

oscInterfaces

wprowadzenie

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

oscInterfaces to kolekcja prostych aplikacji (być może lepiej byłoby nazywać je po prostu zabawkami) napisanych w środowisku Processing (wiele z nich bazuje wprost na przykładach zawartych w dokumentacji środowiska, więcej o Processingu: <http://processing.org>). Aplikacje działają na platformach Mac OSX, Windows i Linux (w przypadku Windows i Linux potrzebna jest maszyna wirtualna Javy: <http://java.com>), korzystają z protokołu OSC (*OpenSound Control* - więcej informacji o OSC: <http://opensoundcontrol.org>) do transmisji danych.

oscInterfaces wykorzystałem po raz pierwszy prowadząc warsztaty MaxMSP/Jitter (<http://cycling74.com>) na Uniwersytecie Muzycznym im. Fryderyka Chopina (<http://www.chopin.edu.pl>), a w kształcie podobnym do obecnego kolekcja "zadebiutowała" podczas warsztatów Pure-Data (<http://puredata.info>) w ramach festiwalu Digital_ia 11 (<http://digitalia.13muz.eu>).

oscInterfaces powstały głównie do celów warsztatowych i umożliwiają tworzenie nietypowych - generatywnych i interaktywnych - struktur audiowizualnych. Wszystkie aplikacje wchodzące w skład kolekcji opierają się o jedną ideę: odejścia od myślenia w kategoriach liniowego interfejsu (linii czasu) i liniowego porządkowania materiału w oparciu o lokowanie zdarzeń (nut, klatek wideo, itp.) w konkretnych punktach czasu - **oscInterfaces** umożliwiają wpływanie na dowolne "podłączone" do nich media w czasie rzeczywistym. Jakkolwiek poprzednie zdania brzmią może nieco zawile, to zadaniem **oscInterfaces** jest uproszczenie tworzenia (lub uczenia się tworzenia) tego typu struktur i umożliwienie łatwego eksperymentowania z nimi.

Poza tym, próbując uczyć korzystania z języków programowania sprofilowanych pod kątem zastosowań związanych z kreacją dźwięku lub form wizualnych (np. wspomniany wyżej Processing), czy środowisk obiektowych (np. wspomniane Pure-Data lub MaxMSP/Jitter) zauważyłem, że dla wielu osób nauka stresująca z prostego powodu: bardzo długo nie mogą sobie wyobrazić ciekawych zastosowań technologii, które poznają, dobrze jest więc umożliwić im skorzystanie z jakiegoś rodzaju dźwigni wyobraźni, która uprzyjemni naukę, a przy okazji wskaże ciekawe pola zastosowań nabywanej wiedzy.

Licencja

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

Instalacja i wymagania systemowe

oscInterfaces wystarczy skopiować do wybranego folderu. W pakiecie znaleźć można skompilowane wersje dla Mac OSX, Windows i Linux. Można też pokusić się o twórczą modyfikację i samodzielną kompilację aplikacji (kod źródłowy jest dołączony do archiwum z programami). **oscInterfaces** wymagają zainstalowanej w systemie maszyny wirtualnej Javy – użytkownicy komputerów Apple są tu w lepszej sytuacji, ponieważ maszyna wirtualna Javy jest standardowo zainstalowanym komponentem

systemu operacyjnego (jakkolwiek ma się to zmienić w przyszłości) natomiast osoby korzystające z Windows muszą odpowiednią dla swojej wersji systemu operacyjnego wersję JRE (*Java Runtime Environment*) zainstalować samodzielnie (środowisko Java jest bezpłatne), najlepiej korzystając ze strony <http://java.com>.

Poza JRE **oscInterfaces** nie wymagają żadnych specjalnych zabiegów i komponentów. Dwukrotne kliknięcie na ikonie pliku wybranej aplikacji ją.

Jak posługiwać się **oscInterfaces**?

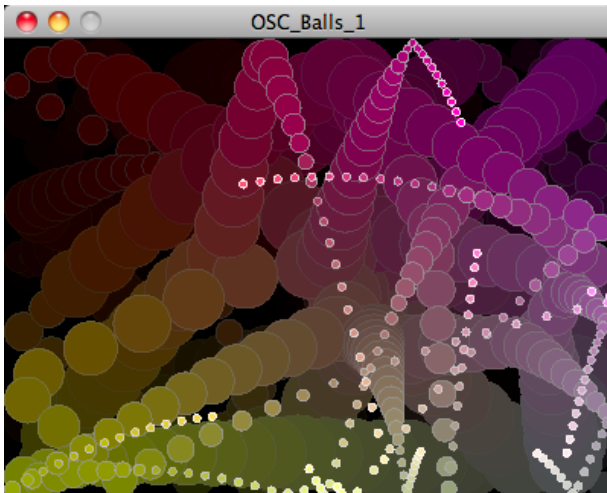
Każda z aplikacji wchodzących w skład kolekcji wysyła - oczywiście, jeśli jest uruchomiona - komunikaty OSC adresowane do numeru **IP 127.0.0.1** (*localhost*) i **portu 5555** - aplikacja odbierająca komunikaty musi więc być skonfigurowana tak, by odbierała komunikaty wysyłane do portu 5555. Komunikaty te mogą być oczywiście interpretowane w dowolny sposób i w dowolnym urządzeniu lub aplikacji korzystającym z protokołu OSC.

UWAGA: w praktyce, zwłaszcza, jeśli pracujemy na systemach Windows, może się okazać, że systemowe lub doinstalowane oprogramowanie antywirusowe blokuje transmisję danych OSC - zwykle na ekranie zobaczymy w takim przypadku stosowny komunikat i pytanie, czy blokować transmisję. Naturalnie - jeśli komunikat dotyczy **oscInterfaces** - musimy odblokować komunikację, by móc sprzągać je z własnymi projektami.

Wciśnięcie klawisza "h" wyświetla informację o składni komunikatów OSC i krótką notkę o sposobie użytkowania dla każdej z aplikacji (ponowne wciśnięcie "h" lub kliknięcie myszą wyłącza wyświetlanie informacji tekstowej). Zasadniczo wszystkie aplikacje w pakiecie obsługujemy za pomocą myszy (czasem dodatkowo używamy też klawiatury - zawsze, gdy jest to wymagane odpowiednia informacja pojawia się po naciśnięciu klawisza "h").

Komunikaty OSC wysyłane przez aplikacje zorganizowane są w pakiety (*bundles*) transmitowane 25 razy na sekundę - 25fps (*frames per second*, klatek na sekundę) wynosi częstotliwość odświeżania obrazu ustawiona dla wszystkich aplikacji pakietu, a nowy pakiet jest wysyłany zawsze przy okazji generowania kolejnej klatki obrazu. Zawartość transmitowanych komunikatów odzwierciedla oczywiście nasze manipulacje "interfejsami" i zależy od tego, z którą z aplikacji pracujemy. Poza tym wszystkie aplikacje wysyłają komunikaty o położeniu kursora myszy. Tak więc każdy pakiet (*bundle*) zawiera pełną informację o stanie interfejsu aplikacji w danej klatce obrazu.

OSC Balls 1



Stosunkowo prosta aplikacja. 12 obiektów o zmiennej prędkości i wielkości („piłek”) odbija się od krawędzi okna i od siebie nawzajem.

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

Klikając myszą w obrębie okna przyspieszamy ruch obiektów, a poruszając myszą powodujemy, że obiekty zwiększają średnicę.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach wszystkich 12 obiektów.

Komunikaty transmitowane przez aplikację OSC:

```
/oscBalls1Mouse x y vx vy button
```

x [float] – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

vy [float] – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

button [int] – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

```
/oscBalls1 index collision x y vx vy diameter
```

index [int] – index obiektu (liczba całkowita pomiędzy 0 a 11);

x [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

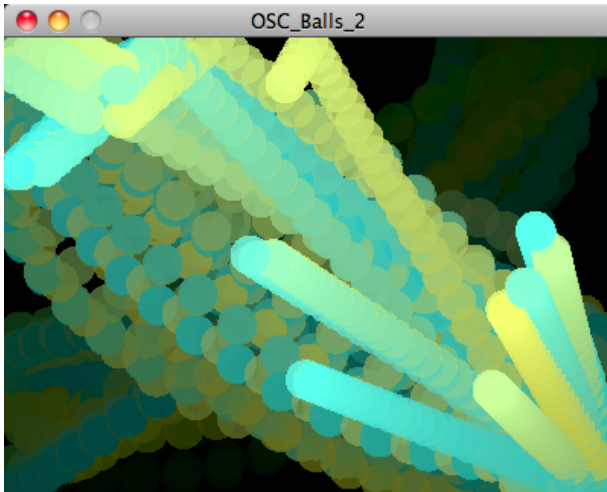
vx [float] – zmiana poziomego położenia obiektu w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” i „√2”);

vy [float] – zmiana pionowego położenia obiektu w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” i „√2”);

diameter [float] – średnica obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”).

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscBalls1Mouse” jest wysyłany jednorazowo na początku każdego *bundle*, a komunikat „/oscBalls1” powtarzany jest 12-krotnie z parametrami odpowiadającymi każdemu z obiektów.

OSC Balls 2



Również bardzo prosta aplikacja. 24 obiekty o zmiennej prędkości przemieszczamy po ekranie przesuwając wciskając przycisk myszy i przemieszczając kursor. Możemy również użyć prawego przycisku (lub kombinacji *ctrl-click* na OSX) by uzyskać inny, niż w przypadku użycia lewego przycisku myszy, charakter ruchu.

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach wszystkich 24 obiektów.

Komunikaty transmitowane przez aplikację OSC:

```
/oscBalls2Mouse x y vx vy button
```

x [float] – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

vy [float] – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

button [int] – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

```
/oscBalls2 index x y vx vy
```

index [int] – index obiektu (liczba całkowita pomiędzy 0 a 23);

x [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy

„0” [zero] a „1”);

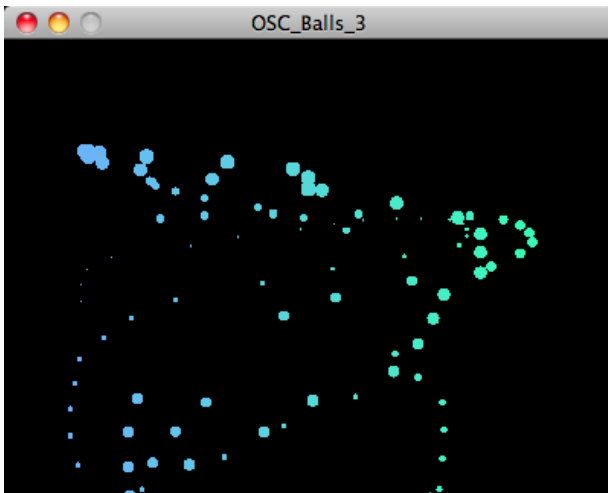
y [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – zmiana poziomego położenia obiektu w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” i „√2”);

vy [float] – zmiana pionowego położenia obiektu w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” i „√2”).

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscBalls2Mouse” jest wysyłany jednorazowo na początku każdego *bundle*, a komunikat „/oscBalls2” powtarzany jest 24-krotnie z parametrami odpowiadającymi każdemu z obiektów.

OSC Balls 3



Klikając myszą (lub przytrzymując wciśnięty przycisk myszy i poruszając kursorem) umieszczamy w oknie aplikacji nowe obiekty: odbijające się od „podłoża” koła – średnica obiektów maleje z czasem (aż do zniknięcia obiektu). Tym samym, w przeciwieństwie do **OSC Balls 1** i **OSC Balls 2** mamy tu do czynienia ze zmienną ilością obiektów (których parametry są oczywiście transmitowane protokołem OSC) – warto na to zwrócić uwagę projektując zastosowania dla tego interfejsu.

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach wszystkich obiektów (o ile jakieś obiekty zostały utworzone).

Komunikaty transmitowane przez aplikację OSC:

/oscBalls3Mouse x y vx vy button

x [float] – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

vy [float] – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

button [int] – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

/oscBalls3Stats num

index [int] – ilość obiektów.

/oscBalls3 index finished x y speed life

index [int] – index obiektu (liczba całkowita pomiędzy „0” [zero] a [ilość obiektów - 1]);

finished [int] – jeśli obiekt ma zostać usunięty w następnej klatce wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

x [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

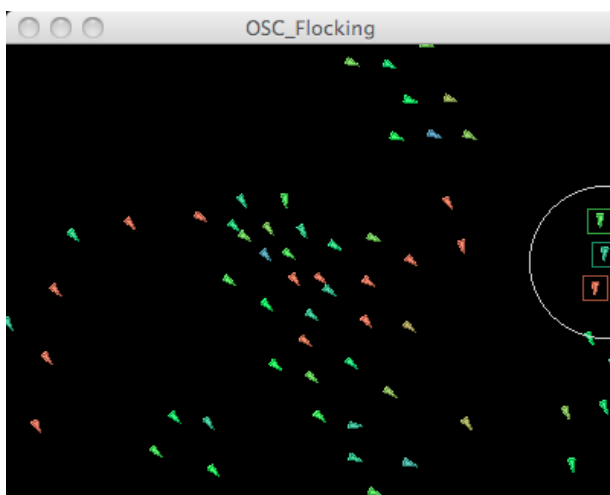
y [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

speed [float] – prędkość obiektu;

life [float] – parametr określający umowny czas egzystencji obiektu (w chwili umieszczenia obiektu na scenie jego wartość wynosi „1”, a tuż przed usunięciem „0” [zero])

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscBalls3Mouse” jest wysyłany jednorazowo na początku każdego *bundle*, następny w kolejności jest komunikat „/oscBalls3Stats”, a po nim komunikat „/oscBalls3” powtarzany jest tyle razy, ile istnieje obiektów (za każdym razem z parametrami odpowiadającymi konkretnemu obiektowi).

OSC Flocking



Aplikacja nieco odmienna od poprzednio omawianych, stanowiąca rozwinięcie jednego z przykładów z dokumentacji środowiska Processing.

Tym razem klikając myszą w obrębie okna aplikacji tworzymy obiekty poruszające się podobnie do stada zwierząt (ławica ryb byłaby tu chyba najlepszym modelem) – znowu mamy więc do czynienia z dynamiczną strukturą o zmiennej ilości obiektów.

Możemy utworzyć do 60 obiektów – po osiągnięciu tej liczby kolejne kliknięcia będą powodowały usunięcie najstarszego obiektu i wstawienie nowego. Wciśnięcie klawisza spacji również spowoduje usunięcie

najstarszego obiektu (wciskając spację odpowiednią ilość razy możemy usunąć wszystkie umieszczone wcześniej objekty).

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach obiektów (o ile jakieś objekty zostały utworzone) znajdujących się w pewnej odległości od kursora myszy (jeśli jakieś objekty znajdują się na tyle blisko kursora, by informacja o ich parametrach była transmitowana, zostaną zaznaczone prostokątnymi „ramkami”, a wokół kursora myszy pojawi się okrąg – parametry obiektów znajdujących się wewnątrz okręgu będą transmitowane *via* OSC).

Komunikaty transmitowane przez aplikację OSC:

`/oscFlockingMouse x y vx vy button`

`x [float]` – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y [float]` – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`vx [float]` – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`vy [float]` – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`button [int]` – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

`/oscFlocking index x y vx vy ax ay ix iy t`

`index [int]` – index obiektu (liczba całkowita pomiędzy „0” [zero] a [ilość obiektów – 1], jednak nie większa, niż 59);

`finished [int]` – jeśli obiekt ma zostać usunięty w następnej klatce wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

`x [float]` – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y [float]` – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`vx [float]` – prędkość obiektu w osi X (w poziomie);

`vy [float]` – prędkość obiektu w osi Y (w pionie);

`ax [float]` – przyspieszenie obiektu w osi X (w poziomie);

`ay [float]` – przyspieszenie obiektu w osi Y (w pionie);

`ix [float]` – poziome (oś X) położenie obiektu w chwili, kiedy

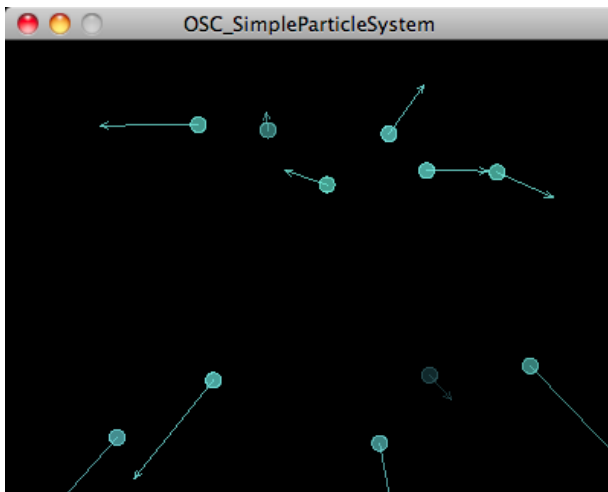
został utworzony (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

iy [float] – pionowe (oś Y) położenie obiektu w chwili, kiedy został utworzony (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

t [int] – parametr określający umowny czas egzystencji obiektu (w milisekundach) mierzony od momentu jego utworzenia.

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscFlockingMouse” jest wysyłany jednorazowo na początku każdego *bundle*, a po nim komunikat „/oscFlocking” (za każdym razem z parametrami odpowiadającymi konkretnemu obiektowi) powtarzany tyle razy, ile istnieje obiektów znajdujących się odpowiednio blisko kursora myszy.

OSC Simple Particle System



Prosty system emisji cząsteczek (podobnie jak opisane powyżej **OSC Flocking** i poniżej **OSC Multiple Particle System** bazujący na przebudowanym przykładzie pochodzącym z dokumentacji środowiska Processing).

Klikając lub po prostu przytrzymując wciśnięty w obrębie okna aplikacji klawisz myszy tworzymy obiekty (cząsteczki). Ilość jednocześnie utworzonych cząsteczek jest ograniczona jedynie naszymi zdolnościami w posługiwaniu się myszą...

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach obiektów (o ile jakieś obiekty zostały utworzone).

Komunikaty transmitowane przez aplikację OSC:

```
/oscParticleSystemMouse x y vx vy button
```

x [float] – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

vy [float] – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu

wartość jest skalowana do zakresu pomiędzy „ $-\sqrt{2}$ ” a „ $\sqrt{2}$ ”);

button [int] – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

/oscParticleSystemObject i x y vx vy ax ay t

index [int] – index obiektu;

x [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – prędkość obiektu w osi X (w poziomie);

vy [float] – prędkość obiektu w osi Y (w pionie);

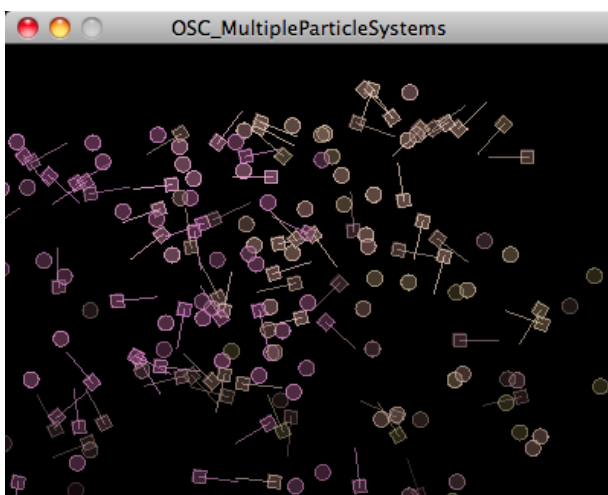
ax [float] – przyspieszenie obiektu w osi X (w poziomie);

ay [float] – przyspieszenie obiektu w osi Y (w pionie);

t [int] – parametr określający umowny czas egzystencji obiektu (w milisekundach) mierzony od momentu jego utworzenia.

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscParticleSystemMouse” jest wysyłany jednorazowo na początku każdego *bundle*, a po nim komunikat „/oscParticleSystemObject” powtarzany tyle razy, ile istnieje obiektów (za każdym razem z parametrami odpowiadającymi konkretnemu obiektowi).

OSC Multiple Particle System



Nieco bardziej skomplikowany (niż **OSC Simple Particle System**) system emisji cząsteczek (podobnie jak opisane powyżej **OSC Flocking** i **OSC Simple Particle System** bazujący na przebudowanym przykładzie pochodzącym z dokumentacji środowiska Processing).

Klikając lub po prostu przytrzymując wciśnięty w obrębie okna aplikacji klawisz myszy tworzymy obiekty (cząsteczki). Ilość jednocześnie utworzonych cząsteczek jest ograniczona jedynie naszymi zdolnościami w posługiwaniu się myszą...

W stosunku do **OSC Simple Particle System** możliwości aplikacji wzbogacają się jednak o drugi typ obiektu („kwadratowe” cząsteczki z rotacją) i wiele systemów cząstek jednocześnie (każde wciśnięcie klawisza myszy tworzy nowy system cząstek). Obiekty (cząsteczki) losowo przybierają w chwili utworzenia formę podstawową lub poszerzoną o rotację.

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią

komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz położeniu i innych parametrach obiektów (o ile jakieś obiekty zostały utworzone).

Komunikaty transmitowane przez aplikację OSC:

`/oscMultiParticleMouse x y vx vy button`

`x` [float] – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y` [float] – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`vx` [float] – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`vy` [float] – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`button` [int] – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

`/oscMultiParticleBasic i1 i2 x y vx vy ax ay t`

`i1` [int] – index systemu cząstek;

`i2` [int] – index obiektu;

`x` [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y` [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`vx` [float] – prędkość obiektu w osi X (w poziomie);

`vy` [float] – prędkość obiektu w osi Y (w pionie);

`ax` [float] – przyspieszenie obiektu w osi X (w poziomie);

`ay` [float] – przyspieszenie obiektu w osi Y (w pionie);

`t` [int] – parametr określający umowny czas egzystencji obiektu (w milisekundach) mierzony od momentu jego utworzenia.

`/oscMultiParticleExtended i1 i2 x y vx vy ax ay a t`

`i1` [int] – index systemu cząstek;

`i2` [int] – index obiektu;

`x` [float] – pozioma współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

y [float] – pionowa współrzędna obiektu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

vx [float] – prędkość obiektu w osi X (w poziomie);

vy [float] – prędkość obiektu w osi Y (w pionie);

ax [float] – przyspieszenie obiektu w osi X (w poziomie);

ay [float] – przyspieszenie obiektu w osi Y (w pionie);

a [float] – rotacja obiektu (wyrażona w radianach);

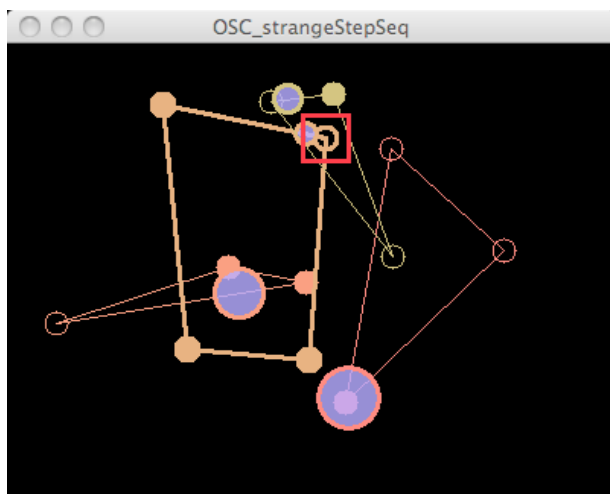
t [int] – parametr określający umowny czas egzystencji obiektu (w milisekundach) mierzony od momentu jego utworzenia.

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscParticleSystemMouse” jest wysyłany jednorazowo na początku każdego *bundle*, a po nim następuje seria komunikatów „/oscMultiParticleBasic” i „/oscMultiParticleExtended” powtarzanych tyle razy, ile istnieje obiektów (za każdym razem z parametrami odpowiadającymi konkretnemu obiektowi).

Komunikaty „/oscMultiParticleBasic” i „/oscMultiParticleExtended” są porządkowane w kolejności odpowiadającej kolejności tworzenia obiektów w ramach poszczególnych systemów cząstek – tak więc najpierw zostaną wysłane dane dotyczące cząstek przynależących do najstarszego systemu cząstek (w ramach tego systemu również najpierw zostaną wysłane dane dotyczące najstarszych cząstek), następnie w analogiczny sposób będą wysyłane kolejne systemy, aż do najnowszej cząstki w najnowszym systemie, której parametry zostaną wysłane na końcu.

Powyższy akapit brzmi może nieco mętnie, dlatego warto chyba w tym miejscu przypomnieć pamiętne słowa bohatera zrealizowanego 1960 roku filmu Andrzeja Munka *Zezowate Szczęście*: *Jeśli mamy trójkąt prostokątny, to sinus jest to stosunek przyprostokątnej do przeciwprostokątnej, a cosinus to też jest stosunek przyprostokątnej do przeciwprostokątnej, ale tej innej przyprostokątnej.*

OSC StrangeStepSeq



Nietypowy sequencer krokowy pozwalający utworzyć do 8 pętli o długości od 3 do 16 węzłów (kroków) i manipulować nimi w czasie rzeczywistym.

Klikając na wybranej pętli (ściślej mówiąc na okręgu reprezentującym węzeł) aktywujemy pętlę, do której węzeł należy. Jeśli którakolwiek pętla jest aktywna kliknięcie na puste miejsce w oknie doda nowy węzeł do pętli. Kliknięcie węzła aktywnej pętli prawym przyciskiem myszy (lub *ctrl-click* na OSX) kasuje go. Jeśli w pętli zostały już tylko 3 węzły kliknięcie

któregokolwiek z nich prawym przyciskiem myszy (lub *ctrl-click* na OSX) kasuje pętlę (ostatniej pozostałej pętli nie można skasować, tak więc zawsze pozostaną nam przynajmniej 3 węzły i jedna pętla do dyspozycji). Nową pętlę z losowo

umieszczonymi 3 węzłami wstawiamy naciskając spację.

Każda pętla posiada swój własny wskaźnik (pulsujący jasnoniebieski okrąg przemieszczający się wzdłuż połączeń pomiędzy poszczególnymi węzłami w ramach pętli) informujący, w którym miejscu sekwencji w danej chwili jesteśmy.

Wciśnięcie klawisza „h” wywołuje lub krótki opis aplikacji i transmitowanych przez nią komunikatów.

W każdej klatce obrazu aplikacja wysyła pakiet komunikatów zawierających dane o kursorze myszy oraz – jeśli w ramach którejś z pętli nastąpiło wyzwolenie kolejnego węzła – informację o zdarzeniu zawierającą index pętli, index węzła (kroku) i jego parametry (położenie).

Komunikaty transmitowane przez aplikację OSC:

`/oscStrangeStepSeqMouse x y vx vy button`

`x [float]` – pozioma współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y [float]` – pionowa współrzędna kursora myszy nad oknem aplikacji (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`vx [float]` – zmiana poziomego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`vy [float]` – zmiana pionowego położenia kursora myszy w stosunku do poprzedniej klatki obrazu (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „-√2” a „√2”);

`button [int]` – jeśli przycisk myszy jest wciśnięty wartość tego parametru wynosi „1”, w przeciwnym razie „0” (zero).

`/oscStrangeStepSeqNote channel step x y`

`channel [int]` – index pętli (liczba całkowita pomiędzy „0” [zero] a „7”);

`step [int]` – index węzła (liczba całkowita pomiędzy „0” [zero] a „15”);

`x [float]` – pozioma współrzędna węzła (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

`y [float]` – pionowa współrzędna węzła (niezależnie od rozdzielczości ekranu wartość jest skalowana do zakresu pomiędzy „0” [zero] a „1”);

Punkt [0, 0] układu współrzędnych, w ramach którego podawane są dane dotyczące położenia kursora myszy i obiektów znajduje się w lewym górnym rogu okna aplikacji, wartość współrzędnej poziomej rośnie wraz z ruchem w prawo, a wartość współrzędnej pionowej rośnie wraz z ruchem w dół. Komunikat „/oscStrangeStepSeqMouse” jest wysyłany jednorazowo na początku każdego *bundle*, a po nim następuje (jeśli status poszczególnych pętli tego wymaga) jeden lub więcej komunikatów „/oscStrangeStepSeqNote” opisujących wyzwolone węzły.