

Przechwytywanie obrazu (016; 25.08.2009; *processing*)

Uzyskanie dostępu do obrazu przesyłanego na żywo z kamery internetowej do środowiska Processing jest niesłychanie proste. Wszystko czego potrzebujemy, to:

- kamera internetowa podłączana na USB lub wbudowana w monitor (zalecam nawet nie instalować do niej sterowników, gdyż u mnie pogarszały one jakość obrazu),
- zainstalowego QuickTime'a (lub WinVDIG w wersji 1.0.1, do pobrania pod adresem: http://www.vdig.com/WinVDIG/WinVDIG_101.exe),
- program Processing, który ściągamy i rozpakowujemy ze strony: <http://processing.org/download/> (aktualna wersja: 1.0.6).

Bardzo prosty kod w Processing'u, pozwalający odczytywać kolejne klatki obrazu z jednej podłączonej kamerki i wyświetlać je w oknie graficznym programu:

```
import processing.video.*;    //import niezbędnej biblioteki
Capture webcam;              //utworzenie obiektu

void setup() {
  size(640, 480);
  webcam = new Capture(this, width, height, 25);
  //deklaracja obiektu przechowującego obraz z kamery
  //o wymiarach takich samych jak okno graficzne i z maksymalną
  //częstotliwością odświeżania obrazu 25 klatek/s
}

void draw() {
  if (webcam.available() == true) {
    webcam.read();           //odczytanie klatki obrazu
    set(0, 0, webcam);      //wstawienie obrazu
    //tu umieszczać polecenia operujące na obrazie
  }
}
```

Gdy w systemie mamy wiele urządzeń przechwytyjących obraz, w sekcji *setup* możemy dodać następujące linie kodu, aby kolejno wyświetlić listę wykrytych urządzeń i ustawić urządzenie drugie na liście (numeracja od zera):

```
print(Capture.list());  
webcam = new Capture(this, width, height, devices[1], 25);
```

Aby w czasie uruchamiania programu otworzyć okno właściwości urządzenia, pod koniec sekcji *setup* dopisujemy linię:

```
cam.settings();
```

Gdy interesuje nas tylko prostokątny wycinek obrazu z kamery, którego lewy górny róg ma współrzędne (10, 20), wysokość wynosi 240, a szerokość 370, na koniec sekcji *setup* wstawiamy następujący kod:

```
cam.crop(10, 20, 370, 240);
```

Opisany wyżej kod bardzo prosto możemy rozszerzać o polecenia operujące na obrazie, korzystając z funkcji *pixels[]*, tak jak opisano to w artykule nr 012. Zwracam uwagę, że wszelki dodatkowy kod operujący na obrazie należy umieszczać wewnątrz funkcji warunkowej, a nie w głównej pętli *draw*, co uchroni program od wielokrotnego wykonywania operacji na tym samym obrazie.