

Disertační práce

Kreativní kódování a generativní algoritmy jako nástroje pro tvorbu interaktivního vizuálního umění

Creative coding and generative algorithms as tools for creating interactive visual art

Autor: **MgA. Aliaksandra Laurova**

Studijní program: P8206 – Výtvarná umění

Studijní obor: 8206V102 Multimédia a design

Školitel: doc. akad. mal. Michal Zeman

Oponenti: doc. PhDr. Miroslav Zelinský, CSc.
Prof. PhDr. Eva Kapsová, CSc.

Zlín, prosinec 2022

© Aliaksanda Laurova

Vydala **Univerzita Tomáše Bati ve Zlíně** v edici **Disertační práce**.
Publikace byla vydána v roce 2022

Klíčová slova: kreativní kódování, vizualizace dat, generativní umění, výpočetní umění, programování, kódování, Processing, WebGL, Shader, JavaScript

Key words: creative coding, data visualization, generative art, computational art, programming, coding, Processing, WebGL, Shader, JavaScript

Práce je dostupná v Knihovně UTB ve Zlíně.

ABSTRAKT

Disertační práce se věnuje programování jako uměleckému nástroji a moderní formě sebevyjádření. Pokrývá téma kreativního kódování, generativních algoritmů, vizualizací dat, analyzuje historii a vývoj této formy umění. Součástí disertační práce jsou ukázky autorských originálních projektů včetně návodu na aplikaci metod kreativního kódování v oblasti vizualizace dat, jehož cílem je dokázat, že kreativní kódování je dostupný nástroj, který může obohatit praxi každého umělce a designéra.

Klíčová slova: kreativní kódování, vizualizace dat, generativní umění, výpočetní umění, programování, kódování, Processing, WebGL, Shader, JavaScript.

ABSTRACT

The dissertation is devoted to programming as an artistic tool and a modern form of self-expression. It covers the topic of creative coding, generative algorithms, data visualization, analyzes the history and development of this art form. The dissertation includes samples of the author's original projects, including instructions for the application of creative coding methods in the field of data visualization, the aim of which is to prove that creative coding is an available tool that can enrich the practice of every artist and designer.

Keywords: creative coding, data visualization, generative art, computational art, programming, coding, Processing, WebGL, Shader, JavaScript.

Ráda bych poděkovala lidem, kterým vděčím za možnost studovat, pracovat a experimentovat a bez kterých by tato práce nevznikla. Děkuji doc. akad. mal. Michalu Zemanovi za veškerou pomoc a vedení během let studia a možnost pracovat v mém vlastním tempu. MgA. Bohuslavu Stránskému, Ph.D. za povzbuzení při prvních krůčcích do světa kreativního kódování. A samozřejmě děkuji doc. PhDr. Miroslavu Zelinskému, CSc. za podporu, motivaci a nekonečnou inspiraci.

OBSAH

0.1	Východiska práce	8
ÚVOD		11
1	CÍLE A METODY DISERTAČNÍ PRÁCE	13
1.1	Cíl	13
1.2	Metodika práce	15
1.3	Spolupráce	17
1.4	Výstupy disertační práce	18
1.5	Přínos práce pro vědu a praxi	20
2	KONCEPT KREATIVNÍHO KÓDOVÁNÍ	22
2.1	Studium zdrojů	22
2.2	Definování pojmů	25
2.3	Kódování versus programování	31
2.4	Kdo je kreativní kodér?	32
2.5	PRŮKOPNÍCI A PRŮKOPNICKÉ TECHNOLOGIE GENERATIVNÍHO UMĚNÍ	33
2.6	Od generativního umění ke kreativnímu kódování	40
	2.6.1 Ben Fry	44
	2.6.2 Casey Reas	47
2.7	Kreativní kódování v moderní době	49
3	NÁSTROJE A PROGRAMOVACÍ JAZYKY KREATIVNÍHO KÓDOVÁNÍ	60
3.1	JavaScript	61
3.2	Processing	63
3.3	p5.js	66
3.4	WebGL	66
3.5	GLSL Shaders	68
3.6	Python	68
3.7	Vizuální programovací jazyky	70
3.8	Jak začít s kreativním kódováním?	71
4	PROCES VIZUALIZACE DAT V KONTEXTU KREATIVNÍHO KÓDOVÁNÍ	74
4.1	Shromažďování dat	75
4.2	Ukládání, organizace a transformace dat	77
4.3	Generativní algoritmy	82

4.4	Import dat	85
4.5	Přidání principu interaktivity	87
4.6	Export vizualizací	88
5	PRAKTICKÁ REALIZACE	89
5.1	Realizované projekty kreativního kódování	89
5.1.1	Vizualizace dat jako výzva estetiky	89
5.1.2	See the wind	95
5.1.3	Everything flows, nothing remains	98
5.1.4	Symposium Art Colony Cered: Zvukový deník	100
5.1.5	Under	102
5.1.6	Bright matter, Superčástice	104
5.1.7	Pánev, kyčle	106
5.1.8	The stars of Cered	107
5.2	Výstava " Undercurrent"	109
5.3	Online portál	112
	ZÁVĚR	115
	SEZNAM POUŽITÉ LITERATURY	116
	SEZNAM OBRÁZKŮ	127
	SEZNAM POUŽITÝCH ZKRATEK	130
	TVŮRČÍ ČINNOST 2017–2022	131
	CURRICULUM VITAE	133

0.1 Východiska práce

“There is an art to science, and a science in art; the two are not enemies, but different aspects of the whole.”

Isaac Asimov

Neobvyklé magické věci se mohou stát, když je člověk otevřený novým příležitostem, nástrojům a technikám, které nám svět se svým pokrokem nabízí, když se umělci nebojí vykročit ze své komfortní zóny a přes sérii neúspěchů a úspěchů vytvoří něco mimořádného, inspirují ostatní. Jak často mluvíme o umění a vědě jako o dvou diametrálně odlišných záležitostech, praktikách, které spolu nesouvisí? Umělec je často prezentován jako závislý na emocích a intuici, bez místa pro rozum a pečlivé plánování. Vědec je tvor chladného světa logiky, kde se nebere ohled na představivost. Podle Isaaca Asimova (1983, s. 116), spisovatele a profesora biochemie, vede tato praxe odpojování k falešným představám. Skutečný umělec spojuje představivost a racionalitu, má plán, kroky, kterými musí projít, ví, co dělá; pokud ne, jeho umění trpí. Skutečný vědec musí být nejen racionální, ale i nápaditý, samotná představivost může někdy vést k inovativním řešením, zatímco rozum vede mnohem pomalejší a delší cestou; pokud je nepřítomen, jeho věda trpí.

Žijeme v době technologického pokroku, který utváří náš pohled na svět. Technologický pokrok vede ke zvýšení produktivity práce v nejrůznějších kategoriích, jako jsou sociální, zdravotní, fyzické, finanční atd. Estetické a intelektuální aktivity by neměly zaostávat a využívat navrhované nástroje pokroku k vytvoření něčeho výjimečného. Jak řekl anglický filozof a státník Sir Francis Bacon (1863, s. 127)¹: *umění a vědy by měly být jako doly, kde je ze všech stran slyšet hluk nové práce a dalších úspěchů.*

Jak bylo uvedeno na briefingu Evropského parlamentu o technologii a umění (2019, s. 1), v průběhu dějin lidstva, s příchodem vynálezů, se je lidé snažili začlenit do různých způsobů sebevyjádření. Od nejstarších známých jeskynních maleb, které byly vytvořeny přibližně před 64 000 lety (červené negativní otisky rukou v jeskyni Maltravieso, Cáceres, Španělsko), přes historii mnoha hudebních nástrojů, až po výrobu moderních foto/video kamer, mikrofonů, záznamových zařízení a samozřejmě k výrobě počítačů. Každý krok technologického pokroku inspiroval a umožnil nové formy umění. Bez tohoto rozvoje by nebylo možné

¹ Latinský originál: *"In artibus et scientiis, tanquam in metalli fodinis, omnia novis operibus et ulterioribus progressibus circumstrepere debent"*, *Novum Organum*, *Knihla 1, XC, sebráno v The Works of Francis Bacon (1826), sv. 8, str. 50-51. Přeložili James Spedding a Robert Leslie Ellis v The Works of Francis Bacon (1863), str. 127.*

produkovat obrazy, textile, fotografie, kino a moderní digitální umělecká díla, která plní muzea a galerie po celém světě.

S každým novým dnem a novým krokem v technologické a vědecké evoluci vzniká mnoho nových neobvyklých uměleckých forem a nástrojů. S rozvojem elektrokomunikací v 60. letech se různé stroje staly dostupnými pro veřejnost, v knihovnách, na univerzitách, na poštách atd. Stroje, původně vynalezené pro podnikání, využívali nápadití tvůrci jako další nástroje umění. Krátce poté, co byly vyrobeny první kopírovací stroje, se objevilo xeroxové umění (xerografie nebo copy art), které zahrnuje použití kopírky k vytvoření nového uměleckého díla. Jeho historie začíná koncem 60. let, kdy několik umělců, Wallace Berman, Esta Nesbitt, Andy Warhol a další začali používat kopírovací stroje k vytváření zkradených obrázků různých předmětů, částí vlastního těla, kreseb, fotografií atd. (*Xerox Art*, 2015). Další svérázná forma umění – fax art (telekomunikační umění nebo telematické umění) – se objevila s rozšířením faxů. 12. ledna 1985 se Joseph Beuys, Andy Warhol a japonský umělec Kaii Higashiyama zúčastnili projektu „Global-Art-Fusion“. Od jednoho umělce k druhému byl zaslán fax a každý přidal ke společnému dílu vlastní kresbu. Trvalo 32 minut, než faxové umělecké dílo procestovalo celým světem – z Düsseldorfu (Německo) přes New York (USA) do Tokia (Japonsko), a bylo přijato ve vídeňském Lichtenštejnském muzeu moderního umění. Tento fax byl prohlášením o míru během studené války v 80. letech (Chahil, 2015).

Celá další kapitola jsou samozřejmě počítače. Když se počítače staly přístupnými veřejnosti, rychle se proměnily v jiný umělecký nástroj, jako je barva a štětec pro malíře nebo hlína pro sochaře. Od 60. let a dále byli umělci vybaveni novým fascinujícím nástrojem s neměřitelným potenciálem. Počítače nám nabízejí nejen efektivnější způsoby práce, ale zásadně změnily naše vnímání společnosti, umění a světa obecně. Následujících několik desetiletí nám přineslo mnoho uměleckých stylů, hnutí a technik. Některé z nich jsou digitální design, umění ASCII, telematické umění, umění nových médií, systémové umění, softwarové umění, umění umělé inteligence, umění algoritmů, fraktální umění, interaktivní umění, vizualizace dat, generativní umění, kreativní kódování atd. Tento seznam může pokračovat dál a dál, protože technologie nezůstávají v klidu, elektrokomunikace se neustále vyvíjejí a přináší nám každým dnem více a více možností tvoření. Technologický pokrok vede k novým logickým systémům, výkonnějším nástrojům, které bude používat nová generace kreativních lidí. Stále se budou objevovat nové styly umění, limity jsou pouze v naší fantazii.

Pouto mezi technologiemi, vědou a uměním je nepochybně silné. Nové nástroje nám umožňují tvořit svobodněji. Důležitá je ale i odvrácená strana této závislosti. Umění pohání inovace a vytváří významnou poptávku po technologiích. Při práci s různými materiály umělci často vyvíjejí nové techniky a posouvají hranice

představivosti, čímž otevírají nové směry technologického rozvoje. Širší umělecké aktivity – od uchovávání a restaurování starověkých děl až po vytváření komplikovaných scén virtuální reality s realistickým prostředím – také generují významnou poptávku po inovacích. Poptávka po technologiích je patrná i ve spotřebě umění, zejména v podobě audiovizuální techniky a obsahu (*Technology and the arts: Past, present and future synergies*, 2019, s. 1).

Technologie jsou tu dnes, dostupnější pro širokou veřejnost a umělce než kdy jindy. Toto těsné spojení mezi uměním, technologiemi a vědou je jasně viditelné v několika podkategoriích umění nových médií, respektive v kreativním kódování a generativním umění. Tyto postupy, jejich historie, vývoj a možnosti jsou hlavní náplní této disertační práce.

ÚVOD

Žijeme v éře digitálních technologií, kde se k vytváření nových uměleckých směrů, stylů a technik používají četné gadgety, počítače a softwarové programy. Jednou z praktik, která si neustále získává na popularitě, je kreativní kódování – způsob využití programování k sebevyjádření. Namísto spoléhání se na existující software, jako jsou Adobe Illustrator, Photoshop, 3ds Max, Blender atd., které mají sadu specifických nástrojů, příkazů a efektů, kreativní kodéři používají k vytváření svých uměleckých děl samotné programování bez jakýchkoli předem určených omezení.

Stále více umělců, architektů a designérů začíná vidět potenciál takového přístupu. Mnoho online platform, portálů a komunit slouží ke sdílení práce a učení se od ostatních. Jednou z největších výhod kreativního kódování je to, že většina programů je uložena online prostřednictvím open source, což znamená, že kdokoli může získat přístup ke kódu, stáhnout si ho, zkopírovat, změnit, použít jako základ pro vytváření něčeho jiného a navázat spolupráci mezi kodéry z celého světa.

Nicméně, když lidé, kteří nejsou obeznámeni s těmito technikami, slyší o kreativním kódování, mohou předpokládat, že je zvládnou pouze opravdoví programátoři, že je to něco, co patří spíše do světa informatiky než výtvarného umění. Ano, kreativní kódování je stále založeno na programování, ale sada různých programovacích jazyků, jako je Processing, byla vytvořena speciálně pro umělecké účely, kde jsou příkazy a instrukce zjednodušené a lze si stáhnout četné knihovny, které nahradí velké části kódu. Kreativní kodér může stejně jako tradiční umělec začít program „skicovat“ bez konkrétního plánu, může přidávat různé objekty, barvy a pohyby, aniž by sám věděl, jaký bude výsledný tvar. Výsledky často nejsou takové, jaké bychom očekávali. Změnou pouze jednoho čísla v kódu můžeme změnit celý vzhled uměleckého projektu. To je také důvod, proč je kreativní kódování tak vzrušující.

Je tradicí moderní společnosti rozdělit tok myšlenek na dva protichůdné proudy: mechanický a organický. Předpokládá se, že matematika a programování patří pouze do studeného světa logiky a přesnosti, zatímco v oblasti umění vládnu emoce a sentimenty. Cílem generativního umění a kreativního kódování je kombinovat tyto pojmy, používat kód a algoritmy jako další umělecký nástroj, stejně jako tužku nebo akvarel. Je fascinující, jak lze několik řetězců kódu převést na živý, pohyblivý obraz, jak tento obraz může pomocí kamer a senzorů interagovat s jakýmkoli fyzickým objektem v místnosti, jak se může měnit a vyvíjet. Umělecká díla vytvořená těmito technikami ukazují krásu kódu a širokou škálu možností, které umělcům přináší, obrátí pozornost lidí na vizuální

programování – rychle se vyvíjející nástroj v uměleckém výkonu, na výrobu elektronické hudby, kulturu VJ a interaktivní instalace (Greenberg, s. 19-20).

Kreativní kódování je každým dnem stále populárnější a stále více umělců hledá inspiraci v této nové formě médií. Disciplína je však stále velmi nová, není mnoho studií, které zkoumají možnosti kreativního kódování ve světě vizuálních médií a propagují je jako další umělecký nástroj, který by mohl použít každý umělec.

Existují praktické kurzy, online lekce a tutoriály, které učí principy této disciplíny. Univerzity a vysoké školy však zaostávají, zejména v České republice. V době psaní této disertační práce zajišťovala kurz kreativního kódování pouze jedna instituce – Vysoké učení technické v Brně. Kurz je určen pro studijní program Architektura a urbanismus a zaměřuje se na kódovací techniky, které pomáhají architektům při tvorbě jejich návrhů. Jeden kurz nestačí. Vzdělávací instituce by měly sledovat pokrok a poskytovat svým studentům nejmodernější nástroje tvorby. Kreativní kódování je disciplína zaměřená nejen na programátory a matematiky, uspět v ní může každý, a i s nulovou znalostí programovacích jazyků, těžit z jejího vyzkoušení může každý umělec, designér, architekt. Kreativní kódování může otevřít dveře do světa interaktivních médií každému, kdo je připraven se učit.

Pokud není uvedeno jinak, překlad A.L.

1 CÍLE A METODY DISERTAČNÍ PRÁCE

Tato disertační práce ukazuje možnosti kreativního kódování v oblasti vizuálních médií. Obsah práce spojuje programování, matematiku a výtvarné umění. Kombinace těchto disciplín přinese nové perspektivy do vývoje multimediální kultury. Rozvoj digitálních technologií přináší obrovský rozsah možností pro výzkum a tvorbu unikátních projektů. Spousta programovacích jazyků a platforem, stejně jako hardwarových nástrojů je k dispozici pro použití, nicméně akademická studia předmětu téměř neexistují (nemáme na mysli příručky o tom, jak se naučit programovat). Jedním z cílů disertační práce je propagovat kreativní kódování mezi kolegy a studenty univerzity a dalších organizací, ukázat krásu kódu na praktických příkladech, ukázat, že programování také patří do světa umění. Kreativní kódování je široký koncept, který kombinuje generativní umění, interaktivní design, vizualizaci informací, zvukovou vizualizaci, interaktivní instalace a představení. Mnoho moderních umělců může prosperovat začleněním kreativního kódování do své praxe. Proto je důležité podporovat tuto disciplínu mezi ostatními umělci, kolegy a studenty.

1.1 Cíl

Hlavním cílem disertační práce je prozkoumat možnosti kreativního kódování v rámci vizuálního interaktivního umění, najít nejefektivnější přístupy ke kreativnímu kódování, aplikovat je v praxi a propagovat výsledek mezi ostatními umělci. Výběr předmětu vychází z osobních, akademických a pracovních zkušeností autorky v programování, matematice, informatice, digitálním designu a výtvarném umění. Spojení těchto různých disciplín v rámci jednoho projektu by mělo otevřít nové perspektivy pro rozvoj multimediální kultury.

Studie prochází teoretickým rozborem kreativního kódování, jeho historií a vývojem, předchozími studii a objevy, umělci, kteří tyto techniky používají. Rovněž jsou zkoumány a analyzovány četné praktické příklady, jako jsou programy, skici, dynamické webové stránky, trojrozměrné objekty, videa a animace vytvořené pomocí kreativních kódovacích technik. Na základě této analýzy je vypracována praktická část disertační práce.

Studie má následující dílčí cíle:

- Poskytnout aktuální přehled principů kreativního kódování, jeho definic, komponent a nejpopulárnějších postupů v rámci interaktivního vizuálního umění, vytvořit teoretický základ technických aspektů a nástrojů;

Kreativní kódování je často spojováno s řadou podobných praktik, jako je generativní umění, algoritmické umění, vizualizace dat, generativní algoritmy. Disertační práce vysvětluje tyto souvislosti a zkoumá nejpobulárnější softwarové nástroje, jako je programovací jazyk Processing a programy Shader.

- Na základě získaných znalostí a praktických zkušeností najít osobní přístup ke kreativnímu kódování pomocí generativních algoritmů a vizualizace dat;

Kreativní kódování je praktická disciplína, která vyžaduje neustálé experimenty a testování. Během let studií bylo vytvořeno velké množství programů, byly vyzkoušeny různé kódovací platformy, uspořádáno mnoho spoluprací. V disertační práci jsou popsány praktické aspekty studia, včetně návodu na práci s daty s důrazem na estetiku.

- Vytvořit online platformu, kde budou prezentovány nejzajímavější autorské projekty v oblasti kreativního kódování;

Programy často obsahují stovky řádků kódu, které nelze vložit a vysvětlit v textu disertační práce. Online platforma navíc ukáže skutečný "živý" obraz, pohyblivou animaci, vizualizaci nebo video. Veškerý kód každého programu bude uložen v open-source, což znamená, že kdokoli, kdo přijde na webovou stránku, bude moci kód číst, kopírovat, měnit a pokračovat ve vývoji. Kód bude doplněn komentáři, aby mu rozuměli i lidé bez zkušeností s programováním.

- Navázat spolupráci s dalšími organizacemi, vědci a umělci;

Techniky kreativního kódování jsou často založeny na kombinaci dat a algoritmů. Algoritmy nebo programovací instrukce pocházejí od umělce/kodéra. Data naopak mohou pocházet odkudkoli, mohou to být obrázky a fotografie, které se kódem promění v nový objekt, mohou to být zvukové materiály, které budou např. vizualizovány programem, mohou to být jakékoli sady databází, čísel, grafů, excelových tabulek, které budou smíchány s kódem k vytvoření uměleckého díla. Různé organizace, instituce a vědci mohou poskytnout taková data ze svých experimentů. Multimediální kultura může těžit ze vzniklé spolupráce mezi umělci a vědci, kteří spolupracují jako partneři s cílem stimulovat synergie. Taková spolupráce může vést k nečekaným uměleckým dílům a dalšímu rozvoji technik kreativního kódování.

- Zúčastnit se výstav a symposií, kde budou prezentována umělecká díla vytvořená během studia, uspořádat vlastní uměleckou výstavu;

Prezentované materiály budou mít podobu tištěné grafiky, videí a interaktivních programů běžících na různých obrazovkách, tabletech a projektorech.

- Propagovat praxi kreativního kódování mezi ostatními kolegy a studenty.

Výstavy a umělecká díla ukážou, co lze vytvořit pomocí kreativního kódování, online portál a text diplomové práce může ukázat, jak tato díla vznikala. Zájem o kreativní kódování stále roste, tato práce může pomoci dalším studentům a kolegům pochopit, kde začít.

1.2 Metodika práce

Podle Jochena Koubka (*Creative Coding*, 2010), profesora digitálních médií na Univerzitě v Bayreuthu, kreativní kódování popisuje použití programovacích jazyků a kódu jako designového nástroje pro produkci médií. Důkladný výzkum je většinou založen na praktických experimentech a vychází z předchozích zkušeností, teoretických základů a interakce s jinými kodéry/umělci, kteří pracují v podobných oblastech. Proto je nezbytné se zapojit do komunity kodérů, účastnit se diskusí, navštěvovat výstavy a analyzovat materiály spojené s touto disciplínou.

Disertační práce a výzkum jsou rozděleny do dvou fází. První, **průzkumná**, je založena na přehledu literatury, článků a dostupných příkladů již provedených výzkumů projektů, spojených s kreativním kódováním, a na studii dostupného softwaru. Všechny zdroje jsou zkoumány, analyzovány a vzájemně porovnávány. Kreativní kódování je praktická disciplína; veškerá dostupná literatura tedy představuje tzv. kódovací manuály a popisuje vybrané programovací techniky. Práce s takovým materiálem vyžaduje jeho neustálé testování na osobním počítači. Veškerá literatura a metody kódování jsou vybírány s cílem vytvořit interaktivní vizuální umění. Tato praxe je stále poměrně mladá, takže není k dispozici tolik materiálů, které by se zaměřovaly na tuto konkrétní oblast, proto první fáze prochází širším výběrem informací. Po experimentování s dostupnými příklady se výzkum přesouvá do druhé **praktické fáze** – vytvoření originálních programovacích skic, které se zaměří na generativní algoritmy a vizualizace dat. Během této fáze bude vytvořen soubor programů, které budou veřejnosti představeny na uměleckých výstavách a sympoziích. Popis programů, včetně vybraných příkladů kódu, bude k dispozici v disertační práci, plné verze programů a kódu budou k dispozici online na speciálně vyvinuté webové stránce pro ostatní studenty/umělce, aby je mohli zkoušet, experimentovat a vyvíjet.

Studie zodpoví na další výzkumné otázky:

1. Jaká je současná praxe kreativního kódování, a jak umělci/programátoři aplikují kreativní metody kódování v oblasti interaktivního vizuálního umění?

Praxe kreativního kódování je stále velmi nová, neexistují striktní hranice toho, co může zahrnovat, ani přesná definice toho, co vlastně představuje. K zodpovězení této otázky studie prochází přehledem již provedených teoretických výzkumů a analýzou moderních metod kreativního kódování v rámci multimediální kultury. Analýzy zahrnují kontextové chápání jevů kreativního kódování, včetně studia literatury, článků a dalších výzkumných prací; dále bude prozkoumán přístup autorů k tvorbě interaktivních vizuálních uměleckých děl, jako jsou interaktivní instalace, animace, vizualizace datových materiálů. Během průzkumu bude zjištěno, které programovací platformy, software a hardware tito autoři používají a proč, kterým nejčastějším problémům čelí. Tento postup může pomoci prozkoumat téma kreativního kódování na více úrovních, získat různé názory a komplexní pohled na předmět a obohatí teoretickou základnu multimediálního designu.

2. Jaký je postup a programovací metody při vytváření interaktivních uměleckých děl se zaměřením na generativní algoritmy a vizualizaci dat?

Při zkoumání tématu digitálního designu, kódování a programování nestačí pouze teoretická analýza. Všechny studované metody a příklady by měly být vyzkoušeny v praxi. Většina disertačních prací, knih a článků popisuje projekty s poskytnutým kódem, aby si je čtenáři mohli vyzkoušet na vlastních počítačích. Existuje mnoho programovacích jazyků, které lze použít pro umělecké účely, každý jazyk má své vlastnosti a parametry, některé se hodí lépe pro složité 3D scény, jiné pracují s jednoduššími animacemi pro dynamické webové stránky. Mezi důležité faktory, které je třeba vzít v úvahu při výběru programovacího jazyka, patří: požadovaná rychlost zpracování (kompilované vs. interpretované jazyky), sady nástrojů dostupné pro práci s vizuálními médii, spolehlivost, dostupné knihovny, složitost metod, kompatibilita s hardwarovým vybavením nezbytným pro interaktivní přístup (kamery, senzory, mikrofony atd.). Mezi programovací jazyky a metody, které byly testovány během let studií, patří:

- Java a Processing;
- JavaScript;
- Shaders a WebGL;
- openFrameworks;
- Python.

Výsledkem testování je výběr dvou programovacích platforem, které jsou kompatibilní a vhodné pro umělce a studenty, kteří nemají hluboké znalosti programování. Jedním z cílů tohoto výzkumu je propagovat kreativní kódování mezi studenty, kolegy a umělci, kteří jsou v této oblasti noví. S velkým počtem možností může být obtížné učinit první krok a zvolit vhodný programovací jazyk. Platformy, které byly vybrány pro praktickou realizaci disertační práce, jsou: Processing a Java, Shaders a WebGL. Tato část disertační práce vysvětlí výběr programovacích jazyků pro tvorbu interaktivního výtvarného umění se zaměřením na generativní algoritmy a vizualizaci dat.

V době psaní této disertační práce nebyly nalezeny žádné akademické studie, které by popisovaly kroky vytváření děl výtvarného umění na základě datových vizualizací a generativních algoritmů. Cílem tohoto výzkumu je zacetit mezeru a poskytnout popis a vysvětlení hlavních kroků potřebných k vytvoření originálních uměleckých objektů.

1.3 Spolupráce

Jedním z cílů studie je navázat spolupráci s dalšími organizacemi, vědci a umělci při vytváření nových originálních projektů. Taková spolupráce byla navázána s několika organizacemi a jednotlivci.

Centrum polymerních systémů UTB. Během spolupráce s Centrem polymerních systémů UTB byl získán velkoobjemový soubor dat, který obsahuje výsledky jejich experimentů s testováním odolnosti pneumatik. Takové testování je důležité pro úspěšný vývoj produktů pro pneumatiky používané v terénu nebo na špatných vozovkách a pro další náročné aplikace pryže. Pro napodobení a kvantifikaci poškození bylo použito speciální zařízení, které řídí a zaznamenává vícenásobná zatížení a posuny během dopadu na povrch vzorku z pevné pryže. Všechny výsledky z těchto testů představují čísla, která jsou uložena v tabulkách aplikace Excel. Tato čísla byla smíchána s autorovými programovacími algoritmy, aby vznikly originální vizualizace. Vizualizace v podobě tištěných materiálů a audiovizuálních materiálů běžících na velkoplošných obrazovkách byly prezentovány na různých výstavách, v galeriích i online, např.: Galerie Hauerova 4, osobní autorská výstava "See the wind"; Galerie G18, osobní autorská výstava "Undercurrent"; Nadace Hollar, Grafika roku 2020; Lucerna TV – online vzdělávání televize; Data Art – autorský online portál pro sdílení kreativních kódovacích projektů; Youtube.

Artcolony Cered. Cered Contemporary International Art Colony je mezinárodně orientovaná dílna a umělecká scéna se sídlem v maďarské vesnici Cered, založená na osobní účasti a je jednou z evropských uměleckých kolonií, která se každoročně koná podle jiného tématu. V průběhu let studia vznikly ve spolupráci s Artcolony a jejími účastníky tři projekty. "To Csaba and Eszter Who

Walked Over the Hills of Cered" - vizualizace audio deníku, který byl použit jako pozvánka na sympozium; "See the Wind" - program, který mapuje vzory proudění vzduchu poblíž vesnice Cered pomocí analýzy a interpretace skutečných údajů o větru, které byly získány z nejbližší meteorologické stanice, a "The Stars of Cered" - vizualizace hvězdných dat (vzdálenost od Země, absolutní a zdánlivá velikost, stupeň a azimut) získaných pomocí mobilní aplikace The Star Chart během sympozia 2022. Projekty byly prezentovány v Ceredu, na osobní výstavě v galerii Hauerova 4, v galerii G18, online na Lucerna TV, portálu Data Art a YouTube.

CREDO Project. The Cosmic Ray Extremely Distributed Observatory (Extrémně distribuovaná observatoř kosmického záření) - projekt testuje jednu z mnoha teorií o tom, co by mohla být temná hmota – superhmotné částice zrozené v raném vesmíru. Vědci z CREDO vytvořili speciální aplikaci, která pomáhá detekovat částice, do projektu se tak může zapojit každý, kdo má chytrý telefon. Data získaná z tisíců chytrých telefonů po celém světě jsou ukládána online a jsou využívána k vytvoření sady vizualizací, které byly veřejnosti představeny na autorské výstavě v galerii G18 ve Zlíně v červnu 2022.

Slezská univerzita v Opavě, Fyzikální ústav. 3D modely přesně zmapovaných částí a orgánů lidského těla, jako je srdce, kyčle, mozek, byly sdíleny a použity jako data pro sadu nových vizualizací. Vizualizace v podobě tištěné grafiky a audiovizuálních materiálů byly prezentovány na autorské výstavě „Undercurrent“ v galerii G18 ve Zlíně.

1.4 Výstupy disertační práce

Důležitou součástí disertační práce a studia je uplatnění všech získaných znalostí a zkušeností do osobních uměleckých projektů, vytvořených pomocí generativních algoritmů aplikovaných v rámci zvoleného programovacího prostředí. Výstupem disertační práce je soubor interaktivních programů a uměleckých projektů, které byly během let studia prezentovány na uměleckých výstavách, festivalech a sympoziích, a také online portál s podrobným popisem každého projektu.

Vývoj každého programu prochází podobnými fázemi:

- Volba programovacího jazyka a platformy;
- Hledání inspirace/tématu pomocí generativních algoritmů, vizualizace dat, vědeckých experimentů, zvukových materiálů atd.
- Vytvoření sady programovacích skic s různými změnami chování, pohybu nebo designu;

- Výběr nejslibnější skici a její přeměna v hotový program;
- Aplikace principů interakce;
- Příprava veškerého materiálu pro fyzickou instalaci a online zobrazení.

Online portál

Jedním z cílů disertační práce je vytvoření online portálu, kam se budou nahrávat všechny probíhající experimenty s kreativním kódováním, datovými a audio vizualizacemi. Tento portál představuje strukturovanou základnu kreativních metod kódování, včetně popisu experimentů s různými programovacími jazyky a platformami. Tato základna je zaměřena na různé typy dat, většinou ve vztahu ke světu vědy, jako jsou generativní algoritmy, výsledky vědeckých experimentů, principy základní fyziky, matematické vzorce a algoritmy, zvukové materiály.

Každé dílo nahrané na portál je prezentováno s jeho podrobným popisem a průvodcem a obsahuje odkazy na kompletní kód každého projektu, aby jej zájemci mohli testovat, používat a dále rozvíjet. Portál obsahuje popis nejužitečnějších způsobů grafického znázornění dat v rámci vizuálního umění pomocí vybraných programovacích platforem, které umožňují vytvářet, manipulovat a interagovat s těmito reprezentacemi. Portál si klade za cíl pomoci umělcům a designérům orientovat se mezi technikami kreativního kódování, získat inspiraci a udělat první kroky do světa kreativního kódování a digitálních vizualizací.

Portál dostal název **Data Art** (datové umění), je plně funkční, obsahuje vybrané projekty kreativního kódování.

Odkaz: <https://antrague.com/>

Výstavy

Během let studia vznikly desítky různých uměleckých projektů vytvořených metodami kreativního kódování. Řada z nich je prezentována online, na portálu Data Art, Lucerna TV – vzdělávací televize, YouTube, OpenProcessing – webová stránka pro sdílení projektů napsaných programovacím jazykem Processing, Shadertoy – online portál pro sdílení OpenGL shaderů, CodePen – portál pro sdílení javascriptových programů. Řada projektů byla prezentována na výstavách, galeriích a sympoziích. Jako jsou:

- Osobní výstava "See the wind" (výstava byla pojmenována podle jednoho z uměleckých projektů), galerie Hauerova 4, Opava. Na výstavě

bylo představeno 16 tištěných grafik a několik audiovizuálních projektů běžících na velkoplošné obrazovce;

- Sympozium Artcolony Cred v Maďarsku, rok 2020 s audio vizualizací "To Csaba and Eszter Who Walked Over the Hills of Cered", rok 2021 jako součást projektu "Next generation", kde bylo poprvé představeno umělecké dílo "See the Wind", rok 2022 v rámci sochařského mezinárodního sympozia, kde byl představen projekt "The Stars of Cered" - vizualizace hvězdných dat ve formě videa obohaceného o audio materiály;
- Galerie Hollar, Grafika roku 2020. Prezentovány byly dvě práce, tištěné verze datových vizualizací: "NR - 32 ms" - velikost A4; "NR - 38 ms" - velikost A1.
- Galerie Hollar, Grafika roku 2021 s tištěnou grafikou z projektu „Everything flows, nothing remains“. Výstava proběhla od 31. března do 30. dubna 2022 v Karolinu, Ovocný trh, Praha 1;
- Mezinárodní výtvarné symposium PANTA RHEI 2021, Galerie kino Hvězda, Uherské Hradiště, Banská Bystrica, Galéria FX. Vytvořený projekt se jmenoval "Everything flows, nothing remains" a představuje fraktál – nekonečný geometrický tvar. Vizualizace objektu je výsledkem spojení programovacích algoritmů a počítačových dat z frézy HWT, kterou tvůrci účastníci se sympozia PANTA RHEI používali pro tvarování svých materiálů, od frézování dřeva po řezání plastů. Vizualizace byla vytvořena pomocí programování WebGL. Kamera prochází pod různými úhly tímto fraktálem a ukazuje jeho nejbližší i nejbližší části.
- Osobní výstava „Undercurrent“ v galerii G18 ve Zlíně. Termín výstavy: 25.05.2022 – 9.06.2022. Na výstavě byly vystaveny vybrané projekty z doby studia, tištěné grafiky a audiovizuální materiály běžící na různých velkoplošných obrazovkách, monitorech, tabletech, počítačích a projektorech.

1.5 Přínos práce pro vědu a praxi

Teoretická a praktická část disertační práce slouží podobným účelům – k **propagaci kreativního kódování** mezi ostatními umělci, kolegy, studenty. Teoretická analýza a srovnání různých uměleckých/kódovacích technik obohatí teoretický základ multimediálního designu. Online portál Data Art, který obsahuje popisy programů včetně odkazů na celý kód každého projektu, by mohl sloužit jako výchozí bod pro ty, kteří se chtějí učit a otestovat své dovednosti v této moderní oblasti vizuálního umění. Tato disertační práce by mohla být

přínosem pro pedagogický proces a pomoci mladým designérům, studentům, umělcům.

Disertační práce zkoumá programování jako další umělecký nástroj, který nepatří jen do chladného světa vědy a matematiky. Předpokládá se, že programovací algoritmy mohou být příliš složité na práci, že patří do světa logiky a přesnosti, zatímco v oblasti umění převažují pocity, emoce a nálady. Pravdou je, že kreativní kódování spojuje tyto dva světy do jednoho. Stejně jako malíři používají akvarely a štětce ke kreslení nebo sochaři proměňují hlinu v rozpoznatelné předměty, kreativní kodéři používají počítače jako další nástroj k vyjádření svých nápadů. Rozdíl mezi různými programy pro digitální umění, jako je Adobe Photoshop, Illustrator, AutoCAD, 3D Max atd., a kreativním kódováním je v tom, že všechny tyto programy mají omezenou sadu nástrojů, příkazů a filtrů a v důsledku toho omezený rozsah možností. Při používání programování si umělec může vytvořit svůj vlastní panel nástrojů a vyvinout algoritmus, který nikdo předtím nepoužil. To umožňuje designérům vybrat si jedinečnou cestu, která povede k širší škále konceptů výtvarného umění.

V moderní době existuje mnoho programovacích jazyků vyvinutých speciálně pro umělce a designéry, jako je Processing, Arduino, openFrameworks, WebGL a další. Tyto jazyky používají zjednodušený způsob, jak popsat myšlenku a proměnit ji v život. Určitá pravidla programování je stále třeba dodržovat, ale začátečník nemusí mít žádné speciální dovednosti, aby mohl začít vytvářet první program.

Disertační práce teoreticky i prakticky ukáže, že kreativní kódování je důležitou oblastí vizuálního umění, že je může využít každý umělec k tvorbě objektů digitálního designu, dynamických webových stránek, animací, kreseb, vizualizací dat, 3D objektů, scén pro videohry, sochy, interaktivních instalací, vizuálních efekty pro divadelní představení, představení VJ, video mapování atd. Potenciál je skutečně obrovský.

2 KONCEPT KREATIVNÍHO KÓDOVÁNÍ

Cílem disertační práce je prozkoumat možnosti kreativního kódování v rámci vizuálních médií. Tento směr zahrnuje studium programování, matematiky, designu a výtvarného umění. Kombinace těchto disciplín přinese nové perspektivy do vývoje multimediální kultury. Rozvoj digitálních technologií přináší obrovský rozsah možností pro výzkum a tvorbu unikátních projektů. Spousty programovacích jazyků a platforem, stejně jako hardwarových nástrojů jsou k dispozici pro použití, nicméně široká volba může přinést nejistotu při rozhodování, který přístup zvolit.

Další kapitoly prozkoumají a analyzují koncept kreativního kódování, jeho vývoj v historii a technologiích, nejvýznamnější jména, zhodnotí moderní nástroje, které umožňují kombinovat programování a umění.

2.1 Studium zdrojů

Důležitou součástí disertační práce je studium dostupných zdrojů, jako je literatura, články, provedené vědecké výzkumy a praktické příklady spojené s touto disciplínou. Každý studovaný výzkum by měl mít solidní teoretický základ, proto je důležité definovat, kdo již na tématu kreativního kódování pracuje, jaké jsou výsledky a závěry, zda má význam ve studiu tímto směrem pokračovat a jak výzkum dále rozvíjet na základě získaných znalostí.

Kreativní kódování se každým dnem stává stále populárnější oblastí expresivních médií a stále více umělců hledá inspiraci v této formě. Disciplína je však stále relativně nová, neexistuje mnoho akademických studií, které by zkoumaly možnosti kreativního kódování ve světě vizuálních médií a propagovaly je jako další umělecký nástroj, který by autoři mohli využít. Tato disertační práce má za úkol tuto mezeru zmenšit.

Není pochyb o tom, že kreativní kódování je praktický předmět, výzkumník by měl vždy otestovat různé metody kódování na osobním počítači, aby ve studiu uspěl; četné knihy dostupné pro veřejnost, zejména průvodce kódování a programování, pomáhají při studiu této disciplíny. Svět digitálních technologií se vyvíjí rychlostí, kterou v jiných oborech nevidíme, a proto mnoho tištěných materiálů nestrhává krok s progresem. Například nová aktualizace softwaru, ke které dochází jednou za několik měsíců, může vést ke změně struktury kódovacích algoritmů. Proto je nesmírně důležité studovat rychle se objevující online články a průvodce, které pomohou držet krok s moderními digitálními nástroji.

Teoretická studie prošla různými zdroji informací, jako jsou:

- Disertační práce související s tématem kreativního kódování;
- Knihy vztahující se ke generativnímu umění, programování a úspěšným kodérům a umělcům;
- Články a výzkumné materiály pro konference;
- Manuály, které popisují programovací techniky;
- Četné online galerie a portály, kde jsou uloženy tisíce programů a skic vytvořených slavnými umělci i absolutními nováčky s otevřeným zdrojovým kódem.

Hlavním cílem teoretického přehledu existujících studií je porovnat a analyzovat různé přístupy umělců, spisovatelů a programátorů k tématu kreativního kódování. Každý z nich si vybírá své vlastní prostředí (jako je programovací jazyk, software, hardware); někteří autoři disertací vysvětlují, proč zvolili tento způsob, často stručně, aniž by uváděli další možnosti.

Jedna z mála existujících akademických studií patří Benjaminovi Frymu (*Computational information design*, 2004). Fry je klíčová postava v generativním umění, kreativním programování a digitalizaci. Jeho disertační práce není příliš nová, ale lze ji považovat za jednu z hlavních studií v této oblasti a pravděpodobně za první. Fry je jedna z mála postav, která posunula svět digitálních médií na novou úroveň. Spolu s Casey Reasem inspiroval Benjamin Fry novou generaci kodérů a dal každé osobě, nejen umělcům, stejnou příležitost vytvořit a oživit své nápady a projekty. Jeho disertační práce je také důležitá, protože představuje základ pro „Processing“ - softwarovou a jazykovou platformu vyvinutou Frym a jeho kolegy, která bude použita v praktické části disertační práce.

Diplomové práce Bena Frye a Alexe McLeana (*Artist-Programmers and Programming Languages for the Arts*, 2011) jsou skvělým příkladem toho, jak se teoretická studie může stát velkým praktickým projektem, který přináší užitek mnoha dalším umělcům. Jennifer Jacobs (*Dynamic Drawing: Broadening Practice and Participation in Procedural Art*, 2017) prošla hlubším teoretickým rozbohem, ale výběr programovacích platforem byl opět poněkud omezený, studie by měla i nadále pokrývat všechny možnosti, které moderní technologie kreativnímu kódu poskytují. Je důležité analyzovat a porovnat různé metody programování a popsat silné a slabé stránky každé z nich.

Všechny studované disertační práce jsou zaměřeny na jeden velký praktický projekt, jehož autoři mají jasnou představu o tom, čeho a jak bude dosaženo, což je připravuje o flexibilitu a experimentování s různými materiály a technikami. Takový přístup není dokonalý, protože kreativní kódování nabízí nepřekonatelný potenciál pro experimentování s nápady, koncepty a návrhy. Teoretické části

těchto disertačních prací rovněž postrádají variabilitu, autoři se většinou zaměřují na jedno téma, které je zajímavé a bude se dále rozvíjet. Dalším problémem je, že jednoduše neexistuje dostatek studií o kreativním kódování. Tyto disertační práce jsou jediné, které v současnosti s tímto tématem souvisí.

Během několika posledních desetiletí bylo vydáno mnoho knih, které se zabývají tématem programování. Existuje výrazně méně zdrojů literatury, které studují možnosti a vývoj kreativního kódování. Studium prošlo literaturou, která je spojena s historií a vývojem kreativního kódování, a také specifickými knihami, které pokrývají základy zvoleného programovacího jazyka.

Část studované literatury je spíše technická, týká se jazyka Processing, openFrameworks a OpenGL, pokrývá další témata:

- Základy programování: proměnné, pole, operátory, řídicí příkazy, objekty a vlastnosti, funkce, smyčky atd.;
- Základy Processingu: hlavní metody, knihovny, uživatelské interakce, externí data, metody exportu atd.;
- Základy Arduino: řadiče, základy aplikace, konstanty, metody, pole, ladicí aplikace atd.;
- Objektově orientované programování s C ++;
- openFrameworks: pole, metody, kreslení proměnných ve 2D, zobrazování videa a obrázků, kompilace a ladění, import knihoven;
- Fyzický vstup: interakce s fyzickými ovládacími prvky, kinetikou, světly, knoflíky, pohybem, vzdáleností, vibracemi atd.;
- Programovací grafika;
- Rastry a pixely;
- Zvuk a audio;
- Protokoly a komunikace;
- Grafika a OpenGL;
- 3D v Processingu a openFrameworks;
- Pohyb a umístění;
- Pohybová gesta;
- Prostory a prostředí.

Tyto zdroje poskytují hlubší pochopení technických parametrů, nástrojů a příkazů jednotlivých programovacích jazyků.

Další část studia prochází přehledem praktických aplikací kreativního kódování. Velkou výhodou většiny programů je, že jsou uvedeny v „open source“, což znamená, že veškerý kód je dostupný online pro ostatní programátory a umělce ke studiu, úpravě a použití. To umožňuje zkoumat a analyzovat každý řádek kódu a obecně principy konkrétního programovacího jazyka. Četné online portály, jako je Codepen, OpenProcessing, Shadertoy atd. obsahují tisíce takzvaných programovacích „skic“, což jsou programy, které obsahují nezkompilované soubory zdrojového kódu a jsou uloženy ve formátu prostého textu. Každý z těchto programů lze spustit přímo na zvoleném programovacím portálu nebo stáhnout pro další použití na osobním počítači.

Příklady kreativního kódování lze nalézt kdekoli: na interaktivní webové stránce, v hudebním videu, v galerii nebo muzeu, ve výzkumném centru, na městských billboardech, během koncertu, divadelního nebo tanečního představení. Kreativní kódování je jedinečná praxe, která dává příležitost lidem z různých oblastí, které nejsou tradičně spojeny s oblastí programování, otestovat sílu kódu, aby mohli sdílet své nápady nebo se jednoduše vyjádřit.

Pro další rozvoj kreativního kódování a jeho propagaci mezi širším publikem je důležité vytvořit obecný přehled nejoblíbenějších a nejslibnějších technik a popsat scénáře pro každou z nich. Základní rozbory jsou uvedeny v kapitole „3. Nástroje a programovací jazyky kreativního kódování“, rady pro začátečníky, kteří se chtějí dostat do sféry kreativního kódování, jsou popsány v kapitole „3.8 Jak začít s kreativním kódováním?“ a „4. Proces vizualizace dat v kontextu kreativního kódování“. Podrobnější a konkrétnější technický manuál je k dispozici online na vytvořeném portálu Data Art.

Veškeré poznatky získané při teoretickém studiu literárních zdrojů a praktickém testování různých programovacích platforem, programů, jazyků a „skic“ jsou aplikovány v praxi a popsány v praktické části disertační práce.

2.2 Definování pojmů

Než se pustíme do jakékoli analýzy, je užitečné definovat pojmy. Co je kreativní kódování, generativní umění a generativní algoritmy? Jaká jsou kritéria? Jak tomu rozumí různí umělci a programátoři?

V průběhu dějin civilizace se lidstvo vždy snažilo vyjádřit sebe, své myšlenky, emoce a názory pomocí různých médií. S rozvojem měst a zemědělství lidé ovládli architekturu a krajinářský design. První známky tradičního vizuálního umění byly nalezeny ve starobylých jeskyních, jako je Chauvet ve Francii, které

se datují přibližně 30.000 let před naším letopočtem. Pravděpodobně byly v té době vytvořeny i první hudební nástroje, stejně jako tance, které byly podporovány hudbou. Básně a divadelní představení nakonec následovaly mýty a rituály. Fotografie, filmy a videa se poměrně nedávno připojily k rodině umění (Greenberg, 2007, s. 13).

V průběhu posledních 60 let se náš svět změnil na digitální. Žádná forma umění nezachytila toto přechodné období – naše období – líp než **generativní umění**. Generativní umění plně využívá všechno, co může počítač nabízet, a produkuje elegantní a působivá umělecká díla, která rozšiřují stejné principy a cíle, které umělci sledují od počátku moderního umění (Bailey, 2018).

Co přesně generativní umění je? Jedna příliš jednoduchá, ale užitečná definice od Jasona Baileye (2018) spočívá v tom, že generativní umění je umění naprogramované pomocí počítače, který úmyslně uvádí náhodnost jako součást svého procesu tvorby. Úlohou umělce je navrhnout nebo ovlivnit tento proces do určité míry.

Tato definice může vést k zavádějícímu závěru, že umělec má nulovou kontrolu a počítač náhodně vytváří návrhy a výkresy; nebo že umělec má úplnou kontrolu a kód je vždy proveden přesně tak, jak je napsán, a generativní umění tak ztratí prvky náhody, chaosu, objevu a spontánnosti, které často dělají umění výjimečným. Pravdou je, že generativní umělci obratně řídí jak velikost, tak umístění náhodnosti zavedené do tvorby.

Kontrolovaná náhodnost může znít protichůdně, ale při pohledu na historii umění si lze všimnout, že lidé vždy vyhledávali způsoby, jak do své práce zavádět náhodnost pro stimulaci kreativity. Proces kódování generativního umění je ve skutečnosti velmi podobný normálnímu malování nebo skicování.

Philip Galanter (2003, s.4), profesor generativního umění a fyzikální výpočetní techniky na Texaské univerzitě A&M, uvádí svou definici: *generativní umění označuje jakoukoli uměleckou praxi, kde umělec používá systém, jako je soubor pravidel přirozeného jazyka, počítačový program, stroj nebo jiný procedurální vynález, který je uveden do pohybu s určitou mírou autonomie přispívající k uměleckému dílu nebo vede k vytvoření nového.*

Termíny „generativní umění“ a „počítačové umění“ se používají v tandemu a nahrazují se při různých příležitostech již od nejranějších dnů počítačové historie. První výstava počítačového umění se jmenovala „Generative Computergraphik“, v češtině – Generativní počítačová grafika, a konala se ve Stuttgartu v únoru 1965. Svou tvorbu zde představil Georg Nies, německý akademik a průkopník počítačového umění a generativní grafiky. O čtyři roky později dokončil svou první doktorskou disertační práci v oboru počítačové umění se stejným názvem

jako výstava (Nees, 1969). Tato teze byla brzy přijata a používána jako příklad a inspirace malou, ale rostoucí komunitou, která spojovala slova generativní a počítač dohromady a na jevišti světového výtvarného umění se začali objevovat generativní umělci (Boden, Edmonds, 2009, s.4).

Od té doby generativní umění produkovalo shluky generativních uměleckých aktivit. Neustálý vývoj digitálních nástrojů přináší celou řadu nových technik a postupů. Vizualní formy jsou vytvářeny pomocí výpočetní logiky, která je spouštěna specifickou sadou pravidel v závislosti na výběru kódového jazyka a programovací platformy. Lze konstatovat, že generativní umění je širší pojem, který se postupem času a pokroku rozdělil na různá hnutí a praktiky, mezi které patří:

- kreativní kódování;
- kódové umění;
- výpočetní umění;
- softwarové umění;
- internetové umění, net art;
- interaktivní umění;
- algoritmické umění;
- procedurální umění;
- vizualizace dat;
- fraktální umění;
- elektronická hudba a algoritmická kompozice;
- živé kódování, atd (více viz obrázek 1).

zaměření většiny příkladů použití programovacích příkazů a řádků kódu, kreativní kódování používá programovací jazyky výhradně pro umělecké účely.

Další částí umělecké rodiny založené na počítači je **algoritmické umění** (v anglickém jazyce „**algorithmic art**“). Vlatko Ceric (2008, s.2) definuje algoritmické umění jako vizuální umění, které je generováno na základě algoritmů, takové algoritmy zcela popisují proces a vývoj generování obrázků. Tento druh umění úzce souvisí s moderním vývojem softwarových technologií a zejména s počítačovým programováním. Další vliv na algoritmické umění souvisí s matematikou, která se používá v algoritmech pro generování obrazu. **Fraktální umění** je příkladem algoritmického umění. Pro obrázek přiměřeně velké velikosti jsou i ty nejjednodušší algoritmy příliš výpočetně náročné na ruční provádění, takže běží buď na jednom počítači, nebo na shluku počítačů. Výsledkem je kresba, obvykle zobrazená na monitoru počítače jako projekce nebo jako tištěná grafika. Variabilita může být zavedena pomocí pseudonáhodných čísel (*Approximating Reality with Interactive Algorithmic Art*, 2001).

Fraktální umění (fractal art) je forma algoritmického umění vytvořená výpočtem fraktálních objektů, výsledky jsou reprezentovány jako digitální obrázky, animace, videa (Bovill, 1996, s. 153). Fraktály jsou nekonečně složité vzory, které jsou sobě podobné v různých měřítcích. Vznikají opakováním jednoduchého procesu stále dokola v konstantní zpětnovazebné smyčce (*What are Fractals*, 2018). Matematická krása fraktálů leží na průsečíku generativního umění a počítačového umění, které se spojují do abstraktních uměleckých děl.

Podobná definice platí pro **kódové umění (code art)**. Jak uvádí ve svém článku Michal Šimkovič, jde o počítačově generované umělecké dílo, které je definováno algoritmy. Nachází se na jedinečném průsečíku designu, umění a programování (Simkovic, 2019).

Další definice popisuje **softwarové umění (software art)**. Softwarové umění lze obecně definovat jako umění, jehož materiálem je formální instrukční kód a/nebo které se zabývá kulturními koncepty softwaru (Cramer, 2002). Ve snaze tuto problematiku dále rozšířit Pete Schultz (2002) popisuje softwarovou kulturu jako živou kulturu programátorů a uživatelů, kteří přebírají role aktivních účastníků ve světě softwaru nebo zprostředkovaného softwaru. Ve svém jádru softwarová kultura ohraničuje oblast intenzivní nehmotné produkce, která vzniká jako reakce na kritickou reflexi každého aspektu lidského života, který je nějakým způsobem řízen či ovládán softwaru. Softwarové umění odráží realitu a potenciál této kultury (*Say it with software art*, nd).

Internetové umění (také známé jako **net art**) je druh umění, který využívá internet jako způsob šíření namísto tradiční galerie. Net art je často interaktivní, může využívat řadu různých médií, jako je zvuk, animace a videa. Tato metoda

umožňuje i malým umělcům sdílet svou práci s velkým publikem (*Internet art*, 2017). Termín internetové umění obvykle neodkazuje na umění, které bylo jednoduše digitalizováno a nahráno, aby bylo možné je zobrazit přes internet, například v online galerii. Net art spoléhá na to, že internet existuje jako celek a využívá takové aspekty, jako je interaktivní rozhraní a konektivita s mnoha sociálními a ekonomickými kulturami (Ippolito, 2002, s.485).

Procedurální umění (Procedural art) představuje umělecké dílo definované výpočtově reprezentovaným systémem pravidel (programovacích algoritmů), vztahů a chování, které umožňuje vytvářet flexibilní díla, která jsou adaptabilní a schopná systematické revize. Popis práce prostřednictvím kódu umožňuje umělcům spravovat složité struktury a velké soubory dat, automatizovat procesy a zobecňovat a znovu používat operace (Jacobs, 2017, s.3-12).

Vizualizace dat (Data visualization) je mezioborovou specializací, která rozebírá fakta a informace a přeformátuje je do stravitelných grafických reprezentací. Takové vizualizace pomáhají sdělovat objemná data, která jsou příliš složitá, komplikovaná nebo monotónní (Gordon, 2021).

Jak popisuje Salah Uddin Ahmed (2018, s. 4), **digitální interaktivní umění (digital interactive art)** se týká žánru uměleckého díla, které je interaktivní a využívá digitální technologii jako nezbytnou součást kreativního nebo prezentačního procesu. Edmonds a Candy (2011, s. 18-32) uvádějí, že interaktivní umění se vyznačuje dynamickým chováním v reakci na vnější stimulaci, jako jsou zvuky nebo pohyb publika. Chování publika může způsobit změnu samotného uměleckého díla. Při vytváření interaktivního umění umělec bere v úvahu nejen estetiku uměleckého díla, ale také způsob, jakým interaguje s publikem.

On-the-fly programování (programování za běhu) neboli živé kódování (live coding) je styl programování, ve kterém je programátor performerem a skladatelem zároveň, který rozšiřuje a upravuje program v reálném čase, bez zastavení nebo restartu, s cílem prosadit výrazné, programovatelné řízení za běhu (Wang G. a Cook, 2017, s.193). Živé kódování je nejprominentnější jako divadelní umění, často se používá k vytváření digitálních médií založených na zvuku a obrazu, stejně jako světelných systémů, improvizovaného tance a poezie (Magnusson, 2013, s.1).

Některé výše uvedené definice se shodují se vzájemným popisem a představují stejné nebo podobné odvětví počítačového umění. Pojmeme, který bude dále v disertační práci používán, je kreativní kódování a v souhrnu studovaných termínů lze s jistotou konstatovat, že kreativní kódování je:

- akt použití počítačového programovacího softwaru k vytváření uměleckých děl, designu, architektury, hudby atd.

- způsob použití kódu k tvorbě umění;
- procedurální způsob tvorby expresivních médií;
- spolupráce umělce a počítače;
- průsečík organického a mechanického světa;
- umění přeměnit sady algoritmů na grafiku, vizualizace a animace.

Kreativní kódování je úzce spjata s jinými formami počítačového umění, může využívat stejné matematické postupy jako algoritmické umění, stejný soubor dat jako při vizualizaci dat, stejné interaktivní principy jako umělci pracující s interaktivními instalacemi.

2.3 Kódování versus programování

Pojmy „kódování“ a „programování“ jsou na dalších stránkách používány poměrně často, je užitečné definovat, co se těmito dvěma disciplínami rozumí.

Kódování a programování jsou často považovány za totožné. Yaduwanshi (2021) definuje **kódování** jako akt překladu lidské komunikace nebo příkazů do strojového jazyka, kterému počítač rozumí, jazyka 0 a 1. Kodér používá různé programovací jazyky jako C, Java, C#, Python, JavaScript atd. Tyto jazyky pomáhají vytvářet a spouštět programy poskytováním instrukcí počítači. Kódování zahrnuje psaní pokynů pro softwarové programy, jako jsou aplikace, webové stránky, hry atd.

Na druhé straně **programování** je *proces navrhování a vytváření spustitelného počítačového programu pro dosažení konkrétního výpočetního výsledku nebo pro provedení specifické úlohy* (Yaduwanshi, 2021). Úkolem programátora je naplánovat a provést celý proces tvorby konkrétního softwaru, od prvotního nápadu až po testování hotového produktu.

Jednoduše řečeno, kodéři píší příkazy v programovacích jazycích, programátoři používají tyto příkazy a další činnosti (recenze dokumentů, vytváření algoritmů, modelování problémů, zpracování dat a řízení projektů) k dokončení softwarového produktu, aplikace nebo webové stránky.

Kreativní kódování zahrnuje obě činnosti. Kódování – pro zápis řádků příkazů, které bude provádět počítač. Následující příklad používá jazyk Processing ke kreslení kruhu, kde 150 (v pixelech) znamená polohu kruhu na x-souřadnici, 160 - pro y-souřadnici a 250 pro šířku a výšku kruhu:

```
void draw() {  
  circle (150, 160, 250);  
}
```

Takže kódování používá příkazové řádky/algoritmy k definování každého objektu, každé barvy, pohybu, animace uměleckého díla. Programování bude zahrnovat přeměnu kódu na kompletní program, který lze přenést z jednoho počítače do druhého, připojení hardwarových nástrojů, kamer, senzorů, projektorů, pokud jsou plánovány jako součást uměleckého díla.

2.4 Kdo je kreativní kodér?

„All good science is art. And all good art is science.“

John Fowles

Když se umění a věda správně spojí v tvůrčím procesu, vyústí v něco transformativně nového. Na rozdíl od funkčního zaměření většiny způsobů práce s programováním a kódem – jako jsou programovací řady mobilních aplikací nebo kódování webových portálů – kreativní kódování využívá programovací jazyky výhradně pro umělecké účely. Estetika zde vždy hraje zásadní roli.

Aby se jednotlivec stal úspěšným umělcem v oblasti kreativního kódování, měl by obsáhnout mnoho disciplín: umění, design, programování a matematiku. Jedno bez druhého nefunguje, samotná znalost kódu bez estetického porozumění nevyústí v umělecké dílo. Na druhou stranu, i když jsou některé programovací jazyky, jako je například Processing, pro umělecké účely zjednodušeny, stále vyžadují pochopení programovacích příkazů.

Proces tvorby projektu často zahrnuje vytvoření originálních algoritmů. Jun Wu, spisovatelka a technologka, se tomuto problému věnuje v článku „Getting Started With Creative Coding“ („Začínáme s kreativním kódováním“). Kreativní kódování vyžaduje nejen matematické základy, ale také umělecký talent k použití algoritmů při vytváření uměleckých děl. „Lidé, kteří jsou opravdu skvělí v kreativním kódování, jsou polyhistori. Hledají smysl jak v kreativním, tak v logickém vesmíru. Hluboko se ponoří do předmětů, se kterými pracují. Nakonec se z nich stanou skvělí programátoři, mnoho jejich projektů jsou softwarové projekty“ (Wu, 2020).

Je pravda, že kreativní kodéři často potřebují rozumět několika programovacím jazykům, aby vytvořili umělecká díla nebo jakoukoli formu kreativního vyjádření. Kreativní kodér může organizovat interaktivní uměleckou instalaci, vytvářet generativní umění, animace, videa, navrhovat prototypy produktů, vizualizovat

data, provádět jevištní představení s prvky vizuálních efektů, vytvářet interaktivní webové stránky nebo vytvářet zvukové umění (Donovan, 2020).

Není to však tak zastrašující, jak se může zdát. Kreativní kódování je pro každého, bez ohledu na znalost jakéhokoli existujícího programovacího jazyka. Stále více designérů po celém světě dělá své první kroky v oblasti kreativního kódování. Četné online galerie obsahují stovky příkladů kódování, které si může kdokoli prohlížet a volně používat. Speciální programovací jazyky zjednodušují proces psaní kódu speciálně pro umělecké účely. Pro ty, kteří jsou ochotni se učit, je k dispozici mnoho návodů, videí, literatury.

2.5 Průkopníci a průkopnické technologie generativního umění

Kreativní kódování, generativní umění, softwarové umění a další související disciplíny spoléhají na jeden hlavní nástroj, kterým je počítač. Počítačové programování, přesněji řečeno. Není divu, že první kroky kreativního kódování byly učiněny v raných letech vývoje počítačových technologií. V 60. letech minulého století byly velké sálové počítače pro širokou veřejnost nepřístupné, nejen že cena jednoho stroje mohla dosáhnout několika milionů dolarů, ale také vážily několik tun a potřebovaly k instalaci prostor celého patra budovy. Přístup k těmto strojům získával pouze velký průmysl a laboratoře podporované armádou USA a vesmírným programem. IBM se stala nezpochybnitelnou jedničkou na trhu v prodeji těchto velkých, drahých a velmi těžko použitelných strojů (*History of Computers*, nd).

Rané počítačově generované umění bylo většinou praktikováno vědci, kteří měli na svých pracovištích praktický přístup k exkluzivní a drahé počítačové technologii. Někteří z nich později zasvětili svou kariéru kreativnímu využití technologií v umění, například významní počítačové umělci Frieder Nake a Georg Nees (Nees, 1969) z Německa. Na druhou stranu, někteří vědci si s počítači a programováním jen hráli ve svém volném čase, tvořili vtipy nebo amatérské umění nebo řešili své vlastní programování a kreativní problémy. Někteří nadšenci experimentující s generativním uměním byli vyškolení umělci (jako Kurd Alsleben a Otto Beckman), zatímco jen velmi málo z nich mělo oba druhy vzdělání, umělecké i vědecké (jako Charles Csurí). Nová forma umění obohatila nebo zmátla tehdejší protagonisty uměleckého světa, jak tomu někdy je i dnes (Fritz, 2011, s. 9-11).

Na úsvitu počítačových dějin neexistovaly žádné softwarové nástroje pro umělce. Software, programy a kódovací rutiny pro počítačové výtvarné umění rostly ruku v ruce s hardwarem. V té době neměly počítače žádnou obrazovku, na které by bylo možné obraz zobrazit. Umělci tedy často používali perové plotry – mechanické zařízení, které drží pero nebo štětec a je propojeno s počítačem, který řídí jeho pohyby k přenosu výsledků své práce na papír. Umělci využívající nové

výpočetní a vizualizační technologie museli buď spolupracovat s inženýry, aby naprogramovali své nápady, nebo vytvořit své vlastní programy a algoritmy. Mezi těmi umělci, kteří jako první implementovali kód a počítače do svých uměleckých děl, byli Hiroshi Kawano, Herbert Frank, Georg Nees, Manfred Mohr, Frieder Nake, Vera Molnar, Edward Zajac atd. Umělci, kteří používali počítače v procesu tvorby umění, byli často nazýváni „počítačovými umělci“ (Verostko, *The Algorists*, nd).

Bell Labs, Americká společnost pro průmyslový výzkum a vědecký rozvoj, byla na počátku 60. let extrémně průkopnická v počátcích digitálního počítačového umění (A. Michael Noll), digitální počítačové animace (Edward E. Zajac, Frank Sinden a Kenneth C. Knowlton) a digitální počítačové hudby (Max V. Mathews a John R. Pierce). Bell Labs měla přístup k IBM 7090 (později aktualizovanou verzi IBM 7094, viz obrázek 1.) – sálovému počítači, který byl navržen pro rozsáhlé vědecké a technologické aplikace. V roce 1960 by se typický systém prodal za 2,9 milionu USD (což odpovídá 27,5 milionu USD v roce 2022) (*7090 Data Processing System*, nd). Zpočátku se badatelé zajímali o možnosti využití počítače v oblasti vizuální produkce, ale výsledky tvůrčí práce vytvořené počítačem v tomto období založily zcela novou generaci počítačových umělců.

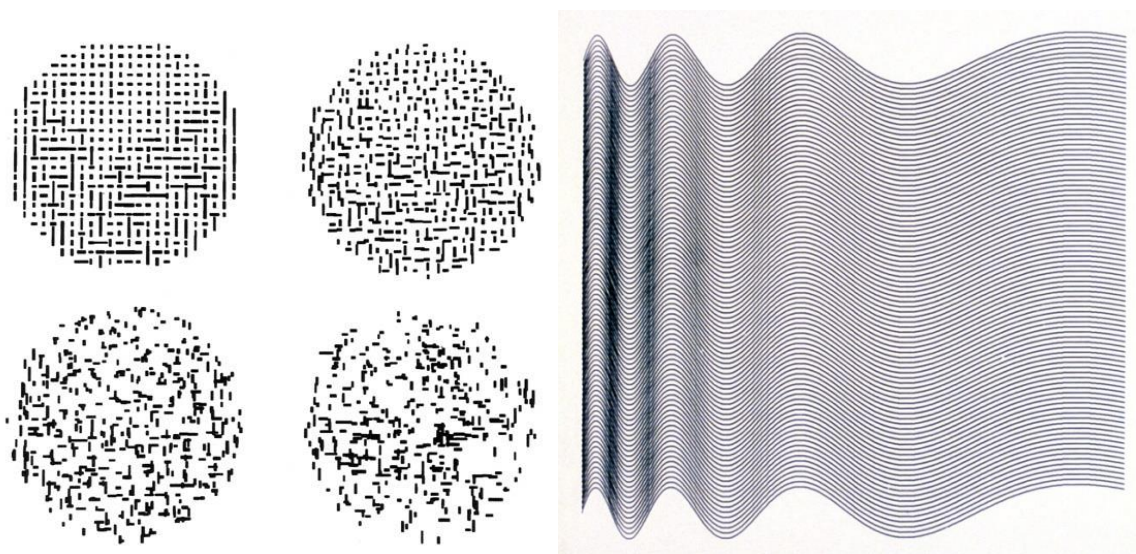


Obrázek 2: pohled na prostor strojovny IBM 7094. Zdroj: Columbia University Archive. <http://www.columbia.edu/cu/computinghistory/1965.html>

FORTTRAN byl prvním univerzálním programovacím jazykem vysoké úrovně, který je zvláště vhodný pro numerické a vědecké výpočty. Fortran byl původně vyvinut společností IBM v 50. letech 20. století pro vědecké a inženýrské aplikace a následně ovládl vědecké výpočty a dokonce i experimenty s výrobou počítačově generovaných uměleckých děl (Backus, 1981, s.6). Používá se již více než šest

desetiletí (dokonce i nyní je FORTRAN zřídka, ale stále používán v průmyslu) ve výpočetně náročných oblastech, jako je numerická předpověď počasí, výpočetní dynamika tekutin, geofyzika, výpočetní fyzika, sledování provozu, krystalografie a výpočetní chemie. Umělci jako Manfred Mohr a Roman Verostko používali FORTRAN k výrobě uměleckých děl, kombinujících strojový kód s plotry pro kreativní tvorbu (Zhai, 2020).

Michael Noll, americký inženýr a profesor, byl jedním z prvních průkopníků digitálního počítačového umění, 3D animace a hmatové komunikace (A. Michael Noll, nd). Noll strávil téměř patnáct let základním výzkumem v laboratořích Bell, kde studoval oblasti jako vliv médií na mezilidskou komunikaci, 3D počítačovou grafiku a animaci, hmatovou komunikaci mezi člověkem a strojem, zpracování řečových signálů a estetiku (Noll, 1964, s. 296-302). Jeho první práce byly naprogramovány v létě 1962, kdy Knoll použil počítač k vytvoření uměleckých vzorů a formalizoval použití náhodných a algoritmických procesů při tvorbě výtvarného umění (Noll, 1967, s. 65-79). Noll jeden z prvních navrhol, že by se digitální počítač mohl stát kreativním uměleckým médiem. Veškeré jeho digitální umění bylo naprogramováno v programovacím jazyce FORTRAN a dalších originálních balíčcích podprogramů, které sám vytvořil. Ukázky Nollových raných děl viz obrázky 3-4.



Obrázek 3: vlevo – Čtyři počítačem generované náhodné vzory založené na kompozičních kritériích Mondrianova "Composition With Lines". Zdroj: https://dam.org/archive/noll/artworks_04.htm

Obrázek 4: vpravo – Devadesát počítačově generovaných sinusoid s lineárně rostoucí periodou. Zdroj: <https://digitalartarchive.siggraph.org/artwork/a-michael-noll-ninety-computer-generated-sinusoids-with-linearly-increasing-period/>

Edward Zajac, inženýr a výzkumník, vytvořil první počítačově generovaný film v Bell Labs v roce 1963. Tento 1,25minutový film nazvaný „Systém řízení

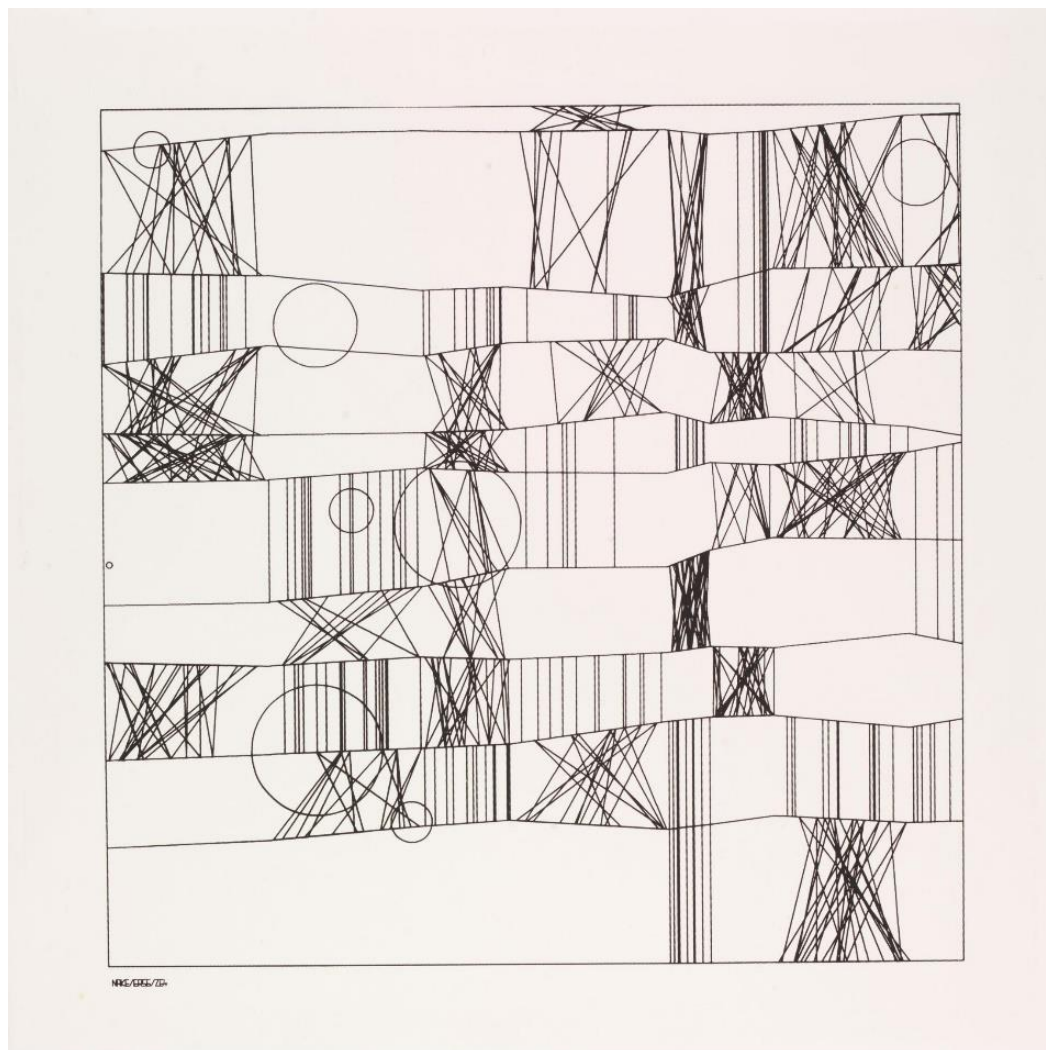
polohy se dvěma gyroskopy s gravitačním gradientem“ (v angličtině „A Two Gyro Gravity Gradient attitude control System“) demonstroval, jak by se určitý typ satelitu pohyboval vesmírem. Film, také namluvený Zajacem, simuluje pohyb a automatické otáčení komunikačního satelitu (viz obrázek 2.). "Zajac naprogramoval výpočty ve FORTRANu, pak použil program napsaný Zajacovým kolegou Frankem Sindinem, nazvaný ORBIT. Původní výpočty byly vloženy do počítače pomocí děrných štítků, poté byl výstup vytištěn na mikrofilm" (Norman, nd). Jeho rané počítačově animované filmy, které v dnešním světě vypadají zastarale a jednoduše, získaly ve své době velký ohlas a ocenění v USA i v jiných zemích a dnes jsou považovány za klasiku (*Edward E. Zajac*, nd).



Obrázek 5: A Two Gyro Gravity Gradient attitude control System. Zdroj: https://youtu.be/GBlQb6Me_1k

V Bell Labs v roce 1963 vynalezl Ken Knowlton – průkopník počítačové grafiky, výtvarník, mozaikář a portrétista, první programovací jazyk pro počítačovou animaci – BELFIX (vyvinutý na sálovém počítači IBM 7090 a založený na programovacím jazyce FORTRAN). Jak vysvětlují Nadia Magnenat-Thalmann a Daniel Thalmann: BELFIX podporoval dvě rozlišení pro výstupní snímky: jemné (252 x 184 pixelů) a hrubé (126 x 92 pixelů). Namísto surového programování BELFIX pracoval s jednoduchými „grafickými primitivy“, jako je kreslení čáry, kopírování oblasti, vyplnění oblasti, přiblížení oblasti, čtení/zápisu pixelů, perspektivní projekce a podobně (1985, s. 19-20). Tyto první kroky při používání programování ke generování uměleckých děl ukázaly potenciál generativního přístupu dalším umělcům.

Jedním ze zakladatelů „počítačového umění“, jak se mu v 60. letech říkalo, byl Frieder Nake, německý matematik a počítačový vědec. Svá první díla vytvořil v roce 1963. Od roku 1963 do roku 1969 prošel řadou stále složitějších programů, v nichž kombinoval kódové algoritmy, perové plotry a papír (Frieder Nake, nd). Obrázek 6. ukazuje příklad jeho práce z roku 1965 s názvem „Hommage à Paul Klee“ (Pocta Paulu Kleeovi).



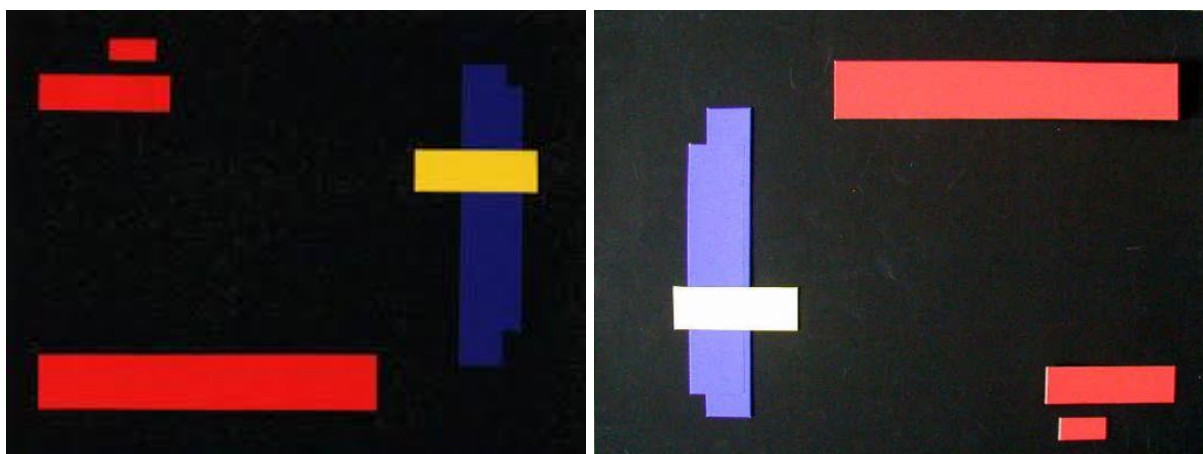
Obrázek 6: *Hommage à Paul Klee*, F. Nake. Zdroj: <https://collections.vam.ac.uk/item/O211685/hommage-a-paul-klee-13965-print-nake-frieder/hommage-%C3%A0-paul-klee-13965-print-nake-frieder/>

Obraz je založen na malbě Paula Klee s názvem „Big Roads and Lane“, 1929. Nake použil Kleeovu studii proporcí a vztahu mezi vertikálními a horizontálními liniemi malby jako výchozí bod pro svůj algoritmus, poté kresbu vygeneroval pomocí plotru. Nake použil svůj algoritmus k nastavení parametrů výkresu k určení horizontální a vertikální struktury, ale také přidal do procesu některé náhodné proměnné, které umožnily počítači učinit určité volby v rámci daného počtu možností (*Hommage à Paul Klee*, 13/9/65 Nr.2, 2009).

Nake významně přispěl k nejranějším projevům počítačového umění účastí na mezinárodních výstavách, jako jsou Cybernetic Serendipity (Londýn, 1968), Tendencias 4 (Záhřeb, 1968), Ricerca e Progettazione (35. bienále v Benátkách, 1970), Arteonica (São Paulo, Brazílie, 1971) (Paul, 2016, s. 57-58). Nike shrnul svou práci s algoritmy v roce 1974 kompletní knihou na toto téma — Ästhetik als Informationsverarbeitung (*Estetika jako zpracování informací*, Nike 1974).

K raným příkladům propojení umělců a programování patří umělecká skupina Compos 68, kterou založili Jan Baptist Bedaux, Jeroen Clausman a Arthur Veen v září 1968 v Utrechtu, kde všichni tři členové studovali. Skupina používala počítač Electrologica X8 Philips umístěný na katedře matematiky Utrechtské univerzity, hlavní myšlenkou bylo spojit estetiku modernistických maleb (jako Mondrian a Klee) a počítačovou simulaci rostoucích stromovitých struktur. Programovací příkazy byly napsány v jazyce ALGOL – algoritmickém jazyce, který Veen používal při svém biologickém výzkumu (*Compos 68*, 2012).

Umělecká díla Series 1, Series 2 a Hobby Box (viz obrázky 3, 4) - ručně barvené počítačové tisky vytvořené plotrem Electrologica X8, byly představeny v letech 1968-69. Fritz popisuje, že: „Bedaux, Clausman a Veen vytvořili systém přiřazování číselných hodnot „barvě“ a „formě“, aby získali čísla pro matematicky formulované estetické teorie“. Tyto rané práce představovaly kombinaci počítačově generovaných vzorů a finálních ručních úprav. Během krátké existence skupiny (1968-1969) stihli Compos 68 prezentovat svou tvorbu na několika významných mezinárodních výstavách počítačového umění, např. „Tendencias 4“ v Záhřebu a „Kunst und Computer“ ve Vídni, a získali významná ocenění (Fritz, 2011, s. 17).



Obrázek 7: vlevo – Hobby box. Zdroj: <http://dada.compart-bremen.de/item/artwork/1132>

Obrázek 8: vpravo – Hobby box 2. Zdroj: https://monoskop.org/Compos_68

Roman Verostko – americký umělec, je známý především svou prací s algoritmickým uměním. Verostko začal experimentovat se svými uměleckými

nápady a programováním na konci 60. let. Na základě těchto zkušeností později vyvinul vlastní program kombinující jej s kreslicím zařízením, dílo se jmenovalo „Kouzelná ruka náhody“ („The Magic Hand of Chance“). Program mohl ovládat kreslicí zařízení, "ruku" stroje známého jako perový plotr, který byl navržen především pro strojírenské a architektonické kreslení. Plotr obvykle kreslí inkoustovými pery, ale Verostko upravil štětce tak, aby vyhovovaly kreslicí ruce, a napsal kódové algoritmy pro napodobování tahů štětcem. Počítačový program řídil vzory, které se objevovaly na papíře (*Roman Verostko and the Cloud of Unknowing*, 2020).

V roce 1995 Verostko spolu s Jean-Pierrem Hébertem založil uměleckou skupinu Algoristé (Algorists). *"Algorista je každý, kdo pracuje s algoritmy. Historicky jsme na algoristy nahlíželi jako na matematiky. Platí to ale také pro umělce, kteří tvoří umění pomocí algoritmických postupů, které zahrnují jejich vlastní algoritmy,"* vysvětluje Verostko (*The Algorists*, nd). Ačkoli byl tento termín přijat v roce 1995, vztahuje se na práce kolegů Verostka, které pocházejí z 60. let a ještě dříve. V 60.-80. letech byl pojem „počítačové umění“ obecně používán pro veškeré umění spojené s počítači. *„Někteří z nás pracovali s algoritmickým postupem asi čtvrt století před naším prohlášením v roce 1995 za algoristy. Vytvoření skupiny „algoristé“ nebylo prohlášením něčeho nového; spíše to dávalo identitu umělecké praxi, která již přinesla radikální změnu a bude i nadále měnit způsob, jakým bychom tvořili umění v 21. století“*, říká Verostko (*The Algorists*, nd).

Jak ukazují fakta, mnoho umělců, kteří začali svou tvůrčí cestu s počítači a kódováním v 60. letech, byli inženýři nebo s nimi museli spolupracovat, protože s prvními dostupnými počítači nebylo snadné zacházet (nebo co je důležitější, získat k nim přístup). *Jsme ohromeni tím, co modely, které běží na počítačích, jsou schopny vyrobit. Náš respekt by však neměl být připisován počítačovému/softwarevému systému, ale lidským výzkumníkům a inženýrům, kteří byli dostatečně kreativní, aby toho dosáhli*, - říká Frieder Nake ve svém rozhovoru (Glenn, 2019, s. 4).

Algoritmické (počítačové) umění by však nemělo být zaměňováno s praxí matematiky. Proces psaní partitury pro kresbu vyžaduje poetické zapojení podobné tomu, které se vyžaduje při komponování partitury pro hudbu. Moje algoritmické kresby se vyvinuly z mé vášně jako malíře, pro snoubení spontánní práce se štětcem a studovaného aranžmá. Se základními programovacími schopnostmi jsem prozkoumal stejné cíle, které jsem si stanovil jako malíř. Je jasné, že programování a matematika umění nevytvářejí. Programování je nástroj, který slouží vizi a vášni umělce, který vytváří postup, vysvětluje Verostko (*The Algorists*, nd).

Harold Cohen, britský umělec, je považován za dalšího průkopníka počítačového a algoritmického umění. Cohen byl také inženýr, jehož přístup pomohl posunout první generaci počítačového umění. V roce 1971 byl Cohen pozván, aby strávil dva roky v Laboratoři umělé inteligence na Stanfordské univerzitě jako hostující vědec. Od té doby jeho dílo bylo zaměřené na integraci umělé inteligence s tvůrčím procesem vizuálního umělce. Proslavil se především vývojem systému AARON (práce na kterém začal ještě dříve, v roce 1968) – jednoho z nejdéle fungujících, neustále udržovaných systémů umělé inteligence v historii. AARON je schopen samostatně vytvářet originální umělecká díla, má svůj vlastní styl, který se v průběhu let mění (Greenberg, s.16). Kariéra Harolda Cohena trvala více než 60 let, jeho díla jsou prezentována ve velkých muzeích po celém světě.

Mnoho počítačových umělců první generace sdílí společnou historii, která vyrostla z konceptuálního umění a zájmu o nové technologie zpracování informací v 60. letech 20. století. Jejich dílo znamenalo zlom v dějinách západní kultury. Algoritmické programovací procedury změnilly vše od toho, jak děláme filmy a hudbu, až po to, jak komunikujeme, řídíme naši armádu a naše zdravotnické systémy.

Četní umělci, inženýři, programátoři a umělecké skupiny zabývající se počítačovým uměním v 60.-70. letech přispěly k pozoruhodnému nárůstu kódovaných postupů prakticky ve všech uměleckých sférách. Začaly se používat další termíny, ale všechny mají jedno společné – algoritmický postup. To platí pro generativní umění, kreativní kódování, kybernetická umění a také širokou škálu interaktivních děl a děl virtuální reality. Jak se toto 21. století rozvíjí, počítačové/algoritmické umění se ukazuje jako hlavní síla ve výtvarném umění (Verostko, nd).

2.6 Od generativního umění ke kreativnímu kódování

Technologie se neustále vyvíjely. Inovace, jako jsou klesající náklady na hardware a rostoucí miniaturizace počítačů, vytvořily v 70. letech nové trhy pro počítače a otevřely nové možnosti pro tvorbu. Soukromé kanceláře by mohly využívat výhod levnějších minipočítačů nebo snadněji získat přístup k sálovým počítačům. Hobbyisté, umělci a nadšenci do programování si také mohli zakoupit počítačové sestavy, konkrétně prodávané pro soukromé účely. Počátkem 80. let se objevila i první masová reklama na počítače. Počítače byly k dispozici pro osobní použití kvůli klesajícím nákladům na stroje (1970S - EARLY 1980S, nd). První dostupné počítače na počátku 80. let stály od 1600 do 6000 amerických dolarů. Jen málo z těchto raných verzí bylo uživatelsky přívětivých ve způsobech, které dnes lidé považují za samozřejmost (Walton, 2007).

Počítače se tak staly dostupnějšími a stále více umělců je začalo začleňovat do tvůrčího procesu. Místa pro vystavování počítačového umění se objevila v 80. letech 20. století, kdy začala různá symposia a konference, jak národní, tak mezinárodní, zahrnovat výstavy a příspěvky související s používáním počítačových postupů v umění. Leonardo, čtvrtletník o umění a technice, sehrál důležitou roli pro vznikající algoritmické umění tím, že zahrnoval eseje a dokumentaci různých výstav. Ars Electronica – kulturní, vzdělávací a vědecký institut v rakouském Linci začal pořádat své každoroční symposium, vytvářel katalogy a ceny, poskytoval veřejnosti příspěvky a příklady s rostoucí komunitou odborníků z praxe. Do konce 80. let 20. století vyvinulo značné množství umělců používajících algoritmy a kód charakteristické originální styly (Verostko, nd).

George Legrady (narozen 1950) – je multidisciplinární umělec digitálních médií a univerzitní profesor v oboru fotografie a umění výpočetních médií, byl jedním z těch, kteří měli snazší přístup k počítačům a začali je začleňovat do své práce. Narodil se v Maďarsku a v raném dětství se svou rodinou emigroval do Kanady, kde později studoval fotografii. Zájem George Legradyho se na počátku 80. let obrátil k počítačům, když začal vytvářet počítačem generované a digitálně upravované obrázky. Prostřednictvím programování začal svou fotografickou práci míchat s algoritmy. *„Jedním z mých cílů při práci s počítači a počítačovém programování bylo představit díla, která by nějakým způsobem testovala hranici mezi uvěřitelným a simulovaným; něco, co by vypadalo jako fotografie nebo vycházelo z fotografické technologie, ale ve skutečnosti by to bylo primárně výsledkem algoritmických procesů,“* - vzpomíná Legrady (Grandon, 2000).

Legrady byl také jedním z prvních umělců, kteří začali do svých děl vnášet interaktivitu. Mezi jeho nejvýznamnější projekty interaktivních digitálních médií patří „Anecdoted Archive from the Cold War“ vytvořený v roce 1993 - interaktivní instalační projekt založený na CD-ROM a počítačových médiích. Legrady vybudoval databázi, ve které se mohli uživatelé pohybovat a která staví do protikladu ikonografii a propagandu kapitalistického Západu s ikonografií a propagandou komunistického Východu. Pro prohlížení této databáze se uživatel pohybuje v osobních i historických dokumentech. Dílo je ukázkou toho, co konfiguruje člověka v daném historickém horizontu a ukazuje, jak se osobní občas prolíná s historickým (Legrady, nd).

Legradyho díla jsou součástí mnoha veřejných sbírek, jako je San Francisco Museum of Modern Art, Whitney Museum of Art, 21c Museum, Cincinnati, Pro Ahlers Art Foundation, Philbrook Museum of Art, Center for Art & Technology (ZKM), American Museum of Art, Smithsonian Institution ve Washingtonu, National Galleries of Canada, Musée d'art contemporain, Montreal a mnoho dalších (George Legrady, nd).

V současné době se Legradyho práce zaměřuje na umělecká a inženýrská studia v oblasti sběru dat, interakcí s mobilními kamerami, analýzy dat a vizualizace prezentované současně v interaktivních instalacích a na internetu. Projekty využívají samoorganizující systémy a algoritmicky generované vizualizace. Legrady je průkopníkem v integraci s výtvarnou fotografickou praxí (viz obrázek 9).

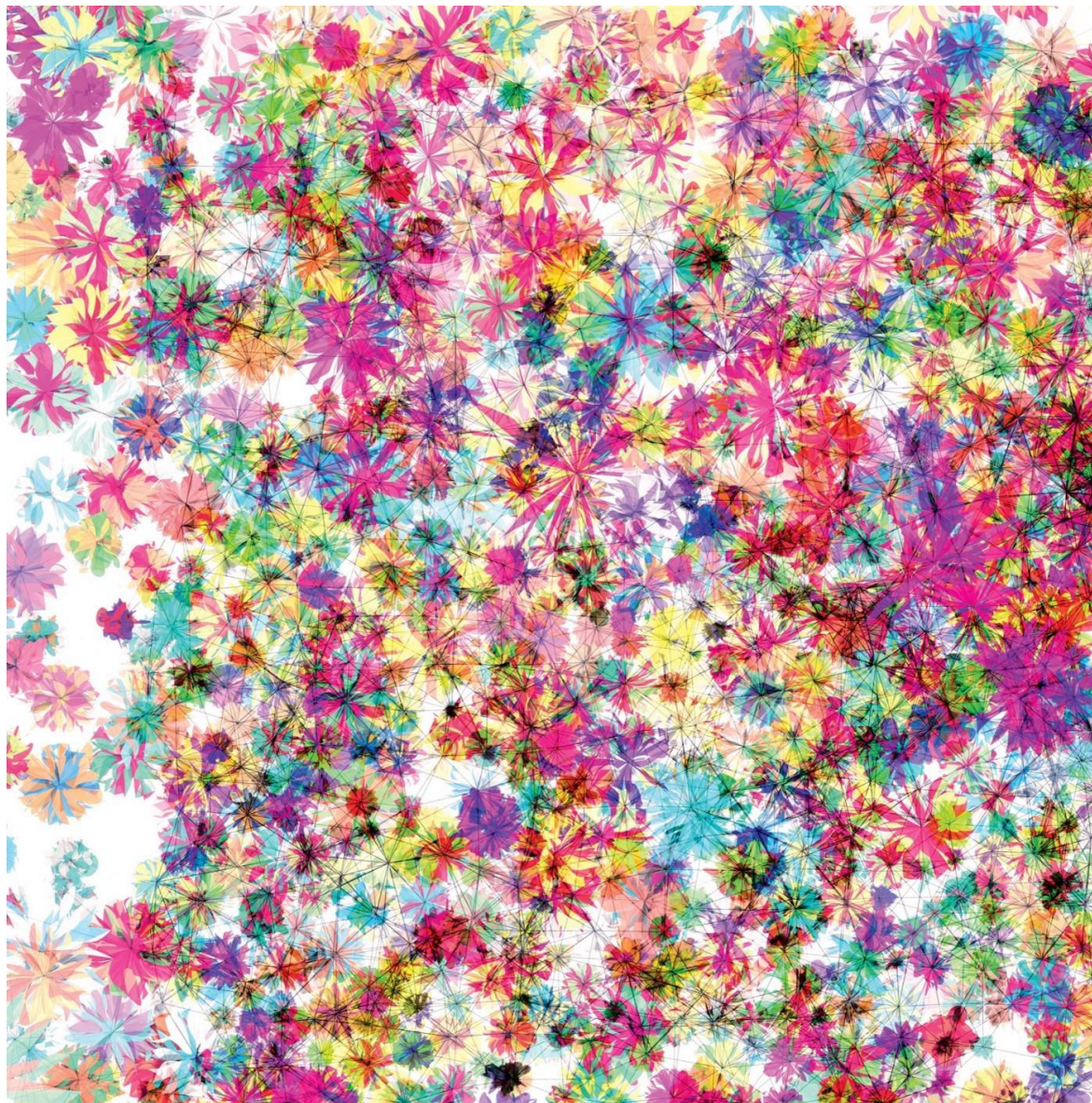


*Obrázek 9: Legrady: Anamorph-Visegrad (2020), 6 černobílých fotografií je komponováno na základě anamorfnní perspektivy a Voronoiovy mozaiky.
Zdroj: <http://www.georgelegrady.com/>*

Již byly zmíněny programovací jazyky 60. let FORTRAN, Algol, BELFIX. Od té doby bylo vyvinuto mnoho dalších jazyků, softwarových platforem. Nový jazyk se zpravidla vymýšlí, když konkrétní problém nelze elegantně vyřešit stávajícími nástroji. První jazyk inicializovaný pro kreativní kódování byl pravděpodobně Design By Numbers (DBN) od Johna Maedy.

John Maeda (narozen 1966) je americký technolog a designér, jeden z nejznámějších průkopníků digitálního umění na světě, jehož práce se zabývá obchodem, designem, technologií a jejich vzájemnou závislostí. Jeho studie, techniky a inovace významně ovlivnily sféru digitálního designu a zejména novou generaci kodérů-umělců. Jeho umělecká praxe kombinovala jak programovací, tak designérské dovednosti a vyvíjela nová média pro umělce (viz obrázek 10.). Maedovo použití elektronických médií jako nástroje pro uměleckou tvorbu

kombinací počítačového programování s tradičními uměleckými technikami položilo základy pro interaktivní pohyblivou grafiku, která je dnes na webu běžná (Jones, 2021).



Obrázek 10: John Maeda, Florida. Zdroj: <https://medium.com/verisart/john-maeda-a-pioneer-of-digital-art-5d8a43e0c16d>

Maeda byl profesorem na MIT Media Lab 12 let, kde podporoval komunitu designérů, kteří uměli kódovat, a inženýrů, kteří uměli navrhovat, nazvanou Aesthetics + Computation Group. Maeda uvádí: *v mediální laboratoři MIT Aesthetics + Computation Group pracujeme na návrhu pokročilých systémových architektur a myšlenkových procesů, které umožňují vytvářet (zatím) nepředstavitelné formy a prostory (The aesthetics + computation group, nd). Právě tam John Maeda v roce 1999 vytvořil Design By Numbers.*

Design By Numbers jsem vytvořil v době, kdy bylo módní zapojit do programování počítače umělce a designéry. Většinu první poloviny 90. let jsem strávil podporou toho, jak je důležité jít za hranice nástrojů a do prostředí samotného programování, - říká Maeda (Maedastudio, 2005).

Jak již bylo zmíněno dříve, většina počítačových / generativních uměleckých projektů dosud musela spojovat umělce s inženýry nebo vědci: umělec má koncepci a technický člověk poskytuje programovací postupy nebo umělec musí mít inženýrské vzdělání, aby mohl pracovat s komplikovanými stroji sám. Jazyk DBN měl umělcům usnadnit způsob, jak začlenit kód do své práce bez jakékoli další pomoci. Programovací jazyk a vývojové prostředí DBN byly dostupné na webu a bylo možné je volně stáhnout nebo spustit přímo v libovolném webovém prohlížeči s podporou JAVA. Tento jazyk, navržený pro umělce a designéry, měl velmi málo příkazů a sestával z prvků podobných těm z mnoha jiných jazyků, jako je C, JAVA, BASIC, Logo atd. Logika a „náročnost“ tohoto jazyka nepřesáhla středoškolskou algebru, jak píše Maeda ve své knize, kde popisuje hlavní principy a příkazy (1999, s. 3). Přestože DBN již není aktivním projektem, ovlivnil mnoho dalších projektů zaměřených na zpřístupnění počítačového programování pro netechniky. Jeho nejúspěšnějším potomkem je Processing – programovací jazyk a prostředí, které vytvořili Maedyho studenti Casey Reas a Ben Fry. S jistotou lze říci, že John Maeda stojí u zrodu moderního kreativního kódování. John Maeda jako umělec pomohl předefinovat programování jako nástroj uměleckého vyjádření.

2.6.1 Ben Fry

Jeden z klíčových kroků v historii kreativního kódování udělali Maedovi studenti Ben (Benjamin) Fry a Casey Reas, když oznámili vynález Processingu.

Benjamin Fry (1975) je americký odborník na vizualizaci dat. Se svým spolupracovníkem Casey Reasem (1972) představuje Fry moderní generaci kodérů a umělců, kteří hledají inspiraci ve výpočetních procesech a generativních strukturách. Fryova vášeň pro počítače a programování začala studiem konceptů spotřební elektroniky. Po získání všech dostupných znalostí v této oblasti Fry přešel na software. Počítačová technika přinesla několik otázek a problémů, Fry začal studovat programování, a to především studiem cizího kódu a jeho úpravou podle svých potřeb. Postupem času bylo snazší napsat svůj vlastní kód od nuly, což vedlo k Fryově plynulosti a pohodlí v kódování, jak je vidět v jeho práci, kterou lze považovat za poetické a pokročilé znázornění výpočetního procesu. Mnoho z jeho současných projektů zahrnuje vizualizaci velkých datových sad a dynamických informačních zdrojů (Reas C., Fry B., 2007, s. 1-2). Fry pracuje na nástrojích pro vizualizaci genetických dat a lidského genomu pro institut Ely a Edith Broad na MIT a Harvardu (Ben Fry, 2009). S tímto výzkumem souvisí i jeho osobní práce. Například jeho slavný projekt Valence, který představuje sadu

softwarových programů o budování reprezentací (vizualizace), které zkoumají struktury a vztahy uvnitř velmi rozsáhlých souborů informací. Project Genome Valence (viz obrázek 11.) je jednou z nejnámějších inkarnací tohoto softwaru. Vizualizuje biologická data a byl vytvořen pro výstavu Whitney Biennial ve Whitney Museum of American Art v roce 2002 (Fry, *Valence*, nd).



Obrázek 11: Ben Fry: *Genome Valence*. Zdroj: <https://benfry.com/genomevalence/>

Fry získal magisterský a Ph.D titul na MIT Media Lab pod vedením Johna Maedy. Jeho doktorská disertační práce „Computational Information Design“ (Design výpočetních informací) může být považována za zlomový bod ve vývoji kreativního kódování. Právě tato disertační práce odstartovala vznik Processingu.

Fryova disertační práce je zaměřena na rozsáhlé soubory dat a moderní metody jejich vizualizace pomocí programování; studie prochází různými sektory designu, informací a výpočtů a zaměřuje se na to, jak se vzájemně podporují. Schopnost shromažďovat, ukládat a spravovat data rychle roste, ale naše schopnost jim porozumět zůstává nezměněna. Ve snaze získat lepší porozumění datům se používají disciplíny, jako je vizualizace informací, dolování dat a grafický design, z nichž každá řeší izolovanou část určitého problému, ale selhává v širším smyslu: existuje příliš mnoho nevyřešených problémů komplexní

vizualizace dat. Jako řešení tato disertační práce navrhuje sloučení jednotlivých oborů v rámci uniformního procesu nazvaného Design Výpočetních Informací. Aby byl tento proces přístupný širšímu publiku, byl představen nástroj, který začátečníkům zjednodušuje výpočetní proces. Nakonec Fry popisuje řadu příkladů, které ukazují, jak lze metodologii a nástroj použít k řešení řady problémů s daty, zejména lidského genomu (Fry, 2004).

Hlavním zdrojem inspirace pro Benjamina Frye a jeho diplomovou práci je biologie. Během desetiletí studií a vědeckých experimentů se biologie rychle stala vědou bohatou na data, kde množství shromážděných dat může přesáhnout rychlost, s jakou je lze analyzovat a následně pochopit. Každý den se shromažďují miliardy a miliardy písmen genetického kódu.

Jedním z hlavních problémů, kterým musel Fry čelit, bylo, jak získat smysluplné informace z datových sad. Množství takových dat ztěžuje získání úplného přehledu a pochopení jejich významu. Problém dále zhoršuje neustále se měnící charakter dat, výsledek přidávání nových informací nebo neustálé zpřesňování starších informací. Velikost databází vyžaduje nové softwarové nástroje a její složitost vyžaduje zvláštní pozornost ve vizuální reprezentaci, aby se zvýraznily prvky podle jejich důležitosti, odhalily se vzory v datech a zároveň se ukázaly vlastnosti existujících dat, která jsou součástí více dimenzí.

Vizuální design – praxe mapování dat do vizuální podoby – pomáhá pochopit, ale obvykle neřeší, jak zacházet s extrémně velkým množstvím dat. Techniky dolování dat mohou zpracovávat velké množství dat, ale nejsou odpojeny od prostředků pro interakci s nimi. Softwarová informační vizualizace přidává stovební kameny pro interakci a reprezentaci různých typů abstraktních dat, ale estetické principy vizuálního designu jsou obvykle považovány za méně důležité nebo dokonce povrchní, spíše než za to, že jejich síla je nezbytná pro efektivní komunikaci, vysvětluje Fry (2001, s.11).

Tato disertační práce je kombinací několika typicky rozdělených disciplín s cílem zaměřit se na proces porozumění datům. Fry představuje a vysvětluje sedm fází vizualizace dat: získávání, analýzu, filtrování, dolování, reprezentaci, zpřesňování a interakci (Fry, 2001, s. 13).

Jedna z hlavních kapitol, která je důležitá pro další studium, se nazývá „Proces“, prochází celým procesem vizualizace dat, krok za krokem, podrobněji popisuje proces tvorby návrhu výpočetní informace. Klíčovým cílem této disertační práce je co nejvíce zpřístupnit tento proces co nejširšímu publiku, a to jak prostřednictvím krátkých úvodů do různých oblastí v textu, tak prostřednictvím online programovacího nástroje – programovacího jazyka a prostředí „Processing“. Při pohledu na tuto disertační práci o dvacet let později lze s jistotou říci, že Fryovy ambice byly plně uspokojeny, protože jeho software

a přístup otevřel dveře tisícům a tisícům umělců, kodérů, designérů a programátorů, kteří pomocí Processing vytvářejí četná umělecká díla a vizualizace.

Disertace Benjamin Fryho je skvělým příkladem dlouhého procesu studia, analýzy a vývoje softwaru. Nejen Fry navrhl nový způsob práce s vizualizací velkých dat, ale také šel mnohem dále. Jednou z výzev této práce bylo zlepšení existujících programovacích nástrojů. Jednou stranou této výzvy bylo vytvořit jednodušší nástroje, které budou rychlejší pro vytváření prototypů a psaní softwarového kódu a zpřístupnit je lidem ze všech druhů specializací. Druhou stranou bylo, že nástroje musí být výkonnější a musí umožňovat sofistikovanější práci. Softwarové nástroje pro design (jako Illustrator, PageMaker a Fontographer) vedly k revoluci v rozvržení, designu a typografii, „Processing“ je stejným průlomem v oblasti vizualizace dat.

Disertační práce Benjamin Fryho je i 20 let po svém vzniku stále důležitá, protože slouží jako první krok ve studiu digitálních vizualizací velkých souborů dat. Fry vynaložil velké úsilí na to, aby se pokusil spojit vědu, programování a umění. Jeho tvorba je zaměřena spíše technicky než umělecky, ale byla zásadním krokem ve studované disciplíně. Vývojem svého softwaru Processing dal Fry příležitost tvořit každému, kdo se zajímá o disciplínu kreativního kódování. Praktická část disertační práce zohlední Fryovy zkušenosti a bude pokračovat v jeho práci s větším důrazem na výtvarnou stránku procesu.

2.6.2 Casey Reas

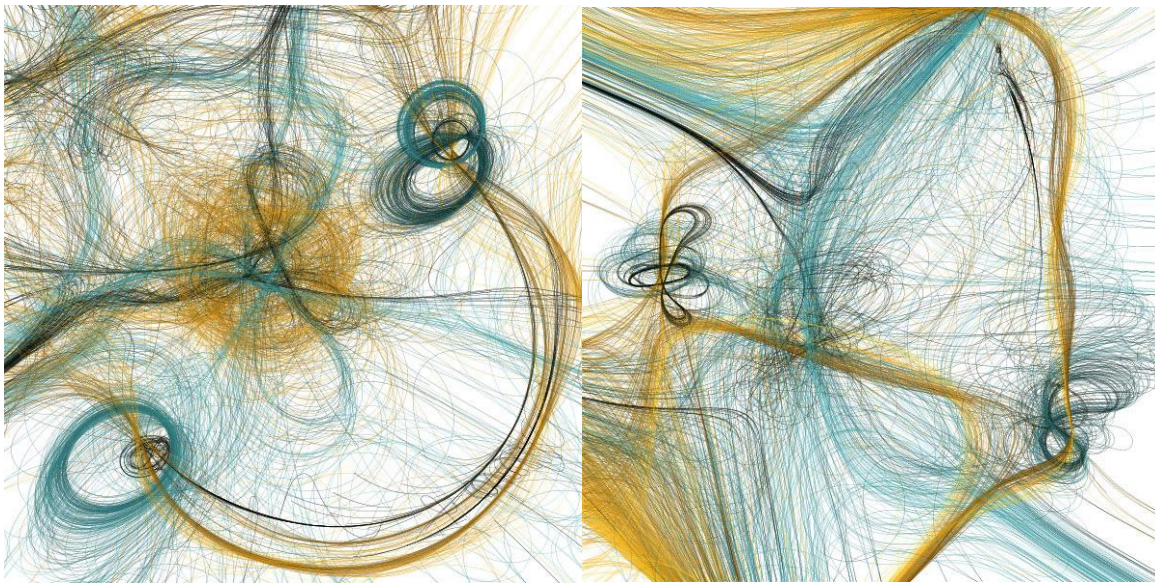
Casey Reas, americký umělec a designér, je další klíčovou postavou mezi kreativními kodéry. Jeho inovativní přístup k používání softwaru vedl k mnoha uměleckým dílům, která ukazují krásu kódu a algoritmických technik. Reas působí jako profesor a přednášející na Kalifornské univerzitě v Los Angeles. Získal bakalářský titul v oboru mediální umění a vědy na Vysoké škole designu, architektury, umění a plánování na univerzitě v Cincinnati a poté magisterský titul na MIT (*Casey Reas*, nd).

Asi nejvýznamnější příspěvek do světa generativního umění přinesl Casey Reas spolu se svým kolegou Benem Fryem. Jak již bylo zmíněno, společně vytvořili populární programovací jazyk s otevřeným zdrojovým kódem a prostředí pro vizuální umění s názvem „Processing“, projekt, který začal v roce 2001.

Reas používá kód jako primární médium tvorby. Píše software pro zkoumání vznikajících propojených systémů. Jeho práce spojuje konceptualismus, teorii systémů, experimentální film a animaci a kresbu. Zatímco jádrem jeho praxe je

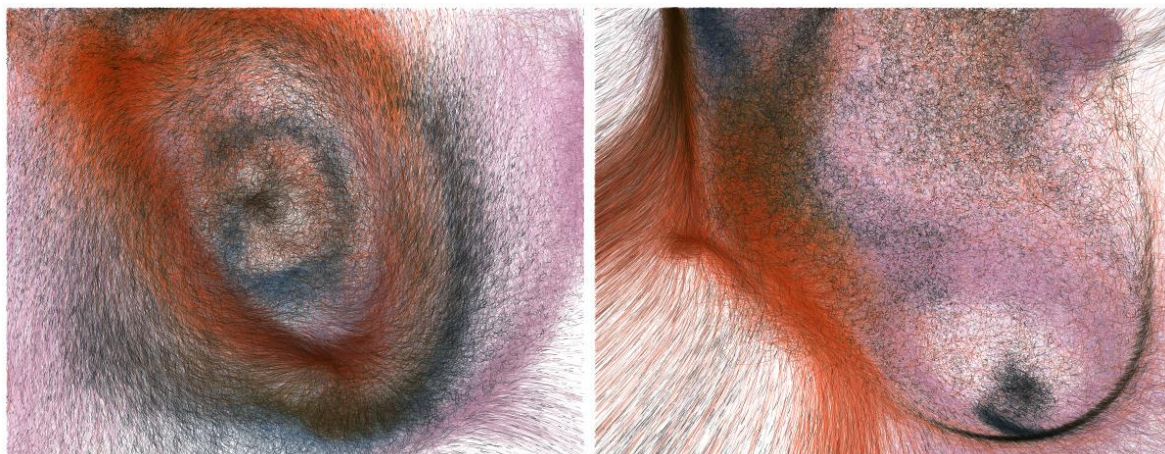
software, jeho práce zahrnuje instalaci, práce na papíře a živé vystoupení (*Reas, biography, nd*).

Ve své kariéře Reas ovšem vytváří sérii tisků na základě vlastního vyvinutého softwaru. Jako příklad může sloužit řada výkresů *The Path* (cesta), které byly vytvořeny z původního softwaru *Path* napsaného v roce 2001 (viz obrázek 12). Tento software je simulací, která znázorňuje pohyb strojů Valentina Braitenberga. Braitenberg byl neuroanatom, který vyvinul sadu minimálních strojů k prozkoumání myšlenek o vývoji nervového systému. Každý řádek výkresu odhaluje historii pohybu jednoho simulovaného stroje, když se pohybuje ve svém prostředí (*Reas, nd*).



Obrázek 12: Casey Reas: *The Path*. Zdroj: https://reas.com/path_p/

Tisky *MicroImage A* (viz obrázek 13) je série osmi obrázků, které dokumentují software *MicroImage*. Stejně jako série prací *Path* je *MicroImage* postaven na myšlenkách neuroanatomy Valentina Braitenberga (*Reas, The MicroImage A prints, nd*).



Obrázek 13: Casey Reas: *MicroImage A*. Zdroj: https://reas.com/microimage_a_p/

2.7 Kreativní kódování v moderní době

Digitální technologie jsou ještě docela mladé, počítač je teprve v prvním století od svého vzniku. Posledních několik desetiletí proměnilo několikatunové, nedosažitelné drahé složité stroje na samozřejmou věc každodenní potřeby, ke které má přístup téměř každý. Nejnovější software dává možnost tvořit každému, kdo je ochoten se učit. Tisíce lidí se snaží otestovat své dovednosti ve světě kreativního kódování. K dispozici je mnoho knih, průvodců a článků. Lidé sdílejí své zkušenosti prostřednictvím webových portálů, online galerií a fór.

V současné době lze použití kreativního kódování aplikovat na jakoukoli podsekcí umělecké praxe. Patří sem hudba, grafika, design, instalace, divadelní představení, interaktivní umělecké objekty, filmový průmysl, reklama, zábava, vzdělávání, firemní identita. Každý den přináší nové technologie, které jen obohacují potenciál pro další rozvoj kódového algoritmického umění. K vytváření originálních uměleckých děl se používá řada programovacích a skriptovacích jazyků, jejichž výběr závisí na rozsahu jejich příkazů, složitosti, možnostech a samozřejmě na předchozích zkušenostech a znalostech umělce v oblasti programování.

Generativní design mění samotné jádro toho, jak design funguje: představte si svět, který vám umožňuje vytvářet příběhy, návrhy a identity značek, které jsou flexibilní, reaktivní, živé a dýchající. Přemýšlejte o dynamických identitách značek, které se v reálném čase přizpůsobují základnímu obchodnímu modelu společnosti, myslte na obaly, které individuálně odrážejí a přizpůsobují se jedinečné směsi historických faktorů životního prostředí, myslte na stovky tisíc jedinečných variací designu vlastní značky šitých na míru každému příjemce, myslte na interaktivní a flexibilní vizuální systémy, které lze použít v celém spektru dotykových bodů. Tvůrčí potenciál je nekonečný, - vysvětluje Patrik Hübner, německý kreativní kodér, který využívá potenciálu generativního

designu při tvorbě materiálů, návrhů, nápadů pro značky, agentury a instituce (Hübner, nd).

Patrik Hübner rozvíjí příležitosti pro značky, aby se odlišily vyprávěním jedinečných a smysluplných příběhů, aplikací nových a experimentálních technik kódové kreativity při práci s generativními a interaktivními designovými systémy. Hübner spolupracoval s mnoha předními mezinárodními agenturami a jeho práce získaly celosvětové uznání a řadu mezinárodních ocenění za design, jeho práce byly prezentovány v kontextu designových festivalů, audio živých vystoupení a interaktivních zážitků (*The one show*, 2020).



Obrázek 14: Patrik Hübner – BRUTE. Zdroj: <https://www.patrik-huebner.com/work/brute-data-driven-wine-brand/>

Patrik našel způsob, jak začlenit svou vášeň pro kódové umění do podnikání v oblasti korporátního a značkového designu. Obrázek 14 ukazuje příklad takového díla. S pomocí JavaScriptu – programovacího jazyka pro web, a vlastního algoritmu Patrik vygeneroval vysoce unikátní obalový design založený na datech o počasí. BRUTE je značka vína, jejíž design destiluje základní prvky počasí jako vítr, déšť a slunce do holistického vizuálního a smyslného zážitku. K utváření své identity používá historická data a data o počasí v reálném čase. Takže jak se každý ročník vyvíjí s klimatem, vyvíjí se i značka. Vizuální styl značky a následné balení je založeno na simulaci částic v reálném čase s desítkami tisíc jednotlivých prvků.

Může generovat nekonečné interpretace stejného algoritmu. Designový systém lze vidět v práci online, kde si každý divák může upravit potenciální povětrnostní podmínky a vytvořit si svůj vlastní jedinečný design (Hübner, nd).

JavaScript používá nespočet umělců a kodérů v mnoha oblastech výtvarného umění, je to také jedna ze základních technologií World Wide Web, která byla vydána v roce 1995, aby se vypořádala s problémem webových stránek, které v té době mohly být pouze statické a postrádaly schopnost dynamického chování po načtení stránky v prohlížeči (*Netscape and Sun announce JavaScript*, 1995). JavaScript umožňuje přidat webové stránce dynamické chování a speciální efekty.

Chelley Sherman ze San Franciska má jiný přístup. Její tvorba se řídí vzory a texturou, které evokují temnou, tajemnou energii. Její praxe se skládá z experimentů v rozšířených a virtuálních realitách, interaktivních zvukových instalacích a performancích. Pomocí výpočetních metod a výzkumu kognitivních procesů vnímání Sherman zkoumá vztah mezi světlem a zvukem a výsledné psychoakustické efekty v prostorových zvukových architekturách a krajinách. Sama Shermanová rozděluje svou tvorbu do tří kategorií: audiovize, virtuální realita a instalace. Kombinuje využití programů pro grafiku a 3D objekty, manipulaci s videem a zvukem, jako je Cinema4D, Substance Designer, s programovacími jazyky a platformami, jako je Processing, TouchDesigner, GLSL, Arduino, JS a další (Sherman, nd).

Sherman říká, že její zkušenost se skoliózou na počátku jejího života vážně ovlivnila to, jak vnímala své tělo, a že jedna z jejích prvních zkušeností ve VR (virtuální realitě) zcela změnila její smysl pro identitu a reprezentaci těla. Sherman viděla objemové zachycení vlastního těla ve VR pomocí technologie Depthkit, která jí dokázala poskytnout zcela nový pohled na tělo v podobě, kterou popisuje jako formu zrcadlové terapie pro fantomové končetiny. Od té doby posouvá hranice vnímání tvorbou umění, které využívá VR k prozkoumání průřezu mysli a těla, se zvláštním zaměřením na terapie traumatu (#797: *VR Artist Chelley Sherman: Heightening Perception with Sonic Architecture, Haptics, & Dynamical Cognitive Science*, 2019).

Jedním z jejích děl s názvem „Dispersion“ (viz obrázek 15.) je audiovizuální instalace ve virtuálním prostředí. Sama popisuje: *je to zvlňné zjevení, které se proplétá ve snovém prostoru a vizualizuje hybnost zvuku vyzařovaného z jeho geometrie. Jeho iluzorní efekty jsou nenápadně vnímány, i když dočasně dezorientovány různými zónami jeho architektury, v závislosti na tom, jak se divák s konstrukcí zapojí a vytváří dojem změn tlaku. Jak se rozptýlí do oblastí za slyšitelným poklesem diváka, zůstávají iluze nepřetržitého zvuku, kde mysl zaplňuje mezery. Tyto artefakty skriptují choreografii a vnímaný pohyb v rámci struktury, stejně jako tělesně pociťovaný rytmus, způsob, jakým mohou vibrační prostory vyvíjet fyzickou odezvu na vnímané zrychlení* (Sherman, 2018).



Obrázek 15: Chelley Sherman – Dispersion. Zdroj: <https://vimeo.com/280815065>

Sean Zellmer je kreativní technolog, jehož cílem je ukázat složitost reality, která nás obklopuje. Abstraktním konceptům a objektům dává formu prostřednictvím technologie, otevírá okno do jiné dimenze a staví sémantický most k ostatním, aby vstoupili do této říše. Se vznikajícími webovými technologiemi, jako je WebVR a WebGL, přináší Sean vysokorozměrné objekty přímo na obrazovky. Seanova práce je silně ovlivněna jeho studiem matematiky a filozofie, používá čísla a kód jako médium k vyjádření své životní filozofie a zkoumání deterministických systémů chaosu (Zellmer, nd).

WebGL (zkratka pro webovou grafickou knihovnu) je technologie, která umožňuje zobrazovat různé druhy 3D grafiky v reálném čase v prohlížeči (Tavares, 2012). Výsledný objekt nebo scéna, vytvořená pomocí programovacích příkazů, může být tak složitá, jak její autor zamýšlel, od jednoduchých geometrických tvarů až po detailní mapy Vesmíru. Obrázek 16 ukazuje jeden ze Seanových experimentů s touto technologií.

Sean prohlašuje, že chce inovovat při navrhování a výrobě holistických webových aplikací a webů, aby vyřešil nové složité problémy pro klienty, aby posunul technologii vpřed. Chce být zastáncem progresivních změn a urychlit je vývojem vysoce kvalitních produktů, které nejen odvedou svou práci, ale udělají to elegantně jak pro uživatele, prostřednictvím zjednodušených pracovních postupů a stručné prezentace informací, tak pro ostatní vývojáře prostřednictvím čitelného kodu (Zellmer, *LinkedIn*, nd).



Obrázek 16 Sean Zellmer. Zdroj: <https://www.fffutures.com/p/forked-and-merged-memories?s=r>

Alida Sun je umělkyně a intersekcionalní futuristka se sídlem v Berlíně i New Yorku. Sun je jedním z prvních moderních umělců, kteří integrovali generativní umění, rozsáhlé instalace, živé vystoupení a blockchain technologii.

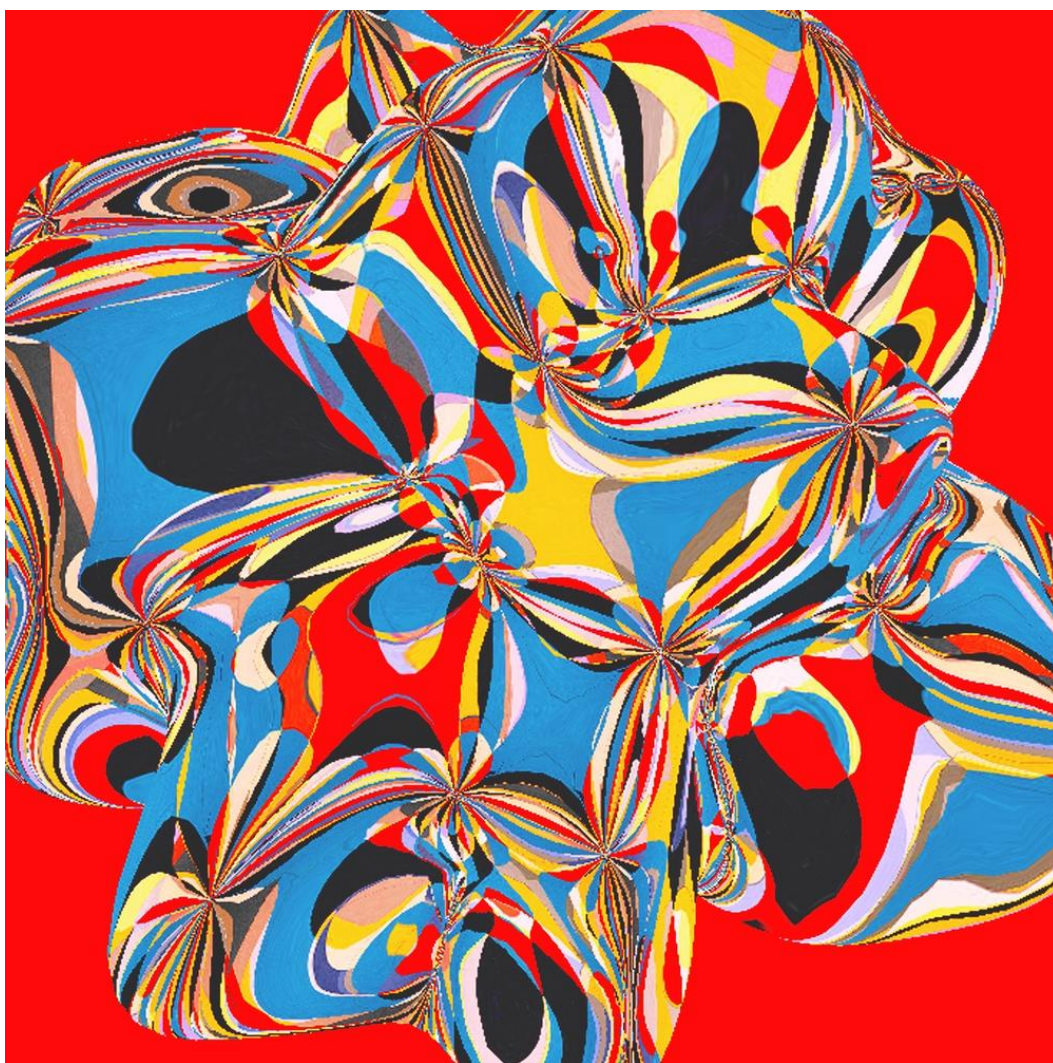
Její současná studiová praxe se zaměřuje na asambláž, dynamiku tekutin, časové krystaly a experimentální humanitní vědy (*Alida Sun: Details for Alida Sun*, nd). Skutečná inovace často pramení z neustálého zkoumání. V samotném jádru Suniny praxe generativního umělce je nezastavitelná touha prozkoumat neomezené možnosti algoritmického umění. Interdisciplinární umělkyně vytvořila přes sto kusů kódového umění. Její inspirace sahají od matematiky po mytologii. Její raná tvorba byla zaměřena na projekční mapování (videomapping). Projekční mapování využívá každodenní videoprojektory, ale místo promítání na

ploché obrazovky nebo stěny je světlo mapováno na jakýkoli povrch a přeměňuje běžné objekty jakéhokoli 3D tvaru na interaktivní displeje. Jinými slovy, projekční mapování je zobrazení obrazu na nerovném nebo nebílém povrchu (*What is projection mapping?*, nd). Taková praxe může, ale nemusí souviset s kreativním kódováním, protože zdroj promítaných obrázků může, ale nemusí být založen na kódu.

Později ve své kariéře se Sun obrátila ke generativnímu umění pomocí open source sad nástrojů, jako je Processing, OpenFrameworks, Pure Data a tak dále. Samouk, pokud jde o kódování, Alida se více zabývá procesem než estetikou. Futuristický a průsečíkový umělec věří, že dovednosti a citlivost zdokonalovaná v tradičnějších médiích – kresbě, malbě, sklářství, animaci, textilu, sochařství – jsou stejně důležité jako zvládnutí nových médií a nových technologií (Bhagat, 2022).

Sun je také NFT umělec. **NFT** je zkratka pro Non-Fungible Token (nezaměnitelný token) - jednotka dat uložená na typu digitální účetní knihy zvané blockchain (systém zaznamenávání informací způsobem, který ztěžuje nebo znemožňuje změnu, hackování nebo podvádění systému.), kterou lze prodávat a obchodovat (*What is blockchain?*, nd). Nezaměnitelný zde znamená, že je jedinečný a nelze jej nahradit něčím jiným. NFT může být cokoli digitálního, jako jsou kresby, hudba, animace atd. Jakékoli digitální dílo, jako je obrázek nebo video, lze stáhnout, kopírovat a sdílet, včetně umění, které je součástí NFT. Ale NFT jsou navrženy tak, aby vám daly něco, co nelze zkopírovat: vlastnictví díla. Řečeno z hlediska fyzického sbírání umění: kdokoli si může koupit výtisk Chagallova uměleckého díla, ale pouze jedna osoba může vlastnit originální dílo (Clark, 2021).

Technologie blockchain a NFT poskytují umělcům, designérům, hudebníkům a všem kreativním jednotlivcům jedinečnou příležitost zpeněžit svou práci. Umělci se například již nemusí při prodeji svých uměleckých děl spoléhat na galerie, aukční domy nebo obchody s uměním. Místo toho jej může umělec prodat přímo spotřebiteli jako jedinečný digitální soubor NFT, což mu umožňuje ponechat si větší zisk. Kromě toho mohou umělci naprogramovat licenční poplatky způsobem, který jim zajistí určité procento z prodeje, kdykoli se jejich umění prodá novému vlastníkovi. NFT mohou mít v jednu chvíli pouze jednoho majitele. To je atraktivní funkce, protože umělci obvykle nedostávají budoucí výnosy poté, co je jejich umění poprvé prodáno (Conti, Schmidt, 2022). Ceny se mohou vyšplhat až na několik milionů amerických dolarů za jedinečný kousek. Například krátké animované video od Bepple (pseudonym Mikea Winkelmana, amerického digitálního umělce) se prodalo za cenu 6,6 milionů dolarů, ačkoli je volně dostupné online (Clark, 2021).



Obrázek 17: Alida Sun – Chromacounterpane #2: pandau. Zdroj: <https://makersplace.com/0x5fcd30a446d465ebb0ff8db85feb19fe98d758f5/chromacounterpane-2-pandau-1-of-1-79092/>

Alida Sun nabízí svá digitální umělecká díla na MakersPlace – online galerii, která prodává a propaguje unikátní digitální výtvoř, podepsané a vydané tvůrcem, pomocí technologie blockchain. Například její umělecké dílo s názvem „Chromacounterpane #2: pandau“ (viz obrázek 17). Jak sama umělkyně vysvětluje, dílo ukazuje jedinečnou iteraci a zkoumání quiltových tradic, které zahrnují divoce podceňované odvětví asamblážového umění, integrované s barevnými studii a dynamikou tekutin (Sun, 2022).

Dalším agitátorem moderních technologií, strojového učení a generativního umění je Marpi Studio, založené v San Franciscu v roce 2019 Marpim, kreativním technologem a umělcem pocházejícím z Polska. Skupinu tvoří šest umělců, techniků a designérů. Ať už se jedná o vlastní režii nebo zakázku, jejich umělecká díla pro veřejný, soukromý a digitální prostor oživují prostředí, předměty a chování pomocí kódu. Studio se snaží začlenit skutečné publikum, aby s ním

spoluvytvářelo, tím, že vstupuje do digitálních instalací, kde se interakce publika mění v avatary, gesta a fyziku, které mění dílo hravými a nečekanými způsoby (*Marpi Studio*, nd).

Marpi Studio ve svých uměleckých dílech využívá strojové učení k vytvoření něčeho skutečně jedinečného. Strojové učení je studium počítačových algoritmů, které se mohou automaticky zlepšovat prostřednictvím zkušeností a pomocí dat. Studie je vnímána jako součást umělé inteligence (Mitchell, 1997). Jinými slovy, strojové učení je věda o tom, jak přimět počítače, aby jednaly, aniž by za ně explicitně programovaly každý krok. V posledním desetiletí nám strojové učení dalo samořídící auta, praktické rozpoznávání řeči, efektivní vyhledávání na webu, personalizovaný online obsah a reklamu a výrazně zlepšilo porozumění lidskému genomu. Strojové učení je dnes tak všudypřítomné, že jej lidé pravděpodobně používají mnohokrát denně, aniž by o tom věděli. Mnoho výzkumníků si také myslí, že je to nejlepší způsob, jak dosáhnout pokroku směrem k umělé inteligenci na lidské úrovni (Ng, nd).

Mezi uměním a technologickými inovacemi existuje zralá tradice práce, která sahá až do 60. let minulého století. Dnes jsme svědky vzniku nových uměleckých forem, v nichž je umělá inteligence a strojové učení nástrojem nebo tématem, a stále více umělců experimentuje s umělou inteligencí, aby podpořili, zlepšili, simulovali nebo replikovali kreativitu. Je poněkud mýtus, že umělou inteligenci a strojové učení lze použít k samostatné tvorbě uměleckých děl, k vytváření umění bez umělců. Někdo (umělec nebo kodér nebo oba současně) stále stojí za procesem a vytváří původní algoritmus nebo sadu příkazů, které počítač používá v tvůrčím procesu.

Příkladem takového uměleckého díla může být „New Nature Digital Petting Zoo“ (viz obrázek 18) od Marpi studios. Jde o výstavu představující pohlcující digitální prostředí přírodou inspirovaných tvorů a rostlin, které ožily díky strojovému učení. Tato digitální „stvoření“ nebo rostliny reagují a přizpůsobují se v reálném čase přítomnosti a pohybu každého návštěvníka v galerii. Generativní vzory vytvořily virtuální terária různých krajin a organismů, které se neustále vyvíjejí, jak se návštěvníci pohybují po prostoru galerie (*New Nature*, 2019).



Obrázek 18: Marpi Studio – New Nature Digital Petting Zoo. Zdroj: <https://www.artechouse.com/program/new-nature/>

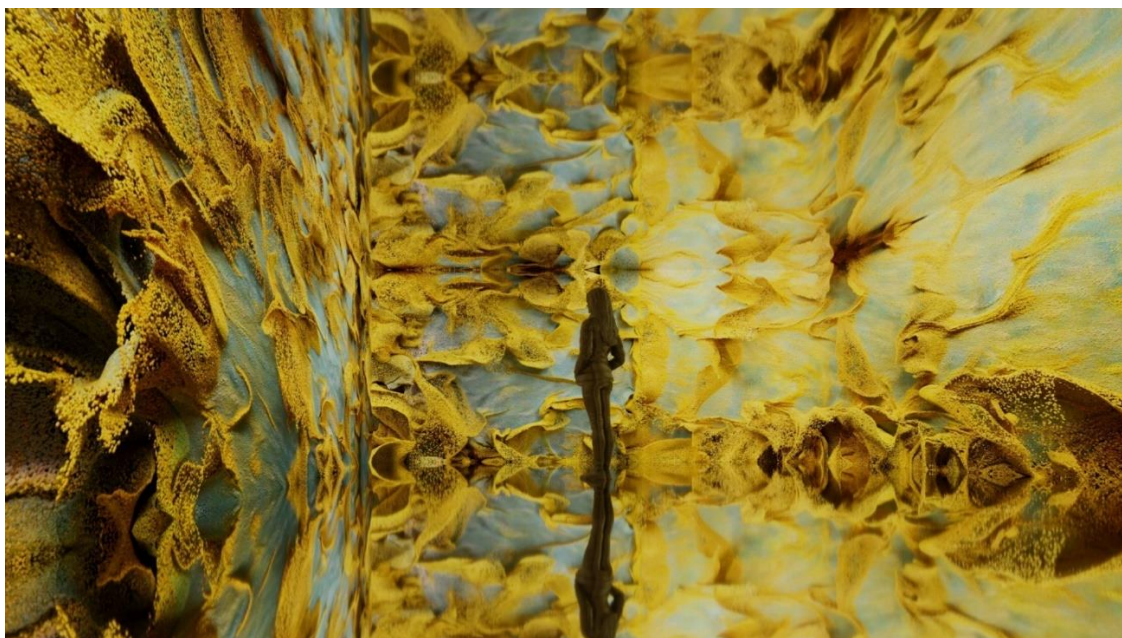
Dalším digitálním umělcem, který používá kód a technologie umělé inteligence, je Refik Anadol. Anadol je mediální umělec tureckého původu, režisér a průkopník v estetice strojové inteligence. V současnosti žije v Los Angeles v Kalifornii, kde vlastní a provozuje Refik Anadol Studio, které objevuje a vyvíjí průkopnické přístupy k datovým narativům a umělé inteligenci. Práce Anadolu se zabývá výzvami a možnostmi, které všudypřítomná výpočetní technika lidstvu vnutila, a tím, co to znamená být člověkem ve věku umělé inteligence. Zkoumá, jak se vnímání a prožívání času a našeho prostředí radikálně mění nyní, kdy stroje dominují našemu každodennímu životu. Anadol je fascinován způsoby, jakými digitální věk a strojová inteligence nabízejí novou estetickou techniku k vytvoření obohaceného pohlcujícího prostředí a jak to vše mění naši zkušenost s uměním. Anadolovy trojrozměrné datové sochy, živé audio a vizuální performance a pohlcující instalace, umístěné na křižovatce umění, vědy a techniky, nabývají různých virtuálních a fyzických podob. Umělec využívá celé podlahy budov k tomu, aby jeho umění ožilo, *kde stěny a stropy mizí donekonečna, dechberoucí estetika se tvoří z velkých pásem dat a to, co bylo kdysi lidskému oku neviditelné, se stává viditelným a nabízí divákům nový pohled na jejich příběh a příběh jiných světů* (Anadol, 2022).

Obrázky 19 a 20 představují jedno z posledních děl Anadolu, nazvané Machine Hallucinations – Space: Metaverse. Jde o umělecké NFT dílo, založené na filmovém zážitku s umělou inteligencí, ve kterém se prolíná minulost a budoucnost vesmírného průzkumu. Dílo představuje 3kapitolové, 15minutové multidimenzionální kino v metaverse – síti 3D virtuálních světů zaměřených na

sociální propojení. Metaverze je často popisována jako hypotetická iterace internetu jako jediného univerzálního virtuálního světa, do kterého lze vstoupit pomocí náhlavních souprav pro virtuální a rozšířenou realitu. Virtuální prostředí, do kterého může jednotlivec vstoupit, místo aby se na něj jen díval na obrazovce (O'Brien a Chan, 2021).

S využitím VR setu mohou diváci „vstoupit“ do Anadolova uměleckého díla a vybudovat si zážitek v architektonickém prostoru, kde se mohou napojit na mysl stroje a spekulovat o alternativním vnímání prostoru. Aby bylo možné vizuálně dosáhnout této komplexní prostorové struktury, je zážitek rozdělen do tří kapitol, z nichž každá se zaměřuje na rozsáhlé soubory dat shromážděné z vizuálních vzpomínek vesmíru zachycených teleskopy Hubble a MRO.

Jednou ze základních otázek, kterou si práce Refik Anadol klade již více než deset let, je možnost navazování vztahů se vzpomínkami, informacemi, databázemi a historií uloženými v nesčetných archivech, ať už digitalizovaných nebo ne. Jeho vize vztahu mezi umělou inteligencí, velkými daty a uměním je následující: *s tolika nástroji, které máme k dispozici, od umělé inteligence po strojové učení, od nejnovějších technik vizualizace dat po uměleckou kreativitu, je odpovědností a zároveň neomezeným spouštěčem představivosti zachovat data o vztazích mezi vesmírem, strojem a člověkem a přenést je do budoucnosti (Machine Hallucinations – Space : Metaverse, 2021).*



Obrázek 19: Refik Anadol – *Machine Hallucinations – Space: Metaverse*. Zdroj: <https://refikanadol.com/works/machine-hallucinations-space-metaverse/?i=d>



Obrázek 20: Refik Anadol – *Machine Hallucinations – Space: Metaverse*. Zdroj: <https://refikanadol.com/works/machine-hallucinations-space-metaverse/?i=d>

Seznam vynikajících technologií, a ještě význačnějších moderních počítačových umělců by mohl pokračovat donekonečna. Technologie se každým dnem vyvíjejí neuvěřitelnou rychlostí, stejně jako způsoby lidské kreativity. Můžeme si jen představovat, jak se v příštích desetiletích změní chápání umění, jaké další žánry vizuální kreativity se budou rozvíjet. Je zásadní zachovat otevřenou mysl, aby společnost mohla mít příležitost ocenit další inovativní formu umění. Kreativní kódování přináší revoluci v tom, co umění je a čím by se mohlo stát. Vzhledem k tomu, že ve fázi našeho vývoje vstupujeme dále do digitálního světa, kreativní kódování může být současným uměleckým hnutím, které potřebujeme k řešení hlavních společenských výzev, kterým čelíme s technologickým pokrokem.

3 NÁSTROJE A PROGRAMOVACÍ JAZYKY KREATIVNÍHO KÓDOVÁNÍ

Pokud jde o moderní nástroje kreativního kódování, umělci i programátoři si musí vybrat z toho, co je k dispozici. Když se umělci seznámí s určitým programovacím prostředím, mohli by přistoupit k vývoji svých vlastních algoritmů, příkazů a softwarových programů, ale rozhodující je výběr počátečního programovacího jazyka. To, co funguje nejlépe, by mělo být určeno tím, čeho chce umělec dosáhnout, pokud je cílem vytvořit interaktivní instalaci nebo 3D scénu pro virtuální realitu nebo projekt vizualizace dat atd., proto je nezbytné důkladně uvažovat o cíli, než se zaměříme na konkrétní sadu nástrojů.

Potenciálně lze pro kreativní účely použít jakoukoli technologii nebo programovací jazyk, historie kreativního kódování to dobře dokazuje, nicméně určité knihovny, software a programovací prostředí byly speciálně vytvořeny tak, aby napomáhaly rychlému prototypování a vývoji kreativních děl, zvláště když umělec má menší nebo nulové zkušenosti s programováním. Mezi často používané softwarové sady nástrojů v této souvislosti patří:

- Processing (Java nebo Python)
- p5.js (JavaScript)
- Three.js (JavaScript)
- D3.js (JavaScript)
- Shaders (WebGL)
- cables.gl (WebGL)
- Python
- openFrameworks (C++)
- Cinder (C++)
- OpenRNDR (Kotlin)
- Max MSP (Vizuální programovací jazyk)
- PureData (Vizuální programovací jazyk)
- vvvv (Vizuální programovací jazyk)
- Babylon.js (TypeScript, JavaScript)

Ze seznamu je zřejmé, že různá programovací prostředí mohou být založena na stejných programovacích jazycích. Jak vyplývá z teoretické analýzy literatury, článků, praktických příkladů mnoha moderních kreativních kodérů a vlastních zkušeností s programováním, nejoblíbenějšími kreativními kódovacími jazyky, které budou podrobněji studovány, jsou: **Processing, JavaScript, WebGL**.

Důležité jsou nejen softwarové komponenty, ale také hardwarová zařízení. Praxe kreativního kódování často zahrnuje hardwarové nástroje pro vkládání dat z okolí, interakci s účastníky a promítání výsledků kódování do prostředí. Příklady běžně používaného hardwaru zahrnují kamery (počítačové webové kamery, hloubkové kamery, infračervené kamery), mikrofony, senzory detektorů pohybu, MIDI ovladače, projektory, LED pásy, tiskárny a plotry. Některé programovací jazyky mají jednodušší funkce a knihovny, aby byly kompatibilní s takovými zařízeními, což je také třeba vzít v úvahu při výběru jazyka.

3.1 JavaScript

Populární volbou pro tvorbu umění pomocí kódu je JavaScript (JS). JS je programovací jazyk, který je vedle HTML a CSS jednou ze základních technologií World Wide Web. Všechny hlavní webové prohlížeče mají vyhrazený JavaScriptový engine pro spouštění kódu na zařízeních uživatelů (Flanagan, 2011, s. 1). JavaScript je nejoblíbenější programovací jazyk na světě. JavaScript již běží v jakémkoli prohlížeči na počítači uživatele, na tabletu a na chytrém telefonu. JavaScript je zdarma k použití pro každého (*JavaScript Tutorial*, nd).

JavaScript se poprvé objevil v prohlížeči Netscape 2.0 v roce 1995 pod názvem LiveScript. Jazyk je lehký a nejčastěji používaný jako součást webových stránek, jejichž implementace umožňují skriptům na straně klienta komunikovat s uživatelem a vytvářet dynamické stránky (*JavaScript overview*, nd). Byl navržen tak, aby přinesl více interaktivity do webových stránek, a proto se stal dalším nástrojem pro kreativní účely. Za zmínku stojí, že JavaScript a Java jsou naprosto nesouvisející programovací jazyky. Důvodem, proč je JavaScript pojmenován po Javě, je nešťastné brandingové a marketingové rozhodnutí učiněné v 90. letech, kdy popularita Javy exponenciálně rostla (Modrzyk, 2018).

Neexistují prakticky žádná omezení toho, co může kodér dělat s JavaScriptem na webové stránce, pokud jde o interaktivitu. Zde je několik příkladů toho, k čemu se JavaScript používá:

- Přidání interaktivního chování na webové stránky – animace libovolných prvků, změna barev, zobrazení/skrytí prvků, přiblížení nebo oddálení obrázku, interaktivní galerie obrázků atd.

- Vývoj her – JavaScript lze použít k vytvoření interaktivních her v prohlížeči.
- Tvorba webových a mobilních aplikací. Existuje mnoho JavaScriptových frameworků (kolekce knihoven kódů JavaScriptu, které obsahují předem napsaný kód k použití pro rutinní programovací funkce a úkoly), které mohou vývojáři použít k vytváření webových a mobilních aplikací.
- Tvorba webových serverů a vývoj serverových aplikací. Kromě webových stránek a aplikací mohou vývojáři také používat JavaScript k vytváření jednoduchých webových serverů a vyvíjet serverové aplikace pomocí Node.js – serverového prostředí s otevřeným zdrojovým kódem (*What is JavaScript used for?*, 2021).

Tato schopnost vnést interakci a pohyb do jakýchkoli objektů na webové stránce je to, co umožňuje použití JavaScriptu pro kreativní kódování – umělci a programátoři mohou kódovat animace, objekty, které reagují na kurzor myši nebo stisknutí tlačítka nebo klávesy. Mnoho takových projektů je uloženo online s otevřeným zdrojovým kódem, například webový portál CodePen je populární online komunita a galerie, kde lze takové příklady nalézt.

CodePen je sociální vývojové prostředí. Většina příkladů kódu zde je veřejná s otevřeným zdrojovým kódem, takže s nimi mohou ostatní lidé a komunita komunikovat, od jednoduchého oslovení přes zanechání komentáře až po kopírování a změny pro vlastní potřeby. Umožňuje komukoli psát kód v prohlížeči a vidět jeho výsledky okamžitě během psaní. CodePen se zaměřuje především na jazyky jako HTML, CSS, JavaScript. Nástroj lze používat zdarma (existuje také premium verze s rozšířenými funkcemi), je to užitečný a rychlý online editor kódu pro tvůrce s jakýmkoli programátorskými dovednostmi, oblíbený mezi lidmi, kteří se učí kódovat (*CodePen, About*, nd).

JavaScript nebyl původně vytvořen pro potřeby kreativního kódování, ale jak již bylo zmíněno, má četné knihovny a frameworky vyvinuté speciálně pro tyto účely. Tyto dodatečné nástroje používají kodéři a umělci k vytváření kreativních uměleckých děl. Framework je struktura, na které lze stavět software, slouží jako základ, takže programátor nemusí začít psát program úplně od nuly. Každý framework má sadu přednastavených příkazů a nástrojů navržených pro konkrétní účel. Programovací knihovnu lze považovat za sbírku kódu, která je vytvořena pro konkrétní účel, takže kdykoli kodér potřebuje provést úlohu, která se vztahuje k tomuto účelu, může použít nástroj z knihovny místo toho, aby si tuto funkci vytvořil sám. Knihovny jsou založeny na základních schopnostech jazyka a rozšiřují je. Pro JavaScript existuje více než sto tisíc knihoven, které umožňují provádět širokou škálu operací (Modrzyk, 2018). Zde jsou některé názvy

frameworků a knihoven vytvořených pro umělecké účely: p5.js, three.js, D3.js, Tensorflow.js, brain.js, Convnet.js, synaptic.js, babylon.js, WebGL Frameworks, Movement.js a mnoho dalších (Mandis, 2021).

3.2 Processing

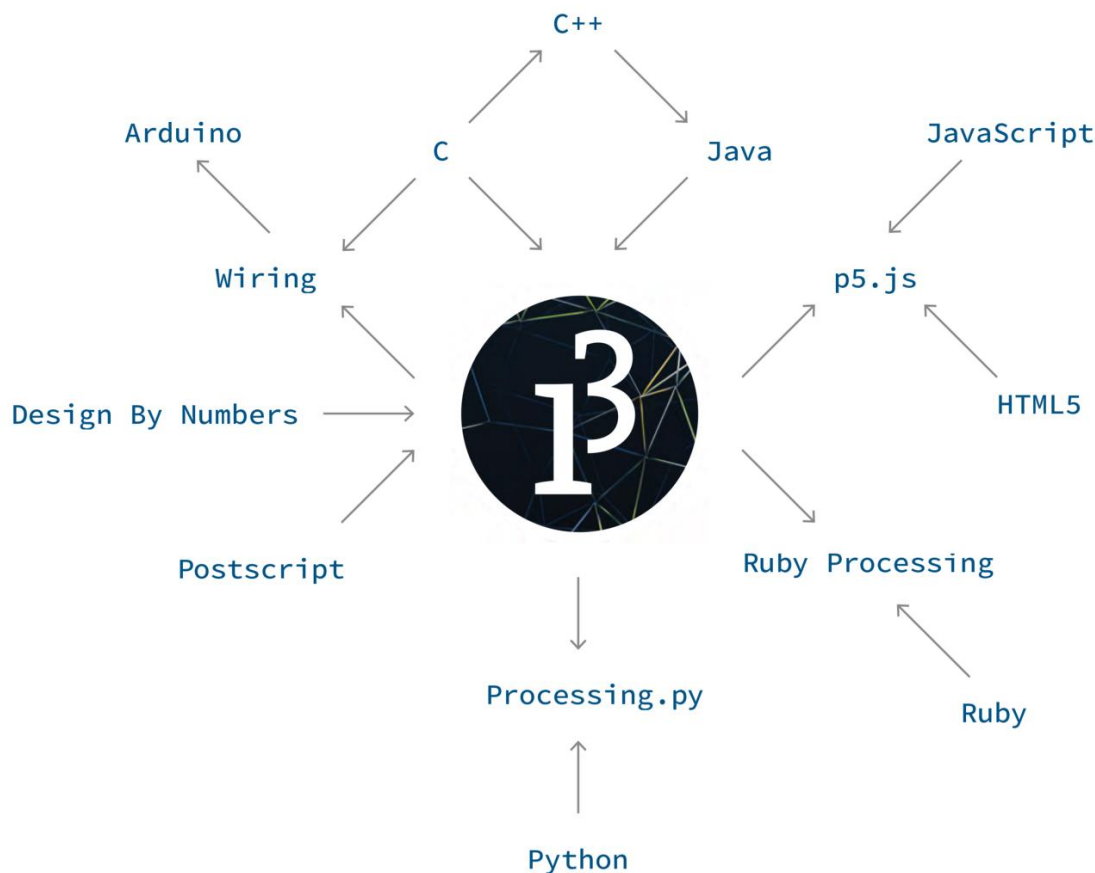
Je pravděpodobné, že nejoblíbenějším a nejznámějším nástrojem pro kreativní kodéry je Processing. Processing je open-source software (software s otevřeným zdrojovým kódem), programovací jazyk a prostředí, které lze spustit na Mac OS, Windows a Linux. Projekt iniciovali v roce 2001 Ben Fry a Casey Reas, oba bývalí členové MIT Media Lab v rámci výzkumné skupiny Johna Maedy pro estetiku a výpočty. Oba zakladatelé uvádějí, že Processing vyrostl z projektu Johna Maedy Design By Numbers, který byl vyvinut v Media Lab v roce 1999. Processing byl speciálně navržen tak, aby byl co nejjednodušší pro začínající programátory a umělce, s vědomím, že jeho jednoduchost bude přínosem i pro zkušenější uživatele. V průběhu let se skutečně vyvinul ve vývojový nástroj pro profesionály. Jazyk Processing byl inspirován bezprostředností dřívějších jazyků, jako jsou BASIC a Logo, a také zkušenostmi Frye a Rease s výukou programování pro širokou škálu studentů s různým profesním zázemím.

Software Processing používají tisíce designérů, umělců, architektů, hudebníků a kodérů jako nástroj pro tvorbu. Processing lze implementovat v každé oblasti kreativity a vizuálního designu: pro scénický design pro taneční rutiny a živá vystoupení, která se promítají na scénu; pro tvorbu obrázků, animace pro hudební videa a filmy; pro tvorbu herního designu a prostředí; pro tvorbu obrázků pro etikety, loga, plakáty, časopisy a knihy; pro tvorbu interaktivních webových stránek a pro vytváření interaktivních instalací různých velikostí v galeriích, muzeích a na ulici. Četná muzea (jako je Exploratorium v San Franciscu, Muzeum moderního umění v New Yorku, Victoria and Albert Museum v Londýně, Centre Georges Pompidou v Paříži a mnoho dalších významných míst) využívají Processing k rozvoji svých výstav a k demonstraci projektů, které byly s ním vytvořeny. (*Overview: A short introduction to the Processing software and projects from the community*, 2021).

Processing umožňuje vytvářet programy a aplikace v rámci principů interaktivity, pohybu a vizuality. Tento systém je směsí programovacího jazyka, softwarového prostředí a vzdělávací platformy. Hlavním účelem programovacího jazyka Processing je vytvářet a upravovat vizuální média. Tento software kombinuje jednoduché grafické uživatelské rozhraní a pokročilou sadu funkcí. I člověk s nulovými zkušenostmi s programováním bude schopen vytvořit první program po přečtení jednoduchých pokynů dostupných na oficiálních stránkách Processingu.

Processing dokáže vykreslit téměř jakýkoli typ digitálních médií: vektor, rastrové obrázky, 3D modely, animace, síťovou komunikaci, vizualizaci dat. Podporuje mnoho vstupních a výstupních zařízení: myš, klávesnici, touchpad, kamery, senzory, audio zařízení atd.

Není pochyb o tom, že Processing předvádí prvotřídní výsledky, ale pravděpodobně nejdůležitější na tomto nástroji je to, jak přilákal novou generaci vizuálních umělců, aby považovali programování za nedílnou součást své tvůrčí praxe.



Obrázek 21: Processing byl ovlivněn jinými kódovacími systémy a ovlivnil další.

Zdroj: <https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85>

Processing má řadu dalších funkcí, které z něj dělají jedinečné prostředí svého druhu:

- platforma s otevřeným zdrojovým kódem;
- zcela zdarma, včetně bezplatného stahování softwaru;
- více než 100 knihoven pro rozšíření funkčnosti, včetně příkladů;

- vhodné pro Windows, Mac OS X, GNU / Linux, Android, ARM;
- interaktivní programy s 2D, 3D nebo PDF výstupem;
- Integrace OpenGL pro zrychlenou 2D a 3D grafiku;
- desítky dostupných knih, článků a pokynů;
- online portál OpenProcessing s tisíci příkladů programování.

Program vytvořený pomocí Processing se nazývá "skica". Pro experimentování s Processingem má uživatel několik možností: buď si stáhne software do počítače z oficiální webové stránky a bude vytvářet skici přímo na počítači; nebo si založí účet na online portálu OpenProcessing, kde je možné vytvářet, upravovat a testovat skici, stejně jako prohlížet, kopírovat a upravovat skici stovek dalších uživatelů. Obě možnosti jsou zcela zdarma.

Skicy vytvořené v Processing lze exportovat jako EXE aplikace pro Windows, Linux a Mac OS X. Dalším způsobem exportu projektu je uložit jej jako webovou stránku. Bylo to zmíněno několikrát, ale je nesmírně důležité, že všechny programy vytvořené online a stovky dalších příkladů a knihoven jsou k dispozici s otevřeným zdrojovým kódem, což znamená, že si jej může kdokoli prohlížet, kopírovat a upravovat, vytvářet něco nového, pokračovat ve vývoji práce někoho jiného, a tedy vytvářet mezinárodní a mezioborovou spolupráci. Především díky této metodě se Processing stal velmi populárním. Webové servery hostí tisíce projektů s přístupem ke zdrojovému kódu. "Naučili jsme se kódovat z velké části tím, že jsme se podívali na kód ostatních, a chtěli jsme, aby tato otevřenost byla ústředním bodem projektu" - říkají jeho zakladatelé (Reas, Fry, *A Modern Prometheus*, 2018).

Processing není jazyk vytvořený od nuly, je to hybrid mezi autentickými prvky a programovacím jazykem Java. Protože Processing je stvořen pro tvorbu vizuálního umění, jazyk obsahuje prvky speciálně pro práci s formou, tvary, barvami, obrázky, animacemi atd. Zároveň lze v Processingu použít jakýkoli kód, který lze použít v Javě. Hlavní myšlenkou je usnadnit vytváření vizuálních prvků, ale také umožnit zkušenějšímu programátorovi dělat komplikované věci v rámci stejné platformy (Reas, Fry, *A Modern Prometheus*, 2018).

Od samého počátku se Processing rozrostl a rozvinul do různých směrů. Právě teď to není jen jazyk, ale spíše umělecky orientovaný přístup k učení, výuce a procesu tvorby vizuálního umění s kódem.

3.3 p5.js

p5.js je jednou z dříve zmíněných JavaScriptových knihoven vytvořených speciálně pro kreativní účely, je založen na Processingu a je jeho interpretací pro moderní weby. p5.js, stejně jako původní Processing, se zaměřuje na to, aby bylo kódování přístupné a inkluzivní pro každého: umělce, designéry, architekty, pedagogy, začátečníky. Je zdarma a s otevřeným zdrojovým kódem, protože jeho tvůrci věří, že software a nástroje k jeho učení by měly být přístupné všem (*p5js*, nd).

Výhodou použití programovacího jazyka JavaScript je jeho široká dostupnost, oblíbenost a všudypřítomná podpora: každý webový prohlížeč má vestavěný interpret JavaScriptu, což znamená, že programy p5.js (také nazývané “skici“) lze spustit na jakémkoli webovém prohlížeči. p5.js je rozšíření JavaScriptu, což je základní jazyk, který poskytuje všechny nástroje pro zabudování jakýchkoli funkcí do webu. Tato knihovna navíc poskytuje některé funkce, které uživatelům JavaScriptu usnadňují kreslení a vytváření uměleckého obsahu online (Sarthak, 2021).

Podle Modrzyka (2018), jedním z nejnáročnějších aspektů učení se programování je najít poutavé příklady, které jsou nejen kreativní a působivé, ale také ilustrují probíranou látku. Používání JavaScriptové knihovny p5.js je skvělé při výuce a učení programování. p5.js umožňuje vytvářet poutavé interaktivní a vizuální kousky a také vybudovat pevný základ pro vývoj softwaru. Vizuální povaha této knihovny umožní vidět výsledky ze skriptů jako grafiku, a tak rozvíjet důvěrné porozumění programovým strukturám. p5.js, stejně jako Processing, je založen na principu skicování. Zde přichází srovnatelnost kódování a kreslení: stejně jako lze skicování považovat za minimální přístup ke kreslení pro sdílení nápadů a návrhů, tak i p5.js je postaven na konceptu psaní kódu s minimálním množstvím a složitostí pro přenos vizuálních nápadů na obrazovku počítače nebo později na projekci, instalaci atd.

p5.js jako rozšíření JavaScriptu nemá omezené možnosti. Tato knihovna má za sebou působivé množství funkcí, historie a mezinárodní komunity, což z ní dělá cennou investici do učení a sofistikovaný nástroj pro ty, kteří vytvářejí nebo chtějí začít vytvářet vizuální umění, design, weby nebo interaktivní díla pomocí kódu. Program vytvořený pomocí p5.js může obsahovat od několika řádků kódu až po tisíce (Modrzyk, 2018).

3.4 WebGL

WebGL (Web Graphics Library) je JavaScriptové aplikační programovací rozhraní pro vykreslování interaktivní 2D a 3D grafiky v jakémkoli moderním webovém prohlížeči bez použití dalších skriptů (Tavares, 2012). Programy

WebGL se skládají z řídicího kódu napsaného v JavaScriptu a kódu shaderu napsaného v jazyce OpenGL ES Shading Language (GLSL ES) a běží na GPU počítače, takže pro rychlejší vykreslování je potřeba lepší grafická karta. WebGL vyvíjí a spravuje nezisková skupina Khronos Group (Delmon, 2018).

WebGL je nový standard pro 3D grafiku na webu. Je navržen tak, aby vykresloval 2D a 3D grafiku a objekty a učinil je interaktivními v kompatibilních webových prohlížečích. Kodéři a umělci pomocí tohoto nástroje mohou vytvářet obrázky jakékoli složitosti, od jednoduchých geometrických až po fotorealistické, rotující objekty, 3D scény, každý z nich může být interaktivní - bude reagovat na počítačovou myš nebo klávesnici, zvuk, mikrofon. WebGL je také navržen tak, aby urychlil funkce webových editorů obrázků a jejich efektů, stejně jako fyzikální simulace. K zobrazení výsledku nebo vykreslení takového objektu/scény není třeba instalovat žádné další zásuvné moduly, což umožňuje uživatelům webu užívat si interaktivní obsah online bez dalších komplikací.

WebGL má několik významných výhod, mezi které patří (*WebGL Tutorial*, nd):

- Snadné nastavení. WebGL je integrován v HTML5, není potřeba žádné další nastavování a instalace. K napsání aplikace WebGL použije kódér textový editor podle svého výběru a moderní webový prohlížeč.
- WebGL je uložen s otevřeným zdrojovým kódem. Je snadné získat přístup ke zdrojovému kódu knihovny a pochopit, jak funguje, co umí a jak byla vyvinuta.
- Programování v JavaScriptu – WebGL je součástí rodiny JavaScript, programy jsou v podstatě psány v JavaScriptu. Takové programy a aplikace mohou snadno interagovat s ostatními prvky webové stránky a jejím HTML kódem.
- WebGL je kompatibilní s dalšími četnými JavaScriptovými knihovnami a technologiemi HTML, které mohou aplikaci obohatit a rozšířit její potenciál.
- Mobilní prohlížeče – WebGL nadále pracuje na podpoře v mobilních prohlížečích, je přístupný prostřednictvím prohlížečů jako iOS Safari, Android Browser a Chrome pro Android.
- Žádná kompilace – Pro spuštění skriptu není potřeba kompilovat soubor, protože se jedná o kombinaci JavaScriptu a HTML. Místo toho lze soubor přímo otevřít pomocí libovolného prohlížeče a zkontrolovat výsledek. Kompilace je proces převodu programovacího jazyka do strojového jazyka, kterému počítač rozumí (*Compilation*, 2017). Jednoduše řečeno, tento proces vyžaduje další čas potřebný k dokončení

celého kroku kompilace před testováním a pokaždé, když došlo ke změně kódu, proces kompilace by měl být spuštěn znovu.

Mezi nevýhody WebGL patří jeho složitost ve srovnání s dříve zmíněnými programovacími jazyky a knihovnamí. Rovněž se uvádí, že je odpovědný za pády prohlížeče způsobené nekompatibilitou pluginů.

3.5 GLSL Shaders

OpenGL Shading Language (GLSL) je vysokoúrovňový jazyk pro vytváření shaderů se syntaxí založenou na programovacím jazyce C. Jak vysvětlil Hergaarden (2011, s. 2), shader je část kódu, která se spouští na grafickém procesoru (GPU), který se obvykle nachází na grafické kartě, za účelem manipulace s obrazem před jeho vykreslením na obrazovku. Shadery umožňují různé druhy vykreslovacích efektů, jako je přidání světelných efektů, všechny druhy textur nebo přidání kreslených obrysů do grafického výstupu.

Existuje několik různých druhů shaderů, ale k vytváření grafiky na webu se běžně používají dva: vertex shadery a fragmentové shadery. Vertex shadery transformují pozice tvaru do 3D souřadnic, na základě kterých se vytváří kresba. Shadery fragmentů řídí vykreslování barev a dalších atributů objektu. GLSL není tak intuitivní jako JavaScript, zahrnuje spoustu matematických algoritmů k definování vektorů a matic. Kód se může stát velmi komplikovaným pro vytváření pokročilejší grafiky (*GLSL Shaders*, 2022). GLSL je součástí jiných grafických API (jako je WebGL), takže pro zvládnutí tohoto jazyka se jej kodér musí naučit společně s poměrně velkým grafickým API, což je často nutné pro nastavení kódu GLSL a požadovaných grafických dat pro grafický program.

Shadery jsou široce používány v designu videoher, post-processingu filmů, počítačových obrázků, hudebních videí k vytvoření řady efektů, jako jsou režimy osvětlení, změna odstínu, sytosti, jasů nebo kontrastu obrazu, efekty rozostření, objemové efekty, osvětlení, bokeh, posterizace, zkraslení obrazu, glitch efekt a mnoho dalších.

3.6 Python

Python je univerzální programovací jazyk vysoké úrovně. Python byl poprvé vydán pro použití na počátku 90. let, ale během posledních dvou desetiletí se stal oblíbeným nástrojem pro programátory, kreativní technology, umělce, designéry a výzkumníky pracující v tak rozmanitých oborech, jako je kreativní programování, vizualizace dat, počítačové vidění, vývoj webu, robotika, vývoj videoher, strojové učení atd. Python se stal uznávanou standardní platformou ve všech těchto a dalších oblastech. Tento jazyk byl také přijat jako vysoce efektivní platforma pro začátečníky, aby se naučili kódovat. Python je známý tím, že se

snadno učí a používá, je jednoduchý a výkonný – pár řádků kódu může být pevným základem pro program, je to bezplatný a otevřený zdroj, což z něj dělá jeden z nejpobulárnějších kódovacích jazyků na světě. Python také běží na jakémkoli počítači, jakémkoli operačním systému, včetně Windows, Mac a Linux. Online komunita a podpora Pythonu je zásadní a mezinárodní, jazyk lze v dnešní době efektivně využít téměř pro jakoukoli kreativní činnost.

Podle shrnutí Passany (2022), mezi výhody Pythonu patří:

- Snadno se učí, snadno se používá, snadno se přizpůsobí. Jazyk je vhodný pro začátečníky. Jedná se o programovací jazyk na vysoké úrovni a jeho syntax se blíží přirozenému jazyku (angličtině). Ve srovnání s jinými programovacími jazyky, jako je Java a C, v Pythonu lze stejný úkol provést s použitím menšího počtu řádků kódu.
- Flexibilita a svoboda. Tento jazyk je velmi flexibilní, umožňuje kodérům pracovat na široké škále projektů. Uživatelé mohou vyvíjet nové druhy aplikací pomocí programovacího jazyka Python, rozšiřovat existující knihovny vytvořením vlastních.
- Zvýšená produktivita. Jednoduchá povaha Pythonu a jeho vysoká produktivita pomáhá vývojářům soustředit se na efektivnější řešení problémů. Pochopení syntaxe a chování programovacího jazyka nezabere mnoho času, takže kodéři mohou trávit čas produktivněji.
- Rozsáhlá knihovna. Python poskytuje rozsáhlou sadu standardních knihoven se všemi možnými funkcemi a operacemi, které může kodér potřebovat, včetně automatizace, analýzy dat, grafických uživatelských rozhraní, mobilních aplikací, multimédií, webových rámců, vědeckých výpočtů, zpracování obrazu a mnoha dalších.
- Online komunita. V průběhu let Python vytvořil podpůrnou mezinárodní komunitu kodérů, vývojářů a umělců, kteří sdílejí své zkušenosti prostřednictvím online portálů, fór, průvodců a tutoriálů.

Nevýhody jazyka Python zahrnují takové nedostatky, jako je rychlost a spotřeba paměti. Ve srovnání s Javou nebo C je Python pomalejší. Python je interpretovaný jazyk, což znamená, že příkazy kódu jsou prováděny přímo, aniž by byly předtím kompilovány do instrukcí ve strojovém jazyce. Proto v případech, kdy je vyžadována rychlá akcelerace, se pro práci volí jiné kompilované jazyky.

Python má také velmi vysokou spotřebu paměti. Využívá velké množství prostoru počítače k ukládání různých typů dat, což může potenciálně vést k nedostatku paměti RAM (Paměť s náhodným přístupem). Vzhledem k tomu, že

není paměťově efektivní a má delší výkon pro zpracování, není Python vhodný ani pro mobilní aplikace.

K vyřešení problému s rychlostí a pamětí lze použít **Cython**. Cython je kompilovaný programovací jazyk, který se používá jako nadmnožina jazyka Python. Příkazy jsou psány v Pythonu s volitelnou další syntaxí a funkcemi inspirovanými jazykem C. Cython se používá k poskytování výkonu jazyka C, aby se urychlil téměř jakýkoli surový kód Pythonu.

3.7 Vizualní programovací jazyky

Většina běžně používaných a známých jazyků v technologii je textová, což znamená, že uživatel musí zapsat příkazy, funkce, operace, aby vytvořil program. Na druhém konci spektra je vizualní programování. Vizualní programování je způsob, jak vytvořit počítačový program manipulací s grafickými objekty namísto psaní řádků kódu. Vizualní programovací jazyk podporuje programování prováděné vizualními výrazy, prostorovým uspořádáním textu, objektů a symbolů, které se používají buď jako prvky syntaxe nebo sekundární notace. Vizualní programovací jazyky pracují s grafickými nebo ikonickými prvky, se kterými kodéři manipulují interaktivním způsobem podle příslušné sady pravidel pro konstrukci programu.

Vizualní programování je často spojeno s vytvořením grafického uživatelského rozhraní, ale taková prostředí se dnes úspěšně používají v průmyslu, vývoji aplikací webových prohlížečů, mobilních aplikacích, programování mikročipů a dokonce i vesmírných raketách.

Pro projekty kreativního kódování se používá několik vizualních jazyků. Max MSP je jedním z nich, jedná se o vyvinutý vizualní jazyk pro hudbu a multimédia. Pure Data je dalším příkladem, který se zabývá interaktivní počítačovou hudbou a multimediálními díly. vvvv je univerzální sada nástrojů, která využívá vizualní programovací rozhraní pro rychlé prototypování a vývoj programu. Specializuje se na syntézu zvuku a videa v reálném čase, pohyblivou grafiku v reálném čase a programování velkých mediálních prostředí s fyzickými rozhraními.

Normálně se vizualní programovací jazyky snadno učí a používají, uživatelská rozhraní lze navrhnout sestavením sady objektů na obrazovce pouze pomocí počítačové myši. Takové jazyky poskytují možnost připojit kód ke každé vizualní komponentě programu, který bude spuštěn, když uživatel interaguje s komponentou rozhraní. Mezi nevýhody patří nízká rychlost spuštění programů a potřeba velké úložné kapacity pevného disku a rychlejšího procesoru počítače.

3.8 Jak začít s kreativním kódováním?

Na rozdíl od funkčního a praktického zaměření většiny kódových projektů – jako je programování webových aplikací nebo aplikace pro chytré telefony – kreativní kódování používá programovací jazyky výhradně pro umělecké účely. 2D, 3D objekty, videa, hudba, interaktivní instalace a vizualizace jsou generovány aplikací algoritmů v programovacím prostředí. Historicky a v širším smyslu se tomu říká generativní umění. V dnešní době se škála různých stylů, výstupů, procesů a tím i terminologie liší. Je tradicí, nebo spíše stereotypem, považovat programování za příliš složité pro někoho bez technických znalostí, stejně jako inženýři mají stigma kolem potíží s kreativním vyjádřením. Tyto disciplíny však již nemusí být samostatnými obory, neboť s rozvojem moderních nástrojů a uměleckých stylů jsou propojeny těsněji, než lidé očekávají.

Pro někoho, kdo se chce pustit do kreativního kódování a už má s programováním zkušenosti, je rada jednoduchá: držte se programovacího jazyka, který znáte, protože téměř každý jazyk lze proměnit v nástroj pro tvorbu umění. Pro ty, kteří nemají žádné předchozí zkušenosti s programováním, ale chtějí vstoupit do oboru, obecná rada je následující:

- absolvovat kurz kreativního kódování, mnoho z nich je k dispozici online ve formě placených i bezplatných video lekcí;
- získat inspiraci od kreativních kodérů;
- prozkoumat četné online galerie a kódovací portály, zejména ty, kde jsou projekty uloženy v open-source;
- vytvořit si účet na jednom nebo několika z těchto portálů, prozkoumat jeho možnosti, zkopírovat projekty jiných kodérů a začít měnit jejich kód, číslo po čísle, řádek po řádku pro vzdělávací účely;
- vyzkoušet různé programovací jazyky, knihovny a frameworky dostupné pro kreativní umělecká díla;
- začít budovat první jednoduché programy, sdílet výsledky online, získat zpětnou vazbu od komunity a pokračovat v pokroku.

Velká rozmanitost možností může být pro začátečníka při výběru programovacího jazyka matoucí. Při výběru programovacího jazyka pro práci je třeba vzít v úvahu několik faktorů: dlouhodobé perspektivy jazyka, jak těžké je naučit se jazyk, jaké projekty na něm lze postavit, jaké technické požadavky klade. Většina programátorů by souhlasila s tím, že skriptovací jazyky na vysoké úrovni se lze relativně snadno naučit. Jednoduše řečeno, jazyk na vysoké úrovni

znamená, že k psaní příkazů používá přirozený jazyk. JavaScript spadá do této kategorie spolu s Pythonem a Processingem.

Existuje mnoho různých frameworků pro kreativní kódování, z nichž nejpopulárnější jsou založeny na JavaScriptu. Processing a p5.js jsou skvělé možnosti pro začátečníky i profesionály. S těmito nástroji je možné vytvořit jakékoli působivé interaktivní a audiovizuální dílo a také pevný základ pro vývoj softwaru. Processing a p5.js patří do stejné nadace Processing a mají fantastickou komunitu, spoustu zdrojů, průvodců, kurzů, knih a bezplatných výukových programů pod vedením profesora Daniela Shiffmana. Veškerý software je zdarma, takže jedinou investicí je čas. Po zvládnutí těchto jazyků může jedinec postupovat dále a zapojit se do jiných rámců. Mezi skvělé frameworky pro kreativní kodéry patří sketch.js, D3, three.js. Jsou také výkonné pro začátečníky a jsou užitečné pro vytváření všech druhů projektů, od 3D objektů a scén po jednoduché 2D kresby.

S rozvojem internetových technologií a globální komunikace je otázkou několika kliknutí zaregistrovat si online profil na programovacím portálu a začít experimentovat, mnoho nováčků nebo prosperujících umělců v dnešní době testuje své kreativní schopnosti pomocí generovaného umění a kreativního kódování tímto způsobem. Existuje řada webových portálů, kde si kdokoli může vytvořit online galerii, nahrát své hotové programy nebo skici. Ostatní uživatelé mohou programy sledovat, testovat, zanechávat komentáře či návrhy, a hlavně si mohou uložit kopii programu do vlastní galerie a kód měnit či vylepšovat, neboť veškerý kód je k vidění přímo na těchto stránkách. V ojedinělých případech licence zakazuje ostatním uživatelům používat tyto programy pro komerční nebo jiné nevhodné účely, ale jinak je kód dostupný komukoli.

Online portály jsou skvělou příležitostí ke spolupráci, učení se od kodérů z celého světa, kladení otázek a získávání cenných rad od kolegů. Další výhodou je, že kdokoli může začít kódovat online, v okně webového prohlížeče; není třeba stahovat žádný software, vše je již na těchto portálech dostupné. Vestavěné kompilátory spouštějí kód buď přímo během procesu zápisu, nebo po stisknutí tlačítka "start", v závislosti na zvoleném programovacím jazyce. Tyto online portály jsou rozděleny podle programovacích jazyků, se kterými pracují, dnes nejpopulárnější jsou:

- OpenProcessing – pro programovací jazyk a prostředí „Processing“;
- Studio Sketchpad – pro programovací jazyk a prostředí „Processing“;
- Processing Subreddit – pro programovací jazyk a prostředí „Processing“;

- CodePen – pro webové stránky vytvořené pomocí JavaScriptu a HTML / CSS;
- DevArt – vícejazyčná platforma, zahrnuje C, C ++, CSS, dart, glsl, go, HTML, Java, JavaScript, Objective-C, php, Processing, Python, Shell;
- ShaderToy – pro programování WebGL;
- GLSL Sandbox – pro programování WebGL;
- ShaderFrog – pro programování WebGL;
- GitHub – hosting pro open-source programovací projekty.

Prohlížení těchto online portálů je skvělý způsob, jak získat inspiraci, najít neobjevené talentované umělce a naučit se nové triky. Studium a testování dostupného kódu je nedílnou součástí jakéhokoli tvůrčího procesu, zapojení s moderními nástroji a komunitami bude pro umělce pouze přínosem bez ohledu na jejich vlastní zkušenosti.

Každý umělec má jedinečný proces generování svého uměleckého díla. Zavedení tohoto procesu je stejně důležité jako vytvoření uměleckého díla. Zde je zobecněný příklad procesu kreativního kódování od spisovatelky a technologa Jan Wu (*Getting Started With Creative Coding*, 2020):

- Nejprve přichází inspirace: slovo, obrázek, předmět, forma, zvuk atd.
- Vytvořte skicu, která bude představovat, jaké bude budoucí umělecké dílo.
- Vizualizujte interakce – s prostředím, s daty, s matematikou atd.
- Vytvořte algoritmus.
- Vytvořte své umělecké dílo s programovacím jazykem dle vašeho výběru.
- Projděte procesem úprav a výzkumu, abyste získali konečný produkt.

4 PROCES VIZUALIZACE DAT V KONTEXTU KREATIVNÍHO KÓDOVÁNÍ

Žijeme ve věku „velkých dat“, kde se každý den generuje bilion řádků informací. Vizualizace dat je trend, který pomáhá organizovat tyto toky informací a reprezentovat je pomocí vizuálních prvků, jako jsou tabulky, grafy, obrázky, animace, videa atd. Vizualizace dat pomáhá vidět propojení mezi různými částmi systému, pochopit jeho chování a analyzovat výsledek. Množství informací, které se denně objevují, se bude díky moderním síťovým technologiím i nadále zvyšovat. Studie ukazují, že lidský mozek obdrží každý den přibližně 32 GB informací prostřednictvím internetu, mobilních telefonů, e-mailu, televize, knih, rádia, sociálních sítí atd. (*Tech 21 century*, nd). Více než 90 % světových dat bylo vytvořeno v posledních několika letech. Každých 24 hodin se vytvoří 2,5 kvintilionu dat a jejich počet neustále roste (*Domo: Data never sleeps*, 2020). Vzhledem k obrovskému množství těchto čísel je nemožné, aby si jeden člověk všudypřítomné informace ani prohlédl, natož aby jim porozuměl. Vizualizace dat pomáhá společnosti vyrovnat se s tímto proudem informací a zpřístupnit je.

Vizualizace dat se stala novou uměleckou praxí, kde jsou informace transformovány do abstraktních estetických forem. Datoví umělci se mohou inspirovat každodenním životem, sociální interakcí, přírodní vědou, předpovědí počasí, literaturou a hudbou. Tito umělci berou dostupná data, často nevyčíslitelné zásoby informací, a přeměňují je v umělecká díla. Některá umělecká díla mohou zprostředkovat složitý svět, ve kterém žijeme, takový, jaký skutečně je, a pomoci ostatním porozumět jeho principům, jako jsou nádherné vizualizace větrných map – interaktivní živé webové stránky, které ukazují proudění větru na základě polohy uživatele. Jiná díla nemají funkční účel, umělci mohou data tlačit do abstraktních neskutečných forem, kde bude téměř nemožné rozlišit, odkud pochází inspirace.

Mnoho moderních populárních umělců se zaměřilo na kreativní kódování a vizualizaci dat. Například Aaron Koblin – tvůrce týmu Data Arts ve společnosti Google. Jedno z jeho nejznámějších děl se jmenuje „Flight patterns“ – představuje vizualizaci leteckého provozu v Severní Americe. 24 hodin záznamu dat a záběrů se přetavilo do série barevných obrázků a minutového videa, kde proudění dopravy představuje dynamická animace. Dalším příkladem je projekt „Flickr Flow“ vytvořený datovými umělci Fernandem Viegasem a Martinem Wattenbergem. Umělci pořídili všechny fotografie místa zvaného Boston Common, které byly během roku zveřejněny na Flickru. Speciální algoritmus analyzoval všechna fotografická data, vypočítal proporce a barevné variace měsíc po měsíci a uspořádal je do živého kola demonstrující změnu barev prostředí v průběhu roku. Mezi další populární jména patří Nathalie Miebach, která

transformuje data o počasí do složitých abstraktních soch a hudebních partitur. Jer Thorp – kanadský umělec, jeho díla se zaměřují na hranice kultury, dat a umění. Je také známý svým algoritmem, který umístil téměř 3000 jmen na památník 11. září na Manhattanu.

Kreativní kódování se stalo účinným způsobem vytváření, manipulace a interakce s datovými vizualizacemi. Četné programovací jazyky, jak je popsáno v předchozích kapitolách, pomocí specifických algoritmů pomáhají transformovat obrovské soubory informací, čísel, zvukových materiálů do vizuálních objektů a zpřístupňovat je prostřednictvím interaktivních webových stránek, grafů, živých a pohyblivých obrázků, videí, instalací atd. Je to další forma umění, která svou mírou autonomie proměňuje chladné logické struktury v hotová umělecká díla. Některé programy, jako je Gephi, RStudio, mohou transformovat velké množství dat do intuitivně srozumitelných grafických vizualizací, které ukazují propojení dat a vzájemnou závislost. Tyto programy se používají pro statistické výpočty, analýzu dat a vývoj statistického softwaru mezi datovými vědci, statistiky, bioinformatiky. Při práci s daty v rámci projektu kreativního kódování nejsou data vizualizována pro informační účely ve formě grafů a tabulek. Data se používají jako tvůrčí síla pro umělecké dílo, které ve své konečné podobě nemusí prozradit, odkud se vzala prvotní inspirace, neboli data.

Proces vizualizace dat v rámci kreativního kódování se u jednotlivých uměleckých děl liší. Nicméně v průběhu let praxe autorkou bylo stanoveno několik obecných kroků, které mohou platit pro většinu projektů, mezi něž patří:

- Výběr programovacího jazyka
- Shromažďování dat
- Ukládání, organizace a transformace dat
- Generativní algoritmy
- Import dat
- Přidání principu interaktivity
- Export vizualizací

4.1 Shromažďování dat

Jak již bylo zmíněno, data pro projekt kreativního kódování mohou pocházet z různých zdrojů. Co se zde rozumí pod pojmem data? **Data** lze definovat jako soubor jednotlivých faktů nebo statistik. Data mohou mít podobu textu, čísel, obrázků, symbolů, zvuků. Data mohou například zahrnovat ceny, vzdálenosti, adresy, věk, jména, teploty, roky atd. Data jsou hrubou formou znalostí a sama o

sobě nemají žádný význam a velmi často strukturu. Jinými slovy, data musí být interpretována a organizována, aby měla význam (Hill, 2015).

Jaká data se často používají ve vizualizacích? S jistou dávkou kreativity lze vizualizovat všechny typy dat: informace z oblastí vědy, jako jsou výsledky vědeckých experimentů, principy základní fyziky, matematické vzorce a algoritmy, astronomická data, data o počasí, zvukové a grafické materiály a mnoho dalších. Data mohou pocházet ze spolupráce s jinými umělci, vědci, organizacemi, institucemi atd. Tyto zdroje mohou poskytovat data organizovaným způsobem, například ve formě tabulky čísel v Excelu, která ukazuje denní změnu teploty vzduchu v určitém městě po dobu jednoho roku. Tento princip s úspěchem využívá Refik Anadol ve svých „datových sochách“ – velkoplošných dynamických vizualizacích meteorologických dat shromážděných vysokofrekvenčním radarem v Marmarském moři poskytovaných tureckou státní meteorologickou službou. Sběr dat o 30denní aktivitě mořské hladiny, který byl zaznamenáván radarem ve 30minutových intervalech, byl převeden do pohyblivé „živé“ vizualizace a vystaven na 12 x 3 metry dlouhé LED mediální stěně.

Data mohou pocházet z vědeckých experimentů a projektů, což je případ japonského vizuálního umělce a elektronického skladatele Ryoji Ikedy a jeho uměleckých děl „data-verse 1, 2, 3“. Práce představuje velké množství vědeckých digitálních dat promítaných na velkoplošnou obrazovku s vysokým rozlišením. Ikeda spojil data shromážděná z CERNu (Evropská organizace pro jaderný výzkum, kde byl rezidentem jako vítěz třetího ročníku Collide Award), NASA a The Human Genome Project se svými vlastními programovacími algoritmy a matematickými kompozicemi. Kombinací dat, matematiky a zvuků Ikeda vytvořil jednotnou vizualizaci. Konečný výsledek nutí diváky porovnat měřítko nejmenších známých částic s ohromností celého viditelného vesmíru. Ikeda také demonstruje data, která si naše mysl normálně vědomě nevšimá kvůli jejich neustálým změnám a rychlému průchodu, šumy a pixely jsou používány jako stavební bloky pro jeho instalace a mění všechna data normálně analyzovaná superpočítači ve vědeckých laboratořích na viditelné objekty. Elektronická hudba na pozadí má diváka dále ponořit do neznámé reality složitosti Vesmíru, který nás obklopuje (Battista, 2019).

Vědecká spolupráce je skvělým způsobem, jak zvýšit povědomí veřejnosti o určitém problému, studii nebo tématu, není to však jediný způsob. Data mohou pocházet od samotného umělce i z okolního světa. Tabulky s čísly a písmeny jsou nejběžnějším typem dat používaným v praxi kreativního kódování a v podstatě jde o formát, do kterého budou muset být převedeny všechny ostatní typy. Ale jako výchozí bod lze jako data pro vizualizace použít jiné materiály. V tomto procesu lze použít obrázky, fotogalerii, video, animaci, 3D objekt, záznam zvuku. Všechny tyto materiály nesou obrovské množství informací, neboli dat, které

budou později smíchány s kódovými algoritmy, aby se tento zásadní kus inspirace proměnil v originální umělecký projekt.

4.2 Ukládání, organizace a transformace dat

V programování čísla ovládají vše: objekty, tvary, postavy, pohyb, směry, rychlost, transformace, a dokonce i barvy. Jedním ze způsobů, jak přenést shromážděná data do programování, je převést je na čísla. Často získaná data jsou již uložena v číselné podobě. Jedním z nejběžnějších způsobů, jak tato data uložit a použít v programech, jsou tabulky v Excelu. Mnoho programovacích jazyků pracuje se souborovým formátem **Excel (XLXS, XLS)** a umí číst čísla přímo z tohoto souboru bez nutnosti kopírování a vkládání mnoha řádků informací do samotného programového kódu, což je velmi výhodné pro rozsáhlé databáze, které ukládají stovky, tisíce nebo dokonce miliony čísel (viz obrázek 22.).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	5	0	101,528	212,2654	2,430725	0,341335	0	0,341335	0,459309	2,090707		39	21,36402	1,971416	1,174	32,218	170,0911	0
21	12	1,343703	105,5185	215,9045	2,529064	0,355996	0	0,355996	-0,02001	2,046129		48	18,30526	2,089769	1,104	32,516	174,7279	0
22	18	2,781153	108,8542	230,1767	2,569649	0,384655	0	0,384655	-0,03361	2,114542		63	-6,30781	2,143949	1,236	32,76133	181,9763	0
23	25	4,046731	113,1901	244,0046	2,587327	0,399348	0	0,399348	0,006574	2,155706		88	16,60532	2,121445	1,181	32,917	192,4994	0
24	32	5,421683	113,8115	246,4245	2,588441	0,415444	0	0,415444	-0,07153	2,165198		107	18,40262	2,20066	1,229	33,11	192,6559	0
25	38	6,70289	116,5661	248,6586	2,57137	0,414848	0	0,414848	-0,08745	2,133198		125	16,8443	2,199508	1,271	33,46	195,6315	0
26	45	8,030968	117,7887	248,7047	2,569677	0,397601	0	0,397601	-0,10862	2,111447		139	16,88322	2,218992	1,382	33,472	193,0417	0
27	51	9,437165	118,6331	249,0389	2,568627	0,419349	0	0,419349	-0,18129	2,099237		161	12,0159	2,290613	1,41	33,474	193,7165	0
28	59	10,85899	121,4897	255,7517	2,586372	0,426535	0	0,426535	-0,16493	2,105131		180	10,86523	2,291995	1,458	33,4148	195,7837	0
29	64	12,15582	119,1271	256,3165	2,811172	0,425839	0	0,425839	-0,0587	2,151623		213	14,0706	2,410568	1,493	33,1944	193,2634	0
30	68	13,4214	124,4667	274,7848	2,910001	0,462112	0	0,462112	-0,01987	2,207697		226	17,41241	2,470563	1,375	32,9728	199,9224	0
31	77	14,79635	125,1254	273,0218	2,948995	0,452314	0	0,452314	0,022242	2,181986		237	17,18804	2,467444	1,444	32,92	201,7437	0
32	81	16,01506	125,4604	299,349	2,990308	0,476087	0	0,476087	0,005827	2,386005		256	9,616808	2,525172	1,563	32,6828	213,699	0
33	87	17,21814	126,0825	303,1574	2,988982	0,472099	0	0,472099	-0,04863	2,404436		275	16,88345	2,576503	1,576	32,5024	215,5716	0
34	97	18,42122	126,807	308,6297	3,0122	0,486388	0	0,486388	-0,04339	2,433853		287	17,20829	2,596276	1,569	32,3484	216,7492	0
35	102	19,73368	126,8649	291,3263	3,014605	0,488373	0	0,488373	-0,00661	2,29635		319	9,368446	2,561906	1,563	32,1452	210,2121	0
36	110	21,10863	129,3196	279,7245	2,80711	0,460097	0	0,460097	-0,12909	2,163048		342	4,625163	2,476888	1,583	32,01	206,6246	0
37	116	22,4367	125,8866	260,5444	2,768476	0,470068	0	0,470068	-0,14674	2,096976		392	18,11292	2,455905	1,583	32,0372	200,367	0
38	120	23,67104	128,874	291,6877	2,7788	0,471693	0	0,471693	-0,16221	2,263355		410	11,13612	2,481703	1,583	32,0592	207,2419	0
39	125	24,85849	130,2541	290,6542	2,810701	0,478911	0	0,478911	-0,12371	2,231439		442	17,48776	2,475102	1,59	32,09	207,7241	0
40	134	26,04595	130,8593	294,455	2,828024	0,452847	0	0,452847	-0,12545	2,250165		460	17,26725	2,494169	1,479	32,1276	209,2237	0
41	140	27,34278	128,6325	281,7549	2,855726	0,449564	0	0,449564	-0,09691	2,190386		486	15,60416	2,493329	1,576	31,8772	199,3107	0
42	147	28,71773	127,8572	285,4033	2,894084	0,4316	0	0,4316	-0,08583	2,232203		499	12,7054	2,520604	1,493	31,608	201,1263	0
43	152	29,98331	124,5506	283,3986	3,073815	0,4769	0	0,4769	-0,02741	2,275368		525	15,63119	2,641918	1,563	31,3448	205,6666	0
44	160	31,24889	125,2119	283,0285	3,067942	0,45692	0	0,45692	-0,01827	2,260397		537	15,91007	2,626902	1,549	31,0648	205,4266	0
45	165	32,48322	125,8376	273,4715	3,051186	0,469264	0	0,469264	-0,0455	2,17321		549	4,097919	2,637381	1,528	30,8384	200,9451	0
46	171	33,73318	127,5566	283,9057	2,808489	0,459962	0	0,459962	-0,14182	2,225723		600	15,08146	2,490999	1,583	30,9664	203,949	0
47	178	35,10813	127,3235	288,5443	2,850619	0,454531	0	0,454531	-0,15808	2,26623		619	16,05925	2,549395	1,458	31,0544	205,5923	0
48	184	36,34246	123,5004	281,6039	3,110782	0,460068	0	0,460068	-0,00808	2,280187		632	12,70155	2,659557	1,59	31,0476	202,488	0
49	191	37,62366	123,9025	286,7261	3,109819	0,464741	0	0,464741	0,037438	2,314126		651	15,3931	2,613073	1,576	31,0824	208,0042	0

Obrázek 22: Excel datový soubor z Centra polymerních systémů UTB

Dalšími oblíbenými formáty souborů pro ukládání dat jsou **CSV** (Comma-separated values, hodnoty oddělené čárkami) - formát prostého textu s řadou hodnot, které jsou odděleny čárkami. Na rozdíl od Excel CSV soubor nemůže provádět operace s daty, slouží pouze pro ukládání. Soubor CSV je běžný formát exportu (viz obrázek 23), protože je tak dobře známý a široce používán na všech typech počítačů. Jeho jednoduchost vedla k tomu, že jej přijalo mnoho programů, pokud jde o export a čtení tabulkových dat.

```

Id,Label,timeset,modularity_class
11,Valjean,,1
48,Gavroche,,8
55,Marius,,6
27,Javert,,7
25,Thenardier,,7
23,Fantine,,2
58,Enjolras,,8
62,Courfeyrac,,8
64,Bossuet,,8
63,Bahorel,,8
65,Joly,,8
24,MmeThenardier,,7
26,Cosette,,6
41,Eponine,,7
57,Mabeuf,,8
59,Combeferre,,8
61,Feuilly,,8
0,Myriel,,0
66,Grantaire,,8

```

Obrázek 23: CSV databáze "Les Misérables" používaná aplikací Gephi jako základní příklad

Další populární formát – **TXT**. Soubory TXT obsahují prostý text, který se ukládá ve formě řádků. Jakýkoli textový editor a nejpopulárnější webové prohlížeče mohou otevřít a pracovat s tímto typem souboru. V systému Windows lze soubor TXT otevřít pomocí programu Microsoft Poznámkový blok nebo Microsoft WordPad, přičemž oba jsou běžnou sadou programů předinstalovaných na počítačích se systémem Windows.

Datový formát **JSON** (JavaScript Object Notation) – je standardní formát pro ukládání textových dat založený na indexu objektů JavaScript. Běžně se používá pro přenos dat ve webových aplikacích. Na rozdíl od CSV JSON umožňuje vytvářet hierarchickou strukturu dat. JSON umožňuje ukládat data v mnoha typech dat včetně řetězců, polí, booleanů, celých čísel atd. (Gupta, 2021).

XML (Extensible Markup Language) je dalším formátem, který se používá k ukládání a popisu dat ve formátu prostého textu. Je to značkový jazyk podobný HTML, který byl navržen pro snadné ukládání, přenos a rekonstrukci libovolných dat.

Obecně platí, že čím jednodušší a populárnější je formát souboru – tím větší je pravděpodobnost, že bude přijat vybraným programovacím jazykem. Jak ukládat data by se mělo rozhodnout podle programovacího prostředí, které bude pro vizualizace použito. Například výběr formátu souboru **CSV** je nejjednodušší způsob, jak importovat data do programu napsaného pomocí Processing – jednoho z nejpopulárnějších programovacích prostředí pro kreativní kódování.

Jiné datové formáty lze obvykle uložit jako soubor .csv a později importovat pomocí jednoduchého příkazu *loadTable("data.csv")*.

Možnost, která se nezabývá žádnými dalšími formáty souborů – **kopírování všech dat** přímo do kódu. Tato metoda má své výhody i nevýhody. Mezi výhody patří: není potřeba řešit další formáty souborů a jejich import. Pokud program běží online na programovacím portálu, bude třeba datový soubor nahrát také online, což tyto online programovací portály někdy podporují, ale prostor, který může jeden uživatel zabírat na serveru, může být omezený. Program nebude vyžadovat žádný další čas k nahrávání a čtení souboru, takže program poběží rychleji. Mezi nevýhody patří zvyšující se počet řádků kódu. Zkopírováním stovek řádků dat do programu je snadné se ztratit mezi veškerým dalším textem, takže při práci s rozsáhlými daty může být rozumnější uložit je do dalšího souboru. Při práci s ne tak velkými soubory dat může být nejlepším rozhodnutím jejich jednoduché kopírování do kódu.

Samozřejmostí je vždy možnost získat data v reálném čase, bez jejich ukládání nebo kopírování, při provozu programu například z online zdroje – webové stránky meteorologické stanice, která umožňuje uživatelům získávat data o počasí ve vybrané oblasti.

Jak již bylo zmíněno, data jsou obvykle transformována do čísel, aby byla součástí vizualizace a ovlivnila vizuální výsledek. Když počáteční informace nepřichází v číselné podobě, existují určité způsoby, jak je přeložit, v závislosti na původním zdroji. Při práci s abecedními a v některých případech speciálními doplňkovými znaky a řídicími kódy existuje na výběr mnoho systémů kódování znaků, které byly vyvíjeny v průběhu let. Tento proces je známý jako **kódování znaků** – praxe přiřazování čísel ke znakům, což umožňuje jejich ukládání, přenos a transformaci pomocí počítačů (*Character Encoding*, 2010). Mezi běžné příklady systémů kódování znaků patří Morseova abeceda, Baudotův kód, Americký standardní kód pro výměnu informací (ASCII) a Unicode.

Číslování písmen tak, že A=1, B=2 atd. je jedním z nejjednodušších způsobů, jak je převést na čísla. Říká se tomu šifra **A1Z26**. Každé písmeno v abecedě je nahrazeno jeho pořadovým číslem v abecedě.

ASCII byla první znaková sada vyvinutá v 60. letech a používaná k odesílání informací mezi počítači na internetu. ASCII přiřazuje standardní číselné hodnoty velkým a malým anglickým písmenům od A do Z, číslicím, interpunkčním znaménkům a dalším znakům používaným v počítačích. Původní sada ASCII obsahuje 128 znaků (*HTML ASCII Reference*, nd).

Umělec/programátor může zakódovat zprávu/data ručně pomocí ASCII tabulky zobrazující číselný ekvivalent každého symbolu, nebo použít jeden z

mnoha online převodníků z anglických písmen do ASCII. Například ASCII verze fráze *Hello, World!* bude vypadat následovně:

104 101 108 108 111 44 32 119 111 114 108 100 33

Další populární možnost – **binární kód**, kterému počítače rozumí a se kterým pracují (také známý jako strojový jazyk), který obsahuje pouze dva znaky: 0 a 1. Každý binární řetězec má osm binárních bitů nul a jedniček v určitém vzoru jedinečném pro každé písmeno slova. Binární verze *Hello, World!*:

01101000 01100101 01101100 01101100 01101111 00101100 00100000
01110111 01101111 01110010 01101100 01100100 00100001

Unicode je standard informační technologie pro kódování textových informací ve většině světových psacích systémů. Standard Unicode poskytuje jedinečné číslo pro každý znak bez ohledu na platformu, zařízení, aplikaci nebo jazyk. Nejnovější verze Unicode dokáže přeložit 144 697 znaků, což zahrnuje písmena z většiny lidských jazyků, historické skripty, speciální symboly, znaky, emotikony atd. Unicode lze uložit pomocí několika různých kódování, která převádějí kódy znaků do sekvencí bajtů (*Unicode® 14.0.0*, 2021).

V různých zemích se používá mnoho metod a kódovacích systémů navržených pro různé jazyky a různá programovací prostředí, díky čemuž je převod textu na sadu číselných dat otázkou několika minut.

Pro práci s **vizualizací 3D modelu**, nikoli ve své podstatě tak, jak byl model vytvořen tak, aby vypadal, ale s daty z něj získanými, je nutné exportovat informace o geometrii 3D objektů. Exportované soubory se často používají pro výměnu informací, CAD a 3D tisk. Existují různé formáty souborů pro export, jako je OBJ, STL, COLLOADA, FBX, X3D, VRML, AMF, 3MF, IGS a další, v závislosti na formátech, které program použitý pro modelování podporuje.

Objekty v souboru **OBJ** představují jednoduchý datový formát, který obsahuje 3D geometrii, polohu každého vrcholu, vztah souřadnic textury k vrcholu, normálu pro každý vrchol a také parametry, které vytvářejí polygony (viz obrázek 24); formát byl vznikl v 80. letech (Schechter, 2021).

v	-5.000000	5.000000	0.000000
v	-5.000000	-5.000000	0.000000
v	5.000000	-5.000000	0.000000
v	5.000000	5.000000	0.000000
vt	-5.000000	5.000000	0.000000
vt	-5.000000	-5.000000	0.000000
vt	5.000000	-5.000000	0.000000
vt	5.000000	5.000000	0.000000
vn	0.000000	0.000000	1.000000
vn	0.000000	0.000000	1.000000
vn	0.000000	0.000000	1.000000
vn	0.000000	0.000000	1.000000
vp	0.210000	3.590000	
vp	0.000000	0.000000	
vp	1.000000	0.000000	
vp	0.500000	0.500000	

Obrázek 24: Část souboru .obj z University of Utah School of Computing. Zdroj: <https://www.marxentlabs.com/obj-files/>

Získávání číselných údajů z jiných zdrojů může být složitější. K exportu číselných dat ze **zvukového souboru** bude zapotřebí audio software, jako je program Audacity. "Sample Data Export" je funkce, která umožňuje exportovat hodnoty amplitudy pro každý vzorek, číst hodnoty po sobě jdoucích vzorků z vybraného zvuku a vkládat tato data do prostého textu, CSV nebo HTML souboru. Při exportu dat lze vybrat jedno z měřítek:

- **dB:** Hodnoty vzorků jsou zobrazeny na stupnici dB vzhledem k plné stupnici. Příklad fragmentu exportovaných dat ze zvukového souboru „Saturn radio waves“ od NASA:

-24.80063, -40.91386, -25.89454, -32.02723, -27.92731, -26.29393, -37.96935, -28.42899, -4.29997, -25.81766, -42.78955, -22.81966, -22.60780, -23.37209, -36.42369, -33.48052...

- **Lineární:** Hodnoty jsou zobrazeny na lineárním měřítku +/- 1. Stejný fragment exportovaný v lineárním režimu:

-0.05423, -0.08103, 0.00651, 0.07717, 0.03275, -0.05028, -0.05859, 0.01191, 0.06962, 0.05950, 0.00213, -0.03906, -0.02410, 0.03162, 0.06136, 0.02548, -0.03313, -0.03738, 0.01605...

Číselné informace lze získat z **obrázků, fotografií a videí**. Jedním ze způsobů, jak toho dosáhnout, je získat barvu každého pixelu tohoto obrázku a převést jej do jeho číselného formátu. Každý pixel na obrazovce je trojicí zdroje červeného, zeleného a modrého světla (model RGB). Barva každého pixelu může být reprezentována jako 6místné hexadecimální číslo nebo trojice desetinných čísel v

rozsahu od 0 do 255. Pro pohodlí a čitelnost kódu jsou barvy reprezentovány jako trojice desetinných míst, jako (červená) (zelená) (modrá). Například (0, 0, 0) je zobrazení bílé barvy nebo (255, 0, 0) červené atd. Dalším speciálním případem je obrázek ve stupních šedi, kde všechny tři barevné kanály pro každý pixel mají stejnou hodnotu jako (56, 56, 56) je světle šedá.

Převod obrázku na čísla a zobrazení jeho „surových“ dat není snadný proces. Ruční výběr každé barvy z každého pixelu bude trvat nekonečně dlouho, k automatizaci tohoto procesu je zapotřebí speciálně napsaný software. Například PIL (Python Imaging Library) dokáže převést obrázek JPG na trojrozměrné pole čísel mezi 0 a 255, které představuje úroveň barev. Výsledek také vyžaduje hodně místa na disku. Několika megapixelový barevný snímek zabere v nekomprimovaném stavu desítky megabajtů (Rohreh, 2019).

K převedení počátečních dat do číselné podoby lze použít i jiné způsoby, nebo lze k vytvoření vizualizací použít i jiná než číselná data. V těchto případech jsou metody využití dat v programech jedinečné pro každý projekt a nelze je doporučit jako obecný přístup k vizualizaci dat. Výše uvedené metody použila autorka k vytvoření vizualizací dat v průběhu let práce a studia. Po získání potřebných dat a jejich uložení ve správném formátu následuje nejsložitější krok – vytvoření programu, do kterého lze tato data vložit.

4.3 Generativní algoritmy

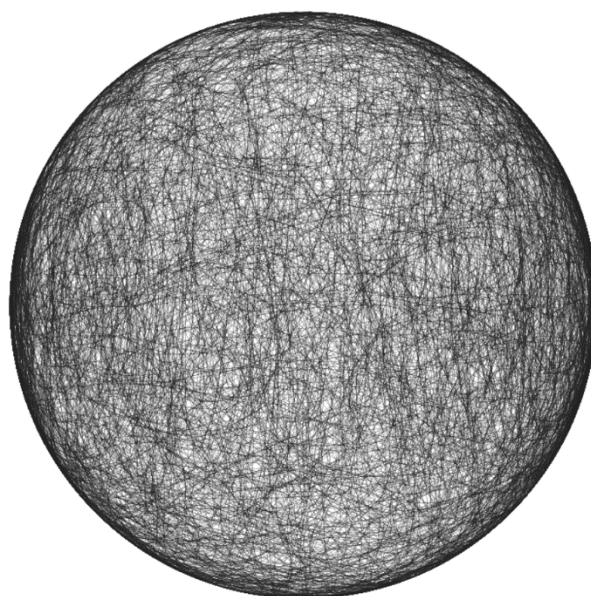
Kreativní kódování neboli generativní umění má hlavní jádro – algoritmus, nebo soubor pravidel, systém. Tato pravidla vytvářejí vše, od estetických kompozic po textury a zvuky. Pojem generativní algoritmus se nevztahuje pouze na programování, zjednodušeně řečeno je to specifický soubor pravidel, systém, podle kterého se objekt chová a vyvíjí.

Generativní algoritmy mohou zahrnovat matematické, biologické, fyzikální přístupy. Například biologie nám poskytuje velké množství různých vzorů odvozených z přírody, jako jsou listy, větve, krystaly a fraktály. Matematika je zodpovědná za geometrické tvary, plochy, křivky a grafy. Fyzika popisuje různé druhy hmoty, její pohyb a chování, prostor a čas, energii a sílu. Každý z těchto přístupů může být použit jako základ uměleckého projektu, jeho inspirace a bude určovat vývoj a vizuální výsledek.

Myšlenkou použití generativních algoritmů v kreativním kódování je prozkoumat design a algoritmická řešení prostřednictvím experimentování. Vzory a sítě v okolním světě, v přírodě, vesmíru, v digitálních databázích mohou být přeměněny na vodítka pro vytváření kódovaného uměleckého díla; složité a komplikované struktury mohou být výsledkem souboru jednoduchých pravidel.

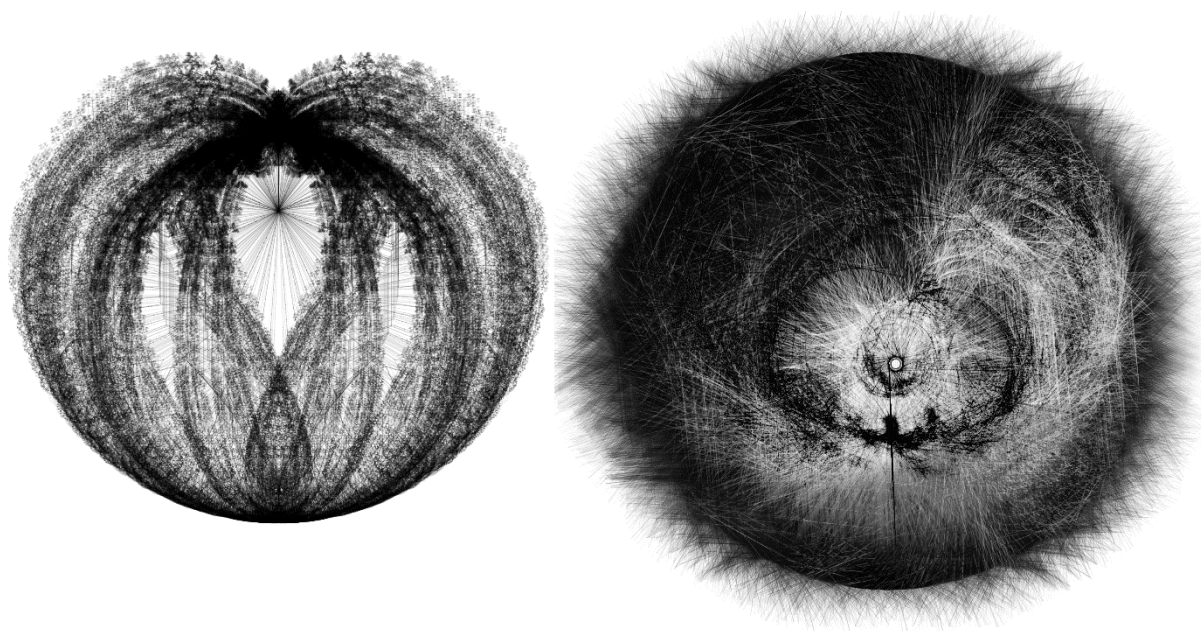
Součástí výzvy je pokusit se duplikovat nějaký vzor nebo pohyb s co nejmenším počtem řádků kódu a co nejjednoduššími pravidly.

Generativní umění obvykle předpokládá, že do procesu tvorby je zapojena určitá úroveň náhodnosti, i když to není podmínkou. Nezáleží na tom, jaký je konečný výsledek, zda jde o hudební kompozici, scénu videohry, animaci, architektonickou strukturu, v každém případě existuje systém, který pomáhá krok za krokem budovat výsledek. Prvním krokem je obvykle aplikace relativně jednoduchého pravidla/algoritmu. Například při práci s tvary mohou být jednoduché geometrické obrazce základem toho, co umělec zamýšlí vytvořit. Kruh může být výchozím bodem pro vizualizaci biologického vzorce. Matematický vzorec pro kruh je jednoduchý, detaily se liší v závislosti na zvoleném programovacím jazyce. Dalším krokem bude zavedení variací, přeměna kruhu na něco jiného, deformace, začlenění pohybů. Celý proces může být vysoce experimentální.



Obrázek 25: Naprogramovaný kruh. Zdroj: vlastní

Umělec může mít prvotní cíl nebo představu, jak by měl výsledek vypadat, ale jako každá jiná umělecká praxe může být kreativní kódování vysoce nepředvídatelné. Nepatrná změna vzorce, experimentální přístup, dokonce i chyba v kódu, záměrná nebo neúmyslná, může vést k zajímavější vizualizaci, než bylo původně zamýšleno. Celý proces vytváření algoritmů pro program je průzkumný, neustálý a výzkumný. Algoritmy samy o sobě neimprovizují, dělají to, k čemu jsou navrženy, ale různé kombinace pravidel, vzorců a chyb vytvářejí jedinečný a často nepředvídatelný zážitek.



Obrázek 26: Zkreslený kruh. Zdroj: vlastní

Programování se zabývá čísly v téměř každém řádku kódu. Další možná experimentování spočívají v povaze těchto čísel – mohou být ovlivněny matematickými operacemi. Co když poloměr kruhu není pevné číslo, ale kosinus pixelu pozice myši na obrazovce? Co když barva tohoto kruhu není červená RGB (255, 0, 0), ale 255 plus počet snímkových frekvencí, které prošly od začátku operace programu, děleno číslem PI? Některé programovací jazyky umožňují mít při práci s čísly větší volnost, některé méně, ale vždy je prostor pro experimentování.

Mezi dobře známými programovacími jazyky, které jsou široce používány, je univerzálně přítomná funkce *random()*, která generuje náhodná čísla. Přesněji řečeno, pseudonáhodná čísla; program stále používá specifický algoritmus k výběru čísla ze sady souborů a taková "náhodná" čísla mohou být předvídatelná v rámci každého programovacího jazyka. Tato funkce může být součástí základního jazyka nebo může být dostupná v rámci dalších knihoven. Funkce *random* nebo její variace umožňují zvolit náhodné číslo v rámci konkrétní množiny, řekněme od 0 do 1000, nebo si vybrat z několika konkrétních hodnot předem nastavených kóděm. Funkce nicméně nechá program „rozhodnout se“, co vloží do kódu, a učiní výsledek experimentálnější a autorem nepředvídatelným.

Jednoduše řečeno, třetím krokem vytvoření vizualizace dat (nebo prvním skutečným krokem, předchozí dva lze považovat za předpřípravu) je nastavení počáteční sady pravidel/algoritmů/vzorců ve formě programovacích příkazů, které se často inspiřují biologickým, fyzikálním nebo matematickým vzorem – pohyb objektu, geometrie tvarů, růst a větvení stromu, mapa mořských proudů,

vzory souhvězdí, chování světelných paprsků, složitost fraktálu atd. Další experimentování s algoritmy odsouvá vizualizaci od původní sady pravidel a přináší do kódu stále více originality.

4.4 Import dat

Po vytvoření první sady pravidel a algoritmů je dalším krokem zavedení dat do programu. Při práci se sadou čísel jsou pole (v angličtině: array) nejběžnějším způsobem, jak je vložit do kódu. V informatice je pole datovou strukturou sestávající ze souboru prvků, hodnot nebo proměnných. Jednoduše řečeno, je to seznam dat. Je možné mít pole jakéhokoli typu dat. Ať už se jedná o přímé vložení čísel z jakékoli databáze, nebo odkaz na online databáze nebo soubor CSV s daty. Každý kus dat v poli je identifikován indexovým číslem představujícím jeho pozici v seznamu. Každé pole má proměnnou délku – počet prvků čísel použitých v seznamu, jejichž počítání při práci s velkými sadami dat může být obtížné.

Uvedme příklad takového pole v programovacím jazyce Processing. Jako data bude použita ASCII reprezentace fráze *Hello, World!*:

```
int[] HelloWorld = { 104, 101, 108, 108, 111, 44, 32, 119, 111, 114, 108, 100, 33};
```

kde *int* znamená celočíselný formát dat, *HelloWorld* je název pole, které je mu předáno a které lze později použít v kódu, a čísla v závorkách jsou naše data.

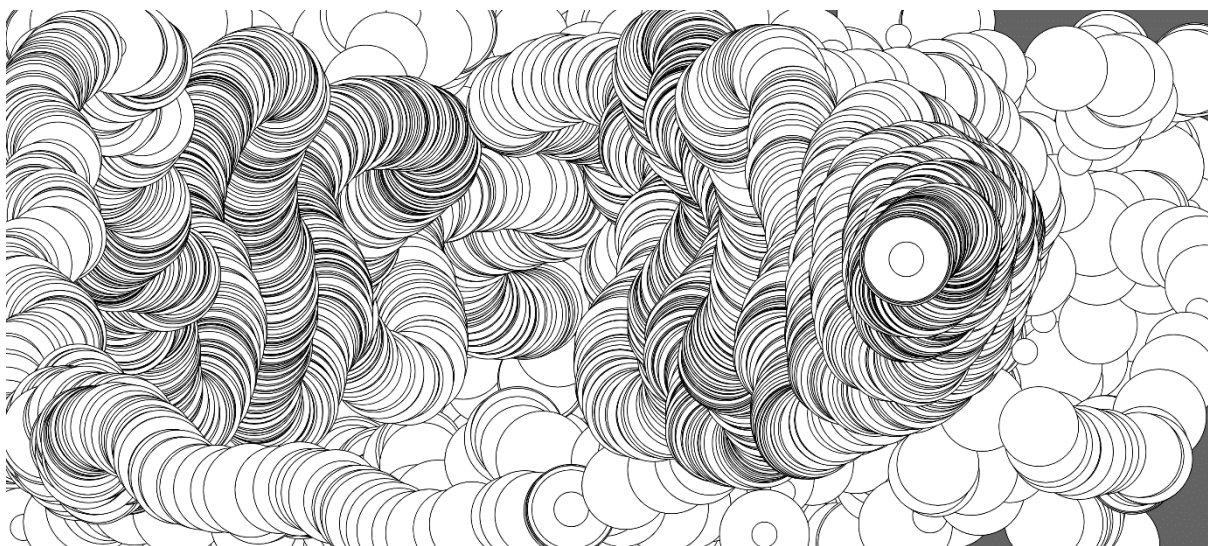
Jakmile je pole zavedeno, lze jej použít kdekoli dále v kódu. Chcete-li například nakreslit (naprogramovat) kruh na obrazovce, budete potřebovat tři parametry – umístění pixelu na azimutu *x* na obrazovce, umístění pixelu na azimutu *y* na obrazovce a průměr kruhu:

```
circle (150, 150, 300);
```

Čísla, souřadnice a průměr kruhu lze libovolně nahradit jakoukoli jinou číselnou informací nebo matematickou operací. Nastavenou pozici na obrazovce můžeme nahradit souřadnicemi kurzoru a místo plného čísla pro průměr kruhu vezmeme náhodné číslo z našeho pole:

```
circle (mouseX, mouseY, random(HelloWorld));
```

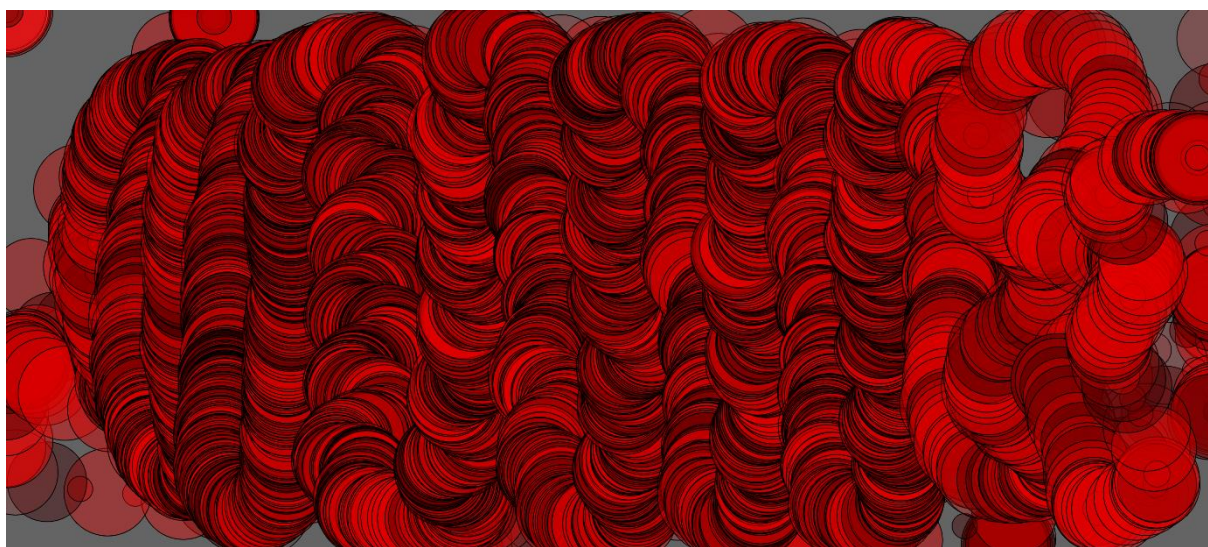
Program v tomto případě nakreslí nový kruh 30krát za sekundu (záleží na nastavení programu pro počet snímků za sekundu) a pokaždé zvolí jiné souřadnice, když pohneme myší, a také nový náhodný průměr z pole:



Obrázek 27: Záplava kruhů. Zdroj: vlastní

Totéž lze provést s barvami, jelikož jsou také reprezentovány čísla. Místo kreslení bílých kruhů nebo jakékoli plné barvy z modelu (červená, zelená, modrá) řekneme, že červená bude reprezentována náhodným číslem z pole vynásobeným dvěma, zelená i modrá budou nula – což nám dá variace červené palety, a poslední vlastnost nastaví neprůhlednost této barvy, také v rozsahu od 0 do 255, ale v našem případě – další náhodné číslo z pole:

```
fill( random(HelloWorld)*2,0,0, random(HelloWorld));
```



Obrázek 28: Záplava červených kruhů. Zdroj: vlastní

Toto je příliš zjednodušený příklad polí číselných dat používaných v kódu programu. Data dokážou mnohem více, lze je použít ve složitějších rovnicích a ovlivnit celkový vzhled vizuálního výsledku, příklady čeho jsou uvedeny v dalších kapitolách.

4.5 Přidání principu interaktivity

Často, ale ne vždy, jsou aplikace vytvořené technikami kreativního kódování založeny na principu interaktivity; program přijímá vstupní data a na základě těchto informací provádí některé úpravy vizuálního výsledku. Detekce pohyblivých objektů se tak stává klíčovou technologií pro takový druh projektů.

Detekce pohybu lze definovat jako proces zachycení změny polohy objektu ve vztahu k jeho okolí a samotnému zařízení. Detektor pohybu je speciální zařízení, které zachycuje pohyb fyzického objektu. Některá zařízení detekují přítomnost osoby v určitém prostoru, jiná zachycují zvuky nebo jakékoli rušení v povrchové rovnováze, jako jsou seismometry.

Systém detekce pohybu (nebo detekce aktivity) je obvykle založen na specifickém monitorovacím algoritmu/funkci vestavěné uvnitř vybraného softwarového programu, který zachycuje fyzický pohyb z kamery a přenáší jej do sady hodnot. Normálně je pohyb zachycen sám; pokročilejší technologie však mohou rozpoznat přesný typ pohybu, jako konkrétní gesto.

Použití systémů detekce pohybu je velmi rozmanité. Je to populární nástroj pro všechny umělecké výkony; mnoho aplikací používá automatické sledování pohybujícího se objektu, aby bylo dosaženo zábavnějších výsledků. Vizuální inteligentní systémy vyžadují přesnější přístup pro výpočet změny polohy objektu. Proces detekce pohybu začíná záznamem video obrazu pomocí kamery/senzoru, snímek po snímku, tyto snímky jsou vzájemně porovnávány a jsou definovány rozlišovací body. Když jsou tyto body programem identifikovány, lze je zpracovat pomocí speciálních technik napsaných v programu nebo filtrů, některé pracují s rozlišením obrazu, jiné s barevnou hloubkou a vyhlazováním, jiné s odstraněním pozadí. (Rathod, Patel, s.71-74, 2015). Systémy detekce pohybu pracují s podporou různých technologických zařízení:

- RGB kamery;
- infračervené senzory;
- laserové senzory;
- akustické senzory;
- mikrofony;
- magnetické senzory;
- rádiové a mikrovlňné senzory.

V kreativním kódování se takové hardwarové komponenty pro vkládání dat z prostředí používají k interakci s divákem a ke změně vizuálu programu. Kamery a senzory mohou zachytit pohyb nebo zvuky nebo jen přítomnost diváka a na základě těchto informací změnit geometrii objektu nebo jeho pohyb, barvy. Není mnoho omezení, kam lze tato dodatečná data umístit, vše závisí na zvolené programovací platformě a knihovnách, které jsou pro daný jazyk k dispozici.

Používají se nižší (jednodušší) formy interakce bez dalšího hardwaru, ale v rámci standardní či běžnější výbavy počítače, notebooku, tabletu či mobilního telefonu – myš, klávesnice, touchpad, dotyková obrazovka. Jednoduchý algoritmus umožní komukoli ovlivňovat obraz na obrazovce dotykem nebo stisknutím určitých kláves na klávesnici, všechny programovací jazyky používané pro kreativní kódování takové interakce podporují, některé již mají vestavěné knihovny, jiné vyžadují dodatečné instalace. Použití počítačové myši nebo kurzoru na obrazovce je nejběžnějším způsobem vložení interakce do programu, je definováno numerickými daty dvou proměnných: pozice x a y na obrazovce, a jak již bylo zmíněno, numerická data mají velký vliv na jakékoli kódovací projekty.

4.6 Export vizualizací

Po dokončení hlavních fází vizualizace dat, jako je získávání a organizace dat, vytvoření počáteční sady pravidel a generativních algoritmů, vložení dat do programovacích příkazů a přidání interaktivity, posledním krokem bude export vizuálního výsledku. Ve svém jádru jsou projekty kreativního kódování podobné – jde o sadu programovacích příkazů. Ale způsob, jakým jsou tyto projekty prezentovány, se může lišit.

V historii generativního umění existovalo mnoho metod používaných k prezentaci díla veřejnosti. První dostupné počítače neměly obrazovky, umělci museli k zobrazení obrázků používat děrné pásy nebo mechanické plotry. V dnešní době je škála dostupných metod široká, nejběžnější a okamžitý způsob, jak vidět výsledek, je na obrazovce počítače. Grafika, animace, 3D objekty atd. mohou být uloženy v počítači, zaznamenány jako video nebo exportovány jako EXE program. Existuje mnoho možností pro online ukládání programů a vizuálů – od mnoha programovacích portálů s otevřeným zdrojovým kódem, kde si každý návštěvník může prohlížet, kopírovat a upravovat příkazy, až po originální webové stránky autora a sociální média. Umělecké projekty lze prodávat jako NTF. Grafiku bylo možné tisknout na jakýkoli možný materiál, na který nám moderní technologie umožňují tisknout, 3D objekty lze tisknout na 3D tiskárnách. Při prezentaci děl na výstavách mohou autoři využít velkoplošné projekce, plátna, monitory, LED stěny apod.

5 PRAKTICKÁ REALIZACE

Disertační práce má několik praktických výstupů:

- 1 umělecká díla vytvořená technikami kreativního kódování, která byla prezentována na různých výstavách, sympoziích (mezinárodních i tuzemských), v galeriích;
- 2 závěrečná výstava s vybranými uměleckými díly vytvořenými během let studia;
- 3 online portál s popisem všech uměleckých děl, jejich metod a zdrojového kódu, který si ostatní mohou prohlédnout a prostudovat.

5.1 Realizované projekty kreativního kódování

Za léta studia vzniklo velké množství projektů. Některé byly představeny veřejnosti na výstavách a sympoziích, jiné spatřily světlo výhradně online. Všechny projekty byly vytvořeny pomocí technik kreativního kódování, za použití generativních algoritmů a několika programovacích jazyků. Mezi používané techniky a nástroje patří: JavaScriptu programování, programování s Processingem, programování p5.js, WebGL a Shadery, HTML a CSS pro webový portál; práce s hardwarem: mikrofony pro audio vstup, projektory, velkoplošné monitory a obrazovky; tištěná grafika. Vybrané projekty jsou popsány v následujících kapitolách.

5.1.1 Vizualizace dat jako výzva estetiky

Vystaveno:

- Výstava "See the wind", Galerie Hauerova 4, Opava, 2021;
- Nadace Hollar, Grafika roku 2021;
- LucernaTV – internetová vzdělávací televize, Slezská univerzita v Opavě, online;
- Výstava "Undercurrent", galerie G18, Zlín, 2022.

Nástroje: Programování WebGL a Shaderů, programování s Processingem, Excel, portály ShaderToy a OpenProcessing.

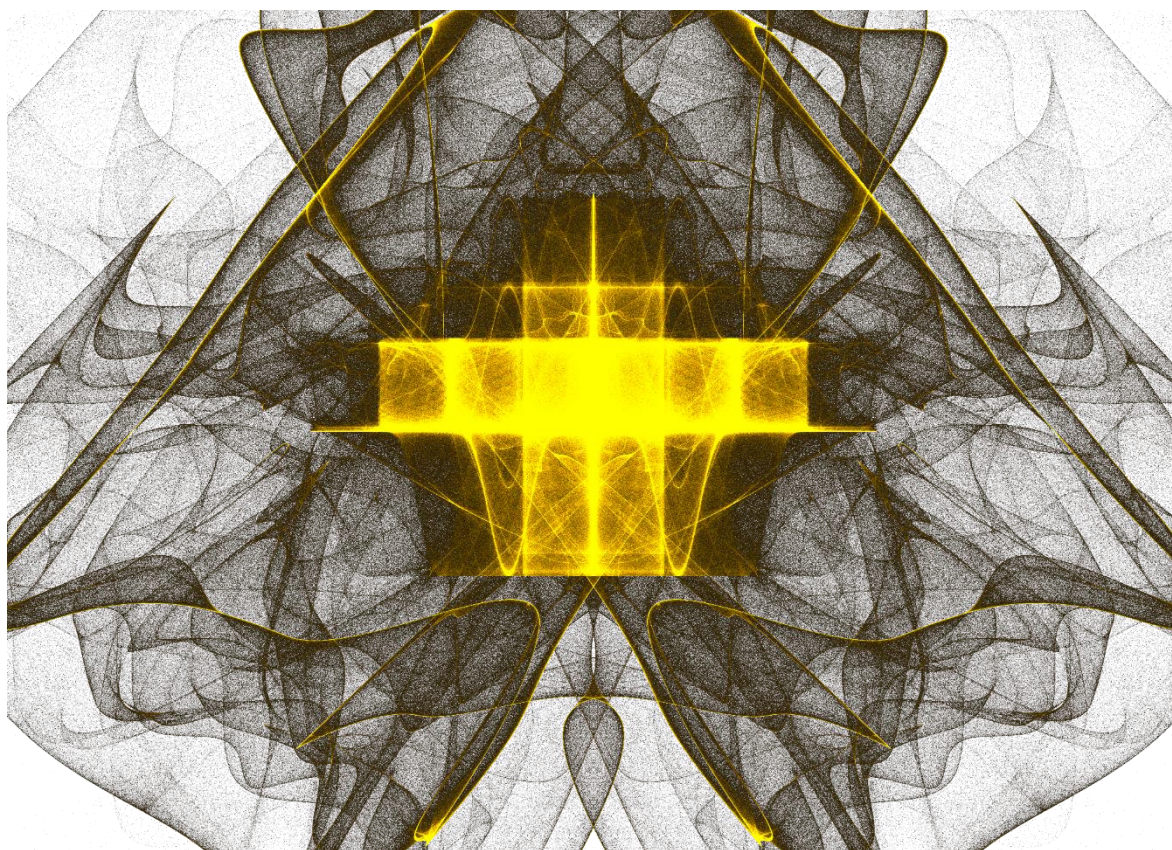
Projekt představuje vizualizaci velkoobjemových dat jako nový trend ve vizuálním umění. Hlavním cílem těchto vizualizací je podívat se na data z jině

perspektivy, přeměnit nudné a laikovi nicneříkající řady informací na umělecké, inspirativní a inspirující dílo.

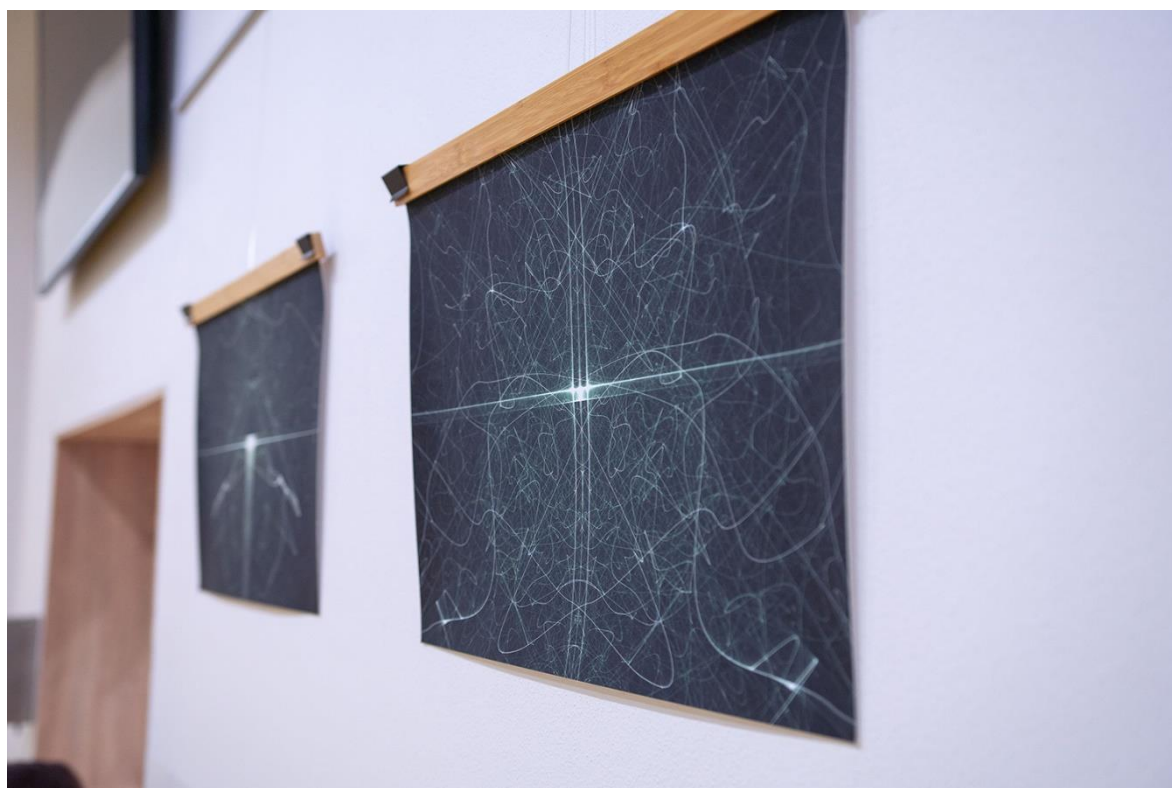
Projekt vznikl ve spolupráci s doc. PhDr. Miroslavem Zelinským CSc. a Centrem polymerních systémů UTB. Jde o první ukázky vizualizací velkého datového toku, který vzniká při analýze pokusů narušit strukturu materiálů pro výrobu pneumatik a dalších pryžových součástí, u kterých dochází k oděru a dalším způsobům opotřebení. Simuluje tak skutečnou realitu běžné jízdy, stavebních prací na velkých mobilních strojích nebo dopravních pásích. Data ukazují chování různých pryžových materiálů, experimenty jsou prováděny na laboratorním zařízení zvaném Instrumented Chip and Cut Analyzer (ICCA).

Tato data byla exportována jako velká sada excelových tabulek čítajících stovky řádků čísel. Projekt byl rozdělen do dvou částí, využívajících různé metody a sady nástrojů. První částí bylo vytvoření statické reprezentace získaných databází, obrázků, které budou později vytištěny jako malá a velká grafika. Zvoleným programovacím jazykem pro tuto část byl Processing – skvělá volba pro práci s jakýmkoliv druhem kreativních projektů, dostatečně rychlý na to, aby přeměnil velké datové sady na statickou vizualizaci.

Sady datových souborů byly umístěny v poli čísel, které určovaly polohu bodů a jejich spojení na obrazovce. Vzorec, který používá toto pole, bral čísla v náhodném pořadí, aplikoval na čísla další matematické operace, aby se dostal dále od standardní vizualizace dat ve formě grafu, který program Excel sám o sobě nabízí. Vznikla tak sada symetrických „zrcadlených“ obrázků a videí ukazujících proces jejich tvorby (viz obrázky 29-30).



Obrázek 29: Vizualizace dat jako výzva estetice1, Zdroj: vlastní



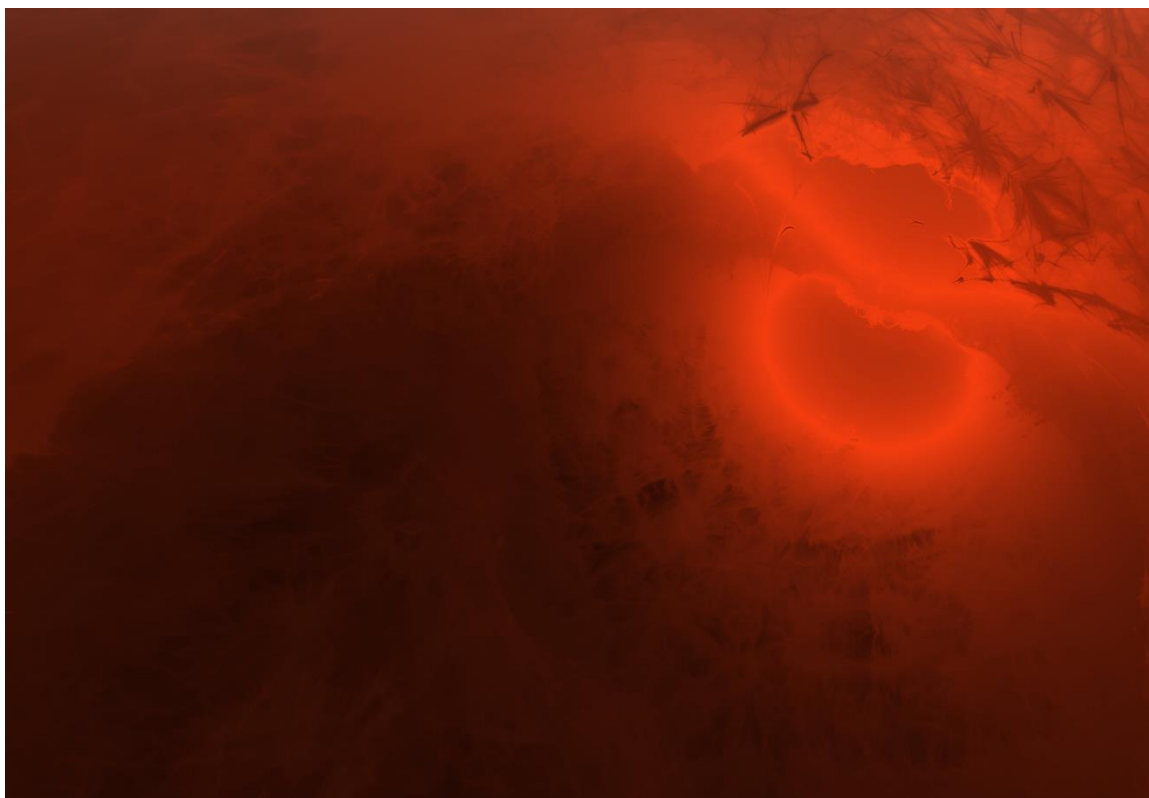
Obrázek 30: Galerie Hauerova 4, Opava. Fotograf: Jan Vlček

Další sada nástrojů byla použita k vytvoření dynamických vizualizací, které se budou pohybovat, rozšiřovat a měnit. Zvoleným prostředím je WebGL, Shaders a online platforma Shadertoy. WebGL se často používá k experimentování s dynamickými scénami, texturami, renderem, 3D objekty a interakcí s nimi. Na rozdíl od Processingu vyžaduje WebGL výkonnější grafickou kartu, protože má na starosti provedení renderu programu, proto byl použit výkonnější počítač.

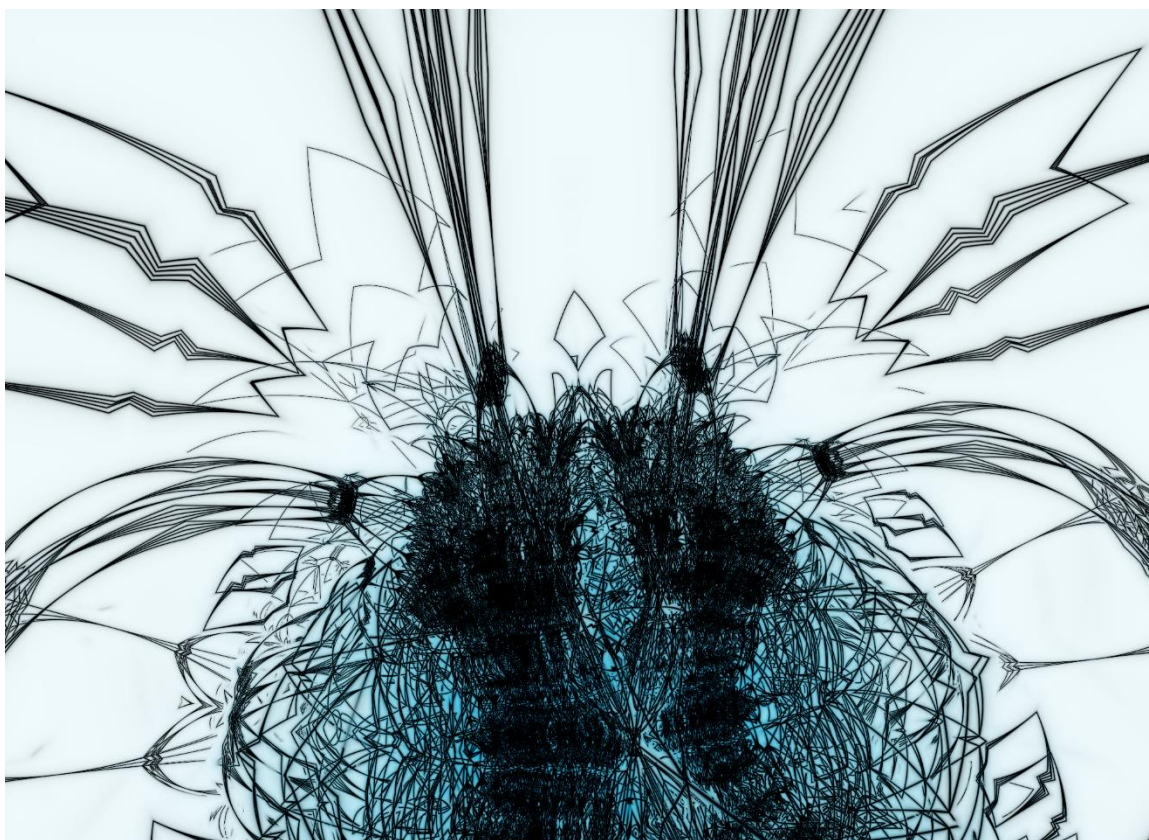
Čísla z datové sady byla vložena do pole a použita ve vzorcích k vytvoření 3D geometrického objektu, připomínajícího fraktál nebo rozsáhlou síť. Opět lze v této vizualizaci detekovat symetrii objektu. Objekt se pohybuje, ukazuje své různé části, přibližuje se k samotnému jádru nebo se vzdaluje, aby byl viděn v celé délce. Vizualizace má několik barevných variací, jedna v červené paletě s rozptýlenými měkkými liniemi, druhá přechází z modré/černé do zářivé zlaté s ostrými liniemi a tvary (viz obrázky 31-33).

Další experimentální částí tohoto projektu je sonifikace dat. **Sonifikace dat** je převod dat do neřečového zvuku za účelem přenosu informací nebo vnímání dat. Je to zvukový ekvivalent běžnější praxe vizualizace dat. Zvuková verze informací má výhody v prostorovém, amplitudovém a frekvenčním vnímání, otevírá další možnosti pro doplnění vizualizačních technik. Sonifikace dat je široce používána v astrofyzice při studiu astronomických objektů, jejich chování a vzdáleností.

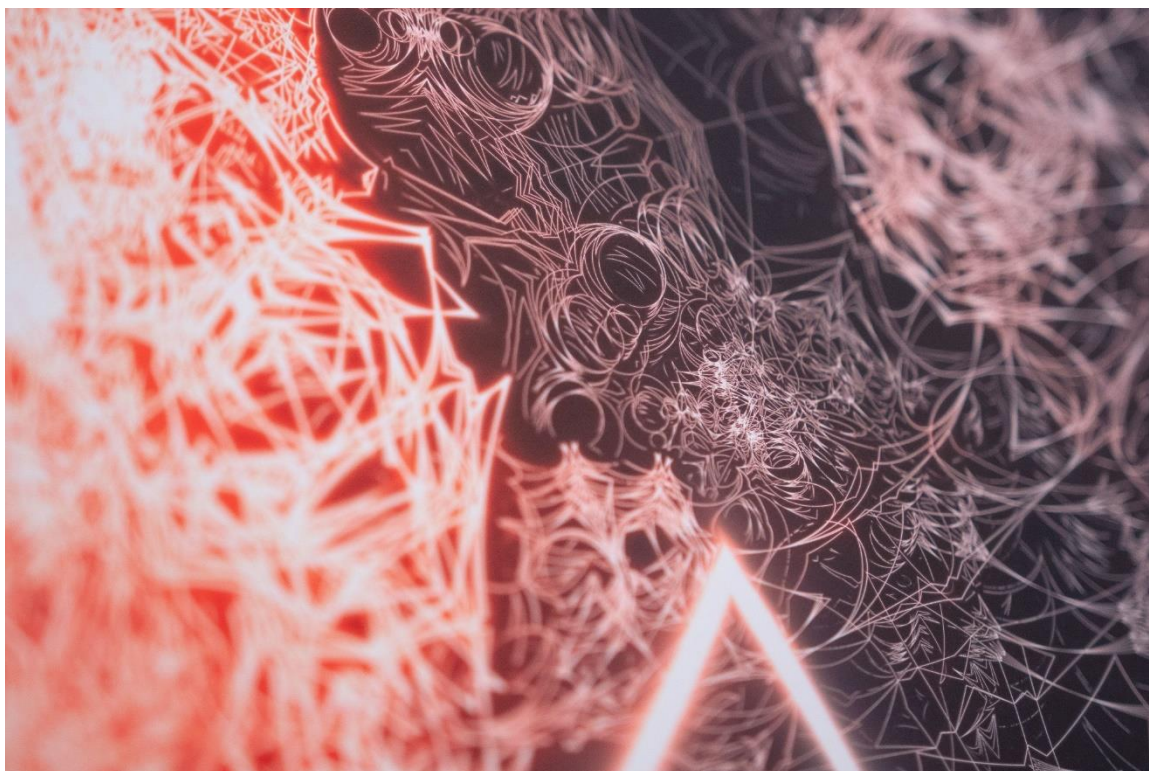
Portál WebGL a Shadertoy umožňuje pracovat se zvukovými informacemi, byly použity další algoritmy pro přeměnu číselných dat na zvuky, které obohatily výslednou vizualizaci. Dvě vizualizace s různými barevnými paletami byly natočeny jako videa a později prezentovány veřejnosti jako velkoplošné projekce na stěny galerie a na velkoplošných obrazovkách. Pomocí sluchátek mohli diváci nejen vidět, jak data testů odolnosti gumy alternativně vypadají, ale také jak zní.



Obrázek 31: Vizualizace dat jako výzva estetiky 1, Zdroj: vlastní



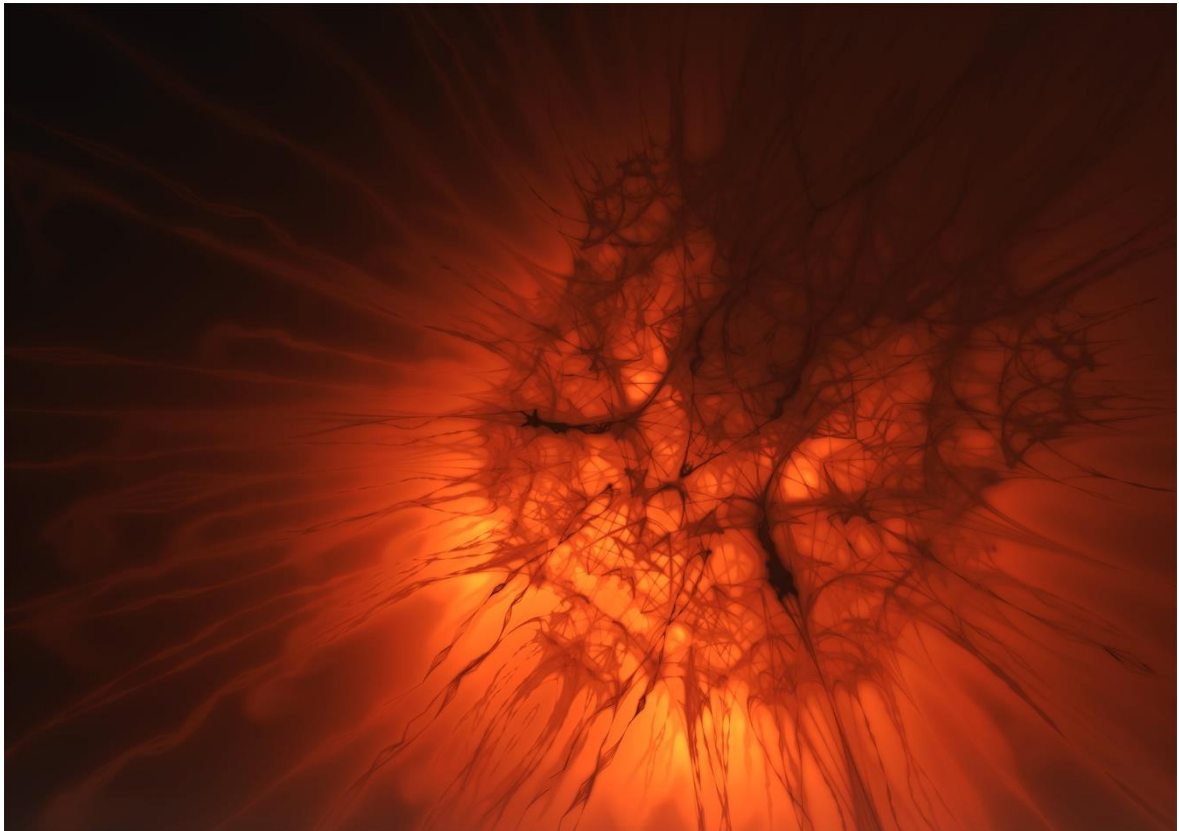
Obrázek 32: Vizualizace dat jako výzva estetiky 2, Zdroj: vlastní



Obrázek 33: Galerie Hauerova 4, Opava. Fotograf: Jan Vlček

Projekt má své pokračování v experimentálním rozšíření, které dostalo název „In my heart“ (V mém srdci). Původní WebGL vizualizace s červenou paletou barev (viz obrázek 31) byla obohacena o další sadu dat. Postupem popsáním v kapitole "4.2 Ukládání, organizace a transformace dat" byly pomocí systému kódování ASCII přeloženy řádky vět do číselné podoby. Tato čísla byla vložena do algoritmu, který určuje tvar objektu. Barvy byly také ovlivněny tímto procesem a posunuly paletu do zlatavějších tónů (viz obrázek 34).

Experiment měl určit, do jakého extrému ovlivní změna čísel uvnitř již tak obrovského pole vizuální podobu programu. Výsledek se lišil od originálu ve smyslu geometrie detailů objektu, od chaotičtějšího rozprostření dostal jasnější tvar. Což vede k závěru, že i nepatrné změny v datech, změna 0 na 1 nebo změna sinusu na kosinus, mohou vést ke vzniku nové dříve netestované reakce. Nová část projektu byla nahrána jako 10 minut dlouhé video, které bylo promítáno na zed' galerie během výstavy Undercurrent a pro nadcházející výstavy.



Obrázek 34: In my heart. Zdroj: vlastní

5.1.2 See the wind

Projekt vizualizace pro Symposium ArtColony v maďarském Ceredu, 2021.

Vystaveno:

- ArtColony Cered, Maďarsko, 2021;
- Výstava "See the wind", Galerie Hauerova 4, Opava, 2021;
- Výstava "Undercurrent", galerie G18, Zlín, 2022.

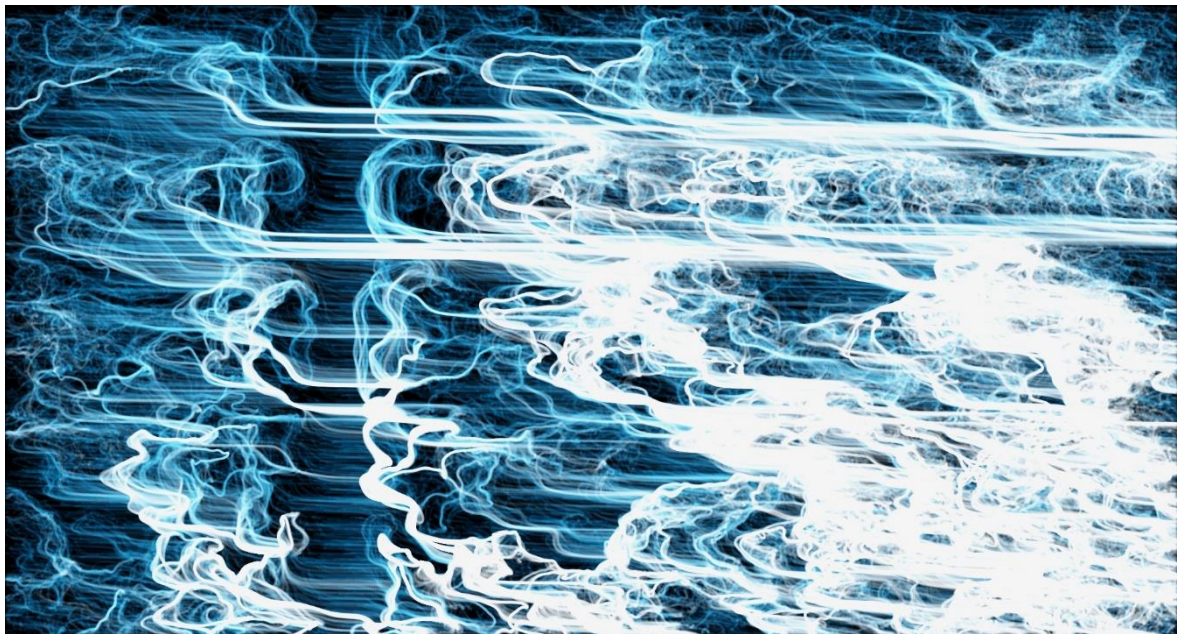
Použité materiály/programy: programování p5.js, portál Openweathermap.org – pro předpověď počasí, zvukové záznamy větru z telefonu, portál OpenProcessing.

Už jste někdy viděli vítr? Jeho hlas slyšíme ve stromech, v trávě a nad poli. Cítíme jeho dotek na naší kůži. Ale co kdybychom skutečně viděli jeho barvy vířit kolem nás? Vizualizace „See the wind“ (Vidět vítr) vznikla během týdenního pobytu v Ceredu v rámci projektu „Next Generation“. Po příjezdu do Ceredu by si měl člověk všimnout silného a hlasitého větru, který otřásá stromy kolem domů.

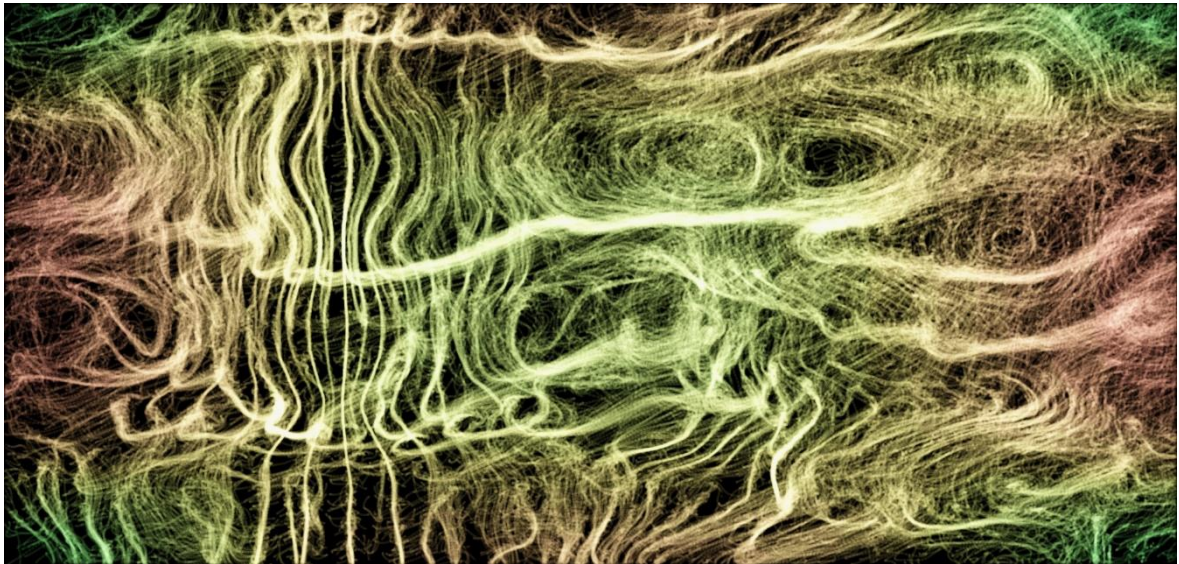
Tento vítr a jeho energie byly podnětem k práci a ukázaly tento fenomén z jiné perspektivy.

K vytvoření této vizualizace bylo použito prostředí Processing s jeho režimem p5.js. Program využívá data z nejbližší meteostanice prostřednictvím portálu Openweathermap. Portál sdílí v otevřeném přístupu údaje o počasí z celého světa, zobrazující teplotu vzduchu, vlhkost, stupeň a rychlost větru, poryvy větru, oblačnost atd. V prvním kroku tvorby algoritmu byla vytvořena mapa 1000-2000 malých bodů s náhodnou pozicí. Jakmile program získá data z meteostanice, mapa se probudí k životu. Vlivem údajů o větru se tečky začnou pohybovat a vykreslují jedinečný vzor větru.

Byly připraveny tři různé záznamy větrných pohybů, které byly zachyceny během pobytu v Ceredu. První je z nočního Ceredu, s cvrčky a slabým deštěm. Druhý, když byla obloha zatažená a v dáli zněla bouřka. Poslední je z chladného slunečného rána. Vítr ovlivňuje také barvy, nejsilnější poryvy zbělají (viz obrázky 35-36). Záznam byl prezentován formou videí obohacených o zvuky deště, vzdáleného hromu a větru.



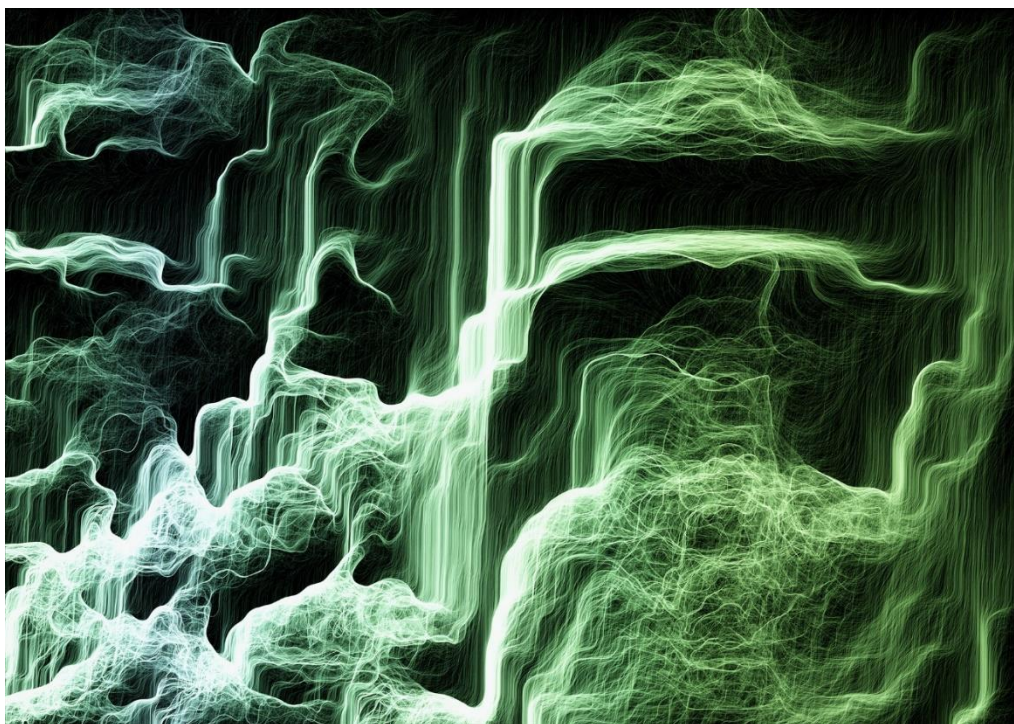
Obrázek 35: See the wind – blue. Zdroj: vlastní



Obrázek 36: See the wind – yellow. Zdroj: vlastní

Později byl projekt rozšířen tak, aby pokryl nejen Cered, ale i další velká i malá města po celém světě. Data je možné získat z jakéhokoli místa, kde je poblíž meteorologická stanice. Mezi studovaná města patří: Zlín, Praha, Brest, Kyjev, Charkov, Toronto, Berlín, Gíza, Jeneau a další. Program běží online prostřednictvím portálu OpenProcessing, kde ke sledování nahraných programů není nutná registrace ani členství. Pro správnou funkci je potřeba pouze internet – pro získání aktuálních dat, která meteorologické stanice aktualizují každých 10 minut, takže mapa a vítr jsou uměleckou reprezentací větrných proudů v reálném čase.

Uživatel může klepnout na obrazovku (v případě dotykových monitorů, tabletů nebo smartphonů), nebo kliknout myší a přepínat mezi různými městy. Pořadí měst je definováno programem a je náhodné.



Obrázek 37: *See the wind – Zlín. Zdroj: vlastní*

5.1.3 Everything flows, nothing remains

Projekt "Everything flows, nothing remains" (Vše plyne, nic nezůstává) pro sympozium PANTA RHEI.

Vystaveno:

- PANTA RHEI 2021, kino Hvězda, Uherské Hradiště;
- PANTA RHEI 2021, Banská Bystrica, Galerie FX;
- Nadace Hollar, Grafika roku 2021 – online;
- Výstava "Undercurrent", galerie G18, Zlín, 2022.

Nástroje: Programování WebGL a Shadery, portál ShaderToy.

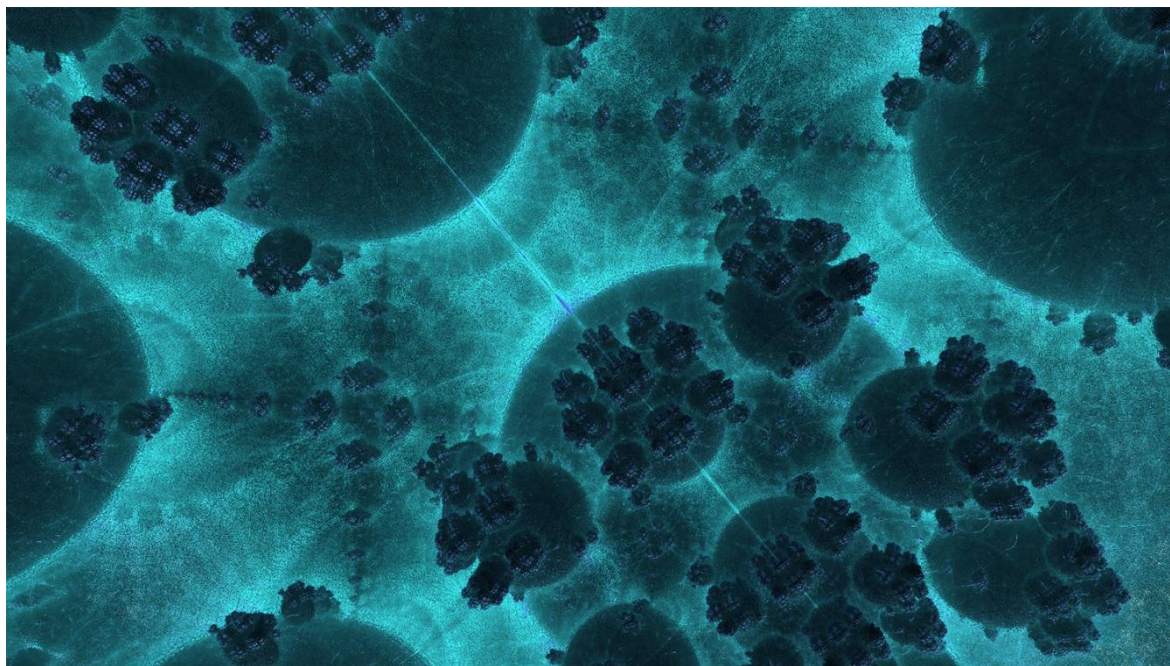
Vizualizace objektu je výsledkem spojení programovacích algoritmů a počítačových dat z frézy HWT, kterou tvůrci účastníci se symposia PANTA RHEI používali pro tvarování svých materiálů, od frézování dřeva po řezání plastů. Vizualizace byla vytvořena pomocí programování WebGL.

Data získaná z frézky byla použita k vytvoření 3D objektu, jehož forma byla také inspirována fraktálem - nekonečným geometrickým tvarem se složitými vzory, které jsou sobě podobné v různých měřítcích. Normálně se fraktály

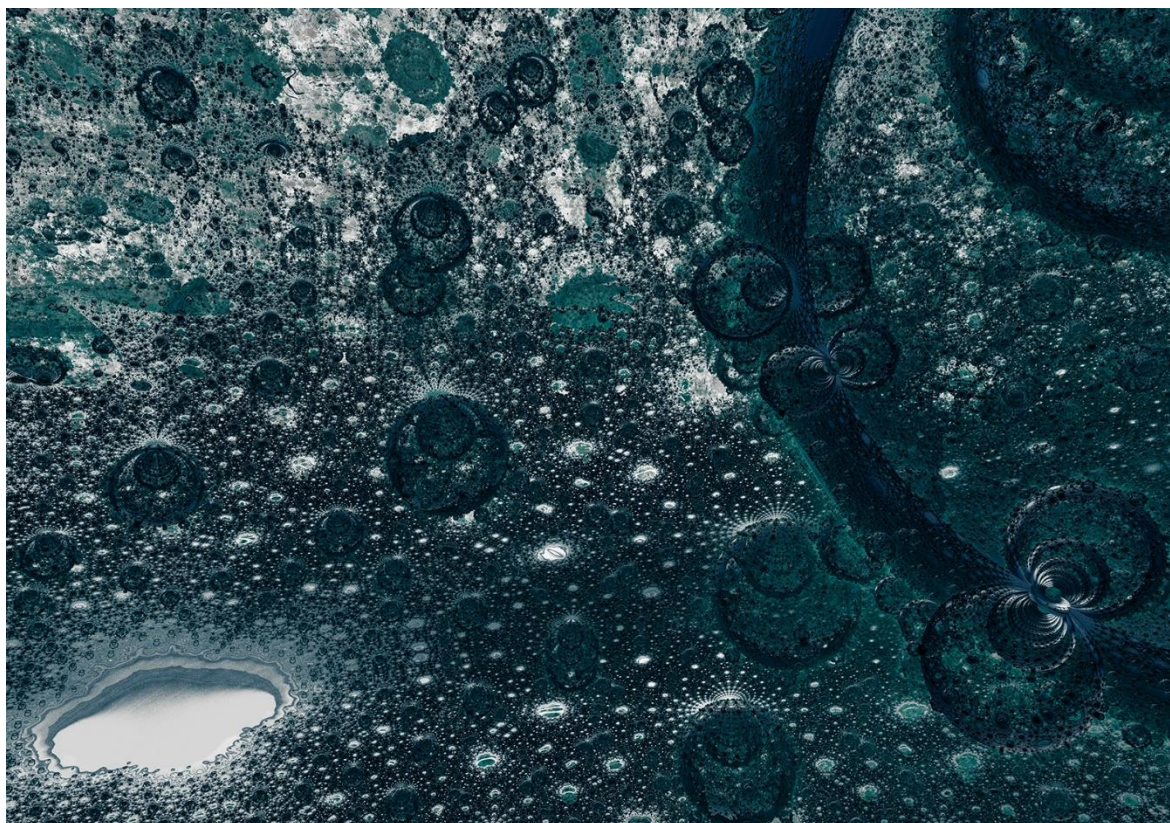
vytvářejí opakováním vzoru znovu a znovu v probíhající smyčce. Takové vzory lze často vidět v přírodě: větve stromů, sněhové vločky, krystaly, blesky a elektrina, rostliny, listy, říční systémy atd.

Fraktál „Everything flows, nothing remains“ je 3D objekt zbarvený do modré palety. Kamera, která zobrazuje výsledek na obrazovce, se však pohybuje mezi jeho různými součástmi, od vzdálených až po blízké části pomocí různých úhlů. Samotný objekt je stabilní, jeho základní geometrický tvar se nemění. Kamera, která zobrazuje výsledek na obrazovce, se však pohybuje jeho nejvzdálenějšími a nejbližšími částmi a zobrazuje objekt z různých úhlů. Při každé změně polohy kamery se používají různá nastavení fraktálu, jako je šířka a počet čar, které spojují hlavní uzly. Takže pokaždé, když kamera skočí skrz fraktál do nové pozice, výsledek vypadá trochu jinak.

Konečným výsledkem je 8minutové video, které ukazuje 16 různých částí a nastavení fraktálu, každé o délce 30 sekund. Vizualizace je doplněna zvuky z Nasa Space Sounds – zvuky vesmírných objektů, planet, satelitů a černých děr, neboť vizuál fraktálu připomíná vesmír ve svých mikro a makro měřítcích.



Obrázek 38: Fraktál 1. Zdroj: vlastní



Obrázek 39: Fraktál 2. Zdroj: vlastní

5.1.4 Symposium Art Colony Cered: Zvukový deník

Projekt vizualizace audio dat pro Symposium ArtColony v maďarském Ceredu, 2020.

Vystaveno:

- ArtColony Cered, Maďarsko, 2020 - online;
- LucernaTV – internetová vzdělávací televize, Slezská univerzita v Opavě, online;
- Výstava "Undercurrent", galerie G18, Zlín, 2022.
- Nástroje: Programování WebGL a Shadery, portál ShaderToy, media portal Soundcloud.

Vizualizace zvukového deníku (viz obrázky 40-41), která slouží jako pozvánka pro další ročník a zároveň tematická výzva výtvarného symposia Art Colony v maďarském Ceredu. Vizualizace vychází ze zvukového materiálu, který doc. PhDr. Miroslav Zelinský CSc. natočil během svého pobytu v Ceredu. Sada speciálně vyvinutých programovacích algoritmů analyzuje zvuky a mění je v abstraktní „živé“ objekty, které se pohybují, mění a vyvíjejí.

Audiovizualizace je praxe vytváření obrázků, videí, scén na základě zvukového souboru. Obvykle se takové vizualizace vykreslují v reálném čase a synchronizují se s přehrávaným zvukovým souborem. Výsledky se mohou lišit co do složitosti, od běžných vizualizací zvukových vln, které mnoho audio přehrávačů provádí automaticky (jako je Winamp, iTunes, Windows Media Player atd.), až po komplexní 3D scény, které kombinují použití jedinečných autorských algoritmů se zvukovými daty, kde práce může být nepředvídatelná. Výsledkem je něco jedinečného a působivého. Smícháním kódu, vizuální grafiky a zvukových dat posouvají autoři naše vnímání umění na novou úroveň.



Obrázek 40: Symposium Art Colony Cered: Zvukový deník. Zdroj: vlastní



Obrázek 41: Symposium Art Colony Cered: Zvukový deník. Zdroj: vlastní

Jedním z nástrojů, který lze použít k vytváření zvukových vizualizací (a byl použit k vizualizaci Audio deníku pro Cered), je Shadertoy – online platforma a komunita pro vytváření, renderování a sdílení procedurálních uměleckých děl, scén a online herních předmětů. Využívá WebGL – knihovnu, která pomáhá vytvářet interaktivní 2D a 3D grafiku v libovolném kompatibilním webovém prohlížeči (Chrome, Firefox atd.). Shadertoy umí pracovat s webovou kamerou a mikrofonem, videem, hudbou, vykreslováním virtuální reality. Shadertoy má vlastní způsob práce se zvukem. Všechny zvukové materiály by měly být nahrány online na mediální portál Soundcloud a poté importovány do programu prostřednictvím odkazů. Shadertoy čte a analyzuje informace o zvukových vlnách, jejich frekvenci, spektrum, tvar vlny a ukládá je jako texturu o velikosti 512 x 512 pixelů (velikost lze změnit). Uvnitř programu lze tuto texturu použít ke změně vzhledu objektu, jako je tomu u audio deníku. Tvary, vlny, čáry, tečky byly vytvořeny pomocí generativních algoritmů, které se začnou pohybovat a měnit při získávání dat ze zvukových souborů.

Konečný výsledek představuje několik minut dlouhé video, které vizualizuje soubor různých přírodních a lidmi vytvořených zvuků z vesnice Cered.

5.1.5 Under

Vizualizace zvukových vln.

Vystaveno:

- Výstava "Undercurrent", galerie G18, Zlín, 2022.

Nástroje: Programování WebGL a Shadery, mediální program Audacity.

Další příklad audiovizualizace s ShaderToy – projekt Under. Zde jde vizualizace jinou cestou. Základem je album Undercurrent jazzového pianisty Billa Evanse a jazzového kytaristy Jima Halla z roku 1962. Album bylo analyzováno softwarovým programem Audacity – bezplatný a open-source digitální audio editor a nahrávací aplikace. Program je schopen exportovat data zvukových vln ve formě číselných informací. Například jako:

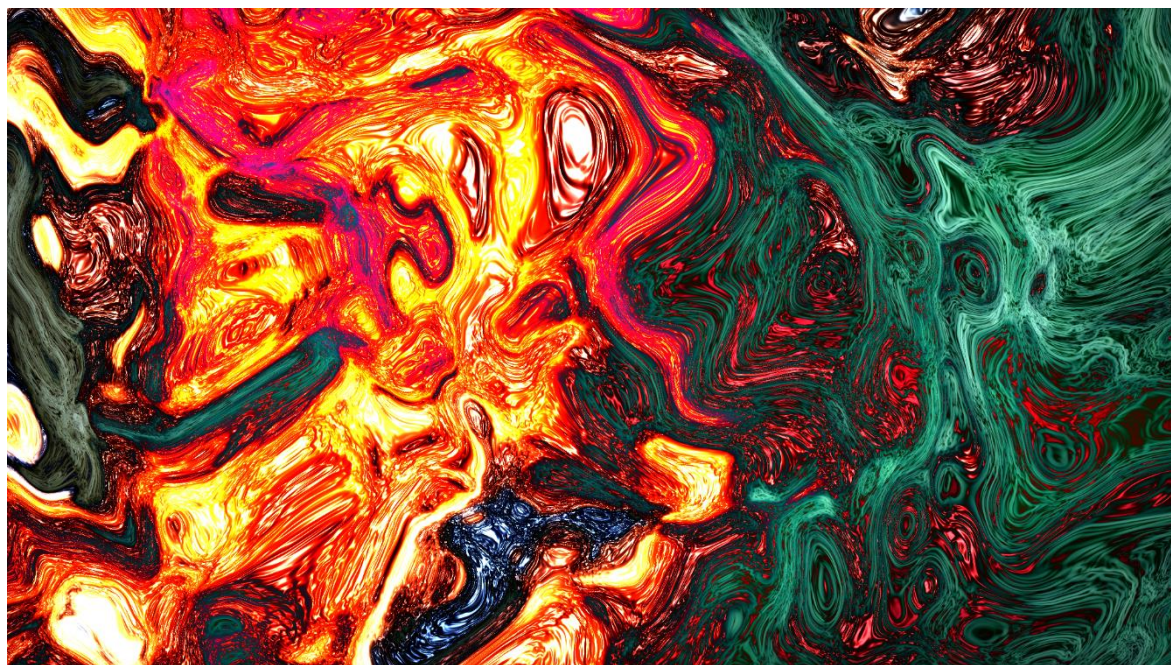
-168.57680 -168.57680

-154.59740 -154.59740

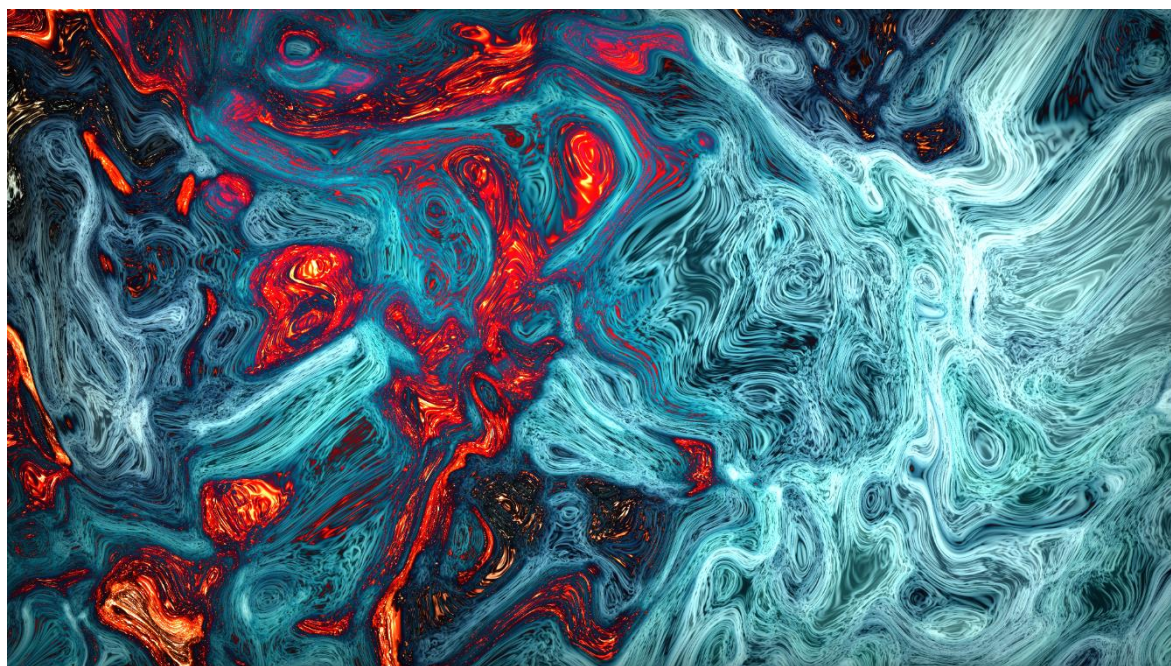
-159.03437 -159.03437

Jak bylo vysvětleno v předchozích kapitolách, při programování čísla ovlivňují vše: tvary, pohyb, rychlost, trajektorii, barvy. Čísla získaná z alba byla vložena

do počátečních programovacích algoritmů, které kreslí čáry a víry různých barev. Velká sada čísel vložených do kódu způsobí, že se víry mění a rostou a vytvářejí pohyblivý živý objekt. Tento projekt neprochází standardní cestou audio vizualizací, víry a linky se nepohybují do rytmu hudby, ale představují data z ní získaná. Vizualizace tedy prošla od zvukového formátu, přes číselné informace až po finální animaci.



Obrázek 42: Under 1. Zdroj: vlastní



Obrázek 43: Under 2. Zdroj: vlastní

5.1.6 Bright matter, Superčástice

Vizualizace dat z mezinárodního projektu CREDO.

Vystaveno:

- Výstava "Undercurrent", galerie G18, Zlín, 2022.

Nástroje: Programování WebGL a Shadery, portál Credo.science, mobilní aplikace CREDO Detector.

CREDO – The Cosmic Ray Extremely Distributed Observatory (extrémně distribuovaná observatoř kosmického záření) - mezinárodní vědecký projekt, jehož cílem je otestovat jednu z moderních teorií o tom, čím může být temná hmota vesmíru.

95 % vesmíru je nám neznámých. Největší část tohoto procenta se nazývá temná energie a je zodpovědná za expanzi vesmíru, kterou nelze jinak vysvětlit. Asi 27 % tvoří temná hmota. Jedna z teorií říká, že temná hmota se skládá ze superhmotných částic, které se zrodily v raném vesmíru, částice samotné nelze vidět ani detekovat, ale občas můžeme pozorovat výsledek jejich rozpadu – vysokoenergetické fotony. Tyto fotony putují vesmírem a procházejí i Zemí. Chytit je však je ošemetná věc, jsou poměrně vzácné a vědci by potřebovali detektor dlouhý několik desítek kilometrů, aby měli štěstí, že by jich pár odhalili.

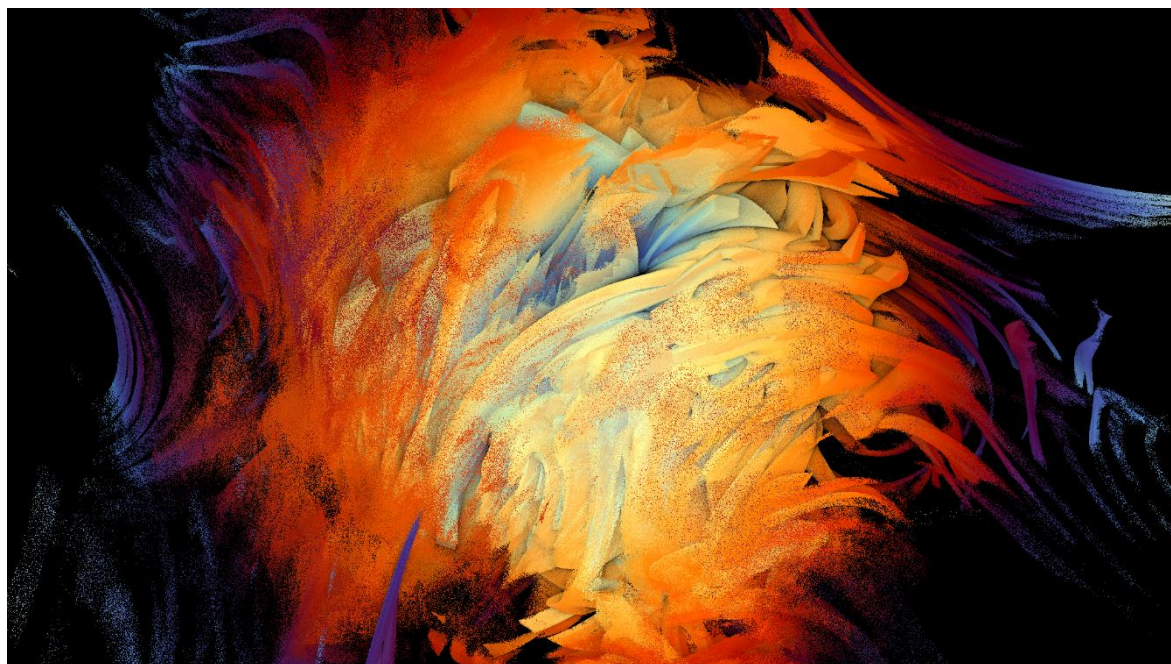
Projekt CREDO nabízí řešení: po celém světě jsou rozmístěny miliony mini detektorů, které lze použít k zachycení těchto částic – a jsou to naše chytré telefony. K tomu je zapotřebí jen chytrý telefon s fotoaparátem a speciální bezplatná aplikace od CREDO – CREDO Detector. Účastníci z celého světa si mohou vyzkoušet a zachytit částice spuštěním této aplikace, nálezy budou nahrány do online databáze na webu projektu. Výsledky, které se nejvíce podobají hledaným částicím, budou analyzovány vědci.

Přístup do této databáze je zdarma pro každého, kdo má o projekt zájem. Některá data byla stažena z hlavní databáze, také zmíněná aplikace byla použita k pokusu o zachycení několika částic. Data byla převedena do číselné podoby a vložena do 3D objektu vytvořeného programováním WebGL na portálu ShaderToy. Počáteční algoritmus kreslí velkou sadu částic různých tvarů a barev, které jsou náhodně rozmístěny na obrazovce. Částice nejsou statické, pohybují se, mění se, narážejí do sebe a vytvářejí chaotický vír. Data ovlivňují trajektorie, po kterých se tyto částice pohybují.

Program je interaktivní, reaguje na „dotek“ diváka. Na osobních počítačích a noteboocích mohou uživatelé používat myš a touchpad, na tabletech, chytrých

telefonech a dotykových obrazovkách – skutečný dotyk. Na základě polohy tohoto dotyku se pohybující objekt změní, po dotyku obrazovky více doprava se objeví více částic s chaotičtějším pohybem, dotykem doleva se částice srazí do ostřejších pevných geometrických tvarů.

Název Bright matter (světlá hmota) je antonymem zkoumané vědecké záhady – temné hmoty, neboť vizualizace ve skutečnosti ukazuje částice (nebo spíše jimi inspirované objekty) v plné barvě.



Obrázek 44: Částice CREDO. Zdroj: vlastní

K vytvoření alternativní verze vizualizace stejných dat byla použita stejná sada nástrojů, ale jiný algoritmus a přístup. Místo vytvoření sady mnoha částic počáteční algoritmus nakreslí jeden 3D objekt sférického tvaru. Data, která jsou vkládána do kódu, začnou měnit geometrii tohoto objektu, roztahovat jej a deformovat, vytvářet další vzory na jeho povrchu (viz obrázky 45-46). Objekt nezůstává stejný, pohybuje se a mění se, téměř se rozpadá a pak se zase dává dohromady v nekonečné smyčce. Tato druhá vizualizace získala název Sepurčástice, protože představuje data získaná z mnoha částic vložených do jednotného objektu.

Vizualizace nepředstavují, jak by částice mohly ve skutečnosti vypadat, jsou inspirací pro projekt, který umělecky představuje sílu Vesmíru. Projekt byl představen veřejnosti na výstavě Undercurrent, kde návštěvníci mohli nejen interagovat s částicemi dotykem na obrazovku, ale také si prohlédnout kód programu, změnit jej nebo přidat pár řádků pomocí klávesnice.



Obrázek 45: Superčástice CREDO. Zdroj: vlastní



Obrázek 46: Superčástice CREDO. Zdroj: vlastní

5.1.7 Pánev, kyčle

Vizualizace dat 3D pánve, kyčlí.

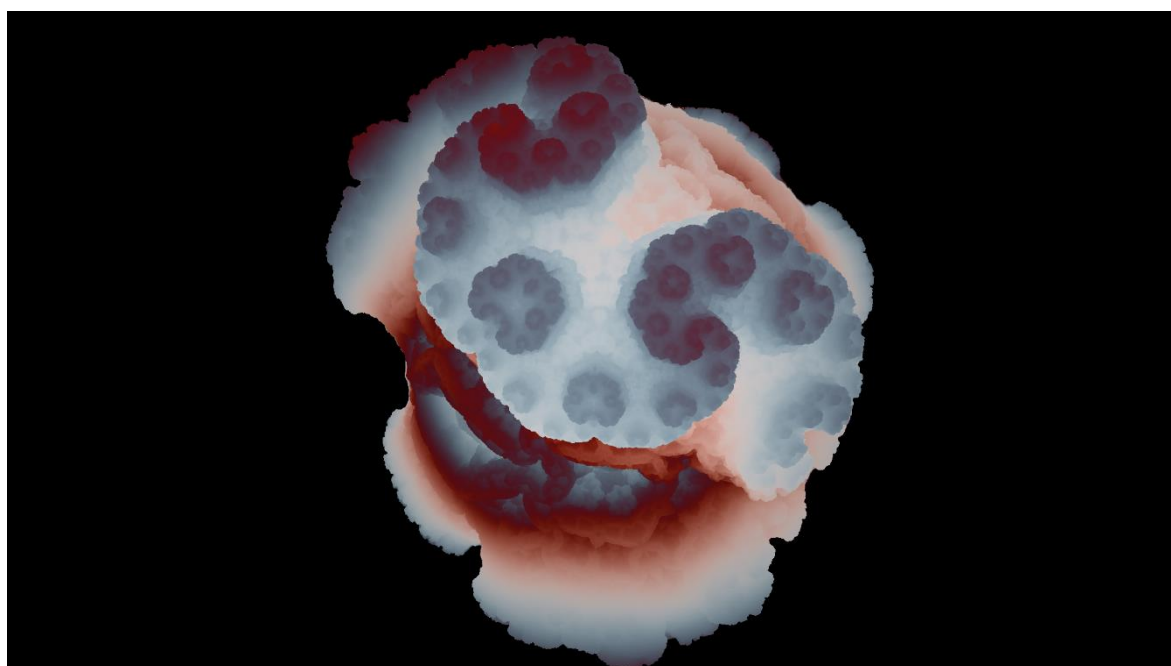
Vystaveno:

- Výstava "Undercurrent", galerie G18, Zlín, 2022.

Nástroje: Programování WebGL a Shadery.

Experimentální vizualizace dat získaných z 3D modelu sdíleného Slezskou univerzitou v Opavě, Fyzikálním ústavem. 3D modely přesně zmapovaných částí a orgánů lidského těla, jako je kyčle a pánev, byly exportovány jako číselná data popisující souřadnice každého bodu objektu.

Podle stejného principu jako u vizualizace Superčástice byl algoritmus sestaven tak, aby vytvořil sférický objekt. Do vzorce tohoto objektu byla vložena číselná data, která vedla ke změně jeho vzhledu. Objekt se pohybuje, otáčí a vytváří dojem „dýchajícího“ tvora, který se v rytmických intervalech rozširuje a stahuje. Objekt je interaktivní, reaguje na dotyk na obrazovce a kurzorem myši. S každým dotykem se mění, buď se stává chaotičtější, nebo se vrací do svého kulového tvaru. Stejně jako předchozí projekt byla tato vizualizace představena veřejnosti na výstavě Undercurrent s dostupným zdrojovým kódem k experimentování.



Obrázek 47: Vizualizace dat 3D pánve, kyčlí. Zdroj: vlastní

5.1.8 The stars of Cered

Vizualizace hvězdných dat pro Symposium ArtColony v maďarském Ceredu, 2022.

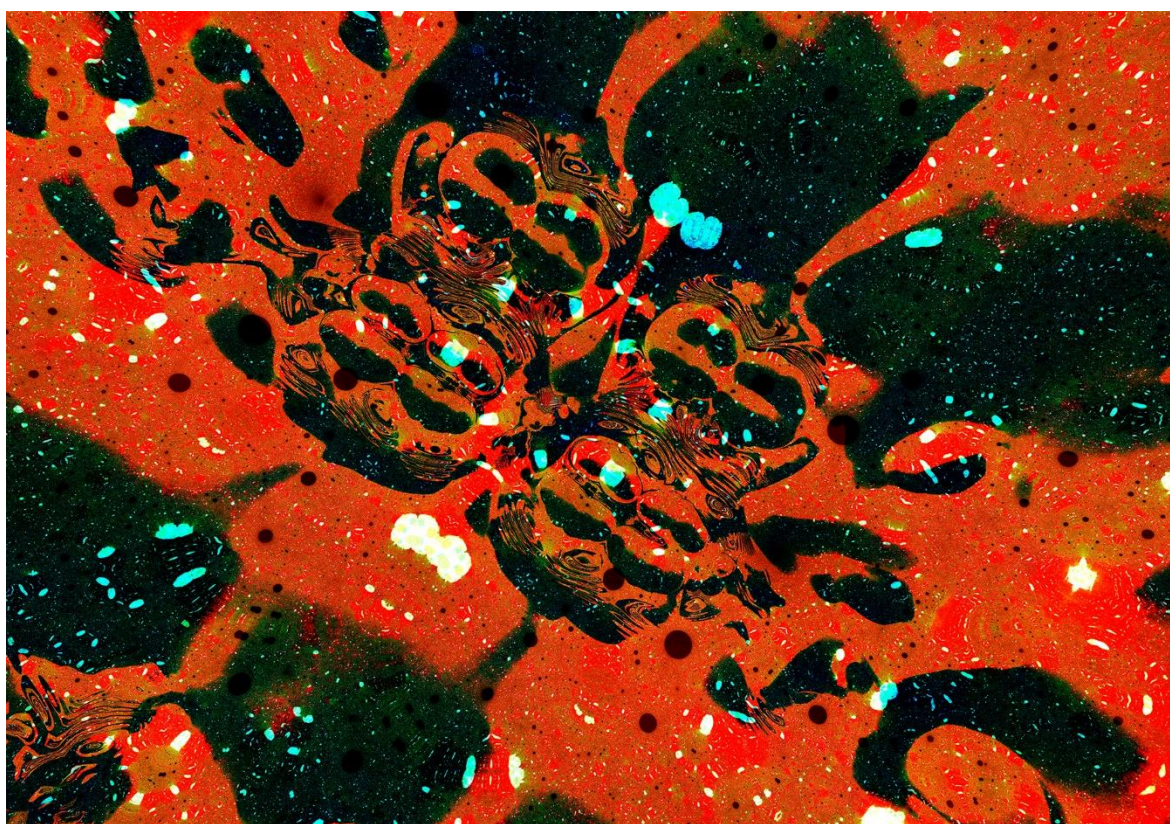
Vystaveno:

- ArtColony Cered, Maďarsko, 2022.

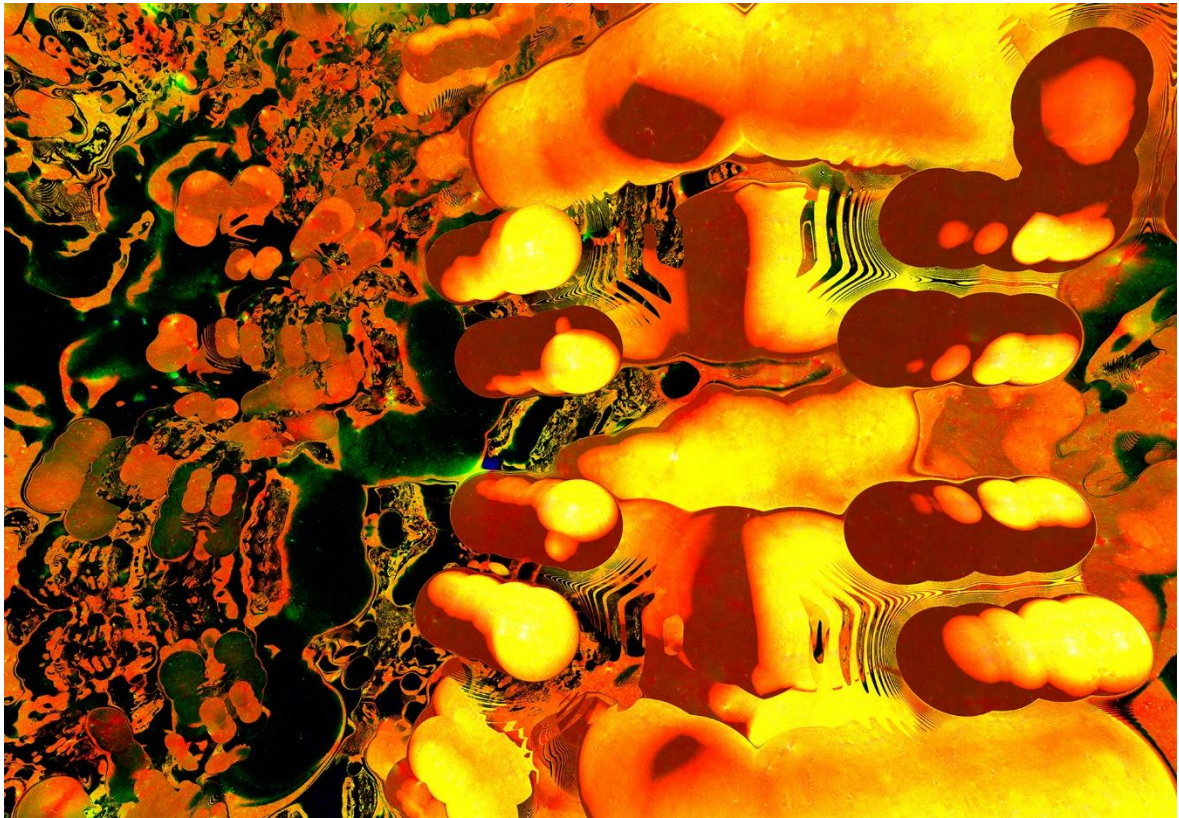
Nástroje: Programování WebGL a Shadery, Mobilní aplikace Star Chart, portál ShaderToy.

Technologie nám umožňují vidět dále, než bylo kdy možné pouhým okem. Mobilní aplikace Star Chart zachycuje vesmírné objekty daleko za horizontem: hvězdy, souhvězdí, galaxie, planety, komety, meteory, mlhoviny. Zobrazuje také vesmírná data: vzdálenost objektu od Země, absolutní a zdánlivou velikost, stupeň a azimut. Pomocí této aplikace lze objekty vzdáleného vesmíru pozorovat v kteroukoli denní i noční dobu.

Během sympozia Art Colony v Ceredu v roce 2022 byly tyto údaje, čtené řady čísel, shromážděny od mnoha hvězd a dalších vesmírných objektů. Byl použit stejný přístup jako u jiných dříve popsaných vizualizací: čísla byla vložena do algoritmů, které vykreslovaly mapu nestejných geometrických tvarů. Čísla ovlivňovala tyto tvary, jejich velikosti a vzdálenosti od sebe, stejně jako jejich pohyby. Kamera prochází bezmeznými kupami těchto objektů nebo nově naprogramovaných hvězd. Program běží online na portálu ShaderToy a může běžet v neomezené smyčce, naprogramovaný vesmír stejně jako ten skutečný je nekonečný. Pro prezentaci výsledku veřejnosti bylo natočeno několikaminutové video obohacené o zvukové efekty z NASA Space Sounds, které ukazuje oblohu Ceredu z jiné perspektivy.



Obrázek 48: *The stars of Cered.* Zdroj: vlastní



Obrázek 49: The stars of Cered. Zdroj: vlastní

5.2 Výstava " Undercurrent"

Hlavní praktický výstup je realizován formou **výtvarné výstavy**. Výstava se konala ve Zlíně, galerie G18.

Název: **Undercurrent**.

Kurátor: doc. PhDr. Miroslav Zelinský CSc.

Termín výstavy: 25.05.2022 – 9.06.2022.

Na stěny galerie bylo pomocí galerijních projektorů promítáno 5 audiovizuálních děl, velkoplošné monitory promítaly díla se zvukem, který mohli návštěvníci slyšet přes sluchátka, byly spouštěny interaktivní programy na 3 menších monitorech připojených k počítači, programy, které mohl každý návštěvník ovlivnit tak či onak dotykem obrazovky a nakonec bylo na výstavě představeno 8 velkoplošných plakátů s projekty vizualizace dat. Některé projekty, které byly na výstavě představeny, vznikly již dříve, jiné jsou dělané přímo pro tuto akci.

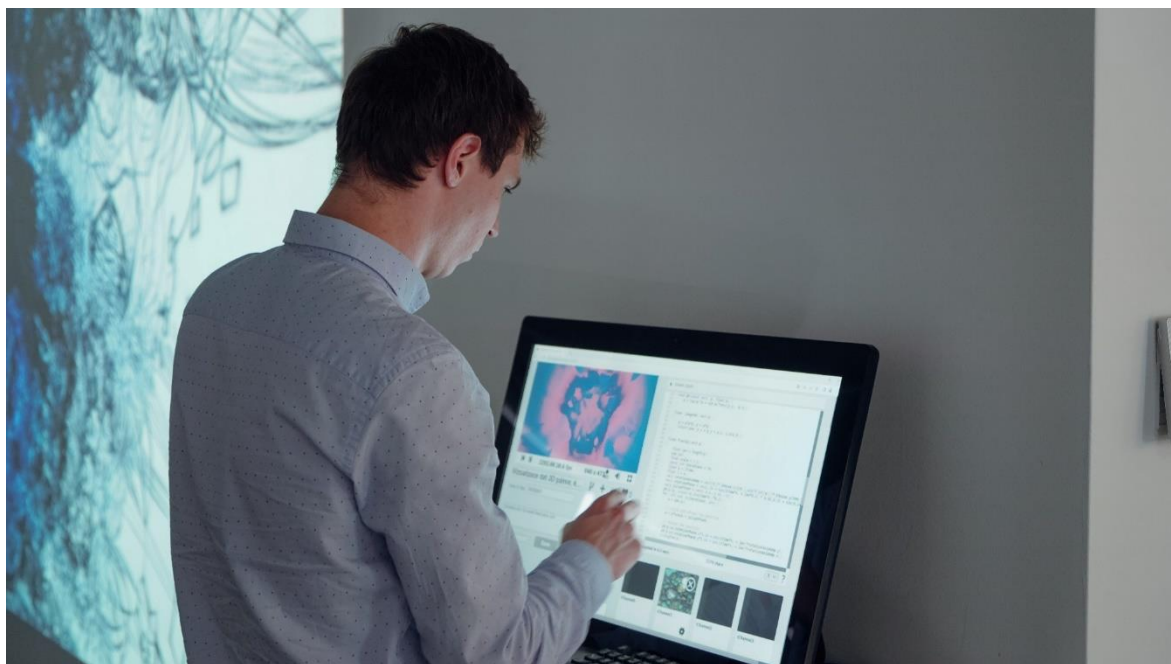
Každé dílo je počítačový program napsaný pomocí technik kreativního kódování. Každý počítačový program ovládá obraz na monitoru a zobrazuje různé

grafické objekty – měnící se, „rostoucí“ fraktální obrázky, trojrozměrné objekty a animace – vše generované ničím jiným než kódem. Některé vizualizace interagují s divákem – mění své tvary, barvy nebo rychlost na základě dotyku diváka na obrazovce nebo klávesnici, což dává divákovi možnost „komunikovat“ s programem. Na dvou monitorech propojených s počítačem se nejen přenášely vizualizace, ale návštěvníkům se zobrazoval i zdrojový kód programu. Každý ochotný návštěvník mohl do programu pomocí dostupné klávesnice přidat několik řádků kódu nebo jen čísel a ovlivnit tak vizuální výsledek. Některé z těchto programů obsahují zvukové vizualizace, a proto byly podpořeny zvukovým materiálem.

Každý předvedený program se liší podle programovacího jazyka, ve kterém je napsán, a tedy podle tvarů, barevné palety, pohybu, principů interakce. Vývoj každého programu prochází několika kroky. Prvním a asi nejdůležitějším krokem je najít myšlenku, téma a inspiraci, z čeho bude program vycházet a co bude reprezentovat. Hlavní inspirací pro tyto programy jsou generativní algoritmy a vizualizace dat. V závislosti na tom, co je vizualizováno – zvukový materiál, vědecké experimenty, přírodní síly, matematické vzorce a algoritmy, trojrozměrné objekty nebo obrázky atd., se vybírá vhodná programovací platforma. Programovací jazyky a platformy zvolené pro vizualizace jsou: Processing, JavaScript, WebGL, Shaders.



Obrázek 50: Výstava Undercurrent, foto: Bc. Eliška Janáková



Obrázek 51: Výstava Undercurrent, foto: Bc. Eliška Janáková



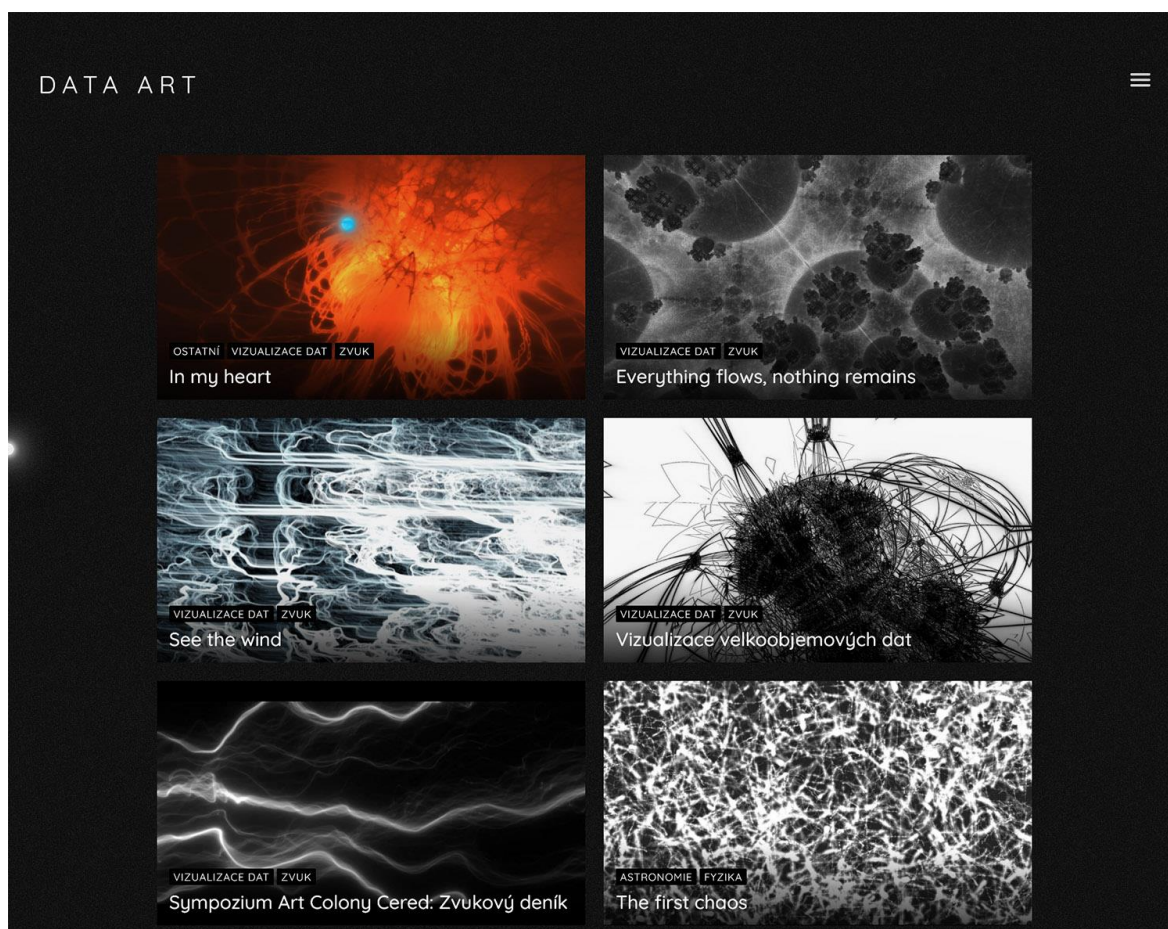
Obrázek 52: Výstava Undercurrent. Zdroj: vlastní

5.3 Online portál

Během studie byl autorkou navržen a naprogramován webový portál, který dostal název Data Art (viz obrázky 53-55). Web je plně funkční, veškeré informace o projektech a kreativním kódování jsou postupně doplňovány.

Adresa: <https://antrague.com/>

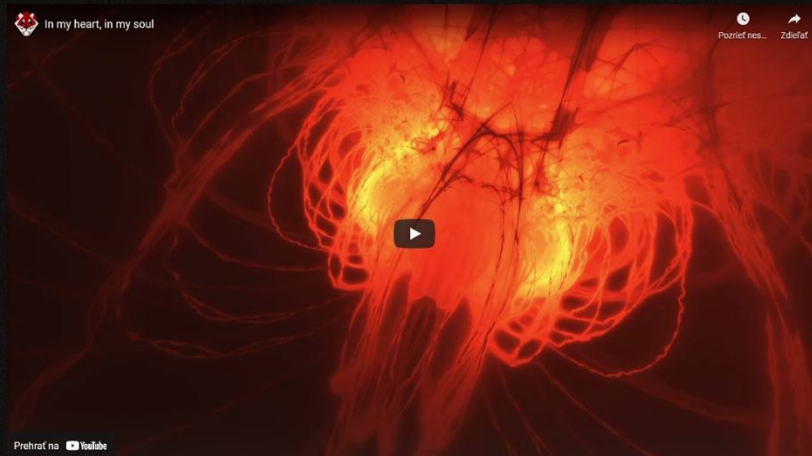
Hlavním cílem webu je ukládat všechny důležité informace o každém vytvořeném projektu, včetně praktických realizací: videa, obrázky, on-page běžící skripty a programy. Většina programů má odkazy na externí kódovací stránky věnované různým programovacím jazykům, kde si ostatní kodéři, umělci nebo studenti mohou prohlížet, kopírovat a používat veškerý kód a algoritmy. Některé projekty mají veškerý programovací kód dostupný přímo na stránce portálu (viz obrázek 54). Web slouží jako průvodce a možný zdroj inspirace pro ostatní. Každý zde nahraný projekt má všechny potřebné informace, včetně volby programovacího jazyka, popisu, metod tvorby. Web má několik různých kategorií, které představují inspiraci a základ každého projektu, jsou to: vizualizace dat, astronomie, fyzika, matematika, zvuk a některé další.



Obrázek 53: Web – Data Art. Zdroj: vlastní

In my heart

Tags: AUDIO FRAKTÁL WEBGL

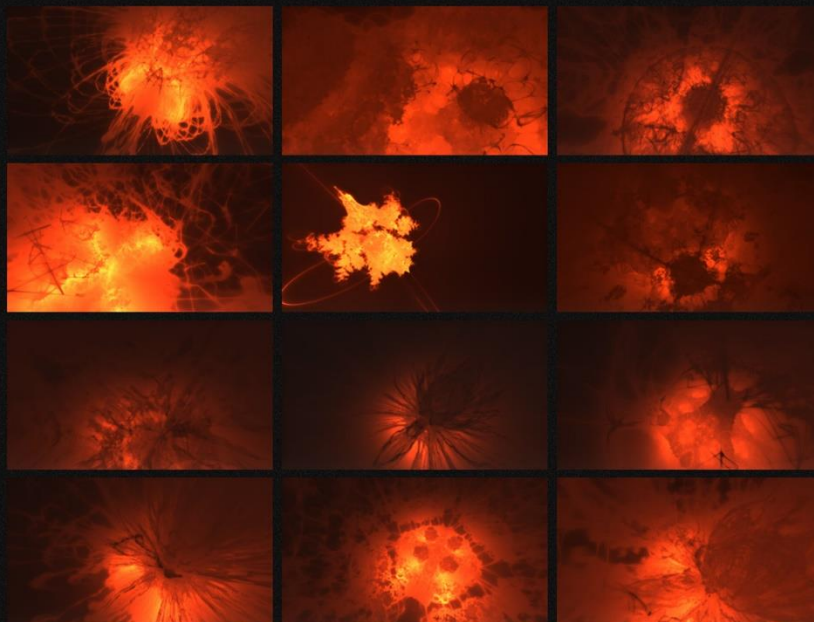


In my heart, in my mind, in my soul... (c) Uriah Heep

Speciálne pro mého L.

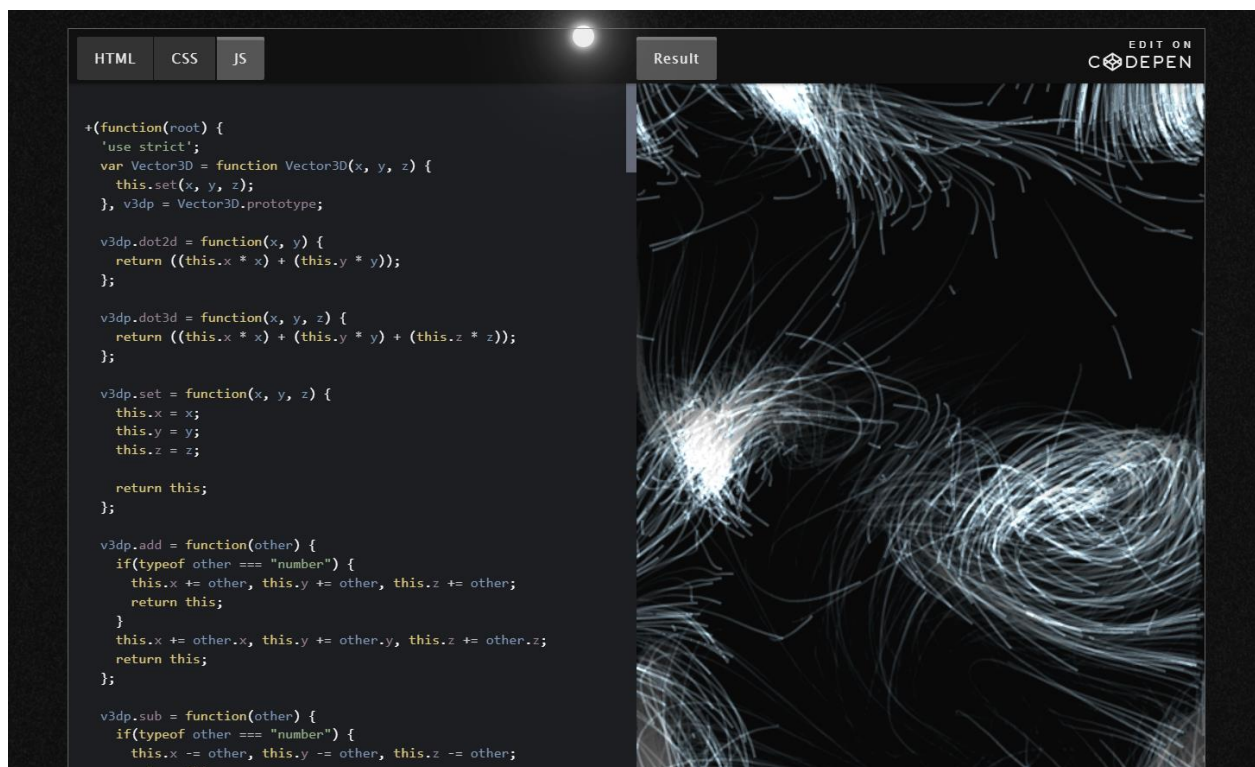
Slova lze měnit na čísla (pomocí ASCII kódů), čísla mohou ovlivňovat tvary, pohyby, dokonce i barvy. Základ tohoto programu naleznete zde: [Vizualizace velkoobjemových dat](#), avšak pozměněný a upravený o několik osobních vět, které byly v číselné podobě doplněny do kódu. I pár slov může znamenat velký rozdíl.

Píseň - "July morning" od Uriah Heep.



(c) 2021 Alex Lavrova

Obrázek 54: Obrázek 22: Web – Data Art. Zdroj: vlastní



Obrázek 55: Web – Data Art. Zdroj: vlastní

ZÁVĚR

"Možnosti jsou tady, musíme je jen využít" - Georg Trogemann a Jochen Viehoff (2005).

Rozvoj technologických možností je neustálým zdrojem inspirace a stimulace pro umělce po celém světě. Před pár lety nebylo možné vytvořit komplexní 3D objekt bez výkonného stroje a mnohahodinového testování. Dnes je to něco, co si může každý vyzkoušet na svém smartphonu. Technický potenciál bude nadále růst a přinášet nové formy umění a vizuálních médií, proto je důležité zůstat otevřený a přijímat nové koncepty. Jak vyplývá z teoretického rozboru a kapitoly „Definování pojmů“, kreativní kódování je akt použití programovacího softwaru k vytvoření děl vizuálního umění, designu, architektury, instalace, zvuku atd. Na rozdíl od tradičního zaměření kódování – funkčního aspektu, estetika zde hraje hlavní roli. Návrháři, kteří používají programování a kód jako své nástroje, jsou stále v menšině, ale tato čísla každým dnem rostou. Programování, kreativní kódování a vizualizace dat se stávají uznávanou formou umění a součástí moderní kultury.

Hlavním cílem disertační práce bylo ukázat možnosti kreativního kódování v rámci vizuálního interaktivního umění. Studie prochází teoretickým rozбором kreativního kódování, jeho historie a vývoje, současných výzkumů a objevů, umělců, kteří tyto techniky používají. V textu disertační práce byl uveden aktuální přehled principů kreativního kódování, jeho definic, komponent a nejpopulárnějších postupů v interaktivním výtvarném umění, teoretický základ pro technické aspekty a nástroje. Vzhledem k tomu, že jde o praktickou oblast umění, byla největší část studia zaměřena na praktické příklady, jako jsou programy, skici, dynamické webové stránky, 3D objekty, videa a animace vytvořené pomocí technik kreativního kódování.

Autorka použila originální přístup ke generativním algoritmům a vizualizaci dat při tvorbě souboru projektů, které byly veřejnosti představeny na výstavách, sympoziích, workshopech, přednáškách a online. Byla vytvořena online platforma, kde jsou prezentovány nejzajímavější projekty, obsahující kód každého díla, který mohou ostatní uživatelé používat, měnit a rozvíjet. Byla navázána spolupráce s dalšími organizacemi, vědci a umělci.

Autorka věří, že představení širšího kontextu, jednotlivých uměleckých projektů a různých aspektů kreativního kódování povede k lepšímu pochopení této disciplíny a podstaty generativního umění. Přehled metod a principů vizualizace dat poskytne poměrně přesnou představu o tvůrčím procesu při kódování a může být využit v pedagogické praxi.

SEZNAM POUŽITÉ LITERATURY

- BACKUS, John. *FORTRAN Session. The history of fortran i, ii, and iii. Association for computing machinery, inc. History of programming languages.* Academic Press, 1981. ISBN 0-12-745040-8.
- BODEN, Margaret a Ernest EDMONDS. *What is generative art?. Digital creativity.* 2009, 20, 32. Dostupné z: doi:10.1080/14626260902867915
- BOHNACKER, Hartmut, Benedikt GROSS a Julia LAUB. *Generative Design: Visualize, Program, and Create with Processing.* 1. Princeton Architectural Press, 2012. ISBN 1616890770.
- BOVILL, Carl. *Fractal geometry in architecture and design.* Boston: Birkhauser, 1996. ISBN 0-8176-3795-8.
- CERIC, Vlatko. *Algorithmic Art: Technology, Mathematics and Art.* Trg J.F.Kennedyja 6, 10000 Zagreb, Croatia: Faculty of Economics and Business, 8. DOI: 10.1109/ITI.2008.4588386.
- COLUBRI, Andrés. *Processing for android.* New York, NY: Springer Science+Business Media, 2017. ISBN 9781484227183.
- EDMONDS, Ernest a Linda CANDY. *Interacting: Art, research and the Creative Practitioner. Interactive Art.* Faringdon: Libri Publishing, 2011, 18–32.
- FLANAGAN, David. *JavaScript: the definitive guide.* Beijing; Farnham: O'Reilly, 2011. p. 1. ISBN 978-1-4493-9385-4. OCLC 686709345.
- FRITZ, Darko. *Mapping the Beginnings of Computer-generated Art in the Netherlands. Beginnings of the Computer-generated Art in the Netherlands.* Amsterdam, 2011.
- FRY, Benjamin Jotham. *Computational information design.* 2004. Disertační práce. Massachusetts Institute of Technology. Vedoucí práce John Maeda.
- GALANTER, Philip. *What is Generative Art? Complexity theory as a context for art theory.* GA2003 – 6th International Conference on Generative Art, 2003.

- GREENBERG, Ira. *Processing: creative coding and computational art*. New York: Distributed to the book trade worldwide by Springer-Verlag, c2007. ISBN 159059617X. p. 13, 19-20, 31.
- IPPOLITO, Jon. *Ten Myths of Internet Art*. *Leonardo*. 2002, 485-498. Dostupné z: doi:10.1162/002409402320774312
- JACOBS, Jennifer. *Dynamic Drawing: Broadening Practice and Participation in Procedural Art*. 2017. Disertační práce. Massachusetts Institute of Technology. Vedoucí práce Mitchel Resnick.
- MAEDA, John. *Design by Numbers*. MIT press, 1999. ISBN 9780262133548
- MAGNENAT-THALMANN, Nadia a Daniel THALMANN. *Computer science.: Theory and practice*. Tokyo: Springer-Verlag, 1985. ISBN 978-4-431-68435-0.
- MAGNUSSON, Thor. *The threnoscope: a musical work for live coding performance. First International Workshop on Live Programming in conjunction with ICSE 2013*. San Francisco, CA, 2013.
- MITCHELL, Tom. *Machine Learning*. New York: McGraw Hill. 1997. ISBN 0-07-042807-7. OCLC 36417892.
- NAKE, F. (2005), 'Computer Art: A Personal Recollection', in L. Candy (ed.) *Creativity and Cognition 2005*. Proceedings of the Fifth Conference on Creativity and Cognition, 12-15.04.2015, Goldsmiths College (New York: ACM Press): 54-62.
- NAKE, Frieder. *Ästhetik als Informationsverarbeitung: Grundlagen und Anwendungen der Informatik im Bereich ästhetischer Produktion*. Wien: Springer-Verlag, 1974. ISBN 3709170982.
- NEES, Georg. *Generative Computergraphik*. Berlin, 1969. Disertační práce. Universities of Erlangen-Nürnberg.
- NOLL, A. Michael. *Computers and the Visual Arts. Design and Planning 2: Computers in Design and Communication*. New York: Hastings House, Publishers, 1967, 36(2), 65–79.
- NOLL, A. Michael. *Short-Time Spectrum and Cepstrum Techniques for Vocal-Pitch Detection*. *Journal of the Acoustical Society of America*. 1964, 36(2), 296–302.

- PAUL, Christiane. *A Companion to Digital Art*. Boston, Massachusetts: Wiley Blackwell, 2016. ISBN 9781118475188.
- RATHOD, Pinky, PATEL, Avani. “*A Novel Approach for Moving Object Detection from Dynamic Background.*” *Int. Journal of Engineering Research and Applications* www.ijera.com ISSN: 2248-9622, Vol. 5, Issue 3, (Part -2) March 2015, p.71-74
- REAS, Casey., FRY, Ben. *Processing: a programming handbook for visual designers and artists*. Cambridge, Mass.: MIT Press, 2007. ISBN 0-262-182629, p. xxiii – xxiv, p. 1-2.
- RICHARDSON, Andrew. *Data-driven graphic design: creative coding for visual communication*. New York: Fairchild Books, an imprint of Bloombury Publishing, 2016.
- SALAH, Uddin Ahmed. *Interaction and Interactivity: In the Context of Digital Interactive Art Installation. Human-Computer Interaction. Interaction in Context*. 2018, 241-257. Dostupné z: doi:10.1007/978-3-319-91244-8_20
- SHIFFMAN, Daniel. *The nature of code: simulating natural systems with processing*. Version 1.0, generated December 6, 2016. New York: Free Software Foundation, 2012. ISBN 9780985930806.
- SHREINER, Dave, KESSENICH, John, SELLERS, Graham. *OpenGL programming guide: the official guide to learning OpenGL, version 2*. 9th ed. Upper Saddle River: Addison-Wesley, 2017. ISBN 0-321-33573-2.
- SMITH, Glenn. *An Interview with Frieder Nake*. *Arts*. 2019, 8(2). Dostupné z: doi:10.3390/arts8020069
- SPEDDING, James, Robert Leslie ELLIS a Douglas Denon HEATH. *The Works of Francis Bacon*. Taggard and Thompson. Boston, 1863.
- *Technology and the arts: Past, present and future synergies*. STOA Options Brief. EU: European Parliament / Science and Technology Options Assessment (STOA), 2019, 4.
- TROGEMANN, Georg, Jochen Viehoff. *CodeArt: eine elementare Einführung in die Programmierung als künstlerische Praktik*. Springer, 2005. ISBN-10: 3211204385.

- WANG, Ge a Perry COOK. *2004: On-the-Fly Programming: Using Code as an Expressive Musical Instrument*. A NIME Reader. 2017, 193-210. Dostupné z: doi:10.1007/978-3-319-47214-0_13.
- WRIGHT, Richard S., SELLERS, Graham, HAEMEL, Nicholas. *OpenGL superbible: comprehensive tutorial and reference*. 7th ed. Boston: Addison-Wesley, 2016. ISBN 9780672337475.

Elektronické zdroje

- #797: *VR Artist Chelley Sherman: Heightening Perception with Sonic Architecture, Haptics, & Dynamical Cognitive Science*. *Voices of VR Podcast* [online]. 2019 [cit. 2022-02-07]. Dostupné z: <https://voicesofvr.com/797-vr-artist-chelley-sherman-heightening-perception-with-sonic-architecture-haptics-dynamical-cognitive-science/>
- *1970S - EARLY 1980S*. *Computer history museum* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.computerhistory.org/brochures/1970s-early1980s/>
- *7090 Data Processing System*. *IBM* [online]. 2021 [cit. 2022-05-07]. Dostupné z: https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP7090.html
- *A Modern Prometheus: The History of Processing by Casey Reas and Ben Fry*. *Processing Foundation* [online]. 2018 [cit. 2022-02-07]. Dostupné z: <https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85>
- *A. Michael Noll*. *USC Annenberg* [online]. [cit. 2022-01-07]. Dostupné z: <https://annenberg.usc.edu/faculty/michael-noll>
- *Alida Sun: Details for Alida Sun*. *Art Blocks Community Wiki* [online]. [cit. 2022-02-07]. Dostupné z: <https://artblocks.wiki/Artists/alida-sun>
- ANADOL, Refik. *Refik Anadol: About*. *Refik Anadol* [online]. 2022 [cit. 2022-02-07]. Dostupné z: <https://refikanadol.com/about>
- *Approximating Reality with Interactive Algorithmic Art*. *MAT: Art and Technology* [online]. 2001 [cit. 2022-02-02]. Dostupné z: https://www.mat.ucsb.edu/~g.legrady/academic/courses/01sp200a/students/brentYokota/200a_fin.html

- BAILEY, Jason. *Why Love Generative Art? Artnome* [online]. 26.08.2018 [cit. 2022-05-07]. Dostupné z: <https://www.artnome.com/news/2018/8/8/why-love-generative-art>
- BARSKY, Michael. *10 innovative creative coding examples that are redefining programming*. Editor X [online]. 3.13.2020 [cit. 2021-10-07]. Dostupné z: <https://www.editorx.com/shaping-design/article/creative-coding>
- BATTISTA, Anna. *Can Data Visualisation Be Beautiful? Yes, It Can, Says Artist Ryoji Ikeda*. Irenebrination [online]. 2019 [cit. 2022-08-17]. Dostupné z: https://www.irenebrination.com/irenebrination_notes_on_a/2019/05/ryoji-ikeda-venice.html
- Ben Fry. *Liu Lectures* [online]. 2009 [cit. 2022-02-07]. Dostupné z: <http://liulectures.stanford.edu/2009/11/18/ben-fry/>
- BHAGAT, Richa. *Futurist and Generative Artist Alida Sun Lives Life on the Internet*. *The crypto times* [online]. 2022 [cit. 2022-02-07]. Dostupné z: <https://www.cryptotimes.io/futurist-and-generative-artist-alida-sun-lives-life-on-the-internet/>
- Casey Reas. *The European Graduate School / EGS* [online]. [cit. 2022-02-07]. Dostupné z: <https://egs.edu/biography/casey-reas/>
- CLARK, Mitchell. *NFTs, explained*. *The Verge* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://www.theverge.com/22310188/nft-explainer-what-is-blockchain-crypto-art-faq>
- *CodePen: About*. *CodePen* [online]. [cit. 2022-02-07]. Dostupné z: <https://codepen.io/about/>
- *Compilation*. *Computer Hope* [online]. 2017 [cit. 2022-03-11]. Dostupné z: <https://www.computerhope.com/jargon/c/compilat.htm>
- *Compos 68*. *Monoskop* [online]. 2012 [cit. 2022-05-07]. Dostupné z: https://monoskop.org/Compos_68
- CONTI, Robyn a John SCHMIDT. *What Is An NFT? Non-Fungible Tokens Explained*. *Forbes Advisor* [online]. 2022 [cit. 2022-02-07]. Dostupné z: <https://www.forbes.com/advisor/investing/nft-non-fungible-token/>
- CRAMER, Florian. *Concepts, Notations, Software, Art*. *Cramer* [online]. 23.03.2002 [cit. 2021-05-17]. Dostupné z:

http://cramer.pleintekst.nl/all/concept_notations_software_art/concepts_notations_software_art.html

- *Creative coding* [online]. Wikipedia, 2014 [cit. 2022-02-02]. Dostupné z: https://en.wikipedia.org/wiki/Creative_coding
- DELMON, Cecil. *WebGL, the wow effect. Joli code* [online]. 2018 [cit. 2022-03-11]. Dostupné z: <https://jolicode.com/blog/webgl-the-wow-effect>
- *Domo: Data never sleeps* [online]. 2020 [cit. 2020-05-01]. Available from: <https://www.domo.com/solution/data-never-sleeps-6>
- DONOVAN, Alexander. *Everything You Need to Know About the Artistic World of Creative Coding. Interesting engineering* [online]. 10.11.2020 [cit. 2022-03-07]. Dostupné z: <https://interestingengineering.com/everything-you-need-to-know-about-the-artistic-world-of-creative-coding>
- *Edward E. Zajac. CompArt daDA: the database Digital Art* [online]. [cit. 2022-02-07]. Dostupné z: <http://dada.compart-bremen.de/item/agent/683>
- *Frieder Nake. CompArt daDA: the database Digital Art* [online]. [cit. 2022-02-07]. Dostupné z: <http://dada.compart-bremen.de/item/agent/68>
- FRY, Ben. *Valence. Ben Fry* [online]. [cit. 2022-02-07]. Dostupné z: <https://benfry.com/valence/index.html>
- *George Legrady. Artnet* [online]. [cit. 2022-02-07]. Dostupné z: <http://www.artnet.com/artists/george-legrady/biography>
- *GLSL Shaders. MDN CONTRIBUTORS. Developer Mozilla* [online]. 2022 [cit. 2022-03-11]. Dostupné z: https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/GLSL_Shaders
- GORDON, Chloe. *Top Five Data Visualization Artists To Follow On Instagram. Printmag* [online]. 13.08.2021 [cit. 2021-10-07]. Dostupné z: <https://www.printmag.com/information-design/top-five-data-visualization-artists-to-follow-on-instagram/>
- GUPTA, Ayush. *Using The Right File Format For Storing Data. Analytics Vidhya* [online]. 2021 [cit. 2022-08-17]. Dostupné z: <https://www.analyticsvidhya.com/blog/2021/09/using-the-right-file-format-for-storing-data/>

- HERGAARDEN, Mike. *Graphics shaders*. VU Amsterdam, 2011, s.2. Files.m2h [online]. 2021 [cit. 2022-08-17]. Dostupné z: <http://files.m2h.nl/LiteraturestudyShaders.pdf>
- HILL, Jon. *Data vs Information: What's the Difference?* Bloomfire [online]. 2021 [cit. 2022-08-17]. Dostupné z: <https://bloomfire.com/blog/data-vs-information/>
- *History of Computers. Department of Computer Science and Statistics College of Arts and Sciences* [online]. [cit. 2021-11-07]. Dostupné z: <https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading03.htm>
- *Hommage à Paul Klee, 13/9/65 Nr.2. Victoria and Albert Museum* [online]. 2009 [cit. 2022-02-07]. Dostupné z: <https://collections.vam.ac.uk/item/O211685/hommage-a-paul-klee-13965-print-nake-frieder/hommage-%C3%A0-paul-klee-13965-print-nake-frieder/>
- *HTML ASCII Reference. W3schools* [online]. [cit. 2022-08-17]. Dostupné z: https://www.w3schools.com/charsets/ref_html_ascii.asp
- HÜBNER, Patrik. *Brute. Patrik-Hübner* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.patrik-huebner.com/work/brute-data-driven-wine-brand/>
- HÜBNER, Patrik. *Method. Patrik-Hübner* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.patrik-huebner.com/method/>
- CHAHIL. *Wien 1985: phänomen fax-art. Beuys, warhol und higashiyama setzen dem kalten krieg ein zeichen. Chahil art consulting* [online]. 13.10.2015 [cit. 2021-10-20]. Dostupné z: <https://andrechahil.com/wien-1985-phaenomen-fax-art-beuys-warhol-und-higashiyama-setzen-dem-kalten-krieg-ein-zeichen/>
- *Character Encoding. Tech Terms* [online]. 2010 [cit. 2022-08-17]. Dostupné z: <https://techterms.com/definition/characterencoding>
- *Internet Art. Technopedia* [online]. 7.01.2017 [cit. 2021-11-17]. Dostupné z: <https://www.techopedia.com/definition/25603/internet-art>
- *JavaScript – Overview. Tutorials point* [online]. [cit. 2022-02-07]. Dostupné z: https://www.tutorialspoint.com/javascript/javascript_overview.htm

- *JavaScript Tutorial. W3Schools* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.w3schools.com/js/default.asp>
- JONES, Vinciane. *John Maeda: A pioneer of digital art. Verisant* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://medium.com/verisant/john-maeda-a-pioneer-of-digital-art-5d8a43e0c16d>
- KOUBEK, Jochen. *Creative Coding: Zur Lage der Kulturtechnik Programmieren* [online]. 2010 [cit. 2022-03-07]. Dostupné z: <https://creativecoding.uni-bayreuth.de/index.html>
- LEGRADY, George. *George Legrady studio* [online]. [cit. 2022-02-07]. Dostupné z: <http://www.georgelegrady.com/>
- *Machine Hallucinations – Space: Metaverse. Sothebys* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://www.sothebys.com/en/buy/auction/2021/machine-hallucinations-space-metaverse/machine-hallucinations-space-metaverse>
- MANDIS, George. *JavaScript for Artists: How to make creative, interactive art projects in JavaScript. JavaScript for Artists* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://javascriptforartists.com/>
- *Marpi Studio: About. Marpi Studio* [online]. 2022 [cit. 2022-02-07]. Dostupné z: <https://www.marpi.studio/about>
- MODRZYK, Nicolas. *Learning how to code using JavaScript and p5.js. Apress* [online]. 2018 [cit. 2022-02-07]. Dostupné z: <https://www.apress.com/la/blog/all-blog-posts/javascript-and-p5-js/15548356>
- *New nature. Artechouse* [online]. 2019 [cit. 2022-02-07]. Dostupné z: <https://www.artechouse.com/program/new-nature/>
- NG, Andrew. *Machine learning. Mlclass* [online]. [cit. 2022-02-07]. Dostupné z: <http://mlclass.stanford.edu/>
- NORMAN, Jeremy M. *Edward Zajac Produces the First Computer-Animated Film. History of Information* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.historyofinformation.com/detail.php?entryid=1002>
- O'BRIEN, Matt a Kelvin CHAN. *EXPLAINER: What is the metaverse and how will it work? ABC News* [online]. Associated Press, 2021 [cit.

2022-02-07]. Dostupné z:

<https://abcnews.go.com/Business/wireStory/explainer-metaverse-work-80842516>

- *Overview: A short introduction to the Processing software and projects from the community. Processing* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://processing.org/overview>
- *P5js. P5.js* [online]. [cit. 2022-03-11]. Dostupné z: <https://p5js.org/>
- PRASANNA. *Advantages and Disadvantages of Python | Python Language Advantages, Disadvantages and Its Applications. AplusTopper* [online]. 2022 [cit. 2022-09-14]. Dostupné z: <https://www.aplustopper.com/advantages-and-disadvantages-of-python/>
- *Press release announcing JavaScript: Netscape and Sun announce JavaScript. Netscape* [online]. PR Newswire, 1994 [cit. 2022-05-07]. Dostupné z: <https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>
- REAS, Casey. *Biography. Bitforms gallery* [online]. [cit. 2022-02-07]. Dostupné z: http://www.bitforms.com/pdfs/biocv/reas_bio.pdf
- REAS, Casey. *Path P. Reas* [online]. [cit. 2022-02-07]. Dostupné z: https://reas.com/path_p/
- REAS, Casey. *The MicroImage A prints. Reas* [online]. [cit. 2022-02-07]. Dostupné z: https://reas.com/microimage_a_p/
- ROHRER, Brandon. *How to Convert a Picture to Numbers. E2eml* [online]. 2019 [cit. 2022-08-17]. Dostupné z: https://e2eml.school/images_to_numbers.html
- *Roman Verostko and the Cloud of Unknowing: From Ideas in Mind to Ideas in Code. Verostko Center* [online]. 2020 [cit. 2022-02-07]. Dostupné z: <http://www.verostkocenter.org/introduction---the-cloud-of-unknowing.html>
- SARTHAK, Ishu. *P5.js Introduction. Geeks for geeks* [online]. 2021 [cit. 2022-03-11]. Dostupné z: <https://www.geeksforgeeks.org/p5-js-introduction/>
- *Say it with software art! Runme* [online]. [cit. 2021-05-07]. Dostupné z: <http://runme.org>

- SHERMAN, Chelley. *Dispersion*. *Vimeo* [online]. 2018 [cit. 2022-06-19]. Dostupné z: <https://vimeo.com/280815065>
- SHERMAN, Chelley. *Chelley Sherman: About. Space is the place* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.spaceisthe.place/about>
- SCHECHTER, Sonia. *Everything You Need to Know About Using OBJ Files. 3D Cloud Marxent* [online]. 2021 [cit. 2022-08-17]. Dostupné z: <https://www.marxentlabs.com/obj-files/>
- SIMKOVIC, Michael. *Creating Art with Code*. *Hacker Rank* [online]. 2019 [cit. 2022-05-07]. Dostupné z: <https://www.hackerrank.com/blog/creating-art-with-code/>
- SUN, Alida. *Chromacounterpane #2: pandau*. *Makersplace* [online]. 2022 [cit. 2022-02-07]. Dostupné z: <https://makersplace.com/0x5fcd30a446d465ebb0ff8db85feb19fe98d758f5/chromacounterpane-2-pandau-1-of-1-79092/>
- TAVARES, Gregg. *WebGL Fundamentals*. *Html5 rocks* [online]. 2012 [cit. 2022-02-07]. Dostupné z: https://www.html5rocks.com/en/tutorials/webgl/webgl_fundamentals/
- *Tech 21 century: The Human Brain is Loaded Daily with 34 GB of Information* [online]. [cit. 2020-05-01]. Available from: <https://www.tech21century.com/the-human-brain-is-loaded-daily-with-34-gb-of-information/>
- *The aesthetics + computation group*. *ACG media* [online]. [cit. 2022-02-07]. Dostupné z: <https://acg.media.mit.edu/>
- *The one show: Patrik Hübner. The one club for creativity* [online]. 2020 [cit. 2022-02-07]. Dostupné z: <https://www.oneclub.org/awards/theoneshow/-judge/3844/patrik-hbner>
- *Unicode® 14.0.0*. *Unicode* [online]. 2021 [cit. 2022-08-17]. Dostupné z: <https://www.unicode.org/versions/Unicode14.0.0/>
- VEROSTKO, Roman. *THE ALGORISTS*. *Verostko* [online]. [cit. 2022-05-07]. Dostupné z: <http://www.verostko.com/algorist.html>
- WALTON, Marsha. *IBM PC turns 25*. *CNN International* [online]. 2007 [cit. 2022-02-07]. Dostupné z: <http://edition.cnn.com/2006/TECH/biztech/08/11/ibmpcanniversary/>

- *WebGL Tutorial. Tutorials point* [online]. [cit. 2022-03-11]. Dostupné z: <https://www.tutorialspoint.com/webgl/index.htm>
- *What are Fractals? Fractal Foundation* [online]. 2018 [cit. 2021-10-07]. Dostupné z: <https://fractalfoundation.org/resources/what-are-fractals/>
- *What is blockchain? Euromoney learning* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.euromoney.com/learning/blockchain-explained/what-is-blockchain>
- *What is JavaScript? Hack Reactor* [online]. 2021 [cit. 2022-02-07]. Dostupné z: <https://www.hackreactor.com/blog/what-is-javascript-used-for>
- *What is projection mapping? Projection Mapping Central* [online]. [cit. 2022-02-07]. Dostupné z: <https://projection-mapping.org/what-is-projection-mapping/>
- WU, Jun. *Getting Started With Creative Coding: When art meets programming and generates design. Better Programming* [online]. 04.01.2020 [cit. 2021-10-22]. Dostupné z: <https://betterprogramming.pub/getting-started-with-creative-coding-16072ff7e778>
- *Xerox Art. Calhoun Press* [online]. 06.10.2015 [cit. 2021-10-19]. Dostupné z: <https://calhounpress.net/blogs/blog/48053955-xerox-art>
- YADUWANSHI, Sukriti. *Difference between coding and programming. Techgig* [online]. [cit. 2022-03-01]. Dostupné z: <https://content.techgig.com/techno-digest/difference-between-coding-and-programming/articleshow/81568031.cms>
- ZELLMER, Sean. *Sean Zellmer: Info. Le Jeune Renard* [online]. [cit. 2022-02-07]. Dostupné z: <https://lejeunerenard.com/info>
- ZELLMER, Sean. *Sean Zellmer: LinkedIn. LinkedIn* [online]. [cit. 2022-02-07]. Dostupné z: <https://www.linkedin.com/in/seanzellmer/>
- ZHAI, Sean. *Creative Coding: Perspectives & Case Studies: How to choose creative coding tools. JavaScript in Plain English* [online]. 30.09.2020 [cit. 2022-02-07]. Dostupné z: <https://javascript.plainenglish.io/all-about-creative-coding-e79268d944e8>

SEZNAM OBRÁZKŮ

Obrázek 1: kategorie počítačového umění. Zdroj: vlastní.

Obrázek 2: pohled na prostor strojovny IBM 7094. Zdroj: Columbia University Archive. <http://www.columbia.edu/cu/computinghistory/1965.html>

Obrázek 3: vlevo - Čtyři počítačem generované náhodné vzory založené na kompozičních kritériích Mondrianova "Composition With Lines". Zdroj: https://dam.org/archive/noll/artworks_04.htm

Obrázek 4: vpravo - Devadesát počítačově generovaných sinusoid s lineárně rostoucí periodou. Zdroj: <https://digitalartarchive.siggraph.org/artwork/a-michael-noll-ninety-computer-generated-sinusoids-with-linearly-increasing-period/>

Obrázek 5: A Two Gyro Gravity Gradient attitude control System. Zdroj: https://youtu.be/GBIQb6Me_1k

Obrázek 6: Hommage à Paul Klee, F. Nake. Zdroj: <https://collections.vam.ac.uk/item/O211685/hommage-a-paul-klee-13965-print-nake-frieder/hommage-%C3%A0-paul-klee-13965-print-nake-frieder/>

Obrázek 7: vlevo – Hobby box. Zdroj: <http://dada.compart-bremen.de/item/artwork/1132>

Obrázek 8: vpravo – Hobby box 2. Zdroj: https://monoskop.org/Compos_68

Obrázek 9: Legrady: Anamorph-Visegrad (2020), 6 černobílých fotografií je komponováno na základě anamorfní perspektivy a Voronoiovy mozaiky. Zdroj: <http://www.georgelegrady.com/>

Obrázek 10: John Maeda, Florada. Zdroj: <https://medium.com/verisart/john-maeda-a-pioneer-of-digital-art-5d8a43e0c16d>

Obrázek 11: Ben Fry: Genome Valence. Zdroj: <https://benfry.com/genomevalence/>

Obrázek 12: Casey Reas: The Path. Zdroj: https://reas.com/path_p/

Obrázek 13: Casey Reas: MicroImage A. Zdroj: https://reas.com/microimage_a_p/

Obrázek 14: Patrik Hübner – BRUTE. Zdroj: <https://www.patrik-huebner.com/work/brute-data-driven-wine-brand/>

Obrázek 15: Chelley Sherman - Dispersion. Zdroj: <https://vimeo.com/280815065>

Obrázek 16 Sean Zellmer. Zdroj: <https://www.fffutures.com/p/forked-and-merged-memories?s=r>

Obrázek 17: Alida Sun - Chromacounterpane #2: pandau. Zdroj: <https://makersplace.com/0x5fcd30a446d465ebb0ff8db85feb19fe98d758f5/chromacounterpane-2-pandau-1-of-1-79092/>

Obrázek 18: Marpi Studio - New Nature Digital Petting Zoo. Zdroj: <https://www.artehouse.com/program/new-nature/>

Obrázek 19: Refik Anadol - Machine Hallucinations – Space : Metaverse. Zdroj: <https://refikanadol.com/works/machine-hallucinations-space-metaverse/?i=d>

Obrázek 20: Refik Anadol - Machine Hallucinations – Space : Metaverse. Zdroj: <https://refikanadol.com/works/machine-hallucinations-space-metaverse/?i=d>

Obrázek 21: Processing byl ovlivněn jinými kódovacími systémy a ovlivnil další. Zdroj: <https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85>

Obrázek 22: Excel datový soubor z Centra polymerních systémů UTB

Obrázek 23: CSV databáze "Les Miserables" používaná aplikací Gephi jako základní příklad

Obrázek 24: Část souboru .obj z University of Utah School of Computing. Zdroj: <https://www.marxentlabs.com/obj-files/>

Obrázek 25: Naprogramovaný kruh. Zdroj: vlastní

Obrázek 26: Zkreslený kruh. Zdroj: vlastní

Obrázek 27: Záplava kruhů. Zdroj: vlastní

Obrázek 28: Záplava červených kruhů. Zdroj: vlastní

Obrázek 29: Vizualizace dat jako výzva estetiky 1, Zdroj: vlastní

Obrázek 30: Galerie Hauerova 4, Opava. Fotograf: Jan Vlček

Obrázek 31: Vizualizace dat jako výzva estetiky 1, Zdroj: vlastní

Obrázek 32: Vizualizace dat jako výzva estetiky 2, Zdroj: vlastní

Obrázek 33: Galerie Hauerova 4, Opava. Fotograf: Jan Vlček

Obrázek 34: In my heart. Zdroj: vlastní

Obrázek 35: See the wind - blue. Zdroj: vlastní

Obrázek 36: See the wind - yellow. Zdroj: vlastní

Obrázek 37: See the wind – Zlín. Zdroj: vlastní

Obrázek 38: Fraktál 1. Zdroj: vlastní

Obrázek 39: Fraktál 2. Zdroj: vlastní

Obrázek 40: Sympozium Art Colony Cered: Zvukový deník. Zdroj: vlastní

Obrázek 41: Sympozium Art Colony Cered: Zvukový deník. Zdroj: vlastní

Obrázek 42: Under 1. Zdroj: vlastní
Obrázek 43: Under 2. Zdroj: vlastní
Obrázek 44: Částice CREDO. Zdroj: vlastní
Obrázek 45: Superčástice CREDO. Zdroj: vlastní
Obrázek 46: Superčástice CREDO. Zdroj: vlastní
Obrázek 47: Vizualizace dat 3D pánve, kyčlí. Zdroj: vlastní
Obrázek 48: The stars of Cered. Zdroj: vlastní
Obrázek 49: The stars of Cered. Zdroj: vlastní
Obrázek 50: Výstava Undercurrent, foto: Bc. Eliška Janáková
Obrázek 51: Výstava Undercurrent, foto: Bc. Eliška Janáková
Obrázek 52: Výstava Undercurrent. Zdroj: vlastní
Obrázek 53: Web - Data Art. Zdroj: vlastní
Obrázek 54: Obrázek 22: Web - Data Art. Zdroj: vlastní
Obrázek 55: Web - Data Art. Zdroj: vlastní

SEZNAM POUŽITÝCH ZKRATEK

- VJ – visual/video jockey – vizuální/videožokej;
- MIT – Massachusetts Institute of Technology – Massachusettský technologický institut;
- JS – JavaScript – skriptovací jazyk;
- HTML – Hypertext Markup Language – značkovací jazyk používaný pro vytváření webových stránek;
- HTML5 – 5. verze značkovacího jazyka;
- CSS – Cascading Style Sheets – tabulky kaskádových stylů;
- VR – Virtuální realita;
- MIDI – Musical Instrument Digital Interface – digitální rozhraní hudebního nástroje;
- LED – Light Emitting Diode – elektroluminiscenční dioda;
- GPU – graphics processing unit – jednotka grafického zpracování;
- RAM – Random Access Memory – paměť s náhodným přístupem;
- CAD – computer-aided design – počítačem podporované projektování;
- EXE – executable – formát spustitelného souboru.

TVŮRČÍ ČINNOST 2017–2022

Projekty:

- *BladeClub, logo, leták* – Návrh loga a letáku pro Blade Club Zlín.
- *Interaktivní banner "Light" (světlo)*. Zákazník – Lucerna TV, internetová vzdělávací televize, Slezská univerzita v Opavě.
- *Vizualizace dat jako výzva estetiky*. Projekt představuje vizualizaci velkoobjemových dat jako nový trend ve vizuálním umění, ve spolupráci s Centrem polymerních systémů UTB.
- *Data art – web, design, logo*. Návrh, design, kódování a programování webového portálu "Data art", zaměřeného na zveřejňování uměleckých děl, programů, videí, vizualizací vytvořených technikami kreativního kódování.
- *Design postavy, ilustrace*. Design postav a ilustrace, které patří do světa online hry – The Elder Scrolls.
- *Videoportál Lucerna TV*. Kódování a programování webového portálu.
- *In my heart – vizualizace dat*. Pokračování projektu vizualizace dat ve spolupráci s Centrem polymerních systémů UTB.
- *Propagační materiál L410*. Návrh propagačních materiálů pro Aircraft industries a.s., který zahrnuje: plakát, samolepky, design tašek.
- *NR - 32 ms*. Umělecký projekt, digitální tisk, 2021. Formát: A4. Technika: počítačová grafika. Grafika představuje vizualizaci dat shromážděných v Centru polymerních systémů UTB.
- *NR - 38 ms*. Formát: A1. Technika: počítačová grafika. Grafika představuje vizualizaci dat shromážděných v Centru polymerních systémů UTB.
- *Under*. Vizualizace zvukových vln.
- *Bright matter, Superčástice*. Vizualizace dat z mezinárodního projektu CREDO.
- *Pánev, kyčle*. Vizualizace dat 3D pánve, kyčlí.

Simpozia a festivaly:

- II. ročník Minifestivalu multimediální scénické tvorby, Napajedla, 18.02.2017. Představení "Imprint" - práce s vizuálními a světelnými efekty, spolupráce se studenty JAMU;

- The International Festival of Theatre Schools. SETKÁNÍ/ENKOUNTER 2017. 29-31.03.2017. *Představení "Imprint" - práce s vizuálními, zvukovými a světelnými efekty, spolupráce se studenty JAMU, dostupné z Youtube.*
- Sympozion art colony cered 2020. *Vizualizace zvukového deníku, která slouží jako pozvánka pro další ročník a zároveň tematická výzva výtvarného sympozia Art Colony v maďarském Ceredu.*
- Sympozion art colony cered 2021. *See The Wind.* Projekt vizualizace větrných proudů.
- Sympozion PANTA RHEI 2021: projekt *Everything flows, nothing remains.*
- Sympozion art colony cered 2022. *The stars of Cered.* Vizualizace hvězdných dat pro maďarský Cered, 2022.

Umělecké výstavy

- *Autorská výstava "See the wind" 2021, Galerie Hauerova 4, Opava, Česká republika.*
- *Grafika roku 2020, Nadace Hollar, online.*
- *Grafika roku 2021, Nadace Hollar, online.*
- *Autorská výstava "Undercurrent" 2022, Galerie G18, Zlín, Česká republika.*

Publikace

- Palagina, E., Laurova A. (2012). *Používání webových technologií v oblasti tvorby a praktické aplikace interaktivního textu v matematice.* Webové programování a internetové technologie Webconf2012: materiály 2. Intern. vědecko-praktické. Conf., 5-7. června 2012, Minsk: Izd. Centrum BSU, 2012. - str. 50-51.
- Laurova, A. (2014). *Online katalog knih.* Diplomová práce. EBU BSU:: SOCIÁLNÍ VĚDY:: Informatika, 2014. BĚLORUSKÁ STÁTNÍ UNIVERZITA, Minsk., <http://elib.bsu.by/handle/123456789/100021>
- Laurova, A. (2018). *Interaktivní instalace.* Diplomová práce. Univerzita Tomáše Bati ve Zlíně (UTB), 2018.
- Laurova, A. ART FOR ART'S SAKE, DIGITAL FOR GOD'S SAKE In S. Gawroński, M. Szewczyk, Ł. Bis (Eds.), *Social and marketing communication in education and didactics*, WSiLiZ, Rzeszów 2022.

CURRICULUM VITAE

MgA. Aliaksandra Laurova

Státní příslušnost: Bělorusko

Datum narození: 06.06.1991

Telefon: (+420)774586251

Adresa: Březnická 5565, 76001 Zlín

E-mail: antrage.antrage@gmail.com

Pracovní zkušenosti:

Web designer

firma Pras.by

30.11.2013 – 30.05.2014

design a rozvoj webových stránek, grafický a komunikační design.

Web designer

Freelance

19.10.2010 – 31.05.2015

design a tvorba webových stránek, webové programování, design loga a vizitek.

Ilustrátor

Freelance

31.05.2014 – Aktuální

více než 500 ilustrací pro zákazníky, návrh postav pro komiksy, ilustrace příběhů.

Přednášející v oboru Interakční design

Univerzita Tomáše Bati ve Zlíně

03.02.2019 – 03.05.2019

Zlín, Česko

Předmět: Interakční design

Přednášející v oboru Visual communications

Univerzita Tomáše Bati ve Zlíně

24.09.2018 – 19.05.2020

Zlín, Česko

Předmět: Kreativní kódování s Processingem

Přednášející v oboru Online marketing

VOŠ Mediální tvorby

31.08.2021 – Aktuální

Ostrava, Česko

Předměty: Online marketing, Digitální marketing

Přednášející v oboru Webmarketing

Panevropská Univerzita

14.1.2021 – Aktuálně

Předměty: Webový marketing, Marketing sociálních médií, Webová analytika, Marketing online vyhledávání.

Vzdělání a odborná příprava

Běloruská Státní Univerzita

Programátor, web designer – vysokoškolské studium (bakalářské + magisterské)

31.08.2009 – 29.06.2014

Fakulta humanitních studií

Obor: Informatyka

Ateliér: Webdesign, počítačová grafika

VUT, Institut celoživotního vzdělávání

Jazykový kurz - Čeština pro cizince

30.09.2014 – 29.06.2015

Univerzita Tomáše Bati ve Zlíně

Magisterské studium – Digitální design

15.07.2015 – 12.06.2018

Fakulta multimediálních komunikací

Obor: Multimédia a design

Ateliér: Digitální design

Univerzita Tomáše Bati ve Zlíně

Chinese Language

30.09.2016 – 25.05.2017

Fakulta Humanitních studií

Public course of Chinese Language and Culture, pre-intermediate level.

Univerzita Tomáše Bati ve Zlíně

Doktorské studium – Výtvarná umění – Multimédia a design

15.07.2018 – Aktuální

Studijní obory: Výtvarná umění – Multimédia a design

Konference a kongresy:

- Mezinárodní vědecká a praktická konference WebConf-2012 (Vývoj webových aplikací a internetových technologií).
- Kongres PTKS. Media i społeczeństwo w erze platform, algorytmów i danych. Gdańsk, Polsko, 2022.

MgA. Aliaksandra Laurova

Kreativní kódování a generativní algoritmy jako nástroje pro tvorbu interaktivního vizuálního umění

Creative coding and generative algorithms as tools for creating interactive visual art

Disertační práce

Vydala Univerzita Tomáše Bati ve Zlíně,
nám. T. G. Masaryka 5555, 760 01 Zlín.

Sazba: MgA. Aliaksandra Laurova

Publikace neprošla jazykovou ani redakční úpravou.

Rok vydání 2022