

“The History of Software Art:
Several Beginnings, Several Histories”

Abstract

This paper introduces an artistic movement known as software art that was born almost twenty years ago. The art movement doesn't just use software as a mere tool for creating art but also works with software itself as an artistic material and fully-fledged means of expression. This paper describes the complex history of the art movement and its various historical roots. Although at first glance, it may seem that software art is a completely new art praxis, we can find several historical predecessors of software art in the world of art. The first art movement that deeply influenced software art was conceptual art. Conceptual art shares several fundamental characteristics with software art such as the shift from the physical object to an abstract idea or the creation of instructions for the realization of artworks. Software art is also inspired by destructive tendencies in art (for example Dadaism, destructive performances, Body Art) because the destruction is one of its most important strategies, realized through the performance of computer viruses. However, the historical roots of software art don't come exclusively from widely recognized art movements but can also be identified in some artworks of early digital art. In the last section of the paper, we describe a revisionist approach to the writing of software art history. In this case, software art is not put into art history as a continuation of the coherent story of art but takes on the role of active participant in the creation of our present. Software art is seen as a new opportunity for the critical reflection of contemporary art and society which are deeply influenced by ubiquitous software.

Klíčová slova

softwarové umění – konceptuální umění – digitální umění – software – počítačový virus – destrukce – instrukce

Keywords

software art – conceptual art – digital art – software – computer virus – destruction – instructions

Autor je doktorandem na Katedře výtvarné výchovy PdF MU v Brně.

416271@mail.muni.cz

DĚJINY SOFTWAREVÉHO UMĚNÍ. NĚKOLIK HISTORIÍ, NĚKOLIK POČÁTKŮ ADAM FRANC

Softwarové umění, až na několik výjimek, nebylo dosud v českém prostředí dostatečně reflektováno,¹ přestože tvoří významnou součást širokého souboru uměleckých přístupů, které konstituují členité pole digitálního umění.² Tyto umělecké přístupy (jako například net art, game art, generativní umění nebo digitální fotografie) můžeme nahlížet jako podmnožinu kategorie digitálního umění. Protože digitální umění je charakteristické tím, že je tvořeno

1

Srov. Jana HORÁKOVÁ, „Softwarové umění: programování excessu“, *Musicologica Brunensia*, roč. 47, 2012, č. 2, s. 23–47; Monika SZÚCSOVÁ, *Softwarové umění: genealogie, historie, jazyk* (diplomová práce), Brno: Masarykova univerzita, Filozofická fakulta 2015.

2

Termín digitální umění, který vzniká v devadesátých letech minulého století, označuje širokou skupinu uměleckých děl, která byla vytvořena za pomoci digitálních technologií. Tento termín tedy odkazuje k jejich specifickému způsobu fungování (metoda digitálního zpracování dat). Můžeme do něj zahrnout imateriální díla, která pracují se softwarem nebo kódem, ale také interaktivní instalace a performativní díla využívající digitální technologie. Digitální umění dále můžeme rozdělit do dvou odlišných kategorií. První kategorie digitálního umění využívá digitální technologie jako nástroje k vytvoření uměleckých děl, aniž by byla blíže reflektována povaha těchto nástrojů. V tomto případě se jedná převážně o díla, která napodobují kvality jiných médií (například digitální fotografie). Druhá kategorie se naopak zaměřuje na průzkum specifických estetických kvalit digitálních médií, které tvoří základ výsledného díla (například generativní umění, net art). Pokud jde o termín umění nových médií, který je od devadesátých let frekventovaně využíván pro označení uměleckých děl, jež pracují s novými médii (video, internet, počítač, virtuální realita, 3d tisk, robotika, biotechnologie apod.), nahlížím jej jako historicky ukotvený pojem, jehož vznik byl podmíněn všeobecným nadšením umělců a teoretiků z nových možností, které přinesly digitální technologie do umělecké tvorby, a potřebou tyto možnosti důkladně prozkoumat. Termín umění nových médií byl podroben kritice za to, že nerozlišuje mezi analogovými a digitálními médii (patří do něj například i videoart) nebo uměleckými díly. Navíc samotný termín nová média je velmi problematický, neboť nelze přesně definovat, co je (stále ještě) novým médiem, a co už je médiem starým, a také proto, že idea novosti obsažená v tomto označení s sebou nese jistou ideologickou zátěž (např. moderní vize technologického pokroku, který povede k vytvoření dokonalejší společnosti, apod.). Více viz Jakub MACEK, „Poznámky k okouzlení novostí nových médií“, *Mediální studia*, roč. 6, 2012, č. 1, s. 6–16. Vzhledem k tomu, že se nyní nacházíme v situaci, kdy se digitální technologie staly banální součástí naší každodennosti a jsou předmětem strážlivější kritické reflexe oproštěné od jednostranně negativních, nebo naopak jednostranně pozitivních náhledů na digitální technologie, či od revolučních vizí, které doprovázely jejich reflexi v devadesátých letech, jeví se pojem digitální umění jako vhodnější pro svůj neutrálnější charakter. Teoretici umění Cat Hope a John Charles Ryan navrhuji používat termín digitální umění v množném čísle (*digital arts*), který odkazuje k širokému souboru digitálních praxí provozovaných umělci a zdůrazňuje diverzitu uměleckých forem a médií, jež můžeme zahrnout do tohoto označení. Viz Cat HOPE – John Charles RYAN, *Digital Arts. An Introduction to New Media*, Londýn: Bloomsbury 2014.

neustále se proměňujícím souborem technik, technologií a konceptů, dochází také k častým proměnám definic či pojmenování stávajících podskupin digitálního umění a k neustálému přibývání nových termínů. Význam těchto označení tedy do značné míry závisí na disciplíně, z níž vychází teoretik či umělec, který je autorem pojmenování, na technologii, kterou využívá, nebo na specifickém časovém rámci, v němž daný umělecký přístup vzniká.

Softwarové umění patří do skupiny uměleckých přístupů, jejichž definice vychází z konkrétního média, které je využito pro tvorbu výsledných děl – v tomto případě se pochopitelně jedná o software. Softwarové umění tedy využívá toto médium jako svůj primární umělecký materiál a věnuje se průzkumu jeho specifických expresivních a estetických kvalit.³ Softwarové dílo nám může přinést plnohodnotný umělecký prožitek, odkrýt manipulativní účinky a stereotypy ukryté ve funkcích mainstreamového softwaru, kritizovat fungování kulturních fenoménů, které jsou do značné míry ovlivňovány působením softwaru, reflektovat estetické vlastnosti digitálních médií nebo plnit úlohu subverzivního prostředku vůči dominantní ideologii. Umělci, kteří se věnují tvorbě v oblasti softwarového umění, usilují o průzkum kreativního potenciálu softwaru, rozvíjení inovativních způsobů jeho užití a vytváření alternativních modelů softwaru.⁴ Reflektují charakter současného umění, jehož produkce, distribuce i recepce je významně ovlivňována digitálními médii. Rovněž se soustředí na odkrývání ideologických efektů softwaru, které významně formují naši kulturu, avšak zároveň berou v úvahu také vliv širšího kulturního kontextu na utváření tohoto fenoménu.⁵

Na první pohled se může zdát, že softwarové umění je novou uměleckou praxí s velmi krátkou historií, která ztělesňuje současné tendence na poli umění nových médií. Když se však podíváme blíže na členitý terén umění

3

Nicméně existuje i skupina děl, která nevznikají v podobě spustitelného počítačového programu, avšak odkazují k estetice a operacím softwaru, proto i tato díla můžeme považovat za softwarové umění nezávisle na médiu, které využívají. Například dílo *.walk* od umělecké skupiny *socialfiction.org*, jehož základ tvoří programové instrukce, které však nejsou určeny pro počítač, ale slouží jako instrukce pro pohyb městským prostorem, který v tomto případě symbolizuje hardware počítače. Jednotlivé instrukce jsou spouštěny během procházky městem. Dílo je kritikou skryté architektury počítače, jejíž charakter a funkce zůstávají běžnému uživateli neznámé, zatímco v případě procházky městem je architektura plně viditelná. Vyzývá nás tedy, abychom zkoumali nejenom městskou architekturu, ale i tu hardwarovou, a zajímali se o to, jak ovlivňuje náš pohyb v digitálním prostoru.

4

Mezi umělce či umělecké skupiny zabývající se tvorbou softwarového umění patří například Amy Alexander, Alex McLean, Radical Software Group, Casey Reas, Geoff Cox, Jaromil, Florian Cramer, Adrian Ward nebo Annet Dekker.

5

Florian CRAMER – Ulrike GABRIEL, „Software Art“, *Netzliteratur*, 2001, http://www.netzliteratur.net/cramer/software_art_-_transmediale.html (cit. 9. 5. 2017).

dvacátého století, nalezneme zde několik paralel, inspirací a počátků, které nelze opomenout. Pokud nechceme ze softwarového umění učinit pouhý krátkodobý trend, který je obklopen dočasnou aurou novosti, jak už to v případě neustále se měnících digitálních médií bývá, je nutné soustředit se nikoliv na psaní jediného historického příběhu této umělecké formy, ale uvažovat nad několika paralelně existujícími historiemi, rozmanitými inspiracemi a počátky, které mohou složit komplexní obraz historických kořenů, z nichž softwarové umění vyrůstá. Tento text si tedy klade za úkol představit různé způsoby, jak uvažovat o historii softwarového umění, které jsou rozvíjeny v oblasti teoretické reflexe digitálního umění, a najít paralely mezi stávajícími uměleckými druhy či praxemi a softwarovým uměním.

SOFTWAREVÉ UMĚNÍ

Ještě předtím, než se budeme zabývat problematikou historie softwarového umění, pokusím se alespoň krátce představit typické strategie softwarového umění a příklady několika klíčových děl, která do značné míry formovala podobu této umělecké praxe.

Mezi často využívané strategie softwarového umění patří tvorba nových verzí tradičních programů (např. grafických editorů), které nevypadají a ani nefungují podle běžných měřítek, jejichž úkolem je nejen zpochybnění našich normalizovaných představ o jejich funkcích a estetické povaze, ale i předložení alternativního modelu softwaru. Například program *Auto-Illustrator* (2001) britského umělce Adriana Warda paroduje a zároveň kriticky reflektuje funkce nejrozšířenějšího grafického programu Photoshop, neboť zčásti automatizuje proces tvorby obrazu – vkládá do něj nové objekty bez vědomí uživatele, sám zakřivuje rovné linie, a provádí další činnosti, které mění výsledné dílo. Tímto software upozorňuje na automatizaci operací v grafických programech, které lze provést jediným kliknutím (vybarvení objektu, aplikace filtrů apod.) a dotýká se problému kreativity v souvislosti s užitím předdefinovaných funkcí grafických programů.

Softwaroví umělci skrze sofistikované aplikace zviditelňují skryté operace kódu, jindy nedostupné soubory či systémové složky. Proto nás díla softwarového umění mnohdy staví před sféru, která přesahuje naše smyslové vnímání, a konfrontují nás s naší vlastní bezmocností plynoucí z pocitu, že se setkáváme s něčím, čemu nerozumíme a co nejsme schopni kontrolovat. Například program *Carnivore* (2016) od umělecké skupiny Radical Software Group sleduje aktivitu uživatele na internetu a v reálném čase ji zobrazuje

jako nekonečný proud kódu, v němž je možné rozeznat jednotlivé činnosti v online prostředí.

Umělci také pracují s počítačovými viry a jinými destruktivními programy a postupy, které vnášejí do uspořádaného a logického digitálního světa fenomény chyby, destrukce a náhody, a ukazují tak křehkost digitálních obrazů ve chvíli, kdy jsou rozloženy počítačovým virem. Například dílo *Forkbomb* (2001) od umělce Alexe McLeana je krátkým kusem programového kódu, který po své aktivaci způsobí nekontrolovatelné větvení procesů spuštěných v počítači. Ve chvíli, kdy již počítač není schopen zajistit běh všech procesů, dojde k jeho zhroucení, které je vždy unikátní událostí, neboť jeho podoba i délka trvání závisí na konkrétním hardwaru počítače.

V softwarovém umění se můžeme také často setkat s využitím literárních strategií. Dílo *Self.pl* (2001) od umělce a teoretika Floriana Cramera je spustitelným počítačovým programem, který má zároveň podobu básně zapsané v programovém kódu:

```
#!/usr/bin/perl
open (IT, "< self");
while (<IT>) {
  push @it, $_
}
close (IT);
open (IT, ">> self");
print IT join (",\n ", @it);
close (IT);
```

Dílo je tvořeno jednoduchými instrukcemi, které stimulují fantazii čtenáře (například *open self*). Svým minimalismem připomíná formu japonské poezie haiku. Báseň reflektuje téma identity stroje a vzájemného splývání přirozeného a strojového jazyka.

Softwarové umění tedy zahrnuje široké spektrum odlišných přístupů a děl, jež kritizují a zviditelňují sociální i politické funkce softwaru, ohledávají vizuální a zkušenostní potenciál softwaru a zkoumají jeho limity.

SOFTWARE V DĚJINÁCH UMĚNÍ

Slovo software je v technologicky orientovaných i všeobecných slovnících a encyklopediích převážně definován jako soubor instrukcí, které řídí a kontrolují

operace hardwaru.⁶ Software tedy popisuje specifické úkoly, které hardware následně vykoná. Z této definice vyplývá, že software je technologickým instrumentem, který musí být vždy spuštěn na počítači nebo jiném hardwaru, aby vyvolal zamýšlený efekt. Tento předpoklad však mnozí teoretici i umělci, kteří se věnují oblasti softwarového umění, zpochybňují, a ve svých teoretických textech hledají paralely mezi softwarem a rozmanitými uměleckými praxemi.⁷ Například německý umělec a teoretik médií Florian Cramer spolu s umělkyní Ulrike Gabriel tvrdí, že stopy softwaru můžeme nalézt mnohem hlouběji v dějinách umění, než je na první pohled patrné.⁸

Cramer a Gabriel vidí jako první a nezbytný krok přehodnocení samotné definice softwaru. Pokud se snažíme definovat software, musíme brát v úvahu, že jeho počáteční existence má textovou podobu. Softwarové dílo je tvořeno souborem algoritmů⁹ napsaných v příslušném programovacím jazyce.¹⁰ Může být zapsáno v textovém editoru, na listu papíru nebo v naší paměti. V této fázi své existence má čistě konceptuální charakter a není závislé na žádném digitálním médiu.¹¹ Ve chvíli, kdy chceme softwarové dílo spustit, vložíme programový kód do příslušného digitálního média, které vykonává jednotlivé instrukce. Program tedy představuje určitý druh logické partitury.¹² Pojem partitura upomíná na tradiční techniku realizace uměleckého díla, která je

6

„Software“, *Techopedia*, 2018, <https://www.techopedia.com/definition/4356/software> (cit. 19. 1. 2018); „Software“, *Encyclopedia Britannica*, 2018, <https://www.britannica.com/technology/software> (cit. 19. 1. 2018); S. M. H. COLLIN – P. H. COLLIN – M. H. COLLIN, *The Dictionary of Information*, Londýn: Peter Collin Publishing 2002.

7

Srov. CRAMER–GABRIEL, „Software Art“; Thomas DREHER, „Conceptual Art and Software Art. Notations, Algorithms and Codes“, *IASLonline*, 2007, <http://iasl.uni-muenchen.de/links/NAKSe.html> (cit. 9. 5. 2017); Meredith HOY, *From Point to Pixel. A Genealogy of Digital Aesthetics*, Hanover, NH: Dartmouth College Press 2017; Saul ALBERT, „Artware“, *Mute*, roč. 1, 1999, č. 14, s. 89–92.

8

CRAMER–GABRIEL, „Software Art“.

9

Algoritmus je tvořen souborem instrukcí, které krok za krokem definují činnost počítačového programu. Když například v grafickém programu zadáme automatickou úpravu barevnosti fotografie, spustí se algoritmus, který provede všechny potřebné barevné korekce dle předem definovaných pravidel. Algoritmus se tedy podobá návodu, který definuje všechny potřebné kroky vedoucí k zamýšlenému výsledku, proto algoritmem nemusí být jen počítačový program, ale v obecnějším smyslu i kuchyňský recept, návod na sestavení nábytku apod.

10

Florian CRAMER, „Concepts, Notations, Software, Art“, *Netzliteratur*, 2002, https://www.netzliteratur.net/cramer/concepts_notations_software_art.html (cit. 9. 5. 2017).

11

Florian Cramer dodává, že pokud je možné spustit různé druhy algoritmů, které tu byly ještě před vynálezem počítače, v naší myslí, pak i software může existovat a být spuštěn bez hardwaru, a udává příklad programovacích příruček, v nichž najdeme vytištěný programový kód. Čtenář, který rozumí programovacímu jazyku, čte jednotlivé instrukce, spouští je ve své myslí, a postupně si vytváří představu o operacích a funkcích počítačového programu.

12

CRAMER, „Concepts“.

běžná v oblasti hudby. Hudebník dostane hudební dílo zapsané v partituře, která obsahuje instrukce pro jeho provedení. Výsledné vystoupení před publikem je specifickou realizací těchto instrukcí v reálném čase. Tento soubor instrukcí můžeme přirovnat ke specifickému algoritmu, který určuje provedení programu. Tím Cramer dokazuje, že charakteristické vlastnosti softwaru můžeme nalézt také v tradičních uměleckých formách. V jeho pojetí tedy není koncept softwaru limitován jen na formální instrukce pro počítač.¹³

I když jsme ukázali, že software není připoután pouze k jedinému typu média, stále zde zůstává otázka, jestli některé z hudebních či literárních děl vykazuje typické znaky, které definují díla softwarového umění. Zatímco softwarové umění velmi často usiluje o odhalení a kritickou reflexi instrukcí, kterými je tvořen software, většina hudebních i literárních děl nerozvíjí kritický přístup k systému instrukcí, které vedly k vytvoření díla, neusilují o zasazení těchto instrukcí do širších uměleckých, společenských nebo ideologických kontextů, ani se nesnaží o jejich kreativní modifikace, přestože existují výjimky.¹⁴

KONCEPTUÁLNÍ SOFTWARE

Při hledání kořenů softwarového umění a jeho možných předchůdců je nezbytné obrátit se k uměleckému směru označovanému jako konceptuální umění, který bývá považován za přímou inspiraci pro softwarové umělce.¹⁵ Konceptuální umění se rodí na počátku šedesátých let a do světa umění přináší důraz na koncept, který je ceněn mnohem více než fyzická realizace díla:

Myšlenky mohou být uměleckými díly; jsou
součástí vývojového řetězce, který případně

13

CRAMER, „Concepts“.

14

Jednu z výjimek tvoří literární skupina Oulipo, která pracuje s konceptuálními postupy v oblasti literatury. Tato skupina, která byla založena v roce 1960 ve Francii (zakladatelé Raymond Queneau a Francois Le Lionnais), se soustředila na hledání hranic literatury, reflexi pravidel, která podmiňují vznik literárního díla, rozmanité experimenty s formou literárního díla a rozvíjení nových přístupů k tvorbě literárních děl. Členové této skupiny často využívali k psaní svých děl soubor předem zformulovaných instrukcí, který určoval výslednou podobu díla. Například román *Zmizení* Georgese Pereca je realizací specifické předem dané instrukce, která spočívá v záměrném vynechání písmena *e*. Zmizení tohoto písmene se pak stává i součástí příběhu románu.

15

Srov. ALBERT, „Artware“; Casey REAS, „Who are the Progenitors of the Contemporary Synthesis of Software and Art?“, in: Arie ALTENA – Lucas van der VELDEN (eds.), *The Anthology of Computer Art. Sonic Acts XI.*, Amsterdam: Sonic Acts Press 2006, s. 9–25; DREHER, „Conceptual Art and Software Art“.

může získat nějakou formu. Všechny myšlenky
nemusí být nutně fyzicky realizovány.¹⁶

Teoretik Henry Flynt, který přichází s pojmem umění konceptu (*concept art*), tvrdí, že hlavním materiálem tohoto umění je jazyk, který operuje s koncepty a ideami.¹⁷ Flynt také uvádí jeho příbuznost s matematikou, kterou považuje za jeden z inspiračních zdrojů pro vznik tohoto druhu umění.

Tematickou náplň klasické fáze konceptuálního umění tvoří zkoumání, zviditelňování a zpochybňování nejrůznějších procesů a postupů spjatých s uměleckými díly. Dalo by se říci, že zde dochází k odkrývání „softwarů“, které řídí jednotlivá umělecká díla. Příbuznost konceptuálního umění se softwarovým uměním je tedy více než zřejmá.

Teoretik umění Jacob Lillemose rozlišuje několik proudů konceptuálního umění a upozorňuje na jejich podobnost s konkrétními strategiemi využívanými v rámci softwarového umění.¹⁸

Prvním z nich je lingvistický konceptualismus – reprezentovaný umělci, jako jsou například Sol LeWitt, skupina Art & Language nebo Joseph Kosuth – zaměřený na reflexi pojmu umění, přičemž umění nahlíží jako určitý sebereflexivní systém, jehož základními stavebními jednotkami jsou psaní, myšlenky a jazyk. Například americký umělec Sol LeWitt vytvořil cyklus abstraktních obrazů *Wall Drawing* (1971–2007), při jehož tvorbě vycházel z variací několika základních elementů malby (linie, oblouk, čtverec apod.).¹⁹ Každý název obrazu zahrnoval soubor instrukcí, které byly návodem pro vytvoření tohoto díla. Pro ilustraci lze uvést jeden z obrazů, který se jmenoval *Wall Drawing #97: Ten thousand straight and ten thousand not straight lines*. Dle tohoto návodu si

16

„Ideas alone can be works of art; they are in a chain of development that may eventually find some form. All ideas need not be made physical.“ Sol LEWITT, „Sentences on Conceptual Art“, *Art-Language*, roč. 1, 1969, č. 1, s. 11–13.

17

Henry FLYNT, „Concept Art“ (1961), in: Kristine STILES – Peter SELZ (eds.), *Theories and Documents of Contemporary Art. A Sourcebook of Artists' Writings*, Berkeley, CA: University of California Press 1996, s. 820–822.

18

Lillemose upozorňuje na to, že trendy konceptuálního umění, které uvádí ve svém textu, se vzájemně překrývají a rozhodně bychom je neměli vnímat jako protikladné. Jejich rozlišení je motivováno zejména snahou poukázat na široký rejstřík uměleckých strategií, které softwarové umění sdílí s konceptuálním uměním, a představit odlišné způsoby, jak nahlížet na softwarové umění. Viz Jacob LILLEMOSE, „A Re-declaration of Dependence—Software Art in a Cultural Context it Can't Get out of“, *artnode.org*, 2004, <http://www.artnode.org/art/lillemose/readme2004.html> (cit. 19. 1. 2018).

19

Ukázky z uměleckého cyklu jsou dostupné na této webové adrese: <http://massmoca.org/sol-lewitt/> (cit. 11. 6. 2018).

každý mohl vytvořit svou malbu. LeWitt zde tematizuje umělecké dílo jako systém pravidel a instrukcí. Instrukce, které připojil ke svým malbám, jsou mnohem podstatnější než samotné obrazy. Uplatňuje zde algoritmický přístup k umění, neboť se soustředí na tvorbu velkého množství variací základních parametrů malby.²⁰ V roce 2004 vytvořil softwarový umělec Casey Reas soubor děl inspirovaný LeWittovými *Wall Drawings*, který pojmenoval *{Software} Structures*. Jedná se o malby, které jsou vytvořeny softwarem, jenž postupuje podle výchozích instrukcí zadaných umělcem. Reas požádal skupinu umělců, aby Lewittovy instrukce zakódovali do různých programovacích jazyků, které počítač interpretuje odlišným způsobem, proto každá realizace LeWittových instrukcí měla jiný vizuální charakter. Zde můžeme spatřovat analogii s LeWittovým tvůrčím postupem, v rámci něhož přenechával realizaci svých děl jiným lidem, kteří rovněž interpretovali jeho pokyny odlišným způsobem. Reas ve svých *{Software} Structures* odkrývá otevřenost a virtuální potenciál softwarového jazyka a poukazuje na bohaté možnosti jeho aktualizace.²¹

Druhý trend je reprezentován umělci, kteří nahlíží umění jako soubor instrukcí a chápou jej jako čistě mentální fenomén.²² Například členové umělecké skupiny Fluxus vytvořili rozsáhlý soubor návodů, které nás podněcují k realizaci hudebních a výtvarných performancí nebo aktivit, jež mají ozvláštnit naši každodenní existenci. Návod Alison Knowles *Color Music #2* nám dává pokyn, abychom pomalovali prostor libovolné ulice třemi druhy barev a ke každé barvě si sami vymysleli odlišný druh pohybu.²³ Chůze po ulici se tak stane určitým druhem tance, který se mění podle barvy, na niž právě narazíme. Umělec Dave Griffiths vytvořil v roce 2005 vizuální 3D programovací prostředí *Fluxus*, které následuje myšlenky této stejnojmenné umělecké skupiny. Uživatel zde zadává instrukce v jednoduchém programovacím jazyce, které v reálném čase generují kontinuální proud grafických struktur a tvarů. Program je určen pro živá vystoupení, kde programátor tvoří a proměňuje virtuální prostředí přímo před očima diváků.

Posledním trendem, který Lillemose zmiňuje, je politicky orientované konceptuální umění, do něhož lze zařadit umělce jako Hans Haacke, Dan Graham

nebo Victor Burgin.²⁴ Tento proud konceptuálního umění kritizuje fungování společenských, byrokratických a technologických systémů, které formují naši imaginaci a sociální vztahy, a snaží se odhalit vnitřní logiku těchto procesů. Jako je tomu například v díle Hanse Haackeho z roku 1970, které bylo tvořeno dvěma hlasovacími nádobami, do nichž návštěvníci vhadzovali své odpovědi na otázku týkající se chodu muzea, v němž bylo dílo vystaveno. Toto dílo tedy představuje kritiku procesů, které stojí v pozadí fungování uměleckých institucí. Podobným způsobem jako zástupci tohoto proudu konceptuálního umění i softwarová umělci zkoumají programový kód, který leží v pozadí fungování digitálních technologií, jako specifický ideologický systém, který může formovat subjektivitu jednice i systém společenských vztahů.²⁵

Jak bylo ukázáno na konkrétních příkladech uměleckých děl a tvůrčích strategií, softwarové umění nelze reflektovat bez jeho zasazení do širšího kontextu umění dvacátého století, které tvoří podhoubí pro vznik uměleckých forem softwaru. Konceptuální umění představuje jednu z hlavních inspirací softwarového umění, neboť klade důraz na postupy, které preferují imateriální a algoritmické aspekty umění a usilují o odkrytí neviditelných procesů a instrukcí, které pohání svět umění.

Umělec a teoretik Casey Reas, který se dlouhodobě zabývá tvorbou i odbornou reflexí v oblasti softwarového umění, se svým textu „Kdo byli předchůdci současné syntézy softwaru a umění?“ dotýká otázky, jaké inspirační vlivy působily na první generaci softwarových umělců, do níž patří osobnosti jako John F. Simon Jr., Alex McLean, Hans Bernhard, Golan Levin a další.²⁶ Za tímto účelem oslovil samotné umělce, a jejich výpovědi zaznamenal. Z jejich odpovědí jednoznačně vyplývá, že jednou z hlavních inspirací pro jejich tvorbu byly postupy využívané v konceptuálním umění. Softwarové umění tedy vědomě přejímá teoretická východiska konceptuálního umění a dále je rozvíjí v digitálním médiu.

20

HOY, *From Point to Pixel*.

21

Ibid.

22

LILLEMOSE, „A Re-declaration“.

23

Ken FRIEDMAN – Owen SMITH – Lauren SAWCHYN (eds.), *The Fluxus Performance Workbook*, Performance Research e-Publications 2002, <http://www.deluxxe.com/beat/fluxusworkbook.pdf> (cit. 9. 5. 2017).

24

LILLEMOSE, „A Re-declaration“.

25

Příkladem může být dílo *Carnivore* od skupiny Radical Software Group zmíněné v úvodu článku v souvislosti s charakterizací softwarového umění.

26

REAS, „Who are the Progenitors“.



DESTRUKTIVNÍ KREATIVITA

Mezi rozmanitými formami softwarového umění mají své pevné místo programy, jejichž funkční charakteristikou je destrukce. Jedná se například o počítačové viry, trojské koně a další druhy malwaru.²⁷ Tento směr softwarového umění navazuje na linii destruktivního umění, která protíná celé dvacáté století.²⁸

K radikalizaci destruktivních tendencí v umění však dochází až v šedesátých letech dvacátého století, kdy začíná být samotný akt destrukce považován za umění.²⁹ Byl realizován jako performance, která probíhala přímo před diváky ve veřejném prostoru, což mu dodávalo i politický rozměr. Umělci využívali široký soubor destruktivních prostředků (nůžky, chemické sloučeniny, oheň, střelné zbraně apod.), s jejichž pomocí demonstrovali proces ničení, který vnímali jako protest proti tehdejšímu vojenskému, sociálnímu nebo politickému násilí.³⁰ V této době dochází také k technologizaci a automatizaci destruktivního umění. Hlavním milníkem pro rozvoj destruktivního umění byl festival Destruction in Art Symposium, který se konal v roce 1966 v Londýně. V rámci festivalu se setkala přes sto umělců z celého světa, kteří zde přednesli své příspěvky nebo realizovali své umělecké performance. Umělci ve svých dílech frekventovaně nahlíželi destrukci jako další dimenzi kreativity, nástroj pro kritiku konvenčních estetických norem a příležitost pro obohacení politického jazyka umění.³¹

Destruktivní počítačové programy a postupy, které jsou využívány softwarovými umělci, bývají často zacíleny na podobný soubor témat: Odkrývají manipulativní účinky neviditelných procesů a struktur skrytých pod grafickým rozhraním našeho počítače, zprostředkovávají nám nové estetické

27

Malware je souhrnným označením pro různé druhy softwaru, jejichž cílem je destrukce, manipulace uživatele, krádež dat nebo instalace nadbytečných programů, které zobrazují nevyžádanou reklamu nebo sledují činnost uživatele.

28

Destruktivní strategie se objevují již v rámci dadaismu, příkladem může být dílo *Indestructible Object* (1923) Mana Raye, které mělo být dle instrukcí zapsaných samotným umělcem zničeno jediným úderem kladiva, nebo v oblasti kinetického umění v podobě sebedestruktivních děl Jeana Tinguelyho.

29

Alan LIU, *The Laws of Cool. Knowledge Work and the Culture of Information*, Chicago: University of Chicago Press 2004.

30

Kristine STILES, „The Story of the Destruction in Art Symposium and the DIAS Affect“, in: Sabina BREITWEISER (ed.), *Gustav Metzger. Geschichte Geschichte*, Vídeň: Generali Foundation 2005, s. 41–65.

31

Ibid.

volby, které digitální médium nabízí, a rozšiřují možnosti aktivisticky zaměřeného umění o přímou intervenci. Umělecká dvojice JODI (Joan Heemskerck a Dirk Paesmans) v rámci své tvorby velmi často pracuje s chybou a destrukcí. V díle *Jet Set Willy Variations* (2002) destrukují konzolovou hru *Jet Set Willy* vydanou v roce 1984 tím, že modifikují kód, v kterém je napsána. Výsledkem jsou nejružnější variace původní hry, které transformují svět hry do souboru abstraktních tvarů, textových znaků nebo rozbitých obrazů. Dílo *The Lovers* (2001) od britské umělkyně Snehly Solanki odkrývá kreativní potenciál počítačového viru jako poetického sdělení. Umělecká instalace je tvořena dvěma propojenými počítači, na jejichž monitorech jsou zobrazeny dvě romantické básně. Do jednoho z počítačů je vložen počítačový virus, který po určité době nakazí i další počítač a postupně deformuje text obou milostných básní, až se stávají téměř nečitelnými. Dílo metaforicky znázorňuje dva milence, kteří se vzájemně nakazí počítačovým virem, tematizuje tedy lásku jako romantickou nákazu. Aktivistický emailový virus *DownJones* (2001) od umělecké skupiny Epidemic dokáže analyzovat emaily, které odesíláme, a nenápadně destruovat význam našich zpráv. Jakmile dokončí analýzu jazykové struktury emailu, pokusí se do něj vložit určité fráze, které nepůsobí příliš rušivým dojmem, a email odešle. Úkolem tohoto programu bylo především upozornit na problém možné manipulace našich sdělení, která denně posíláme, plně automatizovaným softwarem, proto byl program umístěn volně ke stažení na webových stránkách této umělecké skupiny spolu s popisem jeho funkcí.

Jeden z hlavních představitelů destruktivních tendencí v umění, Gustav Metzger, ve svém „Manifestu sebedestruktivního umění“ charakterizuje destruktivní umění jako umění změny, růstu a pohybu.³² Předkládá zde i svou vizi technologického uměleckého díla, jehož pohyb je dopředu naprogramován a posléze realizován, buď zcela bez zásahu člověka, nebo ovládán na dálku pomocí elektronického zařízení. Autodestruktivní umění by dle jeho názoru mělo útočit na kapitalistické hodnoty a vývoj zbraní hromadného ničení. Metzger také zmiňuje široký soubor materiálů a postupů, které lze využít v tvorbě autodestruktivního umění, mezi nimiž nalezneme i zmínku o kybernetice – vědě zabývající se výzkumem řízení a sdělování v živých organismech a strojích. Metzger tedy bere v úvahu i nové počítačové technologie. Ve vizích, které rozvíjí na počátku šedesátých let, lze rozpoznat současnou formu počítačového viru, který funguje na základě předem naprogramovaného

32

Gustav METZGER, „Manifesto Auto-Destructive Art“ (1960), in: STILES–SELZ, *Theories and Documents*, s. 470.

souboru akcí, šíří se digitálním prostředím bez našeho zásahu, a umělci je často používán pro politicky motivované umělecké akce.

Tuto skupinu děl softwarového umění tedy můžeme považovat za pokračování destruktivních tendencí v umění dvacátého století.³³ Stejně jako v šedesátých letech je akt ničení využíván jako prostředek, který umožňuje reflektovat a aktivně reagovat na politickou i společenskou situaci. Nyní však umělci rozkrývají subtilnější způsoby politické manipulace umožněné komplexními softwarovými systémy, které jsou pevnou součástí našich pracovních a soukromých životů. Na rozdíl od předchůdců mohou své strategie aplikovat ve větším měřítku a poskytnout je mnohem širšímu publiku, a to navíc do jisté míry anonymně, neboť tyto strategie jsou realizovány prostřednictvím internetové sítě. Setkáváme se také s díly, která ukazují poetický rozměr destrukce a demonstrují její estetické kvality. Současní tvůrci tedy rozvíjí odkaz svých předchůdců, avšak přizpůsobují jej digitálnímu prostředí, v němž se nyní pohybujeme. Sofistikovanější technologické možnosti, které mají k dispozici, zvyšují variabilitu možných akcí a vyjadřovacích prostředků spojených s aktem destrukce.

ZAPOMENUTÝ POČÁTEK SOFTWARE UMĚNÍ: OSAMĚLÍ EXPERIMENTÁTOŘI

Až do této chvíle byla věnována pozornost uměleckým formám, které nebyly svou povahou digitální, avšak vykazovaly vlastnosti obvykle spojované se softwarovým uměním – jako jsou tvořivá hra s instrukcemi, důraz na procesuální a abstraktní kvality uměleckého díla, destruktivní strategie nebo kritická reflexe fungování politických a společenských struktur. Nyní se přesouváme do oblasti digitálního umění, konkrétně do rané fáze jeho vývoje. Počátky digitálního umění jsou spatřovány v tvorbě skupiny umělců, kteří se přibližně od šedesátých let soustředí na využití počítače jako média vhodného pro tvorbu rozmanitých uměleckých forem, ať už se jedná o počítačovou grafiku, hudbu nebo poezii.³⁴ Umělci z tohoto okruhu, mezi něž patří Frieder Nake, Michael Noll, Georg Nees, zkoumají umělecký potenciál tohoto

33

LIU, *Laws of Cool*;

34

Srov. Wolf LIESER, *Digital Art*, Königswinter: h.f.ullmann 2009; Christiane PAUL, *Digital Art*, New York: Thames & Hudson 2015.

média a v ohnisku jejich zájmu se ocitá strojová estetika a kreativita vycházející ze schopnosti počítače pracovat autonomně, a tedy tvořit dílo nezávisle na člověku.

Autoři, kteří se věnují vymezení pozice softwarového umění v rámci širších dějin digitálního umění, se ve velké většině přiklání k názoru, že tuto ranou fází ve vývoji digitálního umění bychom neměli považovat za počátek či inspirační zdroj pro pozdější vznik softwarového umění.³⁵ Spíše tvrdí, že záměry umělců v šedesátých letech spočívaly na zcela jiných předpokladech než v případě softwarových umělců na počátku nového tisíciletí. Zatímco digitální umění v období šedesátých a sedmdesátých let kladlo důraz na prezentaci uměleckého výstupu vytvořeného počítačem (obraz, zvuk, text) a preferovalo čistě estetickou dimenzi uměleckého díla, které jako by existovalo nezávisle na vnějším světě, softwarové umění klade do popředí obvykle neviditelné procesuální kvality digitální umělecké tvorby, zkoumá charakter a možnosti digitálního kódu a odkrývá politické, kulturní či sociální kontexty, které utvářejí software, a současně jsou softwarem ovlivňovány.³⁶

Kořeny softwarového umění bychom tedy podle mínění teoretiků i samotných umělců neměli hledat v rané fázi vývoje digitálního umění, ale spíše se zaměřit na paralelně se rozvíjející umělecké směry skýtající myšlenkovou východiska, která jsou softwarovému umění mnohem bližší.³⁷ Avšak historie digitálního umění, především ta raná, dosud není zcela zmapována, proto bychom měli odolat pokušení činit předčasné závěry a schematizovat jednotlivá vývojová období.³⁸ Tento přístup by mohl vést například k tomu, že nám unikne určité zajímavé dílo, jež se vymyká naší předem utvořené koncepci. Spíše bychom se měli držet představy historie digitálního umění jako nevyznačeného pole, které může generovat neustále nové možnosti, překvapivé nálezy, a kontinuálně formovat naše poznání týkající se vývoje a rozmanitých podob digitálního umění.

35

Srov. CRAMER-GABRIEL, „Software Art“; Inke ARNS, „Read_me, run_me, execute_me. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text“, in: Rudolf FRIELING – Dieter DANIELS (eds.), *Medien Kunst Netz 2. Thematische Schwerpunkte*, Vídeň: Springer 2005, s. 176–208.

36

HORÁKOVÁ, „Softwarové umění“.

37

REAS, „Who are the Progenitors“; HORÁKOVÁ, „Softwarové umění“.

38

Grant D. TAYLOR, *When The Machine Made Art. The Troubled History of Computer*, New York: Bloomsbury 2014; Paul BROWN – Charlie GERE – Nicholas LAMBERT (eds.), *White Heat Cold Logic. British Computer Art 1960–1980*, Cambridge, MA: MIT Press 2009; Hannah HIGGINS – Douglas KAHN (eds.), *Mainframe Experimentalism. Early Computing and the Foundations of the Digital Arts*, Berkeley, CA: University of California Press 2012.

Jedním z příkladů zapomenutých digitálních děl, která však mohou účinně zpochybnit zavedenou představu o rané historii digitálního umění, je počítačový program *Love Letters Generator*, jehož autorem byl počítačový vědec Christopher Strachey.³⁹ Tento program z roku 1952, určený pro sálový počítač Mark I, měl typické vlastnosti díla, které bychom zařadili do generativního digitálního umění. Jeho autor jej vybavil dostatečnou zásobou nejružnějších slovních obrátů, které používáme, když píšeme milostný dopis, z nichž na základě pravidel pro konstrukci vět vytvářel milostné dopisy automaticky vygenerované počítačem. Teoretik digitálních médií Noah Wardrip-Fruin však upozorňuje, že zavedené vzorce interpretace digitálního umění mohou vést k chybnému rozpoznání významu tohoto raného digitálního díla.⁴⁰ Pokud budeme k dílu přistupovat jako k typickému zástupci raného digitálního umění, zaměříme svou pozornost na analýzu výsledných reprezentací, které počítač vytvořil. Základem díla *Love Letters Generator* však nejsou výsledné dopisy, ale samotný generátor dat. Úkolem programu není prezentovat možnosti strojového vyjadřování pocitů a emocí skrze milostné dopisy ani přinášet potěšení ze čtení, jeho úkolem je něco jiného, co je skryto pod povrchem. Wardrip-Fruin tvrdí, že dílo funguje jako parodie normativního vyjádření touhy, neboť obsahem programu jsou nejružnější klišé, která se v milostných psaních neustále opakují.⁴¹ Výsledné dopisy jsou často humorné, nedokonalé, a přináší neumělé kombinace stereotypních formulací.

Základem díla je tedy samotný generativní proces, který účinně paroduje lidské psaní dopisů a sděluje nám, že psaní konvenčního milostného dopisu je poměrně jednoduchý postup, neboť nevyžaduje velkou slovní zásobu ani složité strukturované věty, a proto jej lze delegovat na počítač. Zároveň dílo reprezentuje proces, který je dopředu odsouzen k selhání (nedokonalost a neumělost dopisů), a tudíž poukazuje na hranice strojové kreativity. Právě tyto dvě navzájem protikladné roviny, přítomné v díle, výstižně ilustrují napětí mezi textovým výstupem – který predestinuje potenciál digitálního média a preferuje estetickou rovinu sdělení, což by dílo řadilo k typickým

zástupcům raného digitálního umění – a hlubší vrstvou na úrovni algoritmu, který zde má subversivní a kritickou funkci – podrývá výsledné textové sdělení a reflektuje samotné fungování a možnosti softwaru, čímž se podobá dílům z oblasti softwarového umění. Analýza, kterou nám předkládá Noah Wardrip-Fruin, ukazuje, že předchůdce softwarového umění můžeme nalézt také v rané fázi digitálního umění, ovšem v tomto případě se jedná o dílo, které vzniklo mimo tehdejší umělecké struktury, v počítačové laboratoři, a jedná se tedy spíše o výjimku.⁴² Není však vyloučeno, že v budoucnu budeme svědky dalších podobných nálezů raných digitálních děl vytvořených v jiných prostředích a kontextech, mimo pozornost uměleckého světa, která se stanou podnětem pro kritickou revizi naší představy o počáteční fázi digitálního umění.

SOFTWARE JAKO NOVÁ METAFORA PRO UMĚNÍ A UMĚLECKÝ AKTIVISMUS

V historii digitálního umění můžeme nalézt dva důležité mezníky, které přinesly první soustavnější reflexi softwaru ve světě umění. Slovo software vstupuje do uměleckého světa v roce 1970, a to prostřednictvím výstavy *Software – Information Technology: Its New Meaning for Art* v Jewish Museum v New Yorku, jejímž organizátorem byl umělec a kritik Jack Burnham. Ve stejném roce také vychází první číslo časopisu *Radical Software* určeného umělcům a aktivistům, kteří pracovali s novými technologiemi.

Výstava Jacka Burnhama byla věnována vztahu mezi současným uměním a softwarem, který měl v tomto případě sloužit jako výstižná metafora pro popis uměleckých praxí, které byly typické pro tvorbu konceptuálních umělců. Burnhamův náhled na konceptuální umění vychází z teoretických východisek, která nastínil ve svém textu *Systems Esthetics*.⁴³ Hovoří zde o posunu současného umění od objektu k systému a přichází s pojmem systémová estetika, který podle něj charakterizuje rozmanité projevy současného umění v podobě happeningů, performancí, konceptuálního umění apod. Umění již netvoří hotové objekty, ale manifestuje fungování systémů, klade důraz na procesualnost a vytváří díla, která se věnují konstrukci a rozvíjení dynamických vztahů mezi objekty, živými organismy a myšlenkami.

39

Noah WARDRIP-FRUIN, „Christopher Strachey. The first digital artist?“, *Grand Text Auto*, 1. 8. 2005, <https://grandtextauto.soe.ucsc.edu/2005/08/01/christopher-strachey-first-digital-artist/> (cit. 9. 5. 2017).

40

Noah WARDRIP-FRUIN, „Digital Media Archaeology. Interpreting Computational Processes“, in: Erki HUHTAMO – Jussi PARIKKA (eds.), *Media Archaeology. Approaches, Applications and Implications*, Berkeley, CA: University of California Press 2011, s. 302–322.

41

Ibid.

42

Ibid.

43

Jack BURNHAM, „Systems Esthetics“, *Artforum*, roč. 7, 1968, č. 1, s. 31–35.

Software charakterizovaný svou procesuálností, imateriální povahou a schopností simulovat chování rozmanitých systémů považoval Burnham za fenomén, který velmi přesně odráží významný posun v současném umění a kultuře.⁴⁴ Proto v rámci výstavy umísťuje vedle sebe díla digitálního a konceptuálního umění.⁴⁵ Software zde tedy plní úlohu metaforu, aniž by byl reflektován jako nové umělecké médium se specifickými kvalitami.

Tvůrci časopisu *Radical Software* Phyllis Gershuny, Beryl Korot, Michael Shamberg a Frank Gillette, předsťirají v úvodním textu hlavní východiska svého myšlení o softwaru, který nahlíží z perspektivy mediálního aktivismu.⁴⁶ Tvrdí, že v současné době moc třímá v rukou ten, kdo má přístup k informacím a k technologiím, které umožňují distribuci těchto informací. Jedná se především o centralizovaná masová média, jejichž hlavním cílem je maximalizace zisku. Aby lidé byli schopni opět kontrolovat své životy, musí vytvořit alternativní informační kanály, které budou reflektovat jejich skutečné problémy a poskytovat jim alternativní náhledy na žitou realitu. K tomuto účelu navrhuji využít tehdy nové médium – videokameru, která vyniká svou dostupností a rychlostí zpracování záznamu, a proto představuje ideální médium pro mediální aktivismus. Kamera umožňuje šíření informací, které mohou modifikovat stávající sociální řád. Stává se ideálním nástrojem, který může pozitivně změnit myšlení komunit i jedinců, řízené sociálním softwarem – souborem instrukcí, pravidel, vzorců chování a dostupných informací, který lze reprogramovat stejně jako technologický software. Tvůrci časopisu *Radical software* tedy používají slovo software jako metaforu, prostřednictvím které nás upozorňují na konstruovanost sociální reality a možnosti její pozitivní proměny.⁴⁷

Podrobnější popis obou zmíněných událostí naznačuje, že vstup softwaru na půdu umění v roce 1970 nelze považovat za počátek softwarového umění. Přestože software tehdy upoutal pozornost umělců i kritiků, tento termín

44
Idem, „Notes on Art and Information Processing“, in: Judith Benjamin BURNHAM (ed.), *Software – Information Technology. Its New Meaning for Art* (kat. výst.), New York: Jewish Museum 1970, s. 10–15.

45
Nelze však opomenout druhou rovinu výstavy, která reflektovala, jakým způsobem software proměňuje naše představy o kreativité, vnímání a limitech umění. Díla digitálního umění prakticky demonstrovala, jakou podobu mohou získat technologie v rukou umělců.

46
Beryl PHYLLIS – Korot GERSHUNY, [úvodní text], *Radical Software*, roč. 1, 1970, č. 1, s. 1.

47
Ibid.

sloužil spíše jako metafora, která byla využita k vyjádření významných posunů v umělecké a kulturní sféře a také aktivistických vizí spojených s novými médii. Jeho potenciál jako samostatného uměleckého vyjadřovacího prostředku nebyl v tomto období reflektován.

VSTUP SOFTWARE DO SVĚTA UMĚNÍ: INSTITUCIONALIZACE SOFTWAREVÉHO UMĚNÍ V LETECH 1999–2003

Na konci druhého tisíciletí se postupně formuje skupina umělců, kteří se soustředí na kritický průzkum počítačových programů a tvorbu kreativních a uměleckých forem softwaru. Je třeba podotknout, že se nejednalo o jednotné hnutí umělců, které by vydalo společný manifest nebo bylo propojeno na základě geografické, institucionální nebo generační příbuznosti. Umělci v oblasti softwarového umění se pohybují spíše v otevřeném poli možností, které umožňuje svobodné ohledávání softwaru jako umělecké formy.

Zrod softwarového umění jako umělecké praxe, která začíná být postupně akceptována teoretiky i umělci v oblasti digitálního umění, můžeme zasadit do období konce devadesátých let, kdy probíhají debaty o konci či brzkém zániku net artu.⁴⁸ Tato umělecká forma dominovala digitálnímu umění v průběhu devadesátých let.⁴⁹ Na konci druhého tisíciletí začíná být mnohými teoretiky i samotnými umělci považována za vyčerpanou.⁵⁰ Alexander Galloway ve svém textu „net.art Year in Review: State of net.art 99“ hovoří o tom, že net artové umění je mrtvé, neboť jeho specifická estetika byla produktem dobových technických

48
Srov. Andreas BROEGGER, „Software Art – an introduction“, *artificial.dk*, 2003, <http://www.artificial.dk/articles/software.htm> (cit. 9. 5. 2017); SZÜCSOVÁ, *Softwarové umění*.

49
Prohlášení o zániku net artu je však nutné brát s určitou rezervou. Neznamená to, že net artová díla zcela zmizela z oblasti digitálního umění, naopak vznikají i v současné době (viz například *Net Art Anthology*, <https://anthology.rhizome.org/> [cit. 16. 8. 2018]), jedná se spíše o to, že tato umělecká praxe na konci devadesátých let pozvolna ustupuje ze středu pozornosti, a na její místo nastupují jiné, aktuálnější směry digitálního umění, které reagují na vývoj technologií i jejich proměňující se společenskou recepci – např. post-internetové umění nebo právě softwarové umění.

50
Tilman BAUMGÄRTEL, „Art on the internet – The Rough Remix“, *nettime.org*, 1998, <http://www.nettime.org/Lists-Archives/nettime-l-9810/msg00082.html> (cit. 9. 5. 2017); Alexander R. GALLOWAY, „net.art Year in Review: State of net.art 99“, *Switch*, roč. 5, 1999, č. 3, <http://switch.sjsu.edu/web/v5n3/D-1.html> (cit. 9. 5. 2017).

omezení internetové sítě.⁵¹ Jednalo se zejména o pomalé a poměrně drahé internetové připojení, jehož rychlost byla nedostatečná pro uspokojivý přenos videí nebo fotografií ve vysokém rozlišení. Proto umělci pracovali s textovými znaky, které skládali do podoby obrazů (ASCII art), programovali webové stránky s minimalistickou grafikou, vytvářeli animovanou textovou poezii apod.⁵²

Na konci devadesátých let už jsou tato omezení minulostí. Umělci v oblasti net artu již nejsou ničím svazováni. Avšak v tuto chvíli, jak tvrdí Galloway, net art ztrácí svou jedinečnost, svou specifickou estetiku.⁵³ Andreas Broegger rovněž prohlašuje net art za mrtvý, neboť se natolik zaměřoval na definování své specifčnosti a vylučování jiných forem umění, že ustrnul ve svém vývoji a nedokázal vstřebat nové podněty, ať už se jednalo o uměleckou nebo technologickou sféru.⁵⁴ Tilman Baumgärtel hovoří o prvním formativním období internetové kultury, které je nyní u konce.⁵⁵

Jak ukazují výše zmíněné úvahy, na konci druhého tisíciletí se prostředí internetu otevírá novým možnostem, vznikají inovativní softwarové aplikace, a prostor pro umělecké experimenty se významně rozšiřuje. Zatímco umělci z oblasti net artu, k nimž patřil Vuk Ćosić, Mark America, Olia Lialina, Heath Bunting a další, pracovali převážně s webovým rozhraním, tedy s HTML kódem, umělci z oblasti softwarového umění jako Adrian Ward, Antoine Schmitt, Jaromil, skupina Radical Software Group a další se nezaměřují pouze na experimenty s jedním médiem, s jedním programovacím jazykem, ale využívají různé programovací jazyky a vytvářejí vlastní software, který odpovídá jejich potřebám a specifickému vidění světa.⁵⁶

Od roku 1999 se začíná rozvíjet diskuze o uměleckém softwaru ve formě článků v odborných periodících, kde se autoři snaží pojmenovat novou uměleckou praxi a zasadit ji do umělecko-historického kontextu.⁵⁷ Avšak

51

GALLOWAY, „net.art Year in Review“.

52

ASCII je označení pro standardizovaný soubor 128 znaků běžně využívaných v informatice. Součástí tohoto souboru jsou písmena anglické abecedy, číslice a jiné speciální znaky.

53

Ibid.

54

BROEGGER, „Software Art“.

55

BAUMGÄRTEL, „Art on the internet“.

56

Zkratka HTML (HyperText Markup Language) označuje jazyk, v němž jsou vytvářeny webové stránky.

57

Srov. ALBERT, „Artware“; Geoff COX – Alexander McLEAN – Adrian WARD, *The Aesthetics of Generative Code*, příspěvek prezentovaný na konferenci Generative Art 00, Milán: Politecnico di Milano University 2000.



nejvýznamnější diskuzní platformu pro úvahy o softwarovém umění představoval mailing list *Nettime*. Tato komunikační platforma hrála významnou roli v debatách o net artu, na které logicky navázala diskuze o umění softwarovém. Díky rychlému způsobu komunikace i publikace teoretických příspěvků dotýkajících se tohoto tématu byla komunita shromážděná kolem služby *Nettime* schopna reflektovat tento fenomén již v jeho počátcích a významně se podílet na vytváření teoretického diskurzu, který určoval, jakým způsobem se bude o softwarovém umění hovořit. Debaty, rozhovory, a eseje teoretiků (Jon C. Ippolito, Tilman Baumgärtel, Robert Atkins) a umělců (Netochka Nezvanova, John F. Simon Jr., skupina I/O/D) naznačují, že v digitálním umění můžeme sledovat nový trend tvorby softwaru, který navazuje na tradici konceptuálního umění, odvrací se od fascinace spektakulárním povrchem digitálních děl a usiluje o průzkum softwarového kódu a procesů probíhajících v digitálních médiích.⁵⁸

V roce 2001 se konal čtrnáctý ročník festivalu umění a digitální kultury *Transmediale*, kde měli umělci poprvé možnost prezentovat svá softwarová díla a účastnit se soutěže o nejlepší dílo tohoto typu. Předsedou odborné festivalové poroty byl teoretik a umělec Florian Cramer, který se věnuje oblasti softwarového umění po praktické i teoretické stránce. V textu, který napsal při příležitosti konání tohoto festivalu, obhájí softwarové umění jako novou formu uměleckého vyjádření.⁵⁹ Vítězná cena za umělecký software byla rozdělena mezi dva umělce – Adriana Warda a Netochku Nezvanovou. Adrian Ward byl oceněn za své dílo *Auto-Illustrator* (viz výše). Netochka Nezvanova získala cenu za svůj experimentální webový prohlížeč *Nebula.M81*, který v reálném čase převádí HTML kód webových stránek do animovaného textu, grafiky a hudby. Tato událost signalizuje, že softwarové umění se začíná profilovat jako nová oblast digitálního umění.

Skutečnost, že softwarové umění bylo definitivně přijato do uměleckého světa a odborného diskursu teorie umění, stvrzuje série událostí, které

58

Jon C. IPPOLITO, „More on Bchner/jodi/formalism“, *Nettime*, 1998, <https://nettime.org/Lists-Archives/nettime-l-9805/msg00062.html> (cit. 9. 5. 2017); Tilman BAUMGÄRTEL, „Interview with John F. Simon“, *Nettime*, 1999, <https://nettime.org/Lists-Archives/nettime-l-9907/msg00067.html> (cit. 9. 5. 2017); Robert ATKINS, „Art And Code: Robert Atkins & John Simon“, *Nettime*, 2001, <https://nettime.org/Lists-Archives/nettime-l-0103/msg00000.html> (cit. 9. 5. 2017); Netochka NEZVANOVÁ, „[ot] [Int] \n2+0\ e. nter. search with Netochka Nezvanova - re: Nebula.m81.Autonomous“, *Nettime*, 2000, <https://nettime.org/Lists-Archives/nettime-bold-0005/msg00326.html> (cit. 9. 5. 2017); Geert LOVINK, „Interview with I/O/D, the Makers of Web Stalker“, *Nettime*, 1998, <https://nettime.org/Lists-Archives/nettime-l-9804/msg00072.html> (cit. 9. 5. 2017).

59

CRAMER–GABRIEL, „Software Art“.



se odehrály v prvních letech nového tisíciletí. V roce 2002 se konal festival *Read_Me*, který byl jako první festival věnován softwarovému umění.⁶⁰ Jeho pořadatelé se v textu, v němž popisují koncept festivalu, vymezují vůči komerčnímu softwaru, který dle nich omezuje kreativní myšlení uživatelů i umělců svými jasně definovanými funkcemi, jejichž podobu nelze měnit.⁶¹ Uživatel i umělec tedy podléhají manipulaci a vykonávají pouze přednastavené úkony. Z tohoto důvodu pořadatelé vyzývají umělce, aby vytvářeli díla v duchu open source softwaru.⁶² Projekty přihlášené na festival by měly hledat alternativní způsoby užití softwaru, modifikovat či dekonstruovat komerční software nebo odhalovat jeho skrytou ideologii. Jedním z vítězných projektů oceněných na festivalu bylo dílo *Deskswap* od Marka Daggetta, které upozorňuje na standardizaci pracovního prostředí našeho počítače. Program funguje jako obyčejný screensaver, který však při své aktivaci pořídí snímek pracovní plochy počítače a odešle jej na server, kde jsou snímky ukládány. Uživatelé si takto mohou vzájemně prohlížet fotografie svých pracovních ploch, jejichž základní prvky vypadají velmi podobně. Dílo je připomínkou toho, že i když se může zdát, že nám pracovní plocha nabízí jistou míru personalizace, ve skutečnosti uživatelé z celého světa využívají podobné programy, ikony, soubory či názvy složek.

Výstava *I love You* uspořádaná v roce 2003 v Muzeu aplikovaného umění ve Frankfurtu poprvé představila specifickou oblast softwarového umění, která je věnována tvorbě destruktivních programů. V tomto případě se jednalo zejména o počítačové viry. Kurátorka výstavy Franziska Nori shromáždila pestrý výběr uměleckých děl s touto tematikou: umělecké počítačové viry, aktivistické softwarové aplikace, básně v digitálním kódu apod. Svá díla zde prezentovali umělci jako Sneha Solanki, Eva a Franco Mattesovi, skupina *epidemiC*, Jarmil a další. Díla byla prezentována prostřednictvím počítačových terminálů, digitálních projektorů, velkoplošných pláten, nebo ve formě umělecké performance. Jedním z vystavených děl byl například virus *Bienalle.py* (2001), jehož

autory byli Eva a Franco Mattesovi spolu se skupinou *epidemiC*, který se šířil mezi dvěma propojenými počítači vybavenými antivirovým programem.⁶³ Oba počítače byly naprogramovány takovým způsobem, aby v nekonečné smyčce demonstrovaly průběh vzájemné nákazy počítačovým virem a jejího následného odstranění prostřednictvím antivirového programu. Zasazení tohoto procesu do uměleckého kontextu vzbudilo nejrůznější reakce mezi návštěvníky i v dobovém tisku, což bylo i cílem umělců. Došlo nejen na debaty týkající se uměleckého potenciálu počítačového viru, ale i k odmítnutí viru jako nebezpečného programu, který by rozhodně neměl být součástí uměleckého světa. Součástí díla byl také kód viru, který se skládal z instrukcí s poetickými názvy (*soul*, *mybody*, *mysoul.close* apod.), a připomínal tak báseň napsanou v počítačovém kódu. Umělci tedy představili virus jako určitý druh umělecké performance probíhající v reálném čase, která skrze reakce, jež vyvolává, odráží kritický diskurs obklopující současné umění, a zároveň jako literární dílo, které se může šířit v internetové síti.

Výstavu doprovázel také katalog, v němž teoretici a umělci popisovali praxi programování počítačového viru jako kreativní proces psaní.⁶⁴ Umělec píše digitální kód, který má své poetické kvality a expresivní potenciál. Skrze destruktivní programy může tvůrce vyjádřit své osobní pocity či politický komentář. Kurátorka Franziska Nori tedy uvedla destruktivní programy do galerijního prostoru a rozšířila povědomí laické i odborné veřejnosti o dalším významném druhu softwarového umění.

Snahu o zprostředkování a uchování softwarového umění reprezentuje internetový archiv *Runme.org*.⁶⁵ Tento archiv od roku 2003 shromažďuje díla z této umělecké oblasti a vnáší jistý řád do nepřehledného světa uměleckého softwaru, který vede svou rozptýlenou existenci v prostředí internetové sítě. Portál *Runme.org*, který má podobu kolaborativní online databáze, neobsahuje striktně vymezené kategorie, které by přesně určovaly, co může být softwarovým dílem a co nikoliv. Autoři jsou si vědomi diverzity softwarového

60

Festival *Read_Me* organizovalo moskevské mediální umělecké centrum Macros Center v letech 2002–2005. Hlavním kurátorem festivalu byla Olga Goriunova, která se zabývá teorií digitálního umění. Kurátorem off-line části festivalu byl Alexej Šulgin, významný představitel net artu.

61

Read_me festival 1.2, 2002, http://web.archive.org/web/20021212040735/www.macros-center.ru/read_me/abouten.htm (cit. 9. 5. 2017).

62

Základní idea open source softwaru spočívá ve volném poskytnutí zdrojového kódu určitého programu komunitě uživatelů, aby jej mohli svobodně modifikovat, využít k jiným projektům nebo dále rozvíjet.

112

63

Toto dílo bylo poprvé představeno na 49. benátském bienále, které se konalo v roce 2001.

64

Franziska NORI (ed.), *I Love You Catalogue* (kat. výst.), 2003, http://www.digitalcraft.org/?artikel_id=283 (cit. 9. 5. 2017).

65

Archiv byl založen organizátory festivalu *Read_Me*. Vznik archivu lze nahlížet jako významnou součást aktivit těchto kurátorů, umělců a teoretiků, jejichž cílem bylo rozšíření povědomí o tomto nově se rozvíjícím druhu digitálního umění. Archiv se nachází na této adrese: <http://runme.org/>

113

umění a respektují jeho proměnlivý charakter a jistou neuchopitelnost či ne-definovatelnost, která plyne z neustálých inovací softwaru a jeho funkcí. Proto zde najdeme široké spektrum kategorií, které zahrnují i díla konceptuálního umění, jež nepracují se softwarem.⁶⁶ Struktura archivu tedy ukazuje, že oblast softwarového umění je velmi široká.

Díky širokému rejstříku teoretických, uměleckých a kurátorských aktivit, které se odehrávaly zejména v letech 1999 až 2003, se softwarové umění stává akceptovanou součástí digitálního umění. V následujících letech se intenzita aktivit kolem softwarového umění snižuje. Softwarové umění se nadále objevuje jako běžná součást výstav, festivalů a uměleckých projektů, avšak počet samostatných výstav významně klesá.⁶⁷ Jedním z důvodů může být skutečnost, že softwarové umění nikdy necítilo silnou potřebu samo sebe vymezovat vůči ostatním uměleckým druhům, definovat svou specifčnost, nebo deklarovat svou příslušnost k jedinému médiu, jak tomu bylo v případě net artu, a dále se kontinuálně rozvíjí a proměňuje na základě nových uměleckých i technologických podnětů.

SOFTWAREVÉ UMĚNÍ JAKO NOVÝ POČÁTEK

Na závěr je nezbytné zmínit koncepci britského umělce a teoretika Geoffa Coxe, bez níž by byl dosavadní přehled možných historií a počátků softwarového umění neúplný, neboť tato koncepce, na rozdíl od těch zmíněných, odmítá samotné snahy o psaní jakékoliv historie softwarového umění. Cox tvrdí, že není nutné psát klasickou historii softwarového umění, která identifikuje jeho příbuznost s jinými uměleckými směry, nalezne jeho počátek a zmapuje jeho vývoj.⁶⁸ Místo toho navrhuje, abychom se pokusili softwarové umění vnímat jako příležitost pro narušení historického kontinua, které může softwarovému umění navrátit jeho revoluční potenciál.

66

Mezi další kategorie patří například umění kódu, konceptuální software, politický a aktivistický software transformace dat nebo selhání systému.

67

V roce 2008 se v Centre del Carme ve Valencii konala výstava *After the Net*, která prezentuje různé směry vývoje softwarového umění. V roce 2010–2011 v kulturním centru MU v Eindhovenu probíhalo sympozium a výstava *Funware*. Tento projekt se zaměřoval na reflexi vztahu mezi softwarovým uměním a konceptem zábavy, který zde byl nahlížen jako kreativní síla, jež propustuje vědu i umění.

68

COX, *Antithesis*.

Cox se vrací do minulosti, aby upozornil na podobné snahy, které byly rozvíjeny v případě avantgardních hnutí vznikajících v průběhu dvacátého století.⁶⁹ Zmiňuje zejména dadaismus, situacionismus nebo destruktivní umění. Tyto avantgardy odmítají dosavadní historii a úlohu umění, a usilují o vytvoření nového umění, které stvoří novou přítomnost, novou životní realitu. Protože tento proces se neobejde bez destrukce tradičního umění a společenských vazeb, charakterizuje každou avantgardu vnitřní násilí, které funguje jako metafora, jejímž úkolem je podnítit změnu v sociální sféře – transformovat prostřednictvím umění myšlení, pocity a konání lidí.

Historie je těmito avantgardami pojímána optikou dialektického materialismu, což znamená, že nenahlíží na historii jako na sílu, která je vůči našemu konání a životu vnější, ale spíše jako na fenomén, který můžeme aktivně proměňovat svým konáním. Jedním z projevů této touhy jsou destruktivní tendence namířené proti komodifikaci umění a proti pojetí uměleckého díla jako fyzického objektu s tržní hodnotou. Revoluční potenciál uměleckých avantgard se však postupně vytrácí, neboť původně revoluční postupy jsou komodifikovány prostřednictvím grafických programů (např. Adobe Photoshop), které absorbovaly typické avantgardní techniky, jako je koláž, recyklace nebo *appropriace*, a učinily z nich základní charakteristiky svého fungování.⁷⁰

Cox si tedy klade otázku, jakým způsobem lze navrátit umění jeho revoluční potenciál.⁷¹ Odpověď nachází v transformativním potenciálu softwarového umění. Pokud je zasadíme do historického kontextu, stane se to, že potlačíme jeho vnitřní dynamiku a neutralizujeme jeho kritický potenciál, neboť je zakomponujeme do schematického příběhu umění a učiníme z něj cosi minulého. Avšak softwarové dílo dokáže vzhledem ke své flexibilitě a otevřenosti absorbovat nejrůznější rozpory, které charakterizují dnešní dobu, zviditelnit je a uvést do vzájemných vztahů, což do značné míry odpovídá záměrům dialektického materialismu. K tomu využívá techniku montáže, která byla zdiskreditována mainstreamovým softwarem.⁷²

69

Ibid.

70

LEV MANOVICH, „Avant-Garde as Software“, in: Stephen KOVATS (ed.), *Media Revolutions*, Frankfurt: Campus 1999, s. 48–52.

71

COX, *Antithesis*.

72

Cox tímto myslí, že v současné době je technika montáže běžnou funkcí implementovanou do všeobecně rozšířených grafických programů, a je tedy přístupná každému, což vedlo k jejímu nadměrnému využívání v digitálním prostředí a došlo k postupné banalizaci této techniky, která dle Coxe měla v rukou umělecké avantgardy kritický a revoluční potenciál.

V případě softwarového umění však získává tato technika nový rozměr, neboť software je schopen simulovat, transformovat a vzájemně propojovat nejrůznější události, procesy či jevy takřka libovolným způsobem. Cox dodává, že pokud si přivlastníme již existující materiály a aplikujeme princip montáže na historii, můžeme porozumět její konstrukci.⁷³ Dochází tedy k závěru, že softwarové umění je ideálním prostředkem pro uplatnění dialektického přístupu ke zkoumání skutečnosti, historie, ideologie i umění.

Další významná vlastnost softwaru, díky níž se může softwarové umění stát radikálním uměleckým nástrojem, spočívá v jeho schopnosti odhalit ideologické principy svého fungování, které nejsou spjaty pouze se sférou softwaru, ale vztahují se také k politice, umění a dalším oblastem našeho života. Cox uvádí příklad počítačového viru, který v sobě skrývá kritický potenciál:

Zdá se, že kulturní forma viru ztělesňuje principy auto-destruktivního umění a negace. Virus manifestuje sebedestruktivní aktivity programu, který se může jednoduše šířit a ovlivňovat ostatní programy, a proto reflektuje strukturální vlastnosti počítače a internetu, na nichž závisí jeho konání.⁷⁴

Zde vidíme, že počítačový virus může obnažit ideologii a fungování systémů, bez nichž si dnes už náš život neumíme představit. Tímto způsobem může softwarové umění vystoupit z historie umění, v níž hraje pasivní roli, a stát se aktivním tvůrcem přítomnosti.

Coxova koncepce je podnětná zejména v důrazu na politický a edukační rozměr softwarového umění, neboť softwarové umění skrze podvratnost svých strategií (hravé parodie a apropriace mainstreamového softwaru, tvorba neužitečného softwaru, experimenty s destruktivními programy nebo odkrývání kódu, který leží za grafickým rozhraním) i jednotlivých děl usiluje o reflexi současnosti a soustředí se na definování role softwaru tady a teď. Jedná se tedy o umění, které má určitý potenciál k tomu, aby budovalo společenské

uvědomění týkající se vlivu digitálních technologií na naše životy a lidskou civilizaci.

Coxův náhled na umělecký software jako na novou revoluční sílu, která dokáže změnit stávající společenské struktury vycházející z dialekticko-materialistického pojetí dějin, však vyznívá až příliš nadšeně a nekriticky, neboť oživuje avantgardní optimismus založený na víře, že umění dokáže změnit společnost, aniž by bral v úvahu spíše minoritní charakter softwarového umění, pevnou zakotvenost softwaru v současných mocenských strukturách a opakovaná selhání uměleckých avantgard při realizaci svého snu o přeměně společnosti za pomoci umění.

Rázné odmítnutí historického uchopení softwarové umění, které Cox rozvíjí ve svém textu, se jeví jako až příliš radikální gesto, neboť psaní dějin tohoto uměleckého směru v sobě skýtá několik podnětných příležitostí: revizi a doplnění historického vývoje digitálního umění jako celku, sledování vzájemného ovlivňování softwaru a umělecké tvorby od šedesátých let do současnosti, a především možnost hlubšího pochopení fenoménu softwaru prostřednictvím mapování jeho historických proměn a specifických podob, kterých nabýval v uměleckém kontextu. Navíc zařazení softwarového umění do historického rámce nemusí automaticky znamenat oslabení jeho politického a edukačního potenciálu, naopak může napomoci podrobněji definovat vazby s jinými uměleckými směry, které sledují podobné cíle (politicky zaměřené konceptuální umění, destruktivní umění), a poskytnout softwarovým umělcům nové podněty či strategie.

SOFTWAREVÉ UMĚNÍ: SPLÉTÁNÍ HISTORIÍ

Rozmanité pohledy na historii softwarového umění svědčí o tom, že obraz tohoto uměleckého směru není zdaleka tak jednotný, jak by se mohlo na první pohled jevit, a podobá se spíše rozbitému zrcadlu, v jehož střepech se odrážejí nejrůznější umělecké a kulturní praxe, které ovlivnily podobu a směřování softwarového umění. Softwarové umění tedy představuje komplexní fenomén, v němž se splétají různé historie, které se dotýkají uměleckých, politických, sociálních, technologických a kulturních oblastí.

Softwarové umění lze nahlížet z odlišných perspektiv: jako rozvinutí tendencí konceptuálního umění v technologické sféře, pokus o uměleckou reflexi pravidel a instrukcí využívaných při tvorbě literárních, hudebních a dalších uměleckých děl, pokračování destruktivních tendencí v umění ve formě

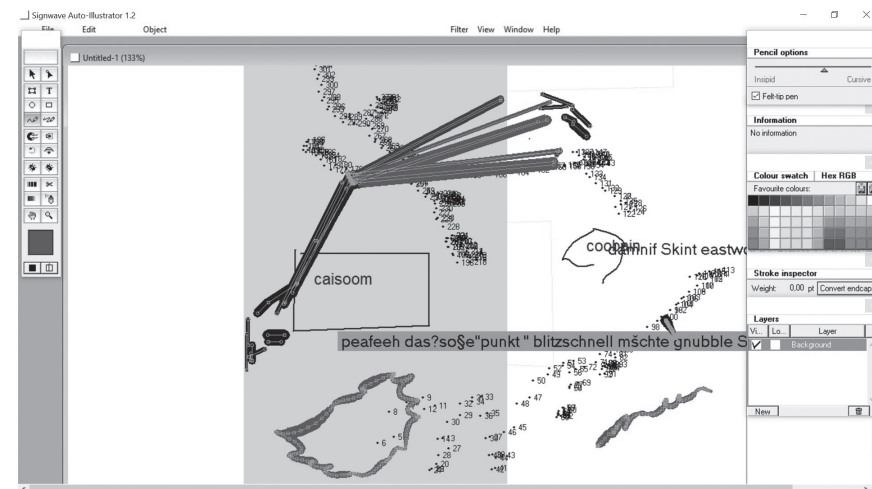
⁷³

Ibid.

⁷⁴

„The cultural form of a virus appears to embody the principles of auto-destructive art and negation. A virus describes the self-reproducing activities of a program that can simply spread and effect other programs, and thereby reflects the structural properties of the computer and the Internet it operates through.“ *Ibid.*, s. 53.

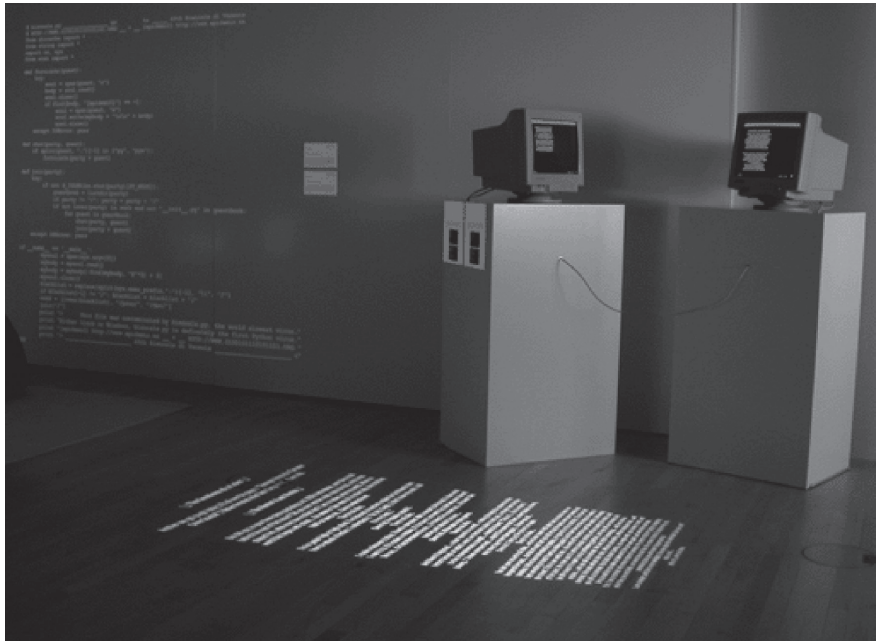
počítačového viru, nebo jako aktivistickou praxi, která přináší jedinečnou možnost reflektovat povahu současné kultury. Všechna tato vlákna je třeba sledovat, abychom mohli uchopit softwarové umění v celé jeho šíři, bohatosti a především proměnlivosti, která je jeho hlavní devizou a činí z něj uměleckou praxi, jejíž potenciál – vzhledem ke konstantnímu technologickému vývoji, jehož zásluhou software mutuje do neustále nových a překvapivých podob – ještě zdaleka nebyl vyčerpán.



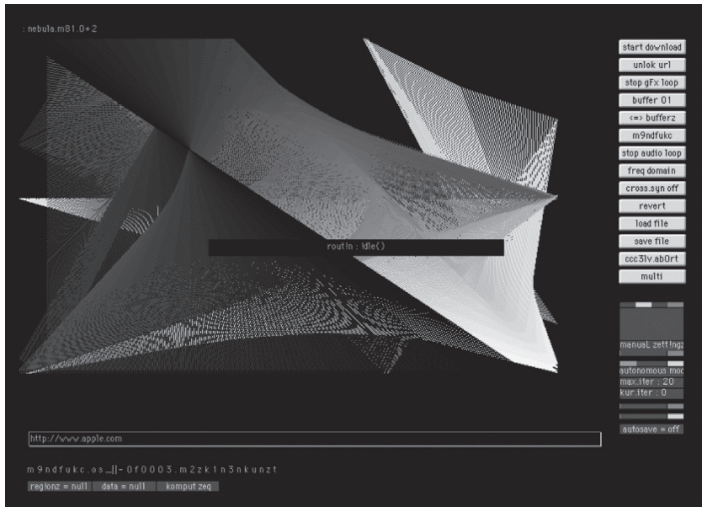
Adrian WARD, *Auto-Illustrator*, 2001, screenshot autora článku



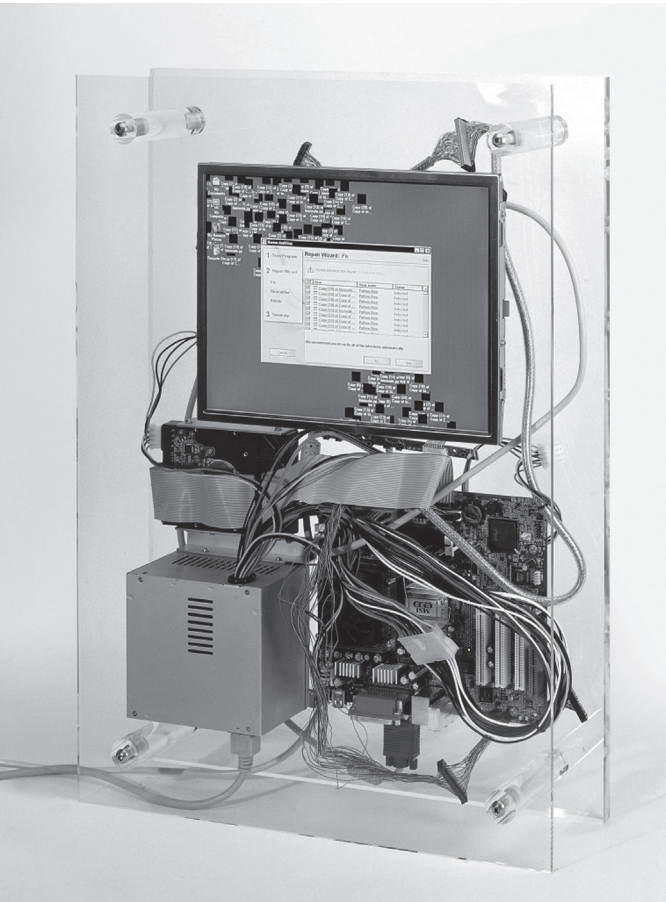
JODI, *Jet Set Willy Variations*, 2002, zdroj: <http://jetsetwilly.jodi.org/> (cit. 9.10.2018)



Sneha SOLANKI, *The Lovers*, Post & Tele Museum, Kodaň, 2004 , zdroj: <http://electronicartist.net/the-lovers/digitalcraft.html> (cit. 9.10.2018)



Netochka NEZVANOV, *Nebula.m81*, 1999, zdroj: <https://transmediale.de/content/netochka-nezvanova-0> (cit. 9.10.2018)



Eva MATTES – Franco MATTES – *EpidemiC*, *Bienalle.py*, 2001, zdroj: <https://0100101110101101.org/biennale-py/> (cit. 9.10.2018)