http://dx.doi.org/10.18778/8088-214-0.15

Monika Górska-Olesińska

Uniwersytet Opolski
Instytut Polonistyki i Kulturoznawstwa

Kolektywne „prze-pisywanie" e-literatury

Generator poetycki *Taroko Gorge* Nicka Montforta to dzieło wyjątkowe, wśród twórców i znawców literatury elektronicznej posiadające status dzieła kultowego. Zostało włączone (wraz z wybranymi jego „prze-pisaniami") do trzeciej edycji antologii *Electronic Literature Collection*[1]. Znalazło się również na liście dwudziestu pięciu kanonicznych dzieł sztuki nowych mediów, sporządzonej przez polskich znawców tego zagadnienia. Piotr Marecki, autor tekstu poświęconego *Taroko Gorge*, opublikowanego w antologii *Klasyczne dzieła sztuki nowych mediów*, argumentuje:

> Podejście do kwestii autorstwa oraz idei dzieła literackiego, prostota i elegancja kodu, nawiązanie do tradycji literackiej oraz nawiązanie do technik pisania pod przymusem (*writing under constraint*) uczyniły utwór Montforta klasycznym dziełem literatury cyfrowej drugiej generacji[2].

Montfort rozpoczął pisanie kodu *Taroko Gorge* w programie Python 8 stycznia 2009 roku podczas wizyty na Tajwanie. Działał pod wpływem chwili, urzeczony krajobrazem majestatycznego, skalistego wąwozu, stanowiącego serce i główną atrakcję Narodowego Parku Taroko. W autokomentarzu towarzyszącym dziełu artysta i teoretyk oświadczył:

> Skoro inni mogli udawać się do miejsc, w których natura objawia swe piękno, i pisać o tych miejscach poematy, czemuż ja nie miałbym [znalazłszy się w podobnej sytuacji – M.G.-O.] napisać dla odmiany generatora poezji? Generator ten – mniej ezoteryczny niż większość moich poprzednich prac, przyjmujący za swój cel raczej sugestywny opis niż prowokację czy parodię – tworzy strofy, które rozpoczynają się oraz kończą wierszem tekstu typu „path" [„ścieżka" – M.G.-O.] i mogą zawierać w sobie jeden, lub kilka [...] wierszy tekstu typu „site" [„miejsce" – M.G.-O.]. Pomiędzy pary takich strof wplatany jest wiersz tekstu typu „cave" [„jaskinia" –M.G.-O.], który odzwierciedla akt stopniowego rozmywania

[1] http://collection.eliterature.org/3/ [dostęp 25.02.2016].

[2] Zob. P. Marecki, *Taroko Gorge*, [w:] *Klasyczne dzieła sztuki nowych mediów*, red. P. Zawojski, Katowice 2015, s. 167–173.

się w ciemności, znikania w mroku tuneli wydrążonych w Parku Taroko przez armię Czang Kaj-Szeka. Lektura generowanego tekstu może być czymś przyjemnym, wydaje się jednak, że równie ciekawą czynnością mogłoby być drążenie kodu i przekształcanie programu w celu wyrażenia odmiennych doświadczeń i idei[3].

Generator, który po uruchomieniu produkuje poemat będący imitacją romantycznych utworów lirycznych oddających hołd potędze przyrody (na przykład tych pisanych przez Williama Wordswortha)[4], powstał w ciągu jednego dnia, pod przymusem twórczym (*constraint*) przyjętym przez Montforta – autora niejednokrotnie nawiązującego w swej praktyce artystycznej do metod pracy OuLiPo.

---

[3] http://collection.eliterature.org/3/work.html?work=taroko-gorge [dostęp 25.02.2016]. Tłumaczenia cytatów, jeśli nie wskazano inaczej, pochodzą od autorki.

[4] „Celem programów Montforta jest imitowanie, podrabianie określonych technik literackich za pomocą algorytmów. Algorytm *Taroko* składa się z około 130 linijek, które cechuje bardzo przejrzysty układ oparty na przeplataniu się wersów opisujących ruch oraz zatrzymanie i zamyślenie. Generowane są trzy typy wersów: 1. «path»: pierwsze i ostanie linijki strofy mają opisywać efekt ruchu, spaceru po ścieżce górskiej i oglądanie efektów natury rozmieszczonych «ponad» i «pod» przechadzającym się (takie kategorie znajdują się w danych wejściowych); 2. «site»: między wersami typu «path» program losuje wersy (dwa lub jeden), które opisują efekt zatrzymania się, podziwiania krajobrazu; 3. «cave»: ostatnim typem jest wers opisujący wrażenie nieoświetlonego i nieskończonego tunelu wyrzeźbionego w górach. Wchodzący do takiego tunelu podmiot liryczny nie ma szansy zobaczyć jego końca. Jako output otrzymujemy następujący przykładowy tekst […] (każde uruchomienie programu będzie generowało odmienne wiersze):

Shapes dream. (1)
The crags hold. (2)
Heights exercise the stones. (1)
direct the objective clear – (3)
Mists trail the rippling. (1)
Basins hum. (2)
The crags sweep the basins. (1)
run the encompassing fine arched – (3)
Height commands the veins. (1)
The crags exercise the veins. (1)
run the straight arched cool dim – (3)
[…]

Na poziomie danych wejściowych Montfort wykorzystał niewielką liczbę rzeczowników i czasowników, które jednak dzięki sprytnemu chwytowi «rozmnożył». Program dodaje do wybranych rzeczowników końcówkę «s» czyniąc z nich liczbę mnogą, podczas gdy z czasowników tworzy trzecią osobę liczby pojedynczej. Ważnym zabiegiem Montforta, budującym rytm wiersza, jest użycie czasowników przechodnich i nieprzechodnich. Program do linijek z klucza «path» losuje czasowniki przechodnie, czyli te, które mogą mieć dopełnienie bliższe. W wersjach «site» losowane są czasowniki z klucza nieprzechodniego, czyli nie mogące mieć dopełnienia bliższego i nie tworzące strony biernej". P. Marecki, *Taroko Gorge*, s. 169–170.

Oryginalny generator *Taroko Gorge* scala geologiczną formację – rozległą i ukazaną w całej jej różnorodności – z potencjałem rekombinacyjnym obliczeń wykonywanych przez komputer. Istnieje analogia pomiędzy sposobem ukazywania człowieka w malarstwie epoki romantyzmu – człowieka małego i bezsilnego wobec sił nieokiełznanej natury – a sytuacją czytelnika *Taroko Gorge*, skonfrontowanego nagle z wyobrażeniem rozległego pejzażu procedur informatycznych, w efekcie działania których z niewielkiego zestawu elementów tekstowych generowany jest niekończący się tekst poematu, wyświetlany linijka po linijce i przesuwający się ku górnej krawędzi ekranu [...]. I tak, jak nie ma dwóch takich samych płatków śniegu, i niemożliwe jest ponowne wejście do tej samej rzeki, tak każda strofa wygenerowana przez algorytm *Taroko Gorge* jest wyjątkowa i niepowtarzalna (skończona liczba kombinacji dalece przekracza bowiem możliwości percepcyjne odbiorcy, żyjącego zbyt krótko, by poznać i przeczytać je wszystkie). Jeśli czytelnik poświęci dłuższą chwilę na lekturę kolejnych strof, dostrzeże ograniczony system leksykalny poematu, którego struktura jest jednak na tyle wielofunkcyjna, że może wyłaniać z siebie nowe poetyckie formy[5].

Jak podkreślają redaktorzy *Electronic Literature Collection* (ELC, Volume 3) – Leonardo Flores, Anastasia Salter, Stephanie Boluk i Jacob Garbe – równie interesujący i znaczący co *output*, tj. poemat pojawiający się w oknie przeglądarki, jest kod *Taroko Gorge*, cechujący się niezwykłą elegancją, zwięzły i pozbawiony jakichkolwiek zbędnych elementów. Wspomniana tu dbałość o estetkę kodu to – o czym należy wspomnieć – jedna z cech wyróżniających programistyczną działalność Montforta. Udostępniony w sieci jako otwarte oprogramowanie, kod *Taroko Gorge* zaintrygował wielu twórców należących do społeczności ELO (Electronic Literature Organization), którzy zawłaszczając, remiksując, przerabiając, modyfikując oraz parodiując oryginalny program Montforta, wpisując się tym samym w nurt pisarstwa niekreatywnego[6], zainicjowali powstanie odrębnego podgatunku w domenie literatury elektronicznej[7].

Tym, co staraliśmy się zachować [włączając *Taroko Gorge* do trzeciego wydania Electronic Literature Collection –M.G.-O.], jest nie tylko minimalizm i elegancja samego poematu, ale owo szersze zjawisko poddawania kodu zabiegom zawłaszczania (*appropriation*) i remiksu. *Taroko Gorge*, który stworzony jako poemat generatywny, wyewoluował stopniowo w rozbudowywaną nieustannie platformę gry poetyckiej[8].

Inicjatorem owej gry, do której odnoszą się redaktorzy ELC Volume 3, był wieloletni przyjaciel Montforta, Scott Rettberg. Zmodyfikował on oryginalny kod generatora, przenosząc akcję poematu z malowniczego

[5] Zob. *Taroko Gorge Remixes: Editorial Statement*, http://collection.eliterature.org/3/collection-taroko.html [dostęp 25.02.2016].

[6] Por. P. Marecki, *Taroko Gorge*, s. 171.

[7] Zob. *Taroko Gorge Remixes: Editorial Statement*.

[8] Tamże.

wąwozu na Tajwanie do garażu zlokalizowanego w centrum tokijskiej metropolii, zastępując słownictwo związane z naturą słownictwem dotyczącym rzeczywistości industrialnej i popkultury, istotnie powiększając przy tym pulę wyrazów wykorzystywanych przez generator. Publikując własną wersję utworu, zatytułowaną *Tokyo Garage*, Rettberg w subwersywnym geście „przekreślił nazwisko Montforta za pomocą komendy <del> i umieścił swoje jako nazwisko autora wraz z przekreślonym nazwiskiem autora oryginału”[9], zapoczątkowując tym samym tradycję, którą konsekwentnie kontynuują twórcy kolejnych „prze-pisań”[10]. Rettberg wspomina:

> Kiedy zobaczyłem *Taroko Gorge* po raz pierwszy, byłem zachwycony mocą tego niewielkiego programu. Doceniłem ironię [ukrytą w dziele Montforta – M.G.-O.] wykorzystującą inwersję – to, co na pierwszy rzut oka wydawało się tradycyjną poezją medytacyjną opiewającą potęgę natury, w rzeczywistości okazywało się automatycznie generowanym cybertekstem. Nigdy nie byłem miłośnikiem wierszy oddających hołd pięknu przyrody, stąd nagły impuls, który kazał mi „poprawić” utwór Nicka. Otwierając poemat *Taroko Gorge* w oknie przeglądarki i wykorzystując opcję „pokaż źródło strony”, uzyskałem dostęp do tekstonów *Taroko Gorge* i przyjrzałem się uważnie strukturze oraz zmiennym programu. Nie prosząc Nicka o zgodę i o niczym go nie informując, przepisałem kod, w pierwszej kolejności zastępując oryginalne słownictwo – według mnie ograniczone – bardziej wyrazistym. Chciałem się przekonać, czy potrafię wykorzystać prostą strukturę utworu generowanego przez program Nicka i odwrócić znaczenia tegoż utworu, przekształcając minimalistyczny poemat obrazujący uporządkowaną i niemal nieskażoną obecnością człowieka naturę w wiersz przedstawiający [...] ogarnięte chaosem miasto, pełne ludzi, którzy swą obecnością generują nieład. Chciałem przełamać refleksyjny, poważny ton utworu Nicka, zastąpić go komizmem i absurdem. I tak w ciągu jednej nocy *Wąwóz Taroko* stał się *Tokijskim Garażem* (*Tokyo Garage*). Zmieniłem wszystkie zmienne, a zatem język utworu, ulepszyłem kilka aspektów programu, zaproponowałem inny schemat kolorów i przyspieszyłem działanie mechanizmu renderowania, przekreśliłem też nazwisko Nicka widoczne w graficznym interfejsie dzieła po prawej stronie i nadpisałem nad nim własne nazwisko, następnie [...] wysłałem do Nicka wiadomość z informacją o adresie URL zmodyfikowanego generatora[11].

---

[9] P. Marecki, *Taroko Gorge*, s. 172.

[10] Redaktorzy trzeciego tomu *Electronic Literature Collection* mówią o dwóch tradycjach zainicjowanych przez Rettberga: pierwsza polega na grze słownej z oryginalnym tytułem generatora (*Taroko Gorge* stawało się w kolejnych inkarnacjach *Tokyo Garage*, *Toy Garbage*, *Taroko Gary*, *Tornedo Gorge*..., a nawet *Wąwozem Kraków*); druga polega na umieszczaniu w graficznym interfejsie generatora listy z przekreślonymi nazwiskami autorów wszystkich poprzednich remiksów. Zob. *Tokyo Garage: Editorial Statement*, http://collection.eliterature.org/3/work.html?work=tokyo-garage [dostęp 25.02.2016].

[11] S. Rettberg, *A Response to Nick Montfort's "Programming for Fun, Together"*. Conference paper. ELMCIP. Conference on Remediating the Social, 2012, s. 2, http://elmcip.net/sites/default/files/files/attachments/criticalwriting/a_response_to_nick_montfort_0.pdf [dostęp 5.05.2015].

Raz przetartym szlakiem szybko podążyli kolejni twórcy – Mark Sample, Maria Engberg, Talan Memmott, Leonardo Flores, Judy Malloy, Helen Burgess, Piotr Marecki i wielu innych – przygotowując kolejne remiksy (i tłumaczenia) *Wąwozu Taroko*. Ci, którzy nie posiadali wystarczających kompetencji informatycznych, ingerowali w kod nieznacznie, zmieniając jedynie słowa w tabelach. Inni czynili takie ingerencje w warstwę kodu, które wymagały większej wiedzy z zakresu programowania i skutkowały zmianami widocznymi na przykład w strukturze składniowej wyświetlającego się na ekranie tekstu. Powstające dzieła, jakkolwiek współdzieliły cyfrowe DNA z oryginalnym programem napisanym przez Montforta, różniły się zarówno tematycznie, jak i stylistycznie[12], prowokując swych interpretatorów do dyskusji o kształtującym kulturę remiksu napięciu, powstającym w miejscach, w których powtarzalność wpisana w procesy apropriacji zderza się z różnicą wynikającą z odczuwanej przez autorów kolejnych przetworzeń potrzeby nadania indywidualnego charakteru własnemu projektowi[13].

Wśród osób zaangażowanych w kolektywne „prze-pisywanie" kodu *Wąwozu Taroko*, które to „prze-pisywanie" scaliło i jeszcze silniej zintegrowało wspólnotę twórców i teoretyków literatury elektronicznej[14], znalazła się Kathi Inman Berens. Poczynione przez nią modyfikacje dotyczyły w pierwszej kolejności osi paradygmatycznej. Oryginalne słownictwo *Taroko Gorge* zastąpiła Berens wyrazami przynależącymi do dwóch, zgoła odmiennych i zdających się nie posiadać elementów wspólnych, pól znaczeniowych. W generatywnym poemacie *Tournedo Gorge*, którego tytuł odsyła poprzez swój pierwszy człon do nazwy jednej z potraw, pochodzącej ze słynnej książki kucharskiej *Mastering the Art of French Cooking* autorstwa Julii Child[15] (pierwszej kobiety, której udało się ukończyć prestiżową szkołę kucharską Le Cordon Bleu oraz wywalczyć swe miejsce nie tylko w zdominowanym przez mężczyzn środowisku szefów kuchni, ale i w świecie gospodarzy programów telewizyjnych, w którym prym również wiedli wówczas mężczyźni), słownictwo związane z czynnościami gotowania

[12] Zob. F. Klink, T. Memmott, N. Montfort, A. Plotkin, S. Rettberg, M. Sample, E. Snodgrass, *Taroko Gorge Remixes: Repetition and Difference in Machine Text*, http://elmcip.net/critical-writing/taroko-gorge-remixed-repetition-and-difference-machine-texts [dostęp 25.02.2016].

[13] Tamże.

[14] Marecki pisze: „[U]twór zainicjowany przez amerykańskiego autora stał się przykładem dzieła zbiorowego, napisanego i sygnowanego przez wspólnotę [...] utwór wraz ze wszystkimi przeróbkami wpisuje się w nurt zespołowego programowania dla zabawy i celów twórczych, czy traktując to szerzej, programowania jako czynności społecznej i scalającej pewne mikrouniwersum". P. Marecki, *Taroko Gorge*, s. 172.

[15] Mowa o przepisie na potrawę Tournedos Sautes Chasseur.

miesza się z komendami języków programowania oraz wyrazami odnoszącymi się do procesu pisania kodu komputerowego.

> Zabawa poematem Montforta była moim pierwszym zetknięciem się z językiem skryptowym JavaScript. Gdy wchodzę w rolę kucharki, [...] zaglądam do lodówki oraz spiżarni i na podstawie tego, co tam znajduję, obmyślam kombinacje składników potrawy, które mogłyby razem dobrze smakować [...]. Kod Montforta potraktowałam jak kuchnię, w której mogę pomieszać ze sobą różne składniki[16].
>
> W kwestii kodu jestem nowicjuszką, kucharką zdającą się na intuicję. Cieszy mnie łączenie tego, co proceduralne/formalne, z tym, co domowe, kobiece, namacalne[17].

Berens, nie mając wcześniejszych doświadczeń z językiem JavaScript (nie ona jedna rozpoczęła swą przygodę z programowaniem od przeróbek *Taroko Gorge*[18]), działała metodą prób i błędów. Eksperymentowała z poszczególnymi elementami składni JavaScriptu, podobnie jak eksperymentuje się w kuchni, gdy szuka się zamienników dla tych składników wyszczególnionych w przepisie, których chwilowo brak pod ręką. Analizowała i porównywała ze sobą kody innych remiksów *Taroko Gorge*, by wywnioskować, jakie operacje wykonują poszczególne operatory. Jej nauce przyświecała idea *Do-It-Yorself*[19]. Jak przyznała, dopiero wgłębienie się w poszczególne warianty kodu, powstałe w efekcie kolejnych artystycznych „prze-pisań" *Taroko Gorge*, pozwoliło jej dostrzec całą złożoność programu napisanego przez Montforta[20]. „Cooks and coders gets their hands dirty" („Kucharze/kucharki i programiści/programistki brudzą sobie ręce")[21]. Owo doświadczenie „brudzenia rąk kodem", towarzyszące odkrywaniu i poznawaniu tajników pisania kodu, wykorzystała Berens w dydaktyce akademickiej, zalecając uczestnikom kursu z zakresu literatury elektronicznej, który prowadziła w 2014 roku na Uniwersytecie w Bergen, przygotowanie własnych remiksów generatora Montforta.

W piśmiennictwie zachodnim wiele uwagi poświęca się zagadnieniom dydaktyki e-literatury, metodom nauczania czytania oraz krytycznej interpretacji jej utworów – utworów, których odbiór wymaga przecież nowych kompetencji (także technicznych), wykraczających znacznie poza

---

[16] K. Inman Berens, *Tournedo Gorge. Artist's Statement*, http://www.alimentumjournal.com/tournedo-gorge#.VVknevntmkp [dostęp 5.05.2015].

[17] http://berens.org/kib/elit/tournedo2.html [dostęp 5.05.2015]. Por. K. Inman Berens, *My first digitally-born poem „Tournedo Gorge"*, http://kathiiberens.com/2012/07/01/tournedo-gorge [dostęp 5.05.2015].

[18] P. Marecki, *Taroko Gorge*, s. 172.

[19] K. Inman Berens, *Tournedo Gorge. Artist's Statement*.

[20] Tamże.

[21] Tamże.

te wyznaczone ramami tradycyjnej lektury. Warto w tym miejscu wspomnieć o publikacjach z serii Media Upheavals wydawnictwa Transcript – między innymi o tomie *Reading Moving Letters. Digital Literature in Research and Teaching. A Handbook* z artykułem Gendolli, Schäfera i Tomaszek, zawierającym przykładowe scenariusze zajęć[22], o sprawozdaniu Marii Engberg z warsztatów „Digital Literature and Pedagogy" włączonym do raportu *Electronic Literature as a Model of Creativity and Innovation in Practice*[23] oraz o materiałach dydaktycznych zamieszczanych na blogach akademików[24] .

> Nauczanie o projektach literackich wykorzystujących komputery i sieć – zauważają autorzy artykułu *Net Literature in the Classroom. Teaching Practice at the University of Siegen* – nie jest łatwym przedsięwzięciem. Nie mamy bowiem do czynienia z kontynuacją nauki o znanych (tradycyjnych) formach literackich [...]. Literatura elektroniczna przynosi ze sobą nowe formy, wobec których dotychczas stosowane metody dydaktyczne tracą swą użyteczność[25].

Dzieje się tak z prostej przyczyny – formy te nie są trwałymi obiektami (tekstami utrwalonymi na materialnych nośnikach), lecz wydarzeniami, programami uruchamianymi i wykonywanymi na komputerze[26]. Sposób „wykonania" każdego programu/dzieła e-literackiego jest uwarunkowany kodem. Uświadomienie sobie „bezpośredniej obecności kodu jest kluczowe dla uchwycenia istoty literatury elektronicznej"[27], o czym przypomina N. Katherine Hayles. Zagłębianie się w kod, badanie tego, jak dzieło zostało zaprogramowane, to działania kluczowe dla rozumienia utworów literatury elektronicznej. Stąd też kształcenie w zakresie e-literatury powinno obejmować także rozwijanie umiejętności czytania kodu oraz umiejętności dostrzegania wzajemnych relacji pomiędzy „technologicznie zapośredniczonymi procesami a osadzonymi

---

[22] P. Gendolla, J. Schäfer, P. Tomaszek, *Net Literature in Classroom. Teaching Practice at the University of Siegen*, [w:] *Reading Moving Letters. Digital Literature in Research and Teaching. A Handbook*, ed. R. Simanowski, J. Schäfer, P. Gendolla, Bielefeld 2010, s. 273–289.

[23] M. Engberg, *Electronic Literature Pedagogies*, [w:] *Electronic Literature as a Model of Creativity and Innovation in Practice. A Report from the HERA Joint Research Project*, ed. S. Rettberg, S. Baldwin, Morgantown 2014, s. 69–84.

[24] Doskonałym źródłem takich materiałów jest prowadzony przez Leonardo Floresa blog www.iloveepoetry.com.

[25] P. Gendolla, J. Schäfer, P. Tomaszek, *Net Literature in Classroom...*, s. 273.

[26] Zob. N.K. Hayles, *The Time of Digital Poetry. From Object to Event*, [w:] *New Media Poetics: Contexts, Technotexts, and Theories*, ed. A. Morris, T. Swiss, Cambridge (Mass.)–London, s. 181–211.

[27] N.K. Hayles, *Electronic Literature: What Is?*, http://eliterature.org/pad/elp.html [dostęp 5.05.2015].

w kulturze praktykami wytwarzania znaczeń”[28]. David Berry, autor książki *Understanding Digital Humanities*, postulował, by adepci kierunków humanistycznych w toku swych studiów, zamiast zajęć z dodatkowego języka obcego, wybierali zajęcia z języka programowania[29]. Idealną sytuacją edukacyjną byłaby taka, w której uczestnicy kursów z zakresu e-literatury sami programowaliby i znali dobrze *software* stojący za utworami analizowanymi i interpretowanymi podczas zajęć. Brak umiejętności kodowania zasygnalizowany przez uczestników takich zajęć nie musi jednak wcale ograniczać procesu edukacyjnego jedynie do dyskusji o dostępnej dla każdego odbiorcy powierzchni dzieła. Można bowiem, w duchu idei przyświecających cyfrowej humanistyce[30], zasypywać przepaść pomiędzy ugruntowanymi w teorii aktami analizy i interpretacji cyfrowych utworów a praktyką wytwarzania kodu, wykorzystując w tym celu i modyfikując kod istniejący, który – tak jak to ma miejsce w przypadku tekstowych maszyn Nicka Montforta, wśród których znajduje się *Taroko Gorge*[31] – bywa udostępniany jako darmowe i otwarte oprogramowanie. Modyfikując kod utworów e-literackich, a następnie obserwując rezultaty wprowadzanych zmian, studenci zdobywają wiedzę o procesach zachodzących pod powierzchnią interfejsu cyfrowych dzieł oraz zyskują lepsze zrozumienie istoty i znaczeń utworów, o których się uczą.

Collective “re-writing” of e-lit

Summary

This article describes *Taroko Gorge*, a short program written by Nick Montfort in 2009 in Python, and later in JavaScript – a program that generates an infinite nature poem inspired by the stunning Taroko

[28] N. Wardrip-Fruin, *Learning to Read Digital Literature*, [w:] *Reading Moving Letters...*, s. 250–252.

[29] Zob. M. Pisarski, *Autor i dzieło w kulturze cyfrowej (na przykładzie twórczości Nicka Montforta)*, „Przegląd Kulturoznawczy” 2014, nr 2, s. 149.

[30] Zob. A. Nacher, *Poza cyfrowość w zwrocie cyfrowym – od humanistyki cyfrowej do spekulatywnej komputacji*, [w:] *Zwrot cyfrowy w humanistyce. Internet, nowe media, kultura 2.0*, red. A. Radomski, R. Bomba, Lublin 2013, s. 91–94.

[31] Zob. http://ha.art.pl/prezentacje/projekty/4039-nick-montfort-po-polsku-utwory--cyfrowe [dostęp 5.05.2015]. Piotr Marecki prezentował szczegółowo utwory Nicka Montforta w wystąpieniu pt. *Nick Montfort's Textual Machines* na konferencji „Post-technological Experiences. Art – Science – Technology” zorganizowanej w Poznaniu przez HAT Center w dniach 23–27 października 2014 roku.

Gorge in Taiwan – and two of its remixes: *Tokyo Garage* by Scott Rettberg, which is the first appropriation of *Taroko Gorge* that initiated along tradition of rewriting Montfort's original code, and *Tournedo Gorge* by Kathee Inman Berens. The article draws attention to the possibilities that the remixing process offers in teaching of electronic literature.