

Anna Paprzycka

Wydział Filologii Polskiej i Klasycznej Uniwersytetu im. Adama Mickiewicza w Poznaniu

Sztuka programowania: Processing jako narzędzie artystyczne i kulturotwórcze

Friedrich Kittler, niemiecki teoretyk mediów i literaturoznawca, jeszcze przed rozpoczęciem XXI wieku powiedział, że w obecnych czasach studenci powinni znać co najmniej dwa języki programowania, żeby móc mówić o tym, czym jest kultura [*Software* 20]. Lev Manovich, badacz mediów rosyjskiego pochodzenia, na początku XXI wieku zmierzył się ze zdefiniowaniem pojęcia „nowe media” w książce *Język nowych mediów*. Zaznaczył w niej potrzebę powstania nowego kierunku badań w humanistyce – takiego, który skupi się na kulturowym znaczeniu oprogramowania. W omawianej publikacji Manovich pisał:

„Żeby zrozumieć nowe media, musimy odwołać się do informatyki. To właśnie tam znajdziemy nowe terminy, kategorie i funkcje charakteryzujące media, które stały się programowalne. Od medioznawstwa zmierzamy w stronę czegoś, co można by nazwać programoznawstwem, czyli od teorii mediów do teorii oprogramowania” [*Język* 117].

W 2007 roku Manovich wspólnie z Matthew Fullerem założyli centrum badawcze Software Studies Initiative, które obecnie posiada dwie jednostki: w Nowym Jorku oraz w mieście La Jolla w Kalifornii. Inicjatywa ta zrzesza badaczy, którzy wzięli sobie za cel prowadzenie krytycznego dyskursu na temat oprogramowania – jego znaczenia, możliwości, zagrożeń i roli w kulturze cyfrowej. Jak widać, oprogramowanie przestaje być aspektem ignorowanym przez badaczy kultury, a w humanistyce sukcesywnie rozwija się nurt naukowy, skupiający się właśnie na tej kwestii. Ten artykuł został poświęcony językowi programowania Processing, który stanowi jedynie element z zasobu możliwości, do których mogą się uciec

artyści chcący łączyć programowanie ze sztuką. Jest on jednak narzędziem wyjątkowym, które może stanowić punkt wyjścia do analizy znaczenia oprogramowania w szeroko pojętej kulturze. Zarysowany tu został w sposób ogólny zasięg jego oddziaływania, a pod uwagę wzięto rozmaite dziedziny, takie jak kultura popularna, sztuka i nauka.

Processing to język programowania, a także związane z nim środowisko programistyczne (IDE)¹ stworzone przez dwóch studentów MIT Media Lab: Bena Frya i Casey Reasa w 2001 roku. Projekt ten powstał jako kontynuacja idei zawartej w języku programowania Design By Numbers (DBN), którego autorem był John Maeda, opiekun funkcjonującej w ramach MIT Media Lab inicjatywy Aesthetic and Computation Research Group, do której należeli również późniejsi autorzy Processingu. Design By Numbers zostało ukończone i upublicznione w 1999 roku. Głównym założeniem, leżącym u podstaw tego projektu, było stworzenie języka programowania, z którego mogliby korzystać ludzie nieposiadający specjalistycznych umiejętności. Prostota języka programowania DBN wynikała z zastosowania jasnego systemu, opartego na manipulowaniu zmiennymi i formułowaniu zdań warunkowych. John Maeda był propagatorem idei, że należy udoskonalić system edukacyjny w szkołach artystycznych i technologicznych od strony programowania, aby podnieść jakość sztuki medialnej i designu. Dlatego DBN było z założenia proste w obsłudze i darmowe, a więc ogólnodostępne na poziomie finansowym i intelektualnym.

¹ IDE (Integrated Development Environment) to aplikacja lub zespół aplikacji umożliwiających pisanie i uruchamianie kodu.

Processing, stworzony przez studentów Maedy, stanowił ideologiczną i technologiczną kontynuację projektu DBN. Jego IDE zostało oparte na podobnym interfejsie. Autorom Processingu także zależało na tym, aby stworzyć ogólnodostępne i łatwe w obsłudze narzędzie do tworzenia wizualizacji za pomocą programowania. Z tego powodu udostępnili ów program w ramach licencji Open Source i umieścili w sieci, skąd każdy może go pobrać na własny komputer i z niego korzystać. Processing został oparty na języku programowania Java, stanowi jego uproszczoną wersję. Autorzy chcieli stworzyć możliwość, aby potencjalnie każdy człowiek mógł się oswoić z tym, co wydaje się skomplikowane i przerażające – z oprogramowaniem². Korzystanie z tego narzędzia nie wymaga od użytkownika zaawansowanej wiedzy matematycznej, choć oczywiście jej znajomość stanowi dodatkowy atut. Processing był adresowny przede wszystkim do artystów i miał pomóc im zacząć myśleć jak inżynierzy, oraz do inżynierów, których z kolei miał zainspirować do działalności artystycznej. Miał on pionierski charakter, bo był jednym z pierwszych narzędzi kreatywnych przeznaczonych do powszechnego użytku, które opierało się na języku programowania o klasycznej syntaksie.

Wspólnoty kreatywne

Processing rozwinął się w szybkim tempie i zyskał popularność na całym świecie. Szybko zaczęły się pojawiać podręczniki stworzone przez samych autorów Processingu (*Processing: A Programming Handbook for Visual Designers*, *Make: Getting Started with Processing*, *Visualizing Data*), ale też pisane przez jego entuzjastów. Z pewnością wielki udział w rozprzestrzenianiu idei kreatywnego kodowania za pomocą Processingu miał Daniel Shiffman, matematyk, filozof i programista, autor dwóch podręczników do nauki tego języka (*Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction* oraz *The nature of Code*), aktywnie działający

² Jak pokazuje internetowy kurs prowadzony przez Daniela Shiffmana na stronie <https://hello.processing.org/>, wystarczy 40 minut, żeby zaprogramować swoją pierwszą interaktywną wizualizację w Processingu.

w internecie pod nickiem The Coding Train. Prowadzi on kanał na portalu YouTube, na którym udostępnia filmy z ciekawymi lekcjami. Dzięki atrakcyjnej, przejrzystej formie jego lekcje mają setki wyświetleń. Daniel Shiffman jest również dyrektorem The Processing Foundation, której celem jest rozpowszechnianie na całym świecie, we wszystkich środowiskach, umiejętności programowania za pomocą tego narzędzia. Fundacji bardziej niż na udoskonalaniu samego programu i poszerzaniu jego bibliotek zależy na pielęgnowaniu zorganizowanej wokół niego społeczności z całego świata. Shiffman napisał:

„Processing dla mnie zawsze był najcudowniejszą rzeczą ze wszystkich. Dał mi misję i pasję, żeby nieść strategię obliczeniową wszystkim: artystom, designerom, muzykom, biologom, lekarzom, tancerzom, animatorom, bankierom, fotografom, bibliotekarzom, projektantom mody, architektom, psychologom, dziennikarzom i pisarzom – wymieniając tylko kilku. Pisanie kodu może się wydawać przerażające, często błędnie rozumiane jako coś zarezerwowanego tylko dla informatyków i inżynierów. Processing pomógł wyeliminować ten lęk, czyniąc programowanie możliwym dla szerszej publiczności, szczególnie dla artystów” [“Introduction” IX]³.

Misją fundacji jest szerzenie świadomości związanej z kodem, a także wspieranie tych, którzy z różnych powodów (takich jak płeć, wiek, zawód, kwalifikacje, klasa społeczna, seksualność itd.) nie mają dostępu do tworzenia za jego pomocą. Jej członkowie myślą o oprogramowaniu jako o medium – narzędziu komunikacji, metodzie na wyrażenie własnego sposobu myślenia i tworzenia – i z tej przyczyny agituja za tym, by było ono otwarte, powszechnie dostępne i darmowe.

Processing jest dystrybuowany na zasadach licencji Open Source, co oznacza, że użytkownicy mają prawo do jego darmowego uruchamiania, kopiowania, modyfikowania, a także rozbudowywania. Dzięki temu nad rozwojem tego projektu zaczęła pracować duża, ciągle rozwijająca się wirtualna społeczność. Początkowo pracowali nad nim tylko jego autorzy.

³ Tłumaczenie własne.

Zainteresowani tym programem i jego działaniem mogli pisać do nich maile z zapytaniami. Dziś na oficjalnej stronie Frya widnieje prośba o to, by nie pisać do niego, ponieważ internet obfituje w fora, gdzie ogromna liczba internautów pomaga sobie nawzajem. Dzięki temu, że narzędzie to było udostępnione za darmo, chętni i kreatywni ludzie z całego świata również za darmo pomogli je rozwinąć. Z biegiem czasu powstały implementacje Processingu w innych językach programowania, takich jak: Javascript, Python czy Ruby.

Przykładem strony internetowej, na której entuzjaści tego języka programowania inspirowali się nawzajem, jest forum Open Processing, gdzie setki użytkowników dzielą się ze sobą napisanym przez siebie kodem⁴. Bardzo istotne, że cały rozwój Processingu opierał się na aktywności jego użytkowników i pisanych przez nich rozszerzeniach dla tego języka. Warunkiem jego istnienia i dalszego rozwoju jest więc współtworzenie go przez rozrastającą się wokół niego społeczność. Język programowania rozwija się wraz z rozwijającymi się wspólnotami kreatywnymi, skupiającymi się wokół niego i komunikującymi się ze sobą w internecie.

Ciekawym przykładem komunikowania się ze sobą użytkowników Processingu jest teledysk piosenki *House of Cards*, który dla zespołu Radiohead stworzył Aaron Koblin, jeden z uczniów Casey Reasa. Programista stworzył animację, która powstała wyłącznie poprzez wizualizację danych uzyskanych za pomocą skanowania trójwymiarowego, zbierającego informacje o kształtach i odległościach obiektów w przestrzeni. W opisie wideoklipu odbiorcy byli zachęceni do tego, by w podobny sposób stworzyć własną wizualizację, korzystając z udostępnionego przez Koblina kodu i instrukcji, a następnie podzielić się publicznie efektem. Dzięki temu powstało wiele alternatywnych, fanowskich wersji teledysku.

Processing jest również wykorzystywany przy realizacji komercyjnych projektów udostępnianych w internecie. Jessica Rosenkrantz, absolwentka MIT, oraz Jesse Louis-Rosenberg, matematyk, stworzyli w 2007

roku platformę, na której można zaprojektować i zamówić biżuterię o bionicznej strukturze⁵. Natomiast Mary Huang, artystka z Kalifornii, stworzyła w Processingu program do projektowania generowanych przez kod ubrań w celu demokratyzacji krawiectwa obejmującego szycie na zamówienie, które należy raczej do luksusowych praktyk⁶.

Użytkownicy Processingu wymieniają się za pośrednictwem internetu napisanym przez siebie oprogramowaniem, które można w tym wypadku traktować jako akt mowy, komunikat, sposób przekazu informacji innej osobie. Geoff Cox podkreślał, że każdy akt mowy przeprowadzony w przestrzeni publicznej ma charakter polityczny, ponieważ jest on wypowiediany wobec społeczeństwa. W ten sposób badacz zwracał uwagę na polityczny charakter udostępnianego w sieci kodu [Cox and McLean 13-14]. Oprogramowanie napisane w Processingu można także określić jako polityczny akt mowy, który traktuje o demokratyzacji kodu, przekazaniu go jako narzędzia twórczego dla wszystkich, którzy chcą poprowadzić samodzielną, świadomą narrację, opartą na cyfrowych narzędziach twórczych. Processing jest więc atrakcyjnym punktem odniesienia dla humanistyki przez wzgląd na sam fakt jego politycznego charakteru. Kod komputerowy może być rozumiany jako kolejny język, ukształtowany w cybernetycznej kulturze. Oznacza to, co również podkreślał Cox, że nie jest on uniwersalnym i niewidzialnym spoiwem świata, ale jest polityczny, zawiera w sobie określoną ideologię i zgodnie z nią kształtuje rzeczywistość. Już sam wybór języka programowania, za pomocą którego się tworzy, jest decyzją o charakterze politycznym, ponieważ wybór ten dotyczy w istocie obieranej ideologii, wobec której podejmuje się działanie⁷.

5 Zob. *Nervous System*. <https://n-e-r-v-o-u-s.com>.

6 Zob. *Mary Huang*. <http://www.rhymeandreasoncreative.com/portfolio/index.php?project=continuum>.

7 Co ciekawe, spór o to, czy sam system, w ramach którego powstaje dzieło sztuki, jest obiektywny, czy nie, był obecny w sztuce jeszcze przed popularyzacją komputerów. Większość conceptualistów (w tym Sol LeWitt) oraz artystów sztuki systemowej uważało ją za uniwersalną, pozbawioną subiektywności. Historyk sztuki Grzegorz Dziamski podkreślał, że już decyzja dotycząca tego, jaki system zastosować w pracy, nosi znamiona subiektywności [71].

4 Zob. *Open Processing*. <https://www.openprocessing.org>.

Wpływ na inne kreatywne narzędzia

Oprócz tego, że Processing sam się rozwija, ma również wpływ na inne kreatywne narzędzia. Za przykład posłużyć może Arduino, które zostało wprowadzone w 2005 roku. Nazwą tą określa się mikrokontrolery, których idea opiera się na, z założenia prostym w obsłudze, hardware i software. Zanim pojawiły się mikrokontrolery, budowanie instalacji artystycznych wiązało się z wysiłkiem umieszczenia w jej obrębie całego komputera przy jednoczesnym zachowaniu funkcjonalności i estetyki kompozycji. Arduino umożliwia budowanie małych podzestawów na płytkach i tworzenie niedużych, ale bardzo praktycznych mikrokontrolerów. Pozwala ono korzystać z poszczególnych funkcji komputera bez konieczności umieszczania go całego w obrębie instalacji. Na płytce Arduino zgrywa się kontrolujące cały podzespół oprogramowanie, które wcześniej trzeba napisać za pomocą Arduino IDE, opierającego się na IDE przygotowanym dla Processingu. Mikrokontrolery Arduino można w stosunkowo prosty sposób łączyć z kodem napisanym w Processingu – istnieją stworzone specjalnie ku temu biblioteki. Zjawiska te są sobie pokrewne – na Arduino można patrzeć jak na kontynuowanie idei Processingu. Różnica między nimi polega na tym, że twórczość związana z Processingiem jest skoncentrowana wokół pisania oprogramowania, natomiast użytkownicy Arduino skupiają się przede wszystkim na budowie fizycznych podzespołów.

Geoff Cox wyznaczył indeterminizm jako cechę oprogramowania. Powołując się na teorię sprawczości aktów mowy, wypracowaną przez Johna Langshawa Austina, Cox określa kod programistyczny jako samospełniającą się przepowiednię [Cox and McLean X]. Oznacza to, że determinuje on rzeczywistość zgodnie z zawartymi w sobie informacjami. Z tej perspektywy można uznać, że oprogramowanie warunkuje kształt przyszłości poprzez zawieranie w sobie danych na jej temat. Jak widać na przykładzie Processingu, języki programowania kształtują również kolejne narzędzia cyfrowe, a więc mają bezpośredni wpływ na sposoby, za pomocą których mogą wyrażać się ich użytkownicy.

Processing jako narzędzie artystyczne

W 2003 roku na światowym festiwalu sztuki nowych mediów, Ars Electronica, Casey Reas, jeden z twórców Processingu, miał dwa wystąpienia konferencyjne, a specjalnie wydzielona część ekspozycji była przeznaczona tylko na prace wykonane za pomocą Processingu. Wówczas ten język programowania nie był jeszcze tak popularny, dlatego trzeba było przedstawić stworzone na jego podstawie dzieła zbiorowo, żeby zwrócić uwagę na samo zjawisko. W 2005 roku, w ramach Prix Ars Electronica, Ben Fry i Casey Reas otrzymali nagrodę Golden Nica, która była uhonorowaniem ich pracy. Dzięki tym wydarzeniom Processing zaczął zyskiwać popularność w obszarze sztuki. Obaj autorzy tworzyli za jego pomocą prace prezentowane w galeriach, przez co przyczyniali się do popularyzowania tego narzędzia wśród artystów.

Prace Frya stanowią serię eksperymentów poświęconych sposobom przedstawiania różnego rodzaju informacji. Artysta najczęściej sięga po dane trudne, nieuchwytnie z powodu poziomu skomplikowania lub przez ogromny rozmiar bazy danych. Jednym z pierwszych jego projektów jest *Valence* (1999–2002), w którym przedstawił za pomocą dynamicznej, trójwymiarowej wizualizacji treść powieści Marka Twaina *Prostaczkowie za granicą*. Napisany przez niego program czyta powieść Twaina liniowo, zdanie po zdaniu, i dynamicznie dodaje każdy następny wyraz do trójwymiarowej przestrzeni, łącząc wszystkie słowa powieści w relacyjną sieć. Każdy wyraz jest połączony wiązką z innymi, z którymi wchodzi w relację. Całość tworzy razem coś na kształt globusa danych. Jest to przestrzenna struktura zmieniająca się w czasie – wraz z pojawiającymi się następnymi elementami powieści. Słowa jako elementy tej sieciowej konstrukcji oddziałują na siebie nawzajem, ale także na bieżąco definiują swoje miejsce w całej kompozycji. Fry zakładał, że najlepszym sposobem na przedstawianie i analizowanie wielkiej ilości danych jest wykorzystanie takiego sposobu wizualizacji, który uwypukli skłonności porządkowania się danych (trendy) i anomalie zachodzące w ramach bazy danych. Przykład analizy danych, będących po prostu zbiorem wyrazów stanowiących treść książki, sam autor tych prac określał jednak jako mało praktyczny,

ponieważ dalej dane te pozostają zbiorem słów i ich badanie nie wnosi wiele ponad liczenie powtarzających się wyrazów. Dlatego w późniejszych latach artysta udoskonalał system Valence i wykorzystywał do przedstawiania różnych informacji, na przykład do aktywności na witrynach internetowych czy do zobrazowania ludzkiego genomu, do czego doszedł w projekcie z 2002 roku pod nazwą *Genome Valence*, w którym przedstawił działanie algorytmu BLAST, najczęściej stosowanego do analiz DNA. Projekt Frya był prezentowany w przestrzeni galeryjnej jako dzieło sztuki, ale rzeczywiście był także stosowany przez naukowców do ich pracy, co więcej, wszedł on także w obieg kulturowy i został wykorzystany w filmie *Hulk* z 2003 roku. Jedną z ostatnich scen przedstawia główną bohaterkę, Betty Ross (graną przez Jennifer Connelly), analizującą próbki na ekranie komputera w swoim laboratorium. Dostrzec można pojawiającą się nazwę programu *Valence* oraz charakterystyczną kulistą wizualizację.

Twórczość Frya jest wielowymiarowa, nie pozwala się jednoznacznie rozumieć jako działalność artystyczna, ale raczej funkcjonuje symultanicznie na różnych płaszczyznach kulturowych, w przestrzeni sztuki, nauki i w powszechnym odbiorze społecznym. Jest to doskonały przykład tego, jak głęboko zanurzony w kulturze może być jeden język programowania.

Innym przykładem wykorzystania Processingu w celach artystycznych jest praca *Unnamed sound-sculpture* (2012) Daniela Frankego i Cedrica Kiefera. Ta rzeźba dźwiękowa jest wyrazem tego, jak można budować dramaturgię wizualną na podstawie przetwarzanych danych. Wizualizacja przedstawiała dane pobrane z trzech kontrolerów Kinect, które skanowały ciało tancerki w ruchu. Program napisany w Processingu przemieniał analizowaną sylwetkę w ruchomą, rozsypującą się chmurę punktów. Połączenie języka programowania Processing z kontrolerami Kinect było możliwe dzięki napisanej przez Daniela Schiffmana bibliotece, rozszerzającej możliwości języka.

Wszelkie dziedziny projektowe, takie jak design i architektura, stoją przed nowym wyzwaniem, które postawiła przed nimi możliwość generowania

nieskończenie gęstych informacji. Architektura komputacyjna opiera się na oddaniu różnych decyzji projektowych oprogramowaniu. Artystą, który wykorzystuje Processing, aby kreować detaliczne formy architektoniczne, jest Michael Hansmeyer. Projektuje on wzory w Processingu, a potem fragmentarycznie drukuje za pomocą techniki druku 3D i następnie składa w całość. Artysta stara się zadać pytanie, czy możemy wyprodukować więcej, niż jesteśmy w stanie zaprojektować. Czy design nie nadażę za możliwościami, które oferuje szybko rozwijająca się technologia?

Processing wobec humanistyki

Podstawowym problemem, na który uwagę zwracali członkowie Software Studies Initiative, był brak odpowiedniego, uniwersalnego języka, który służyłby mówieniu o oprogramowaniu i pozwalałby na propagowanie jego kulturowego znaczenia w sposób zrozumiały dla każdej dziedziny naukowej. Badacze związani z założoną przez Manovicha inicjatywą starali się wypracować tego rodzaju język poprzez analizę kulturową oprogramowania, którą zawarli w swoich książkach. Większość z nich została opublikowana w internecie, dzięki czemu treści w nich zawarte mogły szybciej uzyskać większy zasięg, niż gdyby miały one zostać wydane jedynie w postaci analogowej. Odbiorcami tych lektur były osoby związane ze środowiskiem akademickim. Twórcy Processingu również dążyli do stworzenia języka zrozumiałego dla wszystkich. Fryowi i Reasowi udało się osiągnąć wyznaczony cel. Język programowania Processing stanowi metanarrację na temat oprogramowania i jego znaczenia w kulturze, a przy tym jest niezwykle uniwersalny. Processing jest więc przykładem krytycznego i edukacyjnego dyskursu, funkcjonującego w ramach panującej konsumpcyjnej ery, który rozwinął się szybko, zaangażował bardzo szerokie i zróżnicowane grono odbiorców, a przy tym okazał się łatwo przyswajalny. Patrząc na dającą się liczyć w setkach tysięcy liczbę wyświetleń filmów Daniela Schiffmana na YouTube, można dojść do wniosku, że więcej osób słyszało o Processingu niż o Software Studies Initiative. Istotnym z perspektywy nauk humanistycznych aspektem Processingu jest więc to, że podobnie jak teorie naukowe został on

ukształtowany w środowisku akademickim, jednak o wiele skuteczniej sprostał zadaniu wytworzenia uniwersalnego, interdyscyplinarnego sposobu mówienia o znaczeniu oprogramowania w kulturze. Processing wytworzył uniwersalny metajęzyk, zawierający w sobie ukrytą narrację o znaczeniu kodu, którego ludzie z chęcią się uczyli, ponieważ umożliwiał im kreatywny sposób wyrazu. Warto jednak zaznaczyć, że powszechnie w kulturze ignoruje się obecność języków programowania. Nawet w obszarze sztuki, w opisach prac artystycznych, które mają za zadanie uwzględnić materiały i technikę wykorzystane do wyprodukowania dzieła, takie jak na przykład „olej na płótnie”, bardzo rzadko zwraca się uwagę na języki programowania. W pracach cyfrowych najczęściej ucieka się do ogólnego określenia *custom software*, które oznacza wykorzystanie bliżej nieokreślonego, popularnego oprogramowania. Jest to znamieny przykład tego, że w kulturze nie rozpowszechniły się jeszcze świadomość i wrażliwość na zróżnicowanie języków programowania i ich znaczenie.

Podjęcie rozważań na temat oprogramowania powinno być obecne w pracach badawczych humanistów, ponieważ to ich zadaniem jest wytwarzanie krytycznych narzędzi, które pozwalają na analizę rozwijających się w kulturze zjawisk mających bezpośredni wpływ na egzystującą w systemie kulturowym jednostkę ludzką. Współcześnie kultura cyfrowa obfituje w rozmaitego rodzaju strategie wpływania na człowieka, które bardzo często sprowadzają go do roli biernego konsumenta treści masowych, produkowanych na co dzień w ogromnych ilościach. Dlatego tak potrzebne są teorie sformułowane przez badaczy wywodzących się z nauk humanistycznych oraz wprowadzanie krytycznego dyskursu na temat oprogramowania i jego znaczenia. Niestety rozwój technologiczny postępuje znacznie szybciej niż rozwój nauk humanistycznych. Wobec tego kryzysu przydatne dla nich okazują się narzędzia Open Source, takie jak Processing. Być może przesilenie związane z rozpędzającym się rozwojem technologicznym i nienadążającą za nim humanistyką doprowadzi do zmian w jej obszarze, dzięki którym jeszcze bardziej otworzy się ona na interdyscyplinarność i współpracę z innymi dziedzinami. Językoznawca z Uniwersytetu Jagiellońskiego, Jan Rybicki, w ankiecie przeprowadzonej

dla czasopisma „Czas Kultury” stwierdził, że szanse, jakie niesie cyfrowa humanistyka, wyrażają się w tym, iż dyscyplina ta będzie mogła się otworzyć na atmosferę współpracy i wartość pracy grupowej. Aby bowiem wyjść naprzeciw wymaganiom, które stawia cyfrowa humanistyka, nie wystarczy pojedynczy naukowiec, nawet specjalista w swojej dziedzinie, ale potrzeba tu zróżnicowanego zespołu, w ramach którego każdy będzie prezentował inne kompetencje [Rybicki 5]. Tak samo jak w kulturze przenikają się wpisane w nią sfery życia, takie jak sztuka, społeczeństwo, technologia, tak w naukach humanistycznych, które stają przed wyzwaniem oprogramowania, należy również zainicjować przenikanie się kwalifikacji poprzez tworzenie interdyscyplinarnych zespołów badawczych. Processing natomiast pozostaje medium, które łączy wszelkie te dziedziny i stwarza dla nich przestrzeń kreatywnego porozumienia.

Lista prac cytowanych

- Cox, Geoff, and Alex McLean. *Speaking Code: Coding as Aesthetic and Political Expression*. The MIT Press, 2013.
- Dziamski, Grzegorz. *Przełom konceptualny i jego wpływ na praktykę i teorię sztuki*. Wydawnictwo Naukowe UAM, 2010.
- Fry, Ben. *Visualizing Data*. O'Reilly Media, 2008.
- Manovich, Lev. *Język nowych mediów*. Translated by Piotr Cypriański, Oficyna Wydawnicza ŁośGraf, 2006.
- . *Software takes command*. Bloomsbury Academic, 2016.
- Mary Huang. <http://www.rhymeandreasoncreative.com/portfolio/index.php?project=continuum>.
- Nervous System*. <https://n-e-r-v-o-u-s.com/cellCycle>.
- Open Processing*. <https://www.openprocessing.org>.
- Reas, Casey, and Ben Fry. *Make: Getting Started with Processing*. Maker Media, 2015.
- . *Processing: A Programming Handbook for Visual Designers*. The MIT Press, 2014.
- Rybicki, Jan. "Cyfrowa humanistyka w Polsce: działania i konteksty". Questionary directed by Lucyna Marzec, and Marcin Wilkowski, *Czas Kultury*, vol. 185, no. 2, 2015, p. 5.
- Shiffman, Daniel. "Introduction". Ira Greenberg. *The Essential Guide to Processing for Flash Developers*, Springer, 2009, p. IX.
- . *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*. Morgan Kaufman, 2008.
- . *The nature of Code*. Daniel Shiffman, 2012.

Abstract

Anna Paprzycka

The Art of Programming: Processing as an Artistic and Culture-Forming Tool

The Processing programming language is widely used by artists across the world. This tool, deeply emerged in culture, is an extraordinary bridge between science, technology, art and society. As such, it becomes a starting point for analyses of contemporary culture. The article presents a proposal to analyse programming languages in many aspects, drawing attention to their social, educational or

political dimension. Software transcends contemporary culture and becomes a material context for humanistic research.

Keywords: Processing, generative art, digital art, new media art, creative programming, software studies