

AGNIESZKA PRZYBYSZEWSKA
Uniwersytet Łódzki*

Pisarz (i czytelnik) w laboratorium, czyli od *twórczego pisania* do *twórczego programowania* — przypadek Twine

Writer (and Reader) in a Laboratory, or from *Creative Writing* to *Creative Programming* — Case Study of Twine

Abstract

This paper proposes a reflection on how to introduce some aspects of programming into creative writing study programs. Educational implementation of Twine is an analyzed case study. The problem is discussed in the context of the modern reflection on digital literature pedagogy (methods, topics) as well as of statements on the need of new media literacies. Nick Montfort's conception of *creative/exploratory programming* is also used and discussed.

* Katedra Teorii Literatury Instytutu Kultury Współczesnej
Wydział Filologiczny Uniwersytetu Łódzkiego
ul. Pomorska 171/173, 90–236 Łódź
e-mail: agnieszka_p@smrw.lodz.pl

Między twórczym pisanem a twórczym programowaniem

Szkic ten poświęcony będzie możliwości wykorzystania platformy 'Twine w trakcie kursów *creative writing* (oraz działań edukacyjnych o zbliżonym charakterze), przy czym skupię się nie tylko na kwestiach praktycznych związanych z sięganiem po analizowane narzędzie, lecz i szerszych, kulturowych kontekstach. Prezentowane spostrzeżenia dalyby się dobrze podsumować tytułem artykułu Urszuli Pawlickiej: *To nie jest muzeum, to jest laboratorium*, sprytnie ujmującym istotę przejścia humanistycznych, również literaturoznawczych, badań ku działaniom silniej praktycznym i performatywnym. Współcześnie miejscem „testowania” (bo już nie tylko lektury, analizy i interpretacji) literatury stają się też laboratoria, takie jak choćby opisywane przez Pawlicką Trope Tank na MIT (Massachusetts Institute of Technology) kierowane przez Nicka Montforta¹, pracownia na uczelni technicznej, poświęcona badaniom o wyraźnym zacięciu literaturoznawczym², nastawiona na współpracę, równoległość teorii i praktyki, o której w kontekście nowej humanistyki i jej metodologii pisała (nie tylko) Ewa Domańska (Domańska 2010). Twory takie wydają się naturalną odpowiedzią na przeobrażenie humanistyki w humanistykę cyfrową, na nowy kształt sztuki słowa i zrodzone z niego potrzeby, dotyczące choćby kwestii archiwizacji konkretnych platform lekturowych (a więc umożliwienia czytelnikom poznania/doświadczenia pewnych tekstów w ich oryginalnym, autorsko zaprojektowanym kształcie) bądź stworzenia środowiska dla twórców, w którym mogą oni eksperymentować z technologią. Z racji namysłu nad *twórczym pisanem* ten ostatni wymiar będzie dla mnie zasadniczym punktem odniesienia w tym szkicu³.

¹ Na jego wzór w ramach przyznanego grantu NPRH (*Twórcze programowanie. Laboratorium*) pod kierownictwem Piotra Mareckiego tworzone jest tego typu laboratorium na Wydziale Zarządzania i Komunikacji Społecznej Uniwersytetu Jagiellońskiego. Na rodzimym gruncie takie „techniczne” podejście do literatury i procesu twórczego nie jest jeszcze ani popularne, ani dobrze ugruntowane.

² Sam Montfort wyznaje: „Choć pracuję na MIT i mam doktorat z informatyki, zajmuję się humanistyką i działam w ramach wydziału Comparative Media Studies/Writing” (Montfort i Marecki 2015: 77). A trzeba dodać, że ten informatycznie zaangażowany humanista łączy pracę badawczą z twórczością artystyczną.

³ Świadomie pomijam tu — stanowiący sedno artykułu Pawlickiej — wątek współczesnej humanistyki jako laboratorium w sensie traktowania „tekstu jako laboratorium” (Pawlicka 2015: 61) zaś wspólnoty badaczy jako „kolektywu” pracującego nad mającym realne skutki dla społeczeństwa „projektem” (62–62). Ze względu na to, iż interesuje mnie performatywny wymiar współczesnej humanistyki w kontekście *creative writing* skupiam się przede wszystkim na wątkach, które autorka porusza głównie w ostatniej części swojego szkicu, analizując sposób i formy pracy laboratoriów Montforta i Dene Grigar.

Istotną konsekwencją obserwowanych zmian, szczególnie zaś pojawiania się pewnych form literackich, jest konieczność kształtowania nowych kompetencji medialnych, również literackich i lekturowych, często określanych jako *e-literacies* (Dene Grigar pisała tu o *technological literacy* — Grigar 2008; Roberto Simanowski o *transliteracy* — Simanowski 2010; Henry Jenkins o *new media literacies* — Jenkins i in. 2009). Owocuje ona coraz częstszym przenikaniem treści medialnych, informatycznych do kształcenia literaturoznawczego, czemu nierzadko towarzyszy też przejście od trybu analitycznego do warsztatowo-laboratoryjnego, gdyż najprościej wytłumaczyć i zrozumieć nowe formy niejako „w działaniu”. Głosy o konieczności rozwijania kompetencji ściśle literaturoznawczych w powiązaniu z medialnymi (cyfrowymi) stale pojawiają się w debatach na temat uczenia o e-literackości (por. m.in. Engberg 2014 czy cała część *Teaching Digital Literature* w opracowaniu *Reading Moving Letters* — Simanowski, Schäfer i Gendolla 2010⁴) i wspólnie z tezami, że ten wymiar edukacji (zarówno w zakresie tematyki, jak i metodologii) dopiero jest dopracowywany (zob. np. Ensslin i Pope 2010: 311). Kształcenie w zakresie twórczego pisania, z definicji wpisujące się w model „laboratoryjny”, również kładzie na wspomnianą współzależność silny nacisk, co z kolei prowokuje do pytań o to, jak wiele uwagi należałoby w trakcie studiów tego typu poświęcić kwestiom warsztatowym związanym z technologią.

Zagadnienie nieodzowności takiego dwubiegunowego kształtowania kompetencji interesowało mnie już w szkicu *Medialne uwikłania literatury a twórcze pisanie* (Przybyszewska 2015a), w którym skupiałam się na tym, jak w ramach specjalności twórcze pisanie (prowadzonej w obrębie kulturoznawstwa na Uniwersytecie Łódzkim od roku akademickiego 2011/12) staramy się konkretnymi działaniami programowymi odpowiedzieć na te potrzeby. Wniosków tych nie chciałabym tu powtarzać. Pozostając w kręgu refleksji o *creative writing* w kontekście „laboratoryjnego” wymiaru kształcenia i nauczania o e-literaturze, zamierzam tym razem postawić pytania o użyteczne narzędzia dydaktyczne, więc poniekąd i o — związane z sięganiem po nie — konieczne kompetencje wykładowców⁵. Rozwinę zatem, sygnalizowany w przywołanym szkicu, problem wdrażania w zajęcia o charakterze nawet jeśli warsztatowym, to prymarnie przecież literaturoznawczym, kwestii czysto technologicznych, informatycznych. Wydaje się, iż w kontekście nauczania najnowszych form literackich kształcenie w obrębie *twórczego pisania*, jeśli istotnie ma przygotować absolwentów do tworzenia e-literatury, winno obejmować zarówno rozwój zdolności ściśle literackich, jak i umiejętności technologicznych. Stąd przyczynkiem do tego szkicu stały się pytania o to, czy rzeczywiście wykładowca *creative writing* (a może i mniej warsztatowych w charakterze kierunków literaturoznawczo zorientowanych, gdy porusza kwestie związane z e-literackością) powinien uczyć również programowania oraz jak takie „informatyczne” podejście łączyć z tradycją.

W konsekwencji niniejsze rozważania muszą mieć jeszcze jeden punkt odniesienia: przywołaną już w tytule koncepcję *kreatywnego programowania* (ang. *creative programming*, także *exploratory programming*) zaproponowaną przez Montforta, którą postrzegać można jako swoiste dopełnienie *twórczego pisania* (zob. Montfort 2016; Montfort i Marecki 2015). W jej świetle wypróbowanie najprostszych struktur programistycznych, spojrzenie niejako „od

⁴ Na polskim gruncie problemem tym, poza Górską-Olesińską, zajmowała się też Pawlicka (Pawlicka 2014).

⁵ Na problematyczność tej ostatniej kwestii zwracał uwagę m.in. Simanowski, podkreślając, iż nauczyciele naprawde należący do pokolenia „cyfrowych tubylców” dopiero zostaną wyedukowani (Simanowski 2010: 241).

kuchni” na utwory pisane z użyciem informatycznych narzędzi, może pomóc zrozumieć ich istotę. Takie „laboratoryjne” podejście do tekstów nowomiedialnych wpisuje się też w obszar toczonych dyskusji o znaczeniu kodu (czy wyższych zdaniem Montforta poziomów, jak interfejs czy platforma) dla semantyki utworu. Dlatego niniejsze rozważania traktuję również jako kolejny głos w dyskusji o tym, czy, interpretując teksty nowomiedialne, zawsze musimy zaglądać pod ich „podszewkę”, „czytać kod”, głos przeradzający się w wątpliwość, czy e-literat sam musi ów kod pisać.

W szkicu tym przyjrze się edukacyjnemu wykorzystaniu platformy Twine w kontekście nauczania o formach literatury nieliniowej. Bazą dla prezentowanych wniosków będą doświadczenia dydaktyczne — od 2014 roku w pracy akademickiej korzystam z Twine na różnych poziomach kształcenia: studiach I stopnia (na zajęciach *Współczesne strategie narracyjne* oraz *Literatura i nowe media*, a także autorskim fakultatywnym kursie *Warsztat twórczego pisania — literatura nieliniowa*), II stopnia (w ramach kursu *Kreacja przekazu wielotworzywowego* oraz wspomnianego kursu fakultatywnego), a także różnego typu warsztatów. Prace powstające w trakcie przywołanych kursów oraz opinie ich uczestników stanowią dla mnie istotny punkt odniesienia. Moim celem będzie zaproponowanie sposobu wprowadzania elementów związanych z technologicznym projektowaniem tekstów do treści kształcenia oraz przedstawienie Twine jako narzędzia pozwalającego nie tylko samodzielnie tworzyć teksty nieliniowe, lecz również lepiej zrozumieć ten typ pisarstwa. Ten drugi wymiar wydaje się znacznie ważniejszy i pozwala traktować charakteryzowaną platformę jako narzędzie edukacyjne na każdym typie studiów humanistycznych.

Spróbuj a zrozumiesz — Twine i myślenie mapą tekstu

Twine to darmowa, dostępna na stronie www.twinery.org (w wersji online, a także do zainstalowania na własnym komputerze), platforma służąca w najszerszym ujęciu do tworzenia olinkowanych, a więc interaktywnych, tekstów nieliniowych, często charakteryzowana głównie jako narzędzie do kreacji *interactive fiction*. Zaprojektowana w 2009 roku przez Chrisa Klimasa i nadal rozwijana jako narzędzie *open-source* zyskała wielu zwolenników w dużej mierze dzięki zalecie, którą Montfort ujmuje w stwierdzeniu, że Twine ma „niski poziom wejścia” (Montfort 2015: 83). Innymi słowy: czytanie i pisanie w tym programie nie wymaga niemal żadnych instrukcji, jest niezwykle intuicyjne⁶. Popularność platformy pieczętuje fakt, iż — niezależnie od szerokiej dostępności (ciągle uaktualnianych i rozwijanych) multimedialnych tutoriali — w maju 2016 roku ukazał się drukowany podręcznik Melissy Ford *Writing Interactive Fiction with Twine*⁷.

Zdaniem badaczki, Twine to idealne narzędzie zarówno dla twórców gier (*games*), jak i opowieści (*stories*) (Ford 2016: 2). Dla mnie okazał się platformą użyteczną w pracy dydaktycznej w trakcie omawiania literatury nieliniowej, za której odmianę traktuję również

⁶ Zapewne dlatego (aczkolwiek również z innych względów, wśród których nie sposób nie wymienić: istnienia już wspólnoty miłośników *interactive fiction*, a więc docelowej grupy odbiorczej oraz coraz popularniejszych warsztatów i konkursów upowszechniających narzędzie) wokół Twine wytworzyła się swego rodzaju społeczność.

⁷ Warto uprzedzić, że spora część dostępnych materiałów dotyczy wcześniejszych wersji Twine i nie wszystkie rozwiązania dają się obecnie zastosować.

teksty grywalne, takie jak choćby „analogowe” *gamebooks* (powieści paragrafowe), w których upatruje się źródeł IF⁸. Wprowadzając tego typu formy jako przedmiot kursu, staram się uwrażliwić studentów na fakt, iż nielinearność tekstu nie jest warunkowana jego nowo-medialnym kształtem. Stąd sporo mówimy o tradycji takich form — różnych postaciach pisarstwa symultanicznego (od technik stosowanych przez Filippo Tomasso Marinettiego czy Jana Brzękowskiego po współczesne powieści, jak *Zapiski ze złego roku* Johna Maxwella Coetzee’ego czy *Ludzie z papieru* Salvadora Plascenci), o *shuffle literature*⁹ (w rodzaju *Heart Suit* Roberta Coovera, *Composition no. 1* Marka Saporty czy *Nieszczęsnych* B.S. Johnsona, w kontekście tradycji utworów takich jak choćby Cortazarowska *Gra w klasy*), a także *fiction of scraps*¹⁰ (S. Douga Dorsta i J.J. Abramsa czy *Pamiętnika Cathy* Seana Stewarda i Jordana Weismana) czy, wspomnianych już, powieściach paragrafowych (jak klasyczne *Choose Your Own Adventure* czy książki takie jak *Pajęcza sieć* Robina Waterfielda i Wilfrieda Daviesa, analizowane również w ich uwikłaniu w eksperymenty OULIPO spod znaku *Drzewka teatralnego* Paula Fournela i Jean-Pierre’a Énarada lub *Bajeczki na twoją modłę* Raymonda Queneau¹¹). Dopiero na tym tle rozpatrujemy nowomedialne formy nielinearne, wychodząc od klasycznych hipertekstów i przyglądając się też utworom stworzonym konkretnie w Twine, np. *Depression Quest*¹².

W takim zaplanowaniu kursu kluczowe jest dla mnie to, by studenci skupili się nie na niekonwencjonalnej (nielinearnej) formie utworów, lecz na tym, z czego ona wynika i co daje. W każdym przypadku zadajemy pytania o to, czy daną historię dałoby się opowiedzieć bardziej klasycznie, testując w ten sposób związek — upraszczając — formy z treścią. Moim celem jest to, by na dalszym etapie prac, kiedy studenci sami projektują swoje nielinearne opowieści, owo pytanie o zasadność i celowość użycia nielinearnej struktury towarzyszyło im nieustannie. Bywało, że jeszcze w czasie prac konceptualnych nad konkretnym projektem lub w czasie jego prezentacji wspólnie dochodziliśmy do wniosku, że dana historia nie mieści się zbyt w nielinearnej formule, że w innym kształcie wybrzmiałaby lepiej. Kładę tak duży nacisk na ten etap prac, gdyż uważam umiejętność doboru adekwatnych do przekazu, najlepszych dla utworu środków formalnych (w tym i struktury całej opowieści) za jedną z najważniejszych, jakie powinniśmy kształtować.

⁸ Szerzej o tradycji i kontekstach literatury grywalnej (w jej szerokim, transmedialnym rozumieniu) pisałam w szkicu *Ku literaturze grywalnej (kilka uwag wstępnych)* (Przybyszewska 2014).

⁹ Kategoria *literatury tasowanej*, zaproponowana przez Zuzannę Husárovą i Montforta (zob. Husárová i Montfort 2012), niezwykle obrazowo opisuje teksty, które nie są oddane czytelnikowi w formie zwartej, lecz fragmentów, które odbiorca winien potasować, by „rozdać” historię. *Heart Suit*, trzynastcie kart w kolorze kier, na których odwrotnie wydrukowano fragmenty tekstu (zaś kolejność lektury — poza pierwszą i ostatnią kartą — zależy od czytelnika i — jak wyjaśnia Coover — wynika właśnie z przetasowania i rozdania (*shuffle*) kart) to chyba wręcz idealne ucieleśnienie tej idei.

¹⁰ Pożyczam tu termin, jaki wprowadza Julie Armstrong w swoim *Experimental Fiction. An Introduction for Readers and Writers*, definiując jako taką wszelkie działania artystów, które „przelamują granice tradycyjnej formy narracji poprzez użycie innych mediów i tym samym przelamaniem fizyczności książki” (Armstrong 2014: 181). Traktuję jednak określenie bardziej dosłownie, opisując nim ten rodzaj powieści (drukowanych i elektronicznych), w którym owe różnomedialne dodatki „wysypują się” z klasycznego tekstu, dopełniając go jako „skrawki” właśnie, nie klasycznie zintegrowana z wywodem część utworu.

¹¹ Kontekst działań OULIPO warto aktywować i wcześniej, przypominając choćby Oulipiańskie *Cartes Noires* jako „tasowalny” (*shuffle*) sposób rozgrywania opowieści. Więcej o tej typowo twórczo-pisaniowej, warsztatowej literackiej grze talią 36 symbolicznych kart zob. *Oulipo Compendium* 2005: 261.

¹² Szerzej pisałam o tym utworze w szkicu *Wprowadzenie do literatury (roz)grywalnej* (Przybyszewska 2015b).

Kolejnym krokiem, po analizie i interpretacji przykładowych dzieł, staje się na prowadzonych przeze mnie kursach „laboratoryjna” przygoda z tworzeniem tekstu własnego. W przypadku literatury nieliniowej punktem wyjścia jest znalezienie historii, która się w takiej pokawałkowanej formie dobrze odnajdzie, konceptualne dopracowanie szczegółów rozbicia ciągłości. Na tym etapie zaczyna się warsztatowa praca nad tekstem (lub jego fragmentami¹³) i tu właśnie za pożądane uznałam wykorzystanie narzędzi pozwalających realnie wprowadzić zamiar w czyn i sięgnięcie po platformę Twine. Założone przeze mnie sprzężenie między tym „po co” i „jak”, powiązanie kształtowania kompetencji pisarskich dotyczących kreowania historii, świata, bohaterów z tymi, które dotyczą już *stricte* medialnej formy opowieści wyraźnie widać również w podręczniku Ford. I ona w równym stopniu uczy pracy nad tym, jak zanurzyć czytelnika w świat (poprzez odpowiednie językowe ukształtowanie opowieści, w tym metaforykę, obrazowanie i narrację prowadzoną w drugiej osobie czy właściwą konstrukcję postaci, z którą w przypadku IF czytelnik ma się utożsamić), jak i (zapowiadanych w tytule) kwestii — dosłownie — „technicznych”. Jestem przekonana, że inny tryb pracy nie miałby w ogóle sensu, tak jak nie miałoby go uczenie tworzenia metafor bez zastanawiania się nad ich znaczeniem (skupienie się na samej mechanice procesu).

Dodatkowym impulsem do sięgnięcia po Twine stały się też sygnalizowane przez studentów lub dostrzegane wyłącznie przeze mnie problemy natury czysto technicznej, jak doskwierający brak kontroli nad strukturą całości i poszczególnymi wątkami czy dość losowe wybieranie słów-kotwic. Twine okazał się remedium. Przede wszystkim projektowanie w nim tekstu oznacza spotkanie z mapą: platforma oferuje użytkownikom „tablicę”, na której zminiaturyzowane są wszystkie fragmenty, nad którymi pracujemy (klikając w nie, możemy przejść do ich edycji) oraz, co niezwykle pomocne, połączenia między nimi (w postaci jedno- lub dwukierunkowych strzałek). Taka wizualizacja struktury utworu nie tylko ułatwia organizację tekstu (poszczególne elementy dają się też — przez przeciągnięcie ich po ekranie — łatwo przesunąć, przestawiać), lecz i równoległą pracę nad kilkoma fragmentami — w dowolnym momencie możemy wrócić do każdego kawałka tekstu, śledząc jednocześnie wpływ dokonywanych zmian na cały komunikat.

By stworzyć mapę tekstu (a tym samym działający w oparciu o nią plik HTML, tekst z czytelnie wyróżnionymi aktywnymi hiperłączami), wystarczy opanować zaznaczanie fraz, które chcemy uczynić linkami. Rozpoczynamy pracę z Twine, mając do dyspozycji jedno (startowe) okno na naszej tablicy. I, jak w przypadku każdego kolejnego, możemy je zatytułować (wpisane w rubrykę *title* tytuły, widoczne w trakcie pracy w edytorze jako nagłówki tekstów¹⁴, mają funkcję porządkującą — nie są potem widoczne przy wyświetlaniu tekstu), otagować (takie etykiety, również ukryte później przed odbiorcami, wydają

¹³ Np. w przypadku omawiania powieści paragrafowych zaczynaliśmy od pracy na krótkich fragmentach: każdy miał do przygotowania wprowadzającą scenę zanurzającą oraz jeden, dwa bloki tekstu z wyborami. W niewielkich tekstach łatwiej było nam wylapywać błędy, pracować nad spójnością (zwłaszcza przejściami pomiędzy poszczególnymi fragmentami). Dzięki temu, niezależnie od ograniczeń czasowych, mieliśmy też szansę popracowania nad tekstem każdego z uczestników kursu. Dopiero po tych ćwiczeniach wstępnych studenci przystępowali do prac nad większymi projektami.

¹⁴ W przypadku większych utworów czasem przydatna staje się zmiana trybu wyświetlania mapy tekstu, np. tak, by widoczne były nie i tytuły, i (przynajmniej częściowo) teksty główne, ale właśnie jedynie te pierwsze (w skrajnych przypadkach zaś — wyłącznie struktura tekstu). Umożliwiają to ikony w prawej części dolnego paska zadań.

się niezwykle pomocne przy pracy nad większymi opowieściami, zwłaszcza, gdy mamy kilka paralelnych, ale konkurencyjnych linii fabularnych — to dzięki nim łatwiej pamiętać, że czytelnik, który dotarł do tego fragmentu wie pewne rzeczy, a nie ma prawa wiedzieć innych) i wpisać weń tekst główny, czyli to, co będzie potem wyświetlane na ekranie. Dalejsze kroki to: utworzenie we wpisany tekście linków i przejście do tworzenia kolejnych fragmentów. Na każdym etapie pracy możemy sprawdzić, jak w tym momencie działa nasz tekst (opcja „Test” w prawym dolnym rogu) — w innej karcie otworzy nam się hipertekst, który dotąd stworzyliśmy.

By utworzyć link, wystarczy słowo/a, które ma/mają nim być wziąć w podwójny kwadratowy nawias. Dodatkowym ułatwieniem jest to, że w takiej sytuacji Twine automatycznie utworzy nam kolejne okienka tekstu (więc na mapie od razu pojawiają się kolejne dwa „klocki” oraz połączenia między nimi¹⁵). Jeśli więc w pierwszy blok tekstu wpisujemy słowa: „Tu zaczyna się nasze opowiadanie, które może rozwinąć się w [[jedną]] bądź [[drugą]] stronę”, struktura rozrośnie się do trzech elementów. Nowe bloki automatycznie tytułowane są nazwami linków, zaś klikając na nie, możemy przejść do ich edycji i rozbudowywać projekt. Funkcja „+Passage” (widoczna w prawym dolnym rogu) pozwala również utworzyć nowe niezależne fragmenty, które możemy później, wstawiając linki, połączyć z już istniejącą strukturą. Po zakończonej pracy (lub w jej trakcie) możemy, dzięki opcji „Publish to File” (znajdującej się w menu ukrytym pod strzałką przy nazwie naszego pliku w lewym dolnym rogu ekranu), zapisać plik w formacie HTML, co pozwoli zarówno go dystrybuować (np. przesłać mailem, umieścić na stronie), jak i utworzyć ponownie w programie Twine, by kontynuować pracę¹⁶.

W Twine możemy również zaznaczyć, jakie słowo ma być linkiem, ukrywając nazwę fragmentu, do którego ma odnosić — umożliwia to użycie znaku „|” i wpisanie linku w formie „[[to, co ma być widoczne dla czytelnika|nazwa fragmentu, do którego odnosimy]]”. Wyobraźmy sobie, że pierwsza i druga linia narracji naszego modelowego hipertekstu rozwinęła się dalej, jednak w pewnym momencie chcielibyśmy dać czytelnikowi możliwość przejścia do drugiej z nich. Jeśli umieścilibyśmy w narracji zdanie: „Tak toczy się nasza historia, lecz może chciałbyś wiedzieć, jak by wyglądała, gdybyś wtedy wybrał inną opcję” trudno byłoby w opisany wcześniej sposób przenieść z niego do fragmentu, który zatytułowany jest przecież „drugą”. Oczywiście, w tym akurat wypadku spokojnie moglibyśmy podmienić słowo „inną” na „drugą”, wziąć je w podwójny kwadratowy nawias i problem rozwiązałby się sam przez się, jednak nie w każdej sytuacji dałoby się to zrobić. Wystarczy jednak zastosować wspomnianą tu formułę, tak by wpisane w okienko zdanie miało formę: „Tak toczy się nasza historia, lecz może chciałbyś wiedzieć, jak by wyglądała, gdybyś wtedy wybrał [[inną|drugą]] opcję”.

Opisane tu podstawowe funkcje programu Twine w zupełności wystarczają do zaprojektowania prostego hipertekstu, co potwierdzają też opinie studentów, z którymi pracowałam. Podkreślali oni łatwość obsługi platformy, fakt, że usprawnia pracę nad tekstem, zaś wizualizacja mapy tekstu była jedną z najczęściej podkreślanych zalet. Dodalabym jeszcze to, że program obsługuje polskie czcionki. Tylko raz zdarzyła się sytuacja, gdy studentka

¹⁵ Jeśli stworzymy potem link, który będzie wracał do wcześniejszego bloku tekstu — program automatycznie zaznaczy również dwukierunkowe połączenie między fragmentami.

¹⁶ Gdy ściągniemy Twine na własny komputer, możemy zapisywać historie również bezpośrednio w programie.

przygotowała pracę zaliczeniową nie w Twine, lecz korzystając z opcji hiperłączy w Wordzie. Co ciekawe — wtedy nie tyle ja, co inni uczestnicy kursu stanowczo przekonywali ją, że „utrudniła sobie zadanie”, „dodała pracy”¹⁷. Jednocześnie zaobserwowałam, że po takim doświadczeniu studenci bardziej otwarcie podchodzą do tekstów elektronicznych, wykazując zarazem krytyczną postawę w stosunku do użytych w nich technologicznych środków.

W trakcie prowadzonych kursów dwukrotnie też „zaeksperymentowałam”, odchodząc od przedstawionych tu założeń, co upewniło mnie w przekonaniu, że charakteryzowany sposób i zakres wprowadzania technologii w obręb zajęć o charakterze literaturoznawczym i warsztatowym zarazem dobrze się sprawdza. Grupie studentów II stopnia, którym spodobała się praca z Twine, zaproponowałam, byśmy spróbowali skorzystać z bardziej zaawansowanych funkcji programu¹⁸. Okazało się jednak, że o ile jeszcze włączanie w obręb utworu obrazów, plików muzycznych czy filmowych nie sprawiło nam specjalnej trudności, już korzystanie z linków warunkowych (wyjściowo bardzo interesujące dla studentów) wymagało od nich zdecydowanie większego przestawienia się na myślenie „po informatycznemu” i nie było tak proste. Mimo początkowego entuzjazmu zdecydowanie mniej osób miało ochotę później próbować rzeczywiście korzystać z tych funkcji¹⁹, a tego typu rozwiązania nie pojawiły się w pracach zaliczeniowych. Ogólny wniosek grupy był taki, że wiedza, którą mieli dotąd, pozwalała im przetestować to, co chcieli. „Wyższy stopień wtajemniczenia” niektórych wręcz odciągał od myślenia o literackiej warstwie tekstu, przeszkadzał. Ten jednokrotny eksperyment potwierdził moje przekonanie, że artysta, gdy potrzebuje silniejszego wsparcia technologicznego, może zwrócić się do kogoś z zewnątrz, zaś dla jego własnego rozwoju twórczego ważniejsza niż specjalistyczna wiedza programistyczna wydaje się świadomość, że współpraca w takim zakresie to nic złego, otwarcie na dyskusję ze specjalistami, ewentualnie bycie przygotowanym do pogłębiania wiedzy we własnym zakresie.

Z kolei moją wiarę w to, że wprowadzanie elementów „warsztatu technologicznego” ma sens dopiero wtedy, gdy poprzedzi je namysł nad istotą formy i analiza przykładów (słowem — gdy studenci widzą literackie uzasadnienie dla takiego „obcego” wtrętu), potwierdziło się, kiedy z jedną z grup na studiach I stopnia próbowałam „ugryźć” zagadnienie od drugiej strony i zaczęliśmy od wyjaśnienia, jak działa Twine i prób pisanie tekstów nieliniarnych²⁰. Nigdy na moich zajęciach nie powstały tak kiepskie prace, nigdy też studenci z takim oporem nie podchodzili do prezentowanego im narzędzia (najbardziej z całej grupy otwarty na e-literackość student wycofał się nawet z prób stworzenia tekstu

¹⁷ Uwagi te padły w trakcie kursu *Kreacja przekazu multimedialnego* w roku akademickim 2015/2016. Studenci próbowali też, słusznie jak sądzę, przekonać koleżankę, że jeśli już sięgać po inny program, to w tym wypadku raczej PowerPointa niż Worda.

¹⁸ Twine, czego nie wyjaśniłam na początku, zdaje się mieć podwójnego adresata: inaczej będzie w nim zapewne pracował informatyczny laik, ktoś, kto nie miał wcześniej styczności z programowaniem (takim użytkownikiem są raczej w większości nasi studenci), inaczej ktoś, kto zna jego podstawy. Ten drugi dzięki specyficznej „nakładce” łatwiej skonstruuje hipertekst i będzie mu na pewno wygodniej śledzić jego strukturę, ale do zapanowania nad nią z łatwością użyje też — poznanych wcześniej — bardziej skomplikowanych instrukcji i komend (konkretnych makr).

¹⁹ Co wiąże się również z faktem, że ciągle rozwijany Twine nie jest wcale platformą doskonałą.

²⁰ Taką formułę („sprawdź w działaniu”) mają też prowadzone w duchu *kreatywnego programowania* warsztaty Montforta (np. w Krakowie w 2014 roku czy Victorii w 2016).

w Twine, uznając to za „zbyt trudne” i „bezsensowne”). Odniosłam wrażenie, że studenci zupełnie nie widzą sensu w wykonywanych zadaniach, co niestety przełożyło się później na niechęć do tekstów, które analizowaliśmy (choć studenci wiedzieli, jak zrobić takie teksty, niezbyt potrafili odpowiedzieć na pytanie o to, czemu służyć ma w konkretnych wypadkach zastosowana forma, jak wiąże się z opowieścią). Niechęć do testowanej „technologii” przełożyła się na niechęć do korzystającej z niej twórczości w ogóle. Niezwykle trudno było ich przekonać do tego, że nieklasyczna forma omawianych tekstów może być czymś więcej niż zabawą, że może wiązać się z semantyką utworu (nie mówiąc już o tym, że i oni jako twórcy mogliby w taki właśnie sposób do technologii podejść i stawiać pytania o celowość jej wykorzystania, szukać historii, które wybrzmiały w nieliniarnej strukturze). Poczucie, że im nie udało się stworzyć niczego sensownego przy użyciu Twine, przerodziło się w — niemal bezdyskusyjne — przeświadczenie, że forma nieliniarna w ogóle jest „bez sensu”. Skończyłam kurs przekonana, że przy kolejnej okazji będzie mnie czekała ciężka praca nad tym, by w ogóle otworzyć grupę na eksperymenty z technologią.

Moje dotychczasowe doświadczenia dydaktyczne dowodzą, że wpisujące się w ideę *kreatywnego programowania* oraz uniwersytetu jako *laboratorium* wprowadzanie elementów „warsztatu technologicznego” w kontekście dydaktyki form e-literackich (i innych eksperymentujących z medium sztuki słowa) jest dobrym posunięciem, zwłaszcza na kierunku *creative writing*. Nie tylko pozwala studentom lepiej zrozumieć omawiane formy i zmiany w literaturze, ale też lepiej przygotowuje ich do wejścia na rynek pracy, pogłębiając kompetencje medialne (użyteczne również w innych niż ściśle twórczym wymiarze) oraz przygotowując do kompetentnej rozmowy ze specjalistami z innych dziedzin, gdyby np. zdecydowali się na współpracę, która pozwoli im stworzyć utwór nie tylko o wysokim poziomie literackim, ale i technologicznym (do idei takich duetów twórczych i roli studiów *twórczego pisanie* w przygotowywaniu absolwentów do takiej współpracy wróć jeszcze w zakończeniu szkicu). Jestem przekonana, że do osiągnięcia takich celów nie jest konieczne zagłębianie się w niuanse informatyczne, zaś opanowanie podstawowych funkcjonalności platformy takiej jak Twine nie przekracza kompetencji przeciętnego wykładowcy w XXI wieku.

Jednak kluczowe jest zachowanie równowagi między tym, co w procesie nauczania literackie (i dla studentów już „oswojone”) a (obcą im często i stanowiącą barierę) technologią, a także uwzględniająca predyspozycje studentów kolejność wprowadzania zagadnień. Choć w wielu wypadkach — tu zgodzę się z Montfortem — eksperymentowanie z programem, metoda prób i błędów, pozwala przestać się go bać, odkryć wiele jego funkcjonalności i lepiej go zrozumieć — dla nieznającego zupełnie zasad programowania studenta, który marzy, by zostać pisarzem (raczej niekoniecznie e-literatem) może to być za dużo²¹. Stąd rozsądniej wydaje się zacząć od pokazania, po co ma się przelamywać, że technologia może być nie tylko ciekawą zabawą samą w sobie, ale i interesującym narzędziem do osiągania celów artystycznych na polu literackim.

²¹ Kathi Inman Berens eksperymentująca w ten sposób z kodem *Wąwozu Taroko* (do którego zaraz wrócę) to zupełnie inny przypadek — jako badacz literatury elektronicznej wiedziała, po co eksperymentuje i w punkcie wyjścia była taką „zabawą” zainteresowana (por. Górską-Olesińską 2016). Student pragnący zostać pisarzem, jeśli nie wie, że ta zabawa może mieć coś wspólnego z literaturą — w sumie — nie musi.

Mało to niekoniecznie za mało

Czy Twine wykorzystywany na potrzeby uczenia o formach literatury nielinarnej to jedy-ny przypadek, jaki można by tu poddać analizie? Twórcze programowanie jako swoista od-miana *creative writing* przywodzi na myśl jeszcze co najmniej jeden przykład, zwłaszcza gdy wspomina się Montforta: *Wąwóz Taroko*. Przypomnę, że Scott Rettberg swoim *Garażem w Tokyo* rozpoczął serię „przeróbek” tego wyjściowego prostego utworu generatywnego, testującej zarówno ideę pisania wspólnotowego, jak i pisarstwa *open source*. Fenomen ten został już w Polsce szeroko opisany (zob. Marecki 2015; Górska-Olesińska 2016), zaś *Wą-wóz...* i wybrane jego „wariacje” spolszczone²² oraz opublikowane wraz z komentarzami i tłumaczeniami tekstów Montforta i Rettberga jako „blok specjalny” na lamach „Tech-stów” (*Wąwóz Taroko — blok specjalny* 2016: b.s.), stąd nie chcę poświęcać mu zbyt wiele uwagi. Na potrzeby domknięcia prowadzonych tu rozważań warto jednak wypunktować dwie kwestie. Po pierwsze *Wąwóz...* (a przede wszystkim to, co się z nim dzieje dalej) sta-nowi doskonały przykład sytuacji, w której dotarcie do kodu programu staje się istotnym elementem interpretacji (wręcz: użycia) tekstu. Jeśli rozpatrywać by wyjściowy generator Montforta jako tekst samodzielny, odosobniony (a jako taki był pierwotnie pomyślany), tego typu zagłębienie „pod podszewkę” nie jest konieczne, jednak „gest Rettberga” (po-dejrzenie kodu źródłowego, jawne „pożyczenie” i ingerencja weń) sankcjonuje sytuację, w której pod ową podszewkę należy nie tylko zagłębować, ale i pod nią „majstrować”. Jedno-cześnie, tak jak — dosłownie — przekreśla nazwisko Montforta jako autora utworu (przy czym owo przekreślenie multiplikowane jest w przypadku kolejnych autorskich przeróbek), tak podważa też sensowność rozpatrywania utworu w oderwaniu od (żywej, ciągle się rozrastającej) siatki współ-tekstów, ucieleśnionych (dzięki olinkowanym skreśleniom) jako jego część w każdej kolejnej realizacji. Diagnozę taką potwierdza włączenie do trze-ciego tomu kolekcji *Electronic Literature Collection* (mającej za zadanie ustanowienie kanonu e-literatury) nie wyłącznie generatora Montforta, lecz również wszystkich jego przeróbek.

Bezspornie też trudno bez „grzebania” w tym tekście w ogóle o nim rozmawiać, co potwierdzają nie tylko wypowiedzi wielu „współautorów”, dla których *Wąwóz...* stał się też tematem prowadzonych kursów (jak Kathi Inman Berens), ale i moje własne doświadczenia dydaktyczne. Studenci po lekturze bloku specjalnego poświęconego feno-menowi tego tekstu w „Techstach” (a więc nie tylko generatorów, lecz i komentarzy do nich) niespecjalnie rozumieli ideę tekstu, nie potrafili jej zreferować. Sytuację zmieniło dopiero uzmysłowienie im, co realnie znaczy zajrzenie do kodu źródłowego: nie wyjaśnienie, że to nic skomplikowanego, ale zrobienie tego z nimi — pokazanie właśnie od strony „kuchni”, co tak naprawdę stało się w akcie twórczym. Istota tekstu wybrzmiała dla nich dopiero w tym momencie, zaś próba samodzielnego podmienienia bazy tek-stowej, „zabawa” z kodem pozwoliła też przełamać barierę strachu przed nieznanym, która — w czym nic dziwnego — dla wielu czytelników rozpoczynających dopiero swoją przygodę z e-literaturą stanowi przeszkodę nie do pokonania (bo jak czytać tekst, za który nie wiadomo jak się w ogóle zabrać?). Zatem w tej sytuacji postawa „czytelnika w laboratorium”, łącząca (zgodnie z wieloma charakterystykami ewolucji kategorii autor-stwa w przypadku każdego rodzaju sztuki otwartej na uczestnictwo) *przyjemność czytania*

²² Posługuję się tu dla uproszczenia terminem „spolszczenie” a nie „tłumaczenie”, gdyż kwestia przekładu tego typu tekstów to temat na niejedno pewnie oddzielne opracowanie.

z *przyjemnością pisania*, istotnie pozwoliła uczestnikom kursu zrozumieć nie tylko sens konkretnego utworu (w jego uwikłaniu w relacje intertekstualne), lecz i — w szerszym wymiarze — istotę sztuki generatywnej. Owo lekturowo-warsztatowe „doświadczenie” (nie sięgające bynajmniej informatycznych wyżyn, osiągalne dla każdego laika) w połączeniu z wiedzą o podobnych wcześniejszych eksperymentach (choćby dadaistycznym wyciąganiu słów z kapelusza) dało studentom szerszą panoramę zjawiska.

Pokierowanie w ten sposób procesem dydaktycznym nie wymagało ode mnie specjalistycznej wiedzy z zakresu programowania (tak, jak nadpisywanie kolejnych tekstów generatywnych nad *Wzwożem*... bądź stworzenie własnej opowieści nielinarnej w Twine nie wymagało jej od studentów). W obu tych przypadkach „liźnięcie” technologii pozwoliło lepiej zrozumieć istotę tekstu, dostrzec, jakie możliwości daje artyście warstwa technologiczna (kod bądź inne jej poziomy). Zasiadając do lektury wiersza, mamy zwykle świadomość nie tylko jakimi figurami artysta się posłużył, lecz z jakich nie skorzystał. Wydaje się, że zajrzenie do „laboratorium” e-tekstu pozwala czytelnikowi zdobyć podobne kompetencje w kontekście literatury elektronicznej, co ułatwia proces interpretacji. Mówiąc konkretniej: jeśli wiemy, że artysta miał możliwość skorzystania z linków warunkowych, ale tego nie zrobił — to nieco inna sytuacja niż kiedy mamy świadomość, że nie mógł po takie rozwiązanie sięgnąć. Patrząc na ten problem od strony procesu twórczego: jeśli myślę, że mój tekst takiego typu linku będzie potrzebował — sięgam po oprogramowanie, które umożliwi mi skorzystanie z tego chwytu (dlatego wiedza, że ono istnieje, jest użyteczna dla artysty).

Inną kwestią pozostaje to, czy opanowanie tak podstawowego poziomu warsztatu technologicznego jest dla przyszłego artysty rzeczywiście wystarczające. Z jednej strony nie miałabym nic przeciwko temu, by nasi studenci mieli możliwość uczestniczenia w zajęciach prowadzonych zgodnie z programem opracowanym przez Montforta i — jak marzył w kontekście zakładanego laboratorium Marecki²³ — jako główny podręcznik („biblię”) traktowali choćby przywołaną tu książkę szefa Top Trank Laboratory, by pogłębiali swoją wiedzę o programowaniu. Z drugiej jednak strony jestem przekonana, że — dając im podstawy — zachęcamy ich do dalszych poszukiwań tam, gdzie rzeczywiście będą mieli taką potrzebę. Dla przyjętej przeze mnie perspektywy niezwykle istotne jest również to, że odnoszę się do studiów twórczego pisania rozumianych jako „totalne”, nie zaś skoncentrowanych wyłącznie na nowomediálním pisarstwie. W tym wypadku — podobnie jak w odniesieniu do dowolnych studiów o programie literaturoznawczo zorientowanym — tego typu twórczość staje się jedynie jednym z wielu punktów programu. Nic więc dziwnego w tym, że stopień zanurzenia w technologię będzie tu znacznie mniejszy niż w przypadku kierunków takich jak *Writing For Digital Media*, na którym wykłada Robert Coover czy John Cayley. Ale może tych, których zafascynuje temat, otworzy na marzenia o studiach na Brown University.

Pożądane byłoby na pewno, by zainteresowani tematyką studenci mieli możliwość dalszego rozwijania kompetencji technologicznych — choćby w formie warsztatów, dalszych etapów edukacji czy zwykłej (może z czasem i jako programowo ujętej) współpracy ze wspomnianym laboratorium kierowanym przez Mareckiego²⁴. W przypadku niektórych

²³ Powołuję się na rozmowę przeprowadzoną w czerwcu 2016 r.

²⁴ Zajęcia akademickie są również okazją do wskazywania dalszej bibliografii. W tym wypadku użyteczne może się okazać sygnalizowanie niezwykle częstych w przypadku e-literatury sytuacji, w której autorzy analizowanych tekstów są jednocześnie badaczami, praktycy — teoretykami i nauczycielami. Tym samym,

form e-literackich (np. tych wykorzystujących technologie rzeczywistości wirtualnej), istotnie, właściwie wyłącznie odpowiednio wyposażone technologicznie laboratorium wydaje się miejscem, gdzie można by tego typu zajęcia prowadzić. Jednak, jak przekonują przykłady, które nierzadko stają się podstawą prowadzonych w trakcie kursów dyskusji — mało który artysta decyduje się na całkowitą technologiczną samodzielność²⁵. Za sukcesami artystyczno-technologicznych duetów (typu Amaranth Borsuk — Brad Bouse czy, na rodzimym gruncie, Marta Dzido — Mariusz Pisarski) stoi też świadomość tego, że siła tekstu e-literackiego rodzi się na przecięciu jego wartości literackiej (artystycznej) i technologicznej, że kluczem jest tu współpraca, współbrzmienie tych (jak byłoby idealnie) semantycznie powiązanych ze sobą wymiarów utworu²⁶. Dając studentom („jedynie”) podstawy — a więc pokazując im, w istocie, jak myśleć technologicznie i podkreślając, jak często literaci współpracują z informatykami/grafikami/dizajnerami — ośmielamy ich do wchodzenia w takie układy twórcze. Przygotowanie studentów do tego, by w dialogu z tymi, którzy mogą zagwarantować ich artystycznym projektom najwyższy poziom technologicznego opracowania, nie czuli się niekompetentni, by mieli dobry punkt wyjścia do negocjowania, co jest ich dziełu potrzebne widziałabym jako jeden z celów studiów *creative writing*. I sądzę, że można go osiągnąć m.in. przy pomocy opisywanych tu metod i narzędzi. Zaś tego typu autorska „kolaboracja” wpisuje się już w ideę humanistycznego laboratorium, o jakim pisała Pawlicka.

omawiając konkretny utwór danego artysty warto dodać, gdzie studenci mogą doczytać o jego teoretycznym podejściu do danego zagadnienia (przykład Serge’a Bouchardona, który nie tylko tworzy teksty wykorzystujące figury gestycznej manipulacji, ale i o nich teoretyzuje) a nawet „zaocznie” zostać jego studentami (*casus* Anastasii Salter jako autorki tekstów artystycznych, teoretycznych opracowań oraz — wraz z Johnem Murrayem, z którym współtworzyła też artystyczne projekty — podręcznika programowania we Flashu).

²⁵ Choć, oczywiście, można też przywołać duety w stylu Tender Claws (tj. Samantha Gorman — Danny Cannizzaro), gdzie obydwie artystów czuwa nad każdym z poziomów tekstu. Warto chyba dodać, że są absolwentami kierunku *Writing for Digital Media* na Brown University.

²⁶ Taki wymiar docierania się sensów i dopracowywania ostatecznego kształtu pracy w dialogu uwzględniającym zarówno literackie, jak i technologiczne możliwości utworu bywał wspominany przez Borsuk i Bouse’a, gdy charakteryzowali swoją współpracę nad *Between Page and Screen*.

Bibliografia

- Armstrong Julie (2014), *Experimental Fiction. An Introduction for Readers and Writers*, Bloomsbury, London–New York.
- Domańska Ewa (2010), *Jakiej metodologii potrzebuje współczesna humanistyka?*, „Teksty Drugie”, nr 1–2.
- Engberg Maria (2014), *Electronic Literature Pedagogies* [w:] *Electronic Literature as a Model of Creativity and Innovation in Practice. A Report from the HERA Joint Research Project*, eds. S. Rettberg, S. Baldwin, West Virginia UP, Morgantown, http://dspace.uib.no/bitstream/handle/1956/8939/rettberg_baldwin_elmcip.pdf?sequence=3&isAllowed=y [dostęp: 30.05.2016].
- Ford Meliss (2016), *Writing Interactive Fiction with Twine. Play Inside a Story*, QUE Publishing.
- Górska-Olesińska Monika (2016), *Kolektywne „prze-pisywanie” e-literatury* [w:] *Literatura prze-pisana II. Od zapomnianych teorii do kryminału*, red. A. Izdebska, A. Przybyszewska, D. Szajnert, WUŁ, Łódź.
- Grigar Dene (2008), *Electronic Literature: Where is it?*, „Electronic Book Review” 28.12, <http://www.electronicbookreview.com/thread/technocapitalism/invigorating> [dostęp: 30.05.2016].
- Husárová Zuzana, Montfort Nick (2012), *Shuffle Literature and the Hand of Fate*, <http://www.electronicbookreview.com/thread/electropoetics/shuffled> [dostęp: 30.05.16].
- Jenkins Henry, Purushotma Ravi, Weigel Margaret, Clinton Katie, Robison Alice J. (2009), *Confronting the Challenges of Participatory Culture: Media Education in 21st Century*, MIT Press, Cambridge–London.
- Marecki Piotr (2015), „*Taroko George*” Nick Montfort [w:] *Klasyczne dzieła sztuki nowych mediów*, red. P. Zawojski, Instytucja Kultury Katowice — Miasto Ogródów, Katowice.
- Montfort Nick (2005) *Twisty Little Passages: An Approach to Interactive Fiction*, MIT Press, Cambridge MA.
- (2016), *Exploratory Programming for the Arts and Humanities*, MIT Press, Cambridge MA.
- Montfort Nick, Marecki Piotr (2015), *Materiałność twórczego programowania. Z Nickiem Montfortem o badaniach platform, współczesnych game studies oraz interaktywnej fikcji rozmawia Piotr Marecki*, „Czas Kultury”, nr 2.
- Oulipo Compendium* (2005), eds. H. Mathews, A. Brotchie, Atlas Press — Make Now Press, London–Los Angeles.
- Pawlicka Urszula (2014), *Nauczanie literatury cyfrowej. Trochę wątpliwości i o wiele więcej zysków* [w:] *Polonistyka dziś — kształcenie dla jutra*, red. K. Biedrzycki, W. Bobiński, A. Janus-Sitarz, R. Przybylska, vol. 2, Universitas, Kraków.
- (2015), *To nie jest muzeum, to jest laboratorium*, „Czas Kultury”, nr 2.
- Przybyszewska Agnieszka (2014), *Ku literaturze grywalnej (kilka uwag wstępnych)*, „Przegląd Kulturoznawczy”, nr 2.
- (2015a), *Medialne uwikłania literatury a twórcze pisanie* [w:] *Edukacja medialna w dobie współczesnych zmian kulturowych, społecznych i technologicznych*, red. A. Ogonowska, G. Ptaszek, Impuls, Kraków.
- (2015b), *Wprowadzenie do literatury (roz)grywalnej*, „Poradnik bibliotekarza”, nr 3.
- Simanowski Roberto (2010), *Teaching Digital Literature. Didactic and Institutional Aspects* [w:] *Reading Moving Letters*, eds. R. Simanowski, J. Schäfer, P. Gendolla, Transcript, Bielefeld.
- Simanowski Roberto, Schäfer Jörgen, Gendolla Peter (2010), *Reading Moving Letters. Digital literature in Research and Teaching. A Handbook*, Transcript, Bielefeld.
- Wąwóz Taroko — blok specjalny* (2016), „Techsty”, nr 1, www.techsty.art.pl.