TIM EZIN
Vol. 3, No. 1–2
2012–2013

Umění živého kódování

Autor textu: Ivan Floreš

Anotace: Cílem tohoto článku je představit unikátní způsob tvorby počítačové hudby, který se nazývá živé kódování. V článku se budu zabývat proměnou živých hudebních vystoupení v momentu, kdy v nich figuruje počítač jako hudební nástroj. Budu se snažit popsat zásadní prvky vystoupení, což jsou počítač, programovací jazyk a videoprojekce, která je funkční složkou, jež divákovi odhaluje zdrojový kód a kódování v reálném čase. Zobrazení kódu zdůrazňuje autentičnost, tedy to, že kompozice vzniká „teď a tady". V článku budu vycházet především ze studií autorů Andrewa Browna, Andrewa Sorensena, Nicholase Collinse a Alexe McLeana, kteří mají praktickou zkušenost s živým kódováním a zároveň vnímají počítačovou hudbu jako oblast výzkumu.

Abstract: The aim of this paper is to introduce a unique way of making computer music which is called the live coding. In this article, I will discuss transformation of musical performances from programmable synthesizers to the programmed units. The practice of live coding is also connected to the issue of authenticity. I will try to describe crucial features of this kind of performance which are computer, programming language and the video projection, which is functional component that shows audience the source of music composition and programming in real time. Screening the code accent authenticity. This paper is mainly based on Andrew Brown, Andrew Sorensen, Nicholas Collins and Alex McLean researches. They all have hands-on experience with live coding and at the same time, they perceive computer music as a part of the research.

Klíčová slova: živé kódování, počítačová hudba, algoritmus, programování, hackování

Keywords: live coding, computer music, algorithm, programming, hacking

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606

Úvod

Následující text pojednává o specifické tvorbě hudebních kompozic pomocí počítače. Jedná se o způsob improvizace pomocí programovacích jazyků, přičemž hudba vzniká na základě vytváření a následné úpravy zdrojového kódu. Pro tento proces se také používají anglické označení on-the-fly nebo just-in-time programování. Tato označení vycházejí z vlastnosti programovacích jazyků, které jsou schopny překládat kód do nižšího programovacího jazyka v reálném čase. Velmi důležitý je fakt, že se nejedná o předem připravenou kompozici, ale naopak je tato tvorba součástí živého vystoupení a kompozice vzniká spontánně. Jedná se tedy o konstruktivní proces srovnatelný například s free-style rappem v hip-hopu (Mynarz, 2010) nebo s jakoukoliv jinou improvizací, kdy není dopředu známo, jakou podobu bude mít konkrétní vystoupení. Stejně jako rapper musí efektivně využívat slovní zásobu a výrazně artikulovat, programátor – hudebník musí efektivně popsat generativní proces. Předpokladem pro takovou činnost je především stručné a rychlé psaní kódu, jenž musí být výpočetně efektivní a srozumitelný pro pozdější úpravy.

Podstatou živého kódování je současné vytváření nástroje a hudební kompozice a následné promítání celého procesu na plátno. Podle tvůrců, kteří pracují s živým kódováním, je videoprojekce spíše funkční než estetickou složkou vystoupení. Z pohledu umění nových médií lze na projekci nahlížet jako na taktiku „anti–VJingu", kde se zobrazování desktopového prostředí s odhaleným zdrojovým kódem vymezuje vůči imerzivnímu světu klubové scény.

Z pohledu softwarového umění je živé kódování fúzí tří úrovní, které popsala Inke Arns (2005) jako:

1) zdrojový kód,

2) algoritmus,

3) výsledek (obraz, zvuk), který je generován algoritmem.

Propojení zmíněných vrstev uvádí protagonistu živého kódování do vyvážené role programátora a hudebníka zároveň.

Algoritmy jsou všude

V 70. letech, kdy byly vyvinuty první mikroprocesory, se staly počítače díky své velikosti přenosné. Tím se proces živého kódování stal realizovatelný i mimo výzkumné zvukové laboratoře a mohl jej provozovat téměř kdokoliv, kdo měl přístup k osobnímu počítači. Podobné principy však můžeme hledat i v dřívějších experimentech bez použití elektronických zařízení. Příkladem může být *Composition 1961 No. I, January I* (1961) vytvořená La Monte Youngem. Dále projekt s názvem *An Instructional Game for 1 to many musicians* (1975), který vytvořil Click Nilson, nebo kompozice *The Great learning* (1968–1970), jejímž autorem je experimentální hudebník Cornelius Cardew.

Osmdesátá léta 20. století jsou dalším obdobím, kdy se odehrály důležité momenty ve formování vystoupení v podobě živého kódování. Toto období se dá nazvat **Éra jazyku *FORTH***. Rozšířil se programovací jazyk HMSL neboli Hierarchical Music Specification Language, což byl na Forth jazyce založený objektově orientovaný hudební jazyk s nahrávacím MIDI vstupem a výstupem. (Historical performances, 2011) V roce 1985 se odehrálo první plnohodnotné vystoupení živého kódování, jehož protagonistou byl Ron Kuivila.

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606

Vystoupení se konalo v Amsterdamu v rámci festivalu pořádaného organizací STEIM [1]. V druhé polovině 80. let se objevuje skupina The Hub, která využívala právě jazyk *FORTH*. Na začátku 90. let se na scéně objevilo grafické programovací prostředí, jako je například *Max* nebo *Pure data*, kde se algoritmy schovávají za grafická rozhraní.

Rok 2000 se stal pro živé kódování a celkově pro počítačovou hudbu zlomovým. Počítačové technologie již byly levnější, a tedy i dostupnější. Vyšší takty procesorů umožňovaly rozmanitější práci s digitálním zvukem v reálném čase. V červnu roku 2000 založili Adrian Ward a Alex McLean skupinu *The Slub*. Zpočátku využívali vlastní generativní software, přičemž na plátno promítali obsah pracovní plochy počítače, což umožnilo divákům sledovat tvůrčí proces. Od roku 2005 se skupina *The Slub* zaměřuje výhradně na inscenace živého kódování, v nichž i nadále využívá vlastnoručně napsané programovací jazyky. Užívání videoprojekcí se v průběhu času stalo jedním ze základních charakteristických prvků živého kódování. Díky fenoménu videoprojekcí můžeme nazývat období nového tisíciletí jako ***Projekční éru***. S rozvojem internetu se informace množily daleko rychleji než doposud a rozmach informačních dálnic se odrazil téměř ve všech odvětvích kultury a umění v podobě mixování stylů a aplikování nových postupů. Hudební programování nebylo výjimkou.

Díky novým komunikačním kanálům šel vývoj programovacích jazyků velmi rychle dopředu a lze jen ztěžka určit, který počin byl zásadnější. Nejvýraznějším prvkem byl však *Live coding jam*, který proběhl v roce 2004 v Dánsku. Zde se totiž propojily všechny formy programování hudby v reálném čase s využitím různých jazyků včetně *Max/MSP* a *Pure data*. Na základě této akce se zformovala organizace TOPLAP, sdružující nadšence pro živé kódování.

Kdo píše (kód), ten hraje

Jádrem živého kódování je dynamické programovací prostředí, jež umožňuje interaktivní psaní s překladem v reálném čase: on the fly – za chodu, za letu. Toto prostředí, založené na programovacím jazyku *LISP*, je známé od 60. let 20. století. Populární jsou modifikace prostředí jako *SuperCollider*, *ChucK* nebo *Impromptu*. Ve srovnání s ostatní elektronickou hudbou, tvořenou pomocí počítače, vyžaduje živé kódování vysokou míru programátorských dovedností a umělecké invence. Programovací jazyky živého kódování poskytují flexibilní prostředí pro rozvíjení hudebníkových intelektuálních, technických a kompozičních schopností, přičemž lze dosáhnout vysoké úrovně virtuosity (Brown – Sorensen, 2009). Kreativní část živého kódování vyžaduje především znalost hudební kompozice, pravidel improvizace, hudební nadání, stejně jako orientaci v počítačovém programování a počítačové vědě (Brown – Sorensen, 2009).

Všechny zmíněné prvky jsou velmi důležité, avšak v živém kódování má zásadní význam především aspekt improvizace. Živé kódování nelze vnímat jen jako hudební vystoupení, protože se odehrává na pomezí konceptuální performance. Z manifestu organizace TOPLAP lze vyčíst, že nejde o nástroje, ale o myšlenky v podobě algoritmů (*Manifesto Draft*, 2008). Právě při živé improvizaci se hudebník – programátor nachází ve velmi napjaté situaci, kdy musí věnovat pozornost kódu a zároveň se musí soustředit na hudební kompozici. Zde má velký potenciál kolaborativní kódování. Jeden z programátorů se zabývá rozšířením kódu a druhý upravuje již existující část, která je v přímém styku s hudebním výstupem. Lze tak předejít situacím, kdy se hudba zacyklí ve chvíli psaní složitější části kódu. Tim Perkis, jeden ze zakladatelů skupiny *The Hub*, takovou spolupráci označil doslova za „sochání“ díla (Collins, 2003).

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606

Živé kódování nespočívá pouze v lineárním psaní kódu, ale jak bylo již zmíněno, dochází zpětně k jeho úpravám. Podstata živého kódování tkví v tom, že se nejedná o pouhé aplikování algoritmů a následné mapování na parametry hudby. Zde je algoritmus konstruován a spuštěn ve své přirozené podobě programovacího jazyka, následně modifikován během představení (viz přílohu č. 1). Což také naznačuje, že výsledek generativního procesu není vždy předvídatelný. Kód se v průběhu představení mění, a proto není možné na konci vystoupení prezentovat finální produkt – program. Stejně jak lze kód rozšiřovat a upravovat, je možné jej i mazat. Jedna z možných variant zahájení a/nebo zakončení vystoupení je tedy prázdná obrazovka bez jediného řádku kódu. Pokud je to nutné, je možné průběh tvorby zaznamenat pomocí „logování" celého postupu. Jinými slovy lze veškeré informace o úpravách kódu zaznamenat do textového souboru. Obdobu „logu" lze nalézt ve většině programů s uživatelským grafickým rozhraním v podobě historie.

Živé kódování představuje přístup k počítači jako k novému hudebnímu nástroji, jenž nevytváří simulace klasických hudebních nástrojů jako například u VST [2], ale využívá možnosti digitální technologie. Nejedná se tedy o remediaci, ale o rozšiřování možností samotného média. S tím také souvisí nový způsob vytváření hudebních kompozic. Programátor zde zastává úlohu, ve které popisuje, jakým způsobem se má určitý jev (například posloupnost tónů) vytvořit. Tóny tedy nevytváří přímo, ale pouze pomocí popisu algoritmu. Andrew Brown tento vztah hudebníka k algoritmům vnímá jako intimní a bezprostřední kontrolu (Brown – Sorensen, 2009). Je důležité poukázat na skutečnost, že tento vztah se pojí výhradně s kompozicí. Pro lepší představu lze programátora přirovnat ke skladateli, který tvoří skladbu v reálném čase. Roli instrumentalisty pak zastává počítač: v reálném čase překládá kód a produkuje hudbu. Vliv lidské kontroly na jednotlivé tóny ve skladbě je naopak nepřímý. Jedná se o zprostředkování mezi umělcem a uměleckou formou. Tento proces by se dal v přeneseném významu chápat jako *metakreativita*, kterou popsal Michell Whitelaw (Whitelaw, 2006) Živé kódování totiž poukazuje k částečné autonomii počítače v kreativním procesu. Příkladem je využití generátoru „pseudonáhodných" čísel a jeho aplikace na hudební kompozici. Díky tomu se počítač stává spoluautorem díla.

Ukažte obrazovky!

V době, kdy byla hudba vytvářena výhradně akusticky, byl zřejmý fyzický zdroj zvuku. S nástupem počítačů jakožto hudebních nástrojů se však přímé propojení mezi gestem a výsledným tónem definitivně ztratilo (Schloss, 2002).

Tento jev lze pozorovat především u hudební produkce využívající počítač, kdy není zcela zřejmé, jakým způsobem umělec pracuje, jaká část je produkována na místě a co všechno je připraveno ve studiu. Manifest sdružení TOPLAP na danou situaci reaguje výzvou: *„Tmářství je nebezpečné. Ukažte nám vaše monitory."* (*Manifesto Draft*, 2008). Proto je celý proces kódování promítán na plátno, a divák tak má možnost vidět vznik celého kódu. To je moment, kdy počítačová hudba získává na autentičnosti (příklad podoby vystoupení viz přílohu č. 2). TOPLAP tak navazuje na filozofii hackerské subkultury a na jejich etický kodex, který se zmiňuje o svobodném přístupu k informacím a zdrojovému kódu. Vzniká tak neiluzivní prostředí, jež se neschovává za grafické uživatelské rozhraní či imerzivní obrazy jako například ve VJingu. Hackerský étos podporuje tvorbu umění pomocí počítačů s dodatkem, že není důležitý výsledek, ale samotný program, performativita kódu, může být sám o sobě krásný (Levy, 2010). Podobné vystavování procesu a zdrojového materiálu lze nalézt také u avantgardních filmových tvůrců,

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606

již přiznávají filmový materiál (zobrazení perforace nebo promítacích značek) a odhalují iluzi pohybu (záměrné blikání, smyčkové střídání perspektivy) [3].

S ohledem na publikum, které zatím není dostatečně obeznámeno s procesem programování, jak píše McLean, může nastat situace, kdy vnímá pozitivně snahu ukázat proces a stává se tak jeho součástí, nebo naopak vnímá kód jako změť nesmyslného textu a je tak vyloučeno z procesu performance (McLean, 2011). Podle Collinse totiž živé kódování a softwarové umění vůbec nebylo dostatečně zařazeno do procesu enkulturace (Collins, 2003) a jeho produkce má zatím své místo v klubové scéně.

Závěr

Živé kódování je digitální technologii vlastní způsob hudební produkce opouštějící využívání skeumorfického designu nástroje. Technická omezení však můžou negativně ovlivňovat proces tvorby hudby. Chyby syntaxe programovacího jazyka, neefektivní či příliš náročný algoritmus jsou častými nedostatky, jimž musí programátoři čelit. Mnohdy se však nejedná o záležitost živého kódování, ale o nezralost těch, co jej praktikují. Technické nedokonalosti ovšem nemusí být vždy na škodu. Například pro estetiku chyby, v hudbě obecně nazývánou *glitch music*, je toto nebezpečí v podstatě pozitivní a destruktivní přetváření kódu, či vkládání extrémních hodnot se stává součástí kompozice. Jak poznamenává Alex McLean: *„Někdy elegantní matematické výrazy znějí dobře, jindy nečekané chyby a chaos vyprodukují nejzajímavější výsledky.“* (McLean, 2004) [4]. Živé kódování, jemuž jsem se ve svém článku věnoval, rozpracovává a rozvíjí možnosti počítače jako kreativního média pracujícího v reálném čase a umělci, programátoři, hudebníci s ním pracují způsobem, jaký je počítači vlastní, pomocí algoritmů. Je tedy spíše otázkou kulturního vývoje, kdy se počítačová gramotnost a obeznámenost se základy programovacích jazyků stane součástí širšího povědomí a společnost bude schopna ocenit kompoziční a programátorskou virtuozitu živého kódování (Monks, 2012). Vzhledem k tomu, že programovatelná média zasáhla velkou část euroamerické kultury, bylo by to příjemným překvapením.

Přílohy:

(1) Gibber #6: Resampling The Resamplings. *Vimeo* 2013 [online]. [cit. 2013-03-13]. Dostupné z: vimeo.com/61435762.

(2) Benoît and the Mandelbrots – Bal des Ardents. *Youtube* 2011 [online]. [cit. 2013-03-13] Dostupné z: http://youtu.be/jlx6KNo5Eok.

(3) Outer Space – Peter Tscherkassky. *Youtube* 2011.[online]. [cit. 2013-03-27] Dostupné z: http://youtu.be/yASwqIWjaVI.

Poznámky :

[1] STEIM je centrem pro výzkum a vývoj nástrojů elektronického umění. Viz http://www.steim.org.

[2] VST je zkratka Virtual Studio Technology. Jedná se o modul pro komerční DAW systémy. Jsou to především virtuální nástroje a zvukové efekty. Mnohdy zde najdeme kopie původních analogových syntetizátorů.
Viz http://en.wikipedia.org/wiki/Virtual_Studio_Technology.

[3] Viz přílohu č. 3.

[4] *„Sometimes elegant mathematical expressions sounded great, sometimes unexpected errors and chaos produce interesting results.“* Překlad do češtiny: Ivan Floreš

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606

Použitá literatura:

ARNS, Inke, 2005. 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 (Hg.), *Medien Kunst Netz 2: Thematische Schwerpunkte*. Wien/New York: Springer. ISBN: 3211238719, s. 177-193 (dt.), s. 197-207 (engl.).

BROWN, Andrew a Andrew SORENSEN, 2009. Interacting with generative music through live coding. *Contemporary Music review*, roč. 28, č. 1, s. 17–29.

COLLINS, Nick, 2003. Generative Music and Laptop Performance. *Contemporary Music Review* [online], roč. 22, č. 4, s. 67–79. [cit. 2013-03-10]. ISSN 0749-4467. DOI: 10.1080/0749446032000156919. Dostupné z: http://www.tandfonline.com/doi/abs/10.1080/0749446032000156919.

COLLINS, Nick, Alex MCLEAN, Julian ROHRHUBER a Adrian WARD, 2003. Live coding in laptop performance. *Organised Sound*, roč. 8, č. 03, ISSN 1355-7718. DOI: 10.1017/s135577180300030x. Dostupné z: http://www.journals.cambridge.org/abstract_S135577180300030X.

Historical performances, 2011. *Toplap* [online]. [cit. 2013-03-11]. Dostupné z: http://toplap.org/wiki/HistoricalPerformances.

LEVY, Steven, 2010. *Hackers*. 1st ed. Sebastopol, CA: O'Reilly Media. ISBN 1449388396.

Manifesto Draft, 2008. *Toplap* [online]. [cit. 2013-03-11]. Dostupné z: http://toplap.org/wiki/ManifestoDraft.

MCLEAN, Alex, 2004. Hacking Perl in Nightclubs.. *Perl* [online]. [cit. 2013-03-10]. Dostupné z: http://www.perl.com/pub/2004/08/31/livecode.html.

MCLEAN, Alex, 2011. *Artist-Programmers and Programming Languages for the Arts*. London. Disertční práce. University of London.

MONKS, Keiron, 2012. Rozhovor: Děti by se měly už v šesti letech učit HTML jazyk. *Metro* [online]. [cit. 2013-03-10]. Dostupné z: http://www.metro.cz/rozhovor-deti-by-se-mely-uz-v-sesti-letech-ucit-html-jazyk-p34-/metro-extra.aspx?c=A120315_135337_metro-extra_rab.

MYNARZ, Jindřich, 2010. *Live coding* [online]. Praha: Univerzita Karlova, Filozofická fakulta. [cit. 2013-03-11]. Dostupné z: http://www.slideshare.net/stunome/live-coding.

SCHLOSS, Andrew, W, 2002. Using Contemporary Technology in Live Performance: The Dilemma of the Performer. *Journal of New Music Research*. roč. 31, č. 1. DOI: 0929-8215/02/3101-001.

WHITELAW, Mitchell, 2006. *Metacreation: art and artificial life*. Cambridge, Mass: MIT. ISBN 02-627-3176-2.

TIM ezin je internetový recenzovaný odborný časopis zaměřený na interdisciplinární výzkum a teoretickou reflexi diskurzu, praxe a procesů spjatých s novými médii. Vydává jej každoročně v květnu jako dvojčíslo obor Teorie interaktivních médií při Ústavu hudební vědy Filozofické fakulty Masarykovy univerzity.
www.timezine.cz
ISSN 1805-2606