z listy *MIDI in*), a wybranie *internal* spowoduje, że **wefSeq** będzie korzystał z własnego źródła sygnału „zegara” – tempo w tym trybie możemy ustalić za pomocą pola numerycznego *bpm*. W trybie *MIDI sync* **wefSeq** reaguje na komunikaty MTC: STOP, CONTINUE i START wysyłane przez aplikację lub urządzenie *master*.

Przycisk *reset* pozwala zrestartować zegar – po wciśnięciu tego przycisku wszystkie aktywne ścieżki sequencera wznawiają odtwarzanie od pierwszego kroku.

Ponadto w prawym górnym rogu znajduje się kontrolka migająca w takt impulsów zegara.

out

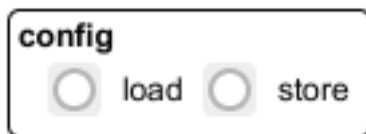


Ta sekcja zawiera kontrolki umożliwiające wybór portu MIDI out oraz portu dla transmisji komunikatów OSC (**wefSeq** wysyła komunikaty adresowane zawsze do *hosta*

127.0.0.1 [*localhost*] – można to zmienić edytując *patch* z projektem). Możemy wybrać transmisję danych poprzez protokoły MIDI lub OSC, albo – co w niektórych sytuacjach może się przydać – aktywować obie metody transmisji jednocześnie (oczywiście możemy też wyłączyć zarówno transmisję MIDI jak i OSC...).

Przycisk *flush* pozwala na „ręczne” wysłanie komunikatów MIDI *note off* i tym samym „zamknięcie” wszystkich „niedokończonych” nut (niektóre starsze sequencery MIDI dysponowały podobną funkcją nazywaną na ogół dość wymownie *panic* – jakkolwiek nie zdarzyło mi się by ta opcja była w pracy z **wefSeq** potrzebna, to, choćby ze względu na eksperymentalno-edukacyjny charakter programu, warto mieć ją do dyspozycji).

config

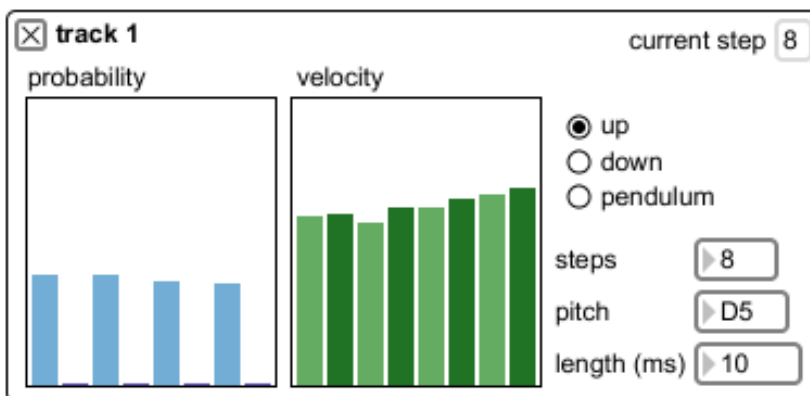


Przyciski dostępne w tej sekcji umożliwiają zapisanie (*store*) lub załadowanie (*load*) zapisanej konfiguracji. Konfiguracja obejmuje takie parametry jak częstotliwość (*bpm*) i źródło synchronizacji zegara, parametry ustawiane w sekcji **out**, parametry sekwencji wszystkich

czterech śladów sequencera. Ostatnio zapisana konfiguracja jest ładowana podczas startu projektu.

Ponieważ konfiguracja zapisywana jest w formie pliku tekstowego („*config.txt*”), można – choć ma to chyba niewiele sensu – edytować ją manualnie. Ewentualnie można też zachować wyjątkowo udaną sekwencję poprzez zapisanie jej do pliku (*store*) i przechowanie pod inną nazwą lub w innym – niż folder z aplikacją – miejscu.

track 1 ... 4



Cztery sekcje **track 1 ... 4** odpowiadają za parametry poszczególnych ścieżek sequencera. Oczywiście parametry przypisywane w obrębie poszczególnych ścieżek odnoszą się tylko do nich.

Każdą ze ścieżek można indywidualnie włączyć lub wyłączyć za pomocą

kontrolki w lewym górnym rogu. W prawym górnym rogu znajduje się natomiast licznik kroków (*current step*) wskazujący, który krok danej ścieżki jest właśnie odtwarzany.

Najważniejszymi kontrolkami każdej z czterech ścieżek są *probability* i *velocity*. O ile przeznaczenie kontrolki *velocity* jest oczywiste (odpowiada ona za parametr *velocity* [czyli „głośność”] w poszczególnych krokach – oczywiście im wyższy „słupek” dla danego kroku, tym większa wartość parametru), o tyle *probability* jest raczej nietypowym parametrem. Im wyższa jest wartość *probability* (im wyższy „słupek”) tym większe jest prawdopodobieństwo, że **wefSeq** wygeneruje zdarzenie (nutę) w momencie odtwarzania kroku, którego dotyczy dana wartość *probability*. Wartość maksymalna (najwyższy możliwy do uzyskania „słupek”) parametru *probability* spowoduje, że zdarzenie związane z tym danym krokiem wystąpi zawsze (jakkolwiek

musimy pamiętać o ustawieniu większej od „0” [zero] wartości parametru *velocity* dla danego kroku), a wartość minimalna (brak „słupka”) spowoduje, że zdarzenie w danym kroku nie będzie generowane. Odpowiednio manipulując *probability* każdego z kroków na ścieżce możemy kreować różne warianty na bazie jednego wzorca rytmicznego.

Kierunek odtwarzania danej ścieżki wybieramy klikając jedną z trzech dostępnych opcji: *up* (tryb „normalny”: kolejność od pierwszego do ostatniego kroku), *down* (odwrócona kolejność: od ostatniego do pierwszego kroku) i *pendulum* (ping-pong: najpierw ścieżka jest odtwarzana od początku do końca, potem od końca do początku, itd.).

Pole numeryczne *steps* umożliwia wybór długości sekwencji (od 1 do 8 kroków) – choć wydaje się to dość banalną funkcją, to warto zauważyć, że w możliwość ustawienia różnych długości sekwencji dla poszczególnych ścieżek pozwala na łatwe generowanie dość skomplikowanych, polirytmicznych przebiegów (zwłaszcza, jeśli jednocześnie wykorzystamy możliwość odtwarzania sekwencji w różnych kierunkach).

Pole *pitch* odpowiada za wysokość (parametr *pitch*) nuty generowanej przez zdarzenia ścieżki. Pole *length* pozwala natomiast na ustalenie (w milisekundach) długości nuty wyzwalanej przez zdarzenia wyzwalane na ścieżce.

OpenSound Control

wefSeq wysyła – jeśli zaznaczymy odpowiednie pole wyboru w sekcji **out** i wybierzemy port – komunikaty adresowane zawsze do *hosta* 127.0.0.1 [*localhost*] – można to zmienić edytując *patch* z projektem).

Składnia komunikatów OSC rozpoznawanych przez **wefSeq** wygląda następująco:

```
/wefSeq trackNumber pitch velocity length
```

```
trackNumber [int 0 ... 3] – numer ścieżki  
pitch [int 0 ... 127] – ustawiony dla ścieżki numer nuty  
velocity [int 0 ... 127] – parametr velocity danego kroku  
length [int] – ustawiona dla ścieżki długość nuty (w milisekundach)
```