

Sem vložte zadání Vaší práce.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

Diplomová práce

## **Metaviz: Průvodce kreativním programováním**

*Bc. Radka Hošková*

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Radek Richtr, Ph.D.

17. dubna 2022



# **Poděkování**

Mě díky Ing. Radku Richtrovi, PhD. za vedení a pomoc při psaní diplomové práce. Díky patří i všem, kteří se mnou práci diskutovali, podporovali mě, nebo už jen byli inspirací.



# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen z části) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. dubna 2022

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií  
© 2022 Radka Hošková. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Hošková, Radka. *Metaviz: Průvodce kreativním programováním*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

# Abstrakt

Cílem je mapování a popularizace kreativního programování, vizualizací a digitálního umění. V rámci práce vznikla webová aplikace Metaviz - digitální výstava dílčích implementovaných vizualizací.

**Klíčová slova** generative art, digital art, creative coding, creative programming, new media art, creative computing, digital morphogenesis

# Abstract

**Keywords** generative art, digital art, creative coding, creative programming, new media art, creative computing, digital morphogenesis



# Obsah

<b>Úvod</b>	<b>1</b>
Cíle práce . . . . .	2
Struktura . . . . .	2
<b>I Teoretická část</b>	<b>3</b>
<b>1 Kreativní programování</b>	<b>5</b>
1.1 Pojmy . . . . .	6
1.2 Vizualizace dat . . . . .	8
1.3 Vývoj umění nových médií . . . . .	8
1.4 Příklady algoritmů a technik . . . . .	11
<b>2 Instituce, komunita a výzvy</b>	<b>15</b>
2.1 SIGGRAPH . . . . .	15
2.2 Datové vizualizace, infografika . . . . .	16
2.3 Demoscene . . . . .	17
2.4 Psaní malého kódu . . . . .	18
2.5 Genuary, Nodevember . . . . .	19
2.6 Webové technologie . . . . .	19
2.7 The Future Sketches group . . . . .	20
2.8 Generativní script . . . . .	20
<b>3 Technologie pro kreativní programování</b>	<b>21</b>
3.1 Processing . . . . .	21
3.2 Technologie pro webový vývoj . . . . .	23
3.3 Vizuální programování . . . . .	24
3.4 Herní enginy . . . . .	25
3.5 Kreativní programování mimo digitální umění . . . . .	25
<b>4 Vybrané projekty</b>	<b>27</b>
4.1 Signal . . . . .	27
4.2 Expo . . . . .	27
4.3 Příklady atypických vizualizačních ploch v Praze . . . . .	28

4.4	Grafové vizualizace . . . . .	29
4.5	Electric Sheep . . . . .	30
<b>5</b>	<b>Výukové kanály a materiály</b>	<b>31</b>
5.1	Processing, p5.js, Three.js, webové technologie . . . . .	31
5.2	Unreal, Blender, Touchdesigner . . . . .	32
5.3	GLSL . . . . .	32
<b>6</b>	<b>Osobní portfolia</b>	<b>35</b>
6.1	Sage Jenson . . . . .	35
6.2	Robert Hodgin . . . . .	35
6.3	Webové a plotter projekty . . . . .	36
6.4	Herní vývoj . . . . .	37
<b>II</b>	<b>Praktická část</b>	<b>39</b>
<b>7</b>	<b>Analýza</b>	<b>41</b>
7.1	Rešerše podobných aplikací . . . . .	41
7.2	Požadavky na aplikaci . . . . .	42
<b>8</b>	<b>Návrh</b>	<b>45</b>
8.1	Doménový model . . . . .	45
8.2	Struktura a architektura . . . . .	45
8.3	Technologie . . . . .	46
8.4	Uživatelské rozhraní . . . . .	46
<b>9</b>	<b>Implementace</b>	<b>47</b>
9.1	Vizualizace 1 – periodicita videa . . . . .	47
9.2	Vizualizace 2 - Crown Shyness . . . . .	50
9.3	Vizualizace 3 - Kensuke Koike . . . . .	51
9.4	Vizualizace 4 - Convex hull . . . . .	51
<b>10</b>	<b>Testování a reflexe nad projektem</b>	<b>53</b>
10.1	Persony . . . . .	53
10.2	Scénáře testování . . . . .	53
10.3	Výsledky testování . . . . .	53
10.4	Aplikace kreativního programování . . . . .	53
10.5	Pokračování projektu . . . . .	53
<b>Závěr</b>		<b>55</b>
<b>Literatura</b>		<b>57</b>
<b>A Seznam použitých zkratek</b>		<b>61</b>





# Seznam obrázků

1.1	Porovnání vyhledávání výrazů <i>creative coding</i> a <i>creative programming</i>	6
1.3	Časová osa	10
1.4	3 oktávy Perlinova šumu (1-dimenziální (1D))	11
1.5	Perlinův šum v 2-dimenziální (2D)	11
1.6	Brownův pohyb, cesta jedné částice ve 3-dimenziální (3D)	11
1.7	Vizualizace 3D dat	12
1.8	Vápenatka mnohohlavá	12
1.9	Conway's Game of Life	12
2.1	Vizualizace <i>Array of chessboards</i> s velikostí 16 bytes	17
2.2	Příklad つぶやき processingu od uživatele yuruyurau	18
4.1	Signal festival 2021	27
4.2	Atypické vizualizační plochy	28
4.3	Příklad výstupu z projektu <i>Seealsology</i>	29
5.1	3D grafika vytvořena pouze pomocí matematiky	33
6.1	Fidenza	37
6.2	DLA	37
6.3	incovergent	37
9.1	Crown shyness	50



# Úvod

Teoretický fyzik Richard P. Feynman popisuje v knize Radost z poznání[1] rozepře se svým kamarádem umělcem o kráse květiny. Ohrazuje se proti umělcově tvrzení, že by vědci svou analýzou ubírali květině na kráse a naopak tvrdí, že studií těchto vnitřních procesů ji na kráse pouze přidávají.

Tuto úvalu si dovoluji půjčit a rozvést jí dále v kontextu dnešních možností. Podobně jako se v kresbě využívají znalosti anatomie lidí či zvířat pro věrnější zachycení pózy, může počítačová grafika usilovat o co nejvěrnější zachycení přírodních jevů studií jejich fungováním. Dnes je možné nejen ocenit vizuální podobnu květiny a její procesy na různých úrovních, ale dokonce tyto procesy do jisté míry simulovat, alternovat, či vymýšlet a vytvářet vizualizace s vlastními zákonitostmi.

Právě kombinace umění a technologie se potkává v digitálním umění, computer artu, kreativním programování. Technologie do umění přináší rozsáhlé možnosti parametrizace a interaktivnosti.

// todo:

tato práce se kromě využití v umění zaobírá praktickým využitím vizualizací

## Cíle práce

Přestože je využití vizualizací často umělecké, není pravidlem, že by umělec a programátor musel být stejný člověk. Cílem práce je strukturalizovat dostupné informace o kreativním programování a zjednodušit tak programátorem i umělcům cestu pro ztvárnění jejich myšlenky. Cílem práce je i vytvořit vlastní set vizualizací využívající znalosti z rešeršní práce a vhodně je prezentovat.

Cílem práce není prezentovat kreativní programování jako něco převratného a přínosného pro každého. Odvětví jakými jsou informatika, počítačová grafika, umění a design by se pravděpodobně bez pojmu kreativní programování obešly. Pojem má za cíl spojit programátory a umělce k vzájemné kolaboraci, případně probouzet umělce v programátora a programátora v umělci.

## Struktura

Diplomová práce je rozdělena na teoretickou a praktickou část. Předmětem teoretické části je rešerše digitálního umění z programátorského hlediska. Nachází se zde popis toho, co to je digitální umění a jaké další pojmy se v této oblasti vyskytují. Mezi ně patří generativní umění a umění nových médií. Následuje část o kreativním programování, která se skládá z rešerší dostupných technologií, analýzou komunity, výukových materiálů a projektů.

V praktické části je poté návrh vlastních vizualizací, kde je každé dílčí vizualizaci věnována samostatná kapitola pro popis návrhu a implementace. Následuje kapitola návrhu a implementace aplikace *Metaviz*<sup>1</sup>. Další kapitoly se věnují testování, debatou nad projektem a jeho možným pokračováním.

---

<sup>1</sup>Název *Metaviz* nesouvisí se společností *Meta*. Aplikace dostala svůj název podle toho, že vizualizuje několik vizualizací, čili od meta vizualizování vznikl název *Metaviz*.

# Část I

## Teoretická část



# KAPITOLA

# 1

## Kreativní programování

Kreativní programování je typ počítačového programování. Vyznačuje se tím, že jeho cílem je spíše než vytvoření něčeho praktického pro zjednodušení lidské činnosti, vytvoření něčeho expresivního či uměleckého. Nejedná se o vědní obor a nemá proto exaktní definici<sup>2</sup>. Kreativní kódování nezahrnuje pouze vizualizace, ale například produkování hudby, VJing, *videomapping*, *live coding*, či různé interaktivní projekty.

Kreativní programování si neklade žádné obecné podmínky pro výstup, užitečnost pojmu spočívá především v agregaci a vzniku technologií či materiálů za společným účelem – poskytnout jednoduchý způsob jak zkoumat určitou myšlenku pomocí počítače, ideálně s odstíněním co nejvíce technických záležitostí, které v daný moment nejsou předmětem zkoumání.

### Kreativní programování vs. kódování

Vyskytují se termíny *kreativní kódování* i *kreativní programování*. V informatici je programování je proces od návrhu řešení problému pomocí výpočetní techniky ke spustitelnému počítačovému programu. Kódování je jednou z činností, který tento proces zahrnuje, tedy zápis zdrojového kódu v cílovém programovacím jazyce.

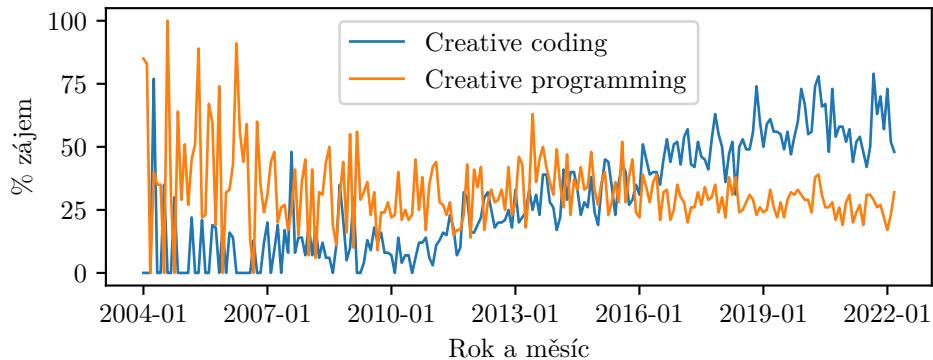
Celosvětové zastoupení těchto pojmu ve vyhledávání přes *Google.com* zobrazuje graf 1.1, kde osa *y* představuje počet vyhledávání vzhledem k nejvyššímu počtu vyhledávání, který je označen jako 100% (nejvyšší zájem byl v srpnu roku 2004 u kreativního programování). Přestože z grafu vyplývá *kreativní kódování* jako termín oproti *kreativnímu programování* na vzestupu, byl vybrán název programování pro jeho význam, tedy pro poskytnutí vyšší obecnosti. Data byla převzata z *Google Trends*.

---

<sup>2</sup>Nebo alespoň nebyla autorkou práce nalezena. Za vědní disciplínu se dá považovat *Creative Computing*.

## 1. KREATIVNÍ PROGRAMOVÁNÍ

---



Obrázek 1.1: Porovnání vyhledávání výrazů *creative coding* a *creative programming*

### 1.1 Pojmy

Paralelně se při spojení domén umění a technologií vyskytuje několik pojmu. Významově se tyto výrazy mohou částmi překrývat a existují mezi nimi nuance. Byly zkoumány často se objevující se pojmy i méně známé. Snahou této sekce bylo nalézt definice pro účely lepší orientace tématem.

#### Digital art

*Digital art* – digitální umění – je obecný termín pro umělecká díla, která využívají digitální technologie. Ty mohou být použity k jeho vytvoření nebo prezentaci. V dnešní době není nutné vytvářet vlastní nástroje, nýbrž umělec může používat mnoho programů pro vizuální umění i hudbu. Obecnějším pojmem do kterého patří digitální umění, je umění nových médií.

#### New media art

S vývojem technologií se mění kdysi tradičně používaná média. Například zpravodajství může místo papírových novin používat televize, pro ukládání dat je možné místo pevných disků používat *cloudová* uložiště a v umění je možné místo plátna používat digitální obrazovky. Lze se domnívat, že není snadné dosáhnout univerzální shody v otázce, co jsou to nová média.

Umění nových médií je velmi obecný pojem, popisující umění produkované elektronickými médií. Patří mezi ně například digitální umění, počítačová grafika, 3D tisk, robotika, videohry, interaktivní umění a další.

#### Computer art

*Computer art* je druh umění libovolně zahrnující počítače. Jak bylo zmíněno v knize [2], termín může konotovat éru průkopníků v této oblasti. Lze se domnívat, že na rozdíl od digitálního umění lze do computer artu zahrnout artefakty

vytvořené pomocí analogových počítačů a mechanických zařízení, jako například osciloskopu.

### Generativní umění

Termín generativní či procedurální umění bývá používán například v kontextu algoritmického komponování hudby, počítačové grafiky a průmyslového designu či architektury. Obecnou definici poskytuje Philip Galanter z roku 2003: „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 v pohyb a s určitou mírou autonomie přispívající nebo vedoucí k hotovému uměleckému dílu.“

### Digitální morfogeneze

Digitální morfogeneze<sup>3</sup> je typ generativního umění, ve kterém algoritmus umožnuje získání komplexního tvaru či morfogeneze. Morfogeneze značí biologický proces, který způsobuje, že buňka, tkáň nebo organismus vyvinou svůj tvar, avšak později se pojednává o rozvinutí do oblastí geologie, geomorfologie a architektury.

V kontextu digitálního umění by se pojednává o rozšíření na objevování tvarů, forem a vzorů pomocí výpočetního modelování a generativních systémů, založené na biologických, chemických a fyzikálních procesech. Je zde možné aplikovat výzkum za prakticky každého odvětví přírodních věd. Pro digitální morfogenezi existuje přehled<sup>4</sup> otevřený příspěvkům veřejnosti.

### Creative Computing

*Creative computing*<sup>5</sup>, volně přeloženo „kreativní výpočetní technika“ je mezinárodní disciplína, kde se střetává kreativní umění a výpočetní technologie. Obor je možné (pod různými názvy) studovat na soukromých i veřejných univerzitách, kde je možné získat i magisterský titul.

**Poznámka:** Existuje mnoho uměleckých směrů, jakými jsou například multimedia art, software art, net art, algorithmic art, jejichž rešerše je nad rámec této práce. Pojem umění nových médií byl shledán jako nejobecnější.

---

<sup>3</sup>Z řeckého *morphē* (tvar) a *genesis* (vytvorárení) – vytváření formy.

<sup>4</sup>Na adrese <https://github.com/jasonwebb/morphogenesis-resources>.

<sup>5</sup>Pojem nebyl přidat do grafu1.1, kvůli shodě v názvu se stejnojmenným časopisem vydávaným v letech 1974 až 1985.

## 1.2 Vizualizace dat

Velkou a podstatnou aplikací počítačové grafiky jsou datové vizualizace. Tyto mají význam ve vědě i pro širokou veřejnost. Jejich složitost se může různě pohybovat na škále od jednoduchých bodových grafů, po (interaktivní) vizualizace zahrnující komplexní znalosti a zkušenosti v oblastech informatiky, matematiky, designu i Human – Computer Interaction (HCI) (interakce člověka s počítačem).

### Fáze vizualizace dat

Ben Fry ve své dizertační práci[3] i knize[4] popisuje 7 kroků procesu tvorby vizualizace dat. Každá datová vizualizace má svoje specifika a tudíž může některé kroky absentovat.

- |   |   |                     |
|---|---|---------------------|
| 1. <i>Acquire</i> – získání dat               | } | informatika         |
| 2. <i>Parse</i> – zpracování dat              |   |                     |
| 3. <i>Filter</i> – filtrování dat             | } | matematika          |
| 4. <i>Mine</i> – statistika                   |   |                     |
| 5. <i>Represent</i> – základní reprezentace   | } | grafický design     |
| 6. <i>Refine</i> – čistší, hezčí reprezentace |   |                     |
| 7. <i>Interact</i> – interakce                |   | } infografika a HCI |

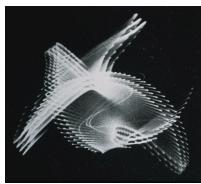
Jedná se o rozsáhlé téma. Těchto sedm kroků zahrnuje vědní disciplíny jakými jsou informatika, matematika a grafický design, případně infografika a HCI. Mezi přínosy vizualizací patří skutečnost, že lidský mozek je schopen přijímat vizuálním kanálem velký objem dat paralelně.

### Historie

Jako disciplína je vizualizace od roku 1987. Jsou známé příklady vizualizací před používáním počítačům například mapa epidemie cholery v Londýně, kde byly do mapy zaznamenávány oběti a díky tomu bylo možné vyvodit spojitost závadu vodovodního potrubí. Dalším příkladem je graf Napoleonova postupu na Moskvu.

## 1.3 Vývoj umění nových médií

Podobně jako jiné oblasti lidské činnosti, i umění prochází vývojem. Po cestě tohoto vývoje se mnozí snažili hledat hranice možností nově se objevujících technologií. Databází pro umění, média a humanitní vědy je stránka Monoskop[5]. Postupný vývoj byl popsán na mnoha místech, například [6], [7], [8],



(a) Ben Laponsky



(b) Vasulka

[9], [10], [11], [12], [13] či [14]. V této sekci se nachází pouze velmi rámcový průřez osobností, objevů a technologií, kteří byli předcházelé tématu kreativnímu programování.

## 19. století

Podle výkladu definice nových médií, která považuje nová média jako cokoliv nového po tradiční architektuře, malování a sochařství, by se dalo považovat za první technologii fotografii, která vznikla technikou heliografie využívající snímků z *camery obscury* roku 1826, jejímž autorem je Nicéphore Niépce. Další technologické pokroky jsou mimo rámcem této práce a nebudou zmíněny.<sup>6</sup>

## 20. století

Za autora první počítačové grafiky se považuje Ben F. Laponsky. Jeho použití osciloskopu pro umění z roku 1950 pro umění je obrázek 1.2a. Mezi další průkopníky patřila například Lillian Schwart a John Whitney, který v padesátých letech vytvářel počítačové animace díky mechanickému analogovému počítači, který vznikl úpravou protiletadlového systému z druhé světové války.

První výstavy *computer artu* byly spíše než umělcí pořádány vědci. Roku 1965 v New Yorkské Howard Wise galerii vystavovali Bela Julesz a A. Michael Noll, v Německu Georg Nees a Frieder Nake v *Galerie Niedlich*. První generace počítačových umělců pracující v *Bell Labs* a *Technische Universität Stuttgart* přispěla množstvím informací k základům počítačové grafiky.

K tvorbě nového umění kromě počítačů byly používány i kamery a televize. Mezi významné umělce *video artu* patří Woody Vasulka pocházející z Brna se ženou Steinou Vasulkou, kteří založili v New Yorku interdisciplinární prostor „The Kitchen“, kde se pořádali výstavy, promítání, koncerty. Prostor je aktivní i v roce 2022. O umění Vasulkových byly natočeny dokumenty a příklad jejich díla je vidět například na obrázku 1.2b.

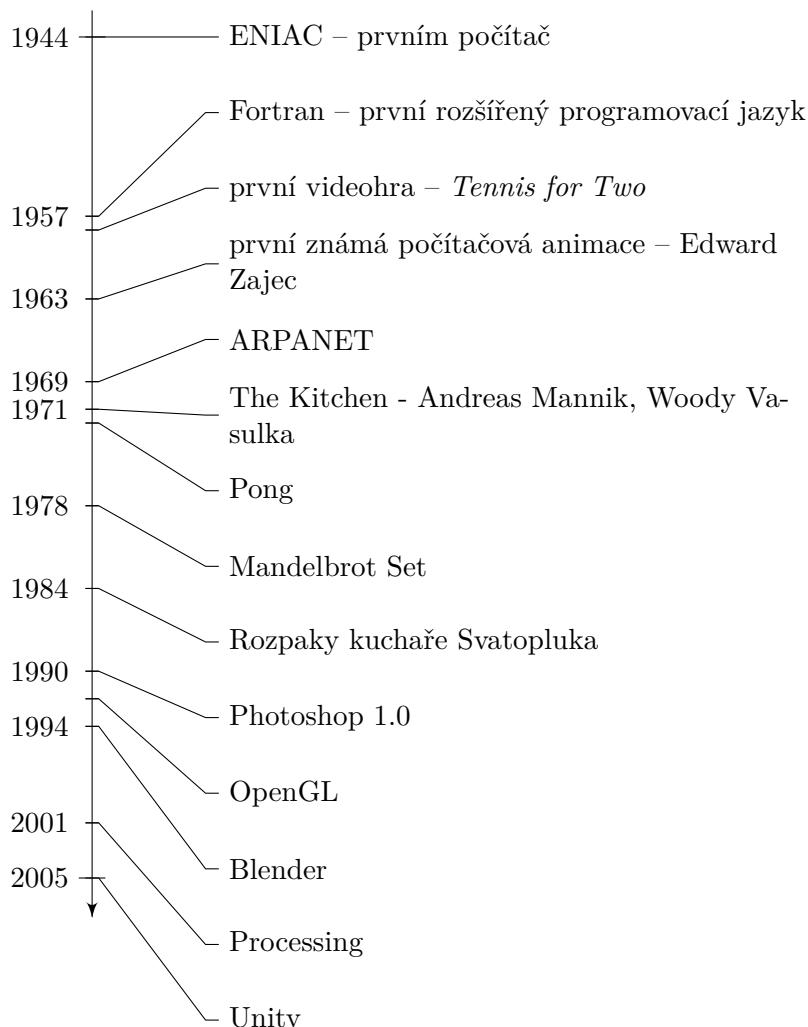
Jedny z prvních pokusů začlenit československé publikum učinil seriál Rozpaky kuchaře Svatopluka, kde občané pomocí rozsvěcování žárovek hlasovali o osudu hlavní postavy. Časová osa na obrázku 1.3 zobrazuje výběr z letopočtů vývoje technologií a událostí od roku 1944 do roku 2005.

---

<sup>6</sup>Některé z nich (jako je například mechanická televize) popisuje například kanál *Technology connections* na platformě *Youtube*.

## 1. KREATIVNÍ PROGRAMOVÁNÍ

---



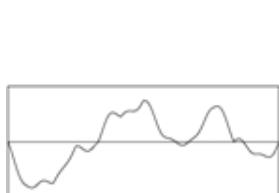
Obrázek 1.3: Časová osa

### 21. století

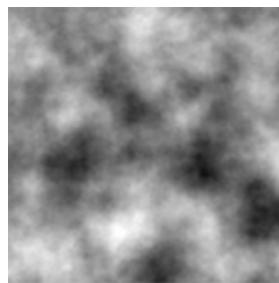
Jako autor prvního jazyku pro kreativní programování se označuje John Maeda z Massachusetts Institute of Technology (MIT) Media Lab, roku 1999. Následně jeho studenti Casey Reas a Ben Fry oznámili vytvoření Processingu. V současném umění nových médií existuje mnoho možností, jaké technologie použít z nichž některé jsou popsány v kapitole 3.<sup>7</sup>

---

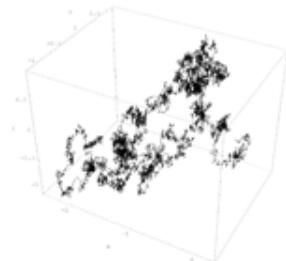
<sup>7</sup>O rozdílech práce v oblasti umění nových médií na přelomu 80. a 90. let a nyní ho-  
vorí například Lucie Svobodová – <https://artycok.tv/45434/vidis-tedy-jsem-lucie-svobodova>.



Obrázek 1.4: 3 oktaovy Perlinova sumu (1D).



Obrázek 1.5: Perlinuv šum v 2D.



Obrázek 1.6: Brownuv pohyb, cesta jedne partie ve 3D,

## 1.4 Příklady algoritmů a technik

Existují volně dostupné i komerční knihy a kurzy, týkající se tématice kreativního programování. Pod licencí CC BY-NC 3.0<sup>8</sup> je dostupná kniha[15] Daniela Shiffmana z roku 2012. Tato sekce jí bude inspirována.

V této sekci budou zmíněny příklady technik, využívaných převážně generativními umělci. Častým zdrojem inspirace jsou přírodní jevy a procesy, jako například chemické reakce a růst rostlin. Kromě inspirace v přírodě je možné inspirovat se tradičním uměním a umělci.

### Náhodnost

Základním a užitečným nástrojem je použití náhody pomocí procedurálního šumu. Jak bylo psáno v práci [16]: „šum je náhodný generátor čísel počítačové grafiky.“ Má široké spektrum použití od procedurálního texturování, přes simulace mraků a vln, po obecnou snahu rozbití symetrie a přidání náhodnosti. Výhodou je i typicky rychlý výpočet a kompaktnost (například oproti alternativnímu přístupu – přidání šumu pomocí obrázku).

Jedná se o funkci, typicky s výstupními hodnotami mezi 0 – 1. V závislosti na definičním oboru, čili vstupních parametrech, může být šum 1D, 2D, nebo 3D. Jedním ze známých algoritmů je Perlinův šum, původně vyvinutý pro potřeby filmu *Tron* začátkem osmdesátých let 19. století, jeho hodnoty lze promítnout do různých dimenzí, jak ukazují obrázky 1.4 a 1.5<sup>9</sup>

Místo přiřazování hodnot celému prostoru, lze zobrazovat například pohyb částic. Tomuto přírodnímu jevu se přezdívá Brownův pohyb (náznak pohybu je na obrázku 1.6<sup>10</sup>), matematicky ho popisuje Wienerův proces.

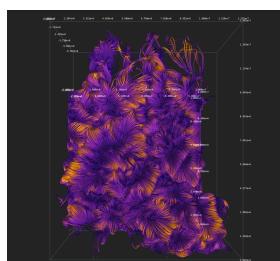
<sup>8</sup>Uvádění původu, nekomerční použití.

<sup>9</sup>Zdroj: <https://rtouti.github.io/graphics/perlin-noise-algorithm>.

<sup>10</sup>Zdroj: [https://en.wikipedia.org/wiki/Brownian\\_motion](https://en.wikipedia.org/wiki/Brownian_motion).

## 1. KREATIVNÍ PROGRAMOVÁNÍ

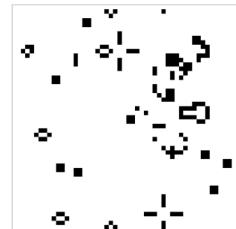
---



Obrázek 1.7:  
Vizualizace 3D dat.



Obrázek 1.8: Vápenatka  
mnohohlavá



Obrázek 1.9:  
Conway's Game  
of Life

### Reprezentace pohybu, vektorová pole

Jedním ze stavebních kamenů užitečným pro reprezentování reálných i imaginárních světů je euklidovský vektor s definovanou velikostí a směrem. Pomocí vektorů lze vyjádřit působení sil na objekty ve scéně v čase, simulovat částicové systémy a tak dále. Živé příklady minimalistického použití základních konceptů včetně potřebné matematiky lze prohlížet online v již zmíněné knize.<sup>11</sup> Možnostmi vizualizace 3D vektorových dat a vektorovými poli se věnovala práce [17], jejíž příklad je na obrázku 1.7.

### Autonomní agenti, chování a systémy

Pro určité jevy jakými jsou například termitiště a pohyb hejna ptáků, je výhodné zavést pojem *autonomní agent*, kladoucí tři požadavky:

1. autonomní agent má limitovanou schopnost vnímat okolí,
2. autohomní agent zpracovává informace z okolního prostředí a využívá svojí akciid další,
3. autonomní agent nemá centrálního vůdce.

Tvoření komplexních systémů pomocí emergence z jednoduchých lokálních pravidel se zabýval například Craig Reynolds, který roku 1986 vyvinul program *Boids*, simlující shlukování ptáků. Chování některých jevů je popsáno, například růst houby anglickým názvem *Physarum polycephalum* (na obrázku 1.8<sup>12</sup>) popisuje článek [18].

---

<sup>11</sup>Na adrese <https://natureofcode.com/book/>.

<sup>12</sup>Zdroj: <https://www.sciencealert.com/this-creeping-slime-is-changing-how-we-think-about-intelligence>.

## Celulární automat

Dalším způsobem modelování systémů je celulární automat (CA), které definovali Stanisław Ulam ve snaze zkoumat růst krystalů a John von Neumann, ve snaze o sebe-replikující roboty. CA se skládají z:

- pravidelné mřížky buněk,
- kde každá buňka má konečný počet stavů
- a pro každou buňku je definované okolí.

Příkladem CA je Hra života J. H. Conwaye (na obrázku 1.9<sup>13</sup>). Hra, či spíše simulace, má několik pravidel a její vývoj je závislý na výchozím stavu. Existuje mnoho implementací s komunitou hledající zajímavé vzorce.<sup>14</sup>

Pomocí CA lze simulovat například *Belousov-Zhabotinsky* reakci, či *Diffusion-Limited Aggregation*.<sup>15</sup> Reakce *Belousov-Zhabotinsky* reakce příkladem nerovnovážné termodynamiky, což vede ke vzniku nelineárního chemického oscilátoru. *Diffusion-Limited Aggregation* je proces difuze částic a agregace do fraktálních struktur. O CA a o mnohem dalším pojednává například kniha *A New Kind of Science* od Stephena Wolframa, zakladatele *Wolfram Research*.

## Vzory v přírodě

Alan Turing se v roce 1952 zajímal v práci[19] vznikem vzorů v přírodě z homogenních stavů. Nyní nazývané Turingovy vzory lze generovat pomocí modelů reakční difuze a *Gray-Scott* modelem. Zde je výčet dalších příkladů typů vzorů.

- Symetrie
- Stromy, fraktály
- Spirály
- Chaos, tok, meandry
- Vlny, duny
- Bubliny, pěna
- Teselace
- Praskliny
- Skvrny, pruhy

---

<sup>13</sup>Zdroj: <https://www.wolfram.com/language/gallery/implement-conways-game-of-life/>.

<sup>14</sup>Jedním ze zajímavých počátečních stavů má rekurzivní vlastnost a tedy po oddálení (změně měřítka) se objevuje stejný vzor jako v počátečním stavu (zdroj: <https://www.youtube.com/watch?v=uRIHR55tyko>).

<sup>15</sup>Zdroj: [www.hermetic-systems.com/compsci/cellular\\_automata\\_algorithms.htm](http://www.hermetic-systems.com/compsci/cellular_automata_algorithms.htm).



# 2

## KAPITOLA

# Instituce, komunita a výzvy

Tato kapitola se věnuje grafickým komunitám a skupinám, které mohou být užitečné i pro oblast kreativního programování. Vědecké publikace a výzkum je možné sledovat pomocí konferencí Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH). Příkladem dlouholetého fenoménu je komunita *Demoscene*. Další skupiny se sdružují například kolem konkrétních technologií, kde mohou sdílet své výtvory, nebo pořádat soutěže.

## 2.1 SIGGRAPH

SIGGRAPH je každoroční konference o počítačové grafice, skládající se z akademických přednášek i průmyslového veletrhu. Součástí konferencí jsou vzdělávací kurzy, panelové diskuze a různé interaktivní techniky. Hlavní konference se pořádá or roku 1974 v Severní Americe a druhá v od roku 2008 v Asii. SIGGRAPH uděluje několik cen za příspěvky v počítačové grafice.

### ACM SIGGRAPH

Konference jsou pořádané skupinou *ACM SIGGRAPH*. ACM SIGGRAPH je speciální zájmová skupina (special interest group) *ACM. Association for Computing Machinery* je mezinárodní učená společnost působící v oblasti výpočetní techniky. Byla založena v roce 1947 a jedná se o největší vědeckou a vzdělávací informatickou společnost.

## 2.2 Datové vizualizace, infografika

Kromě čistě uměleckého záměru, lze počítačovou grafiku využít například pro zobrazení informací a dat. I přes rozsáhlost tohoto oboru, budou ve stručnosti některé webové stránky zabývající se vizualizacemi dat a informační grafikou v této sekci zmíněny. Byly nalezeny příklady webů, kde je možné podle zadání přispět datovými vizualizacemi.

**storytellingwithdata.com** Webová stránka, kterou provozuje Cole Nussbaumer Knaflík nese stejné jméno, jako její kniha *Storytelling with data*. Je zde možné nalézt články, podcasty a připojit se ke komunitě jejíž cílem je sdělit myšlenku pomocí dat. Přispět ke komunitě je možné například sdílením na platformě *Twitter* s #SWDchallenge.

**makeovermonday.co.uk** Eva Murray a Andy Kriebel, autor knihy *#MakeoverMonday*, provozují stejnojmennou webovou stránku, kde v letech 2016-2021 byly zveřejňovány každé pondělí datové sady s instrukcemi pro vytvoření datových vizualizací, kde byl zaznamenáván počet přispěvatelů. Přes skutečnost, že momentálně v roce 2022 není stránka aktivní, jsou jejich datové sady používány,

**kaggle.com** Kaggle je dceřinná společnost *Google* a online komunita pro *data science* a strojové učení. Nabízí prostor pro inspiraci i soutěže, zahrnující různá téma například od vylepšení gest pro *Microsoft Kinect* po vylepšení hledání Higgsonova bosonu v *CERNu*.

### Inspirace

Další weby je možné sledovat pro inspiraci kreativní způsoby použití grafiky a téma, která popisují.

**pudding.cool** Webovou stránku *Pudding* obsahující vizuální eseje zaměřené na interaktivitu, provozuje Matt Daniels.

**visualcapitalist.com** Zakladatel *Visual capitalist* Jeff Desjardins spolu s týmem publikují datové vizualizace zaměřené na téma jakými jsou technologie, energie a globální ekonomie.

**informationisbeautiful.net** Webovou stránku provozuje David McCandless, autor několika infografických knih. Spolu s týmem sdělují informace pomocí přehledných barevných vizuálů, vytvořených interním nástrojem *VizSweet*.

## 2.3 Demoscene

*Demoscene* ([20], [21], [22]) je fenomén na pomezí digitálního umění, kreativní tvorby a undergroundu. Jedná se o mezinárodní decentralizovanou a nekomerční subkulturnu, která se zaměřuje na produkování digitální audiovizuální tvorby. Artefakty – *dema* – jsou typicky několikaminutové, počítacem generované sekvence animací, kombinace videí, textu a 3D grafiky.

Události kde soutěžící *demosceneři* živě produkují svá *dema* se nazývají *demoparty*. Výtvory musí být při prezentaci generovány v reálném čase. Soutěže mají kategorie, kde je nutné splnit různé limity. Například některá *dema* musí mít maximálně čtyři kilobyty, nebo musí být spuštěna na konkrétních počítacích, například C64.

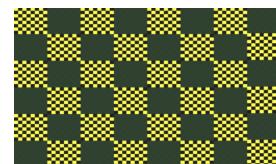
Fenomén *demoscene* byl přijat jako kulturní dědictví organizací *UNESCO* například ve Finsku (v roce 2020) a v Německu (v roce 2021). Anglickým portálem pro databázi *dem* je *pouet.net*. Počátky *demoscene* jsou v osmdesátých letech 19. století v *cracker* kultuře, kdy počítacový nadšenci přidávali své podpisy do her zbavených protekce proti kopírování, odkud se odloučili na nezávislé vyvýjení *dem*.

Demoscéna měla dopad na oblasti jako vývoj počítacových her a umění nových médií. S demoscénou souvisí například vznik *JavaScriptové knihovny Three.js* *demoscenerem* Ricardem Caballo<sup>16</sup>, nebo platformy *Shadertoy* pro sdílení a upravování vizuálů generovaných v reálném čase pomocí jazyka *OpenGL Shading Language (GLSL)*. Díky hardwarovému pokroku nemusí již být malá velikost programu nutností, avšak zkušenosti s kompaktním a efektivním kódem jsou nadále ceněné.

### Memories

Jedním příkladem takového dema je MS DOS program *Memories* od uživatele *HellMood*. Velikost programu je 256B a vyhrál soutěž "PC 256 byte" události *Revision* komunity *demoscene* v roce 2020 i cenu popularity. Analýzu tohoto programu popisuje článek<sup>17</sup> na stránce *Sizecoding wiki*. Demo obsahuje několik efektů, které se do sebe prolínají a hudbu. Příklad jednoho z efektů ukazuje 2.1.

```
xchg dx,ax ; get XY into AX
sub ax,bp ; subtract time from row
xor al,ah ; XOR pattern (x xor y)
or al,0xDB ; pattern for array of boards
add al,13h ; shift to good palette spot
```



Obrázek 2.1: Vizualizace *Array of chessboards* s velikostí 16 bytes.

<sup>16</sup>O vzniku hovoří v přednášce <https://www.youtube.com/watch?v=LXWYOF4VibE>.

<sup>17</sup><http://www.sizecoding.org/wiki/Memories>

## 2.4 Psaní malého kódu

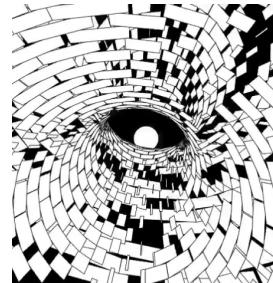
Programování co nejkompaktnějším způsobem bez ohledu na čitelnost rozehodně nepatří mezi nejlepší praktiky, avšak i tak má koncept svou přidanou hodnotu. Kompaktní programování může motivovat programátory objevovat hranice jazyka i algoritmů, nebo už jen sloužit k pobavení a výzvě mezi programátory.

### つぶやき processing, つぶやき GLSL

Na sociální síti *Twitter* se vyskytuje hashtag '# つぶやき Processing'.<sup>18</sup> Jedná se o výzvu<sup>19</sup> v tvoření pomocí *Processingu*, kdy je kód limitovaný délkou znaků.<sup>20</sup> Součástí příspěvku bývá i video náhled výsledku. Dále se výzva rozšířila i na jazyk GLSL.

Na příkladu 2.2<sup>21</sup> je vidět příklad kódu a (původně animovaný) vizuál, který kód generuje. Program lze spustit pomocí knihovny p5.js, což umožňuje zkomprimovat *JavaScript* do jinak nezkompilovatelné podoby. Je například možné vynechat klíčové slovo *let* u definice proměnných a *function* u definice funkcí (dokonce je použita *arrow function*). Pro vynechání *setup* funkce je použit operátor *||*, díky kterému se vytvoří *canvas* pouze při prvním vykreslení funkce *draw*.

```
t=0;draw=$=>{t||createCanvas(540,600);background(0);circle(270,270,50);for(j=-4+(t+=s=PI/20)/10%;j<4;j+=s)for(i=0;i<6;i+=s*2)quad(...[[i,j],[b=i+s*sin(t+i*j)*2,j],[b,j+s],[i,j+s]].flatMap(([u,v,e=v**2*40+80])=>[e*cos(u+j)+270-v*19,v*80+250+e*sin(u+j)]))}
```



Obrázek 2.2: Příklad つぶやき processingu od uživatele yuruyurau.

### Dwitter, Twigl

Podobně jako předchozí příklady, se stránky *Dwitter.net* a *Twigl.app* věnují výzvě psaní krátkého kódu s různou limitací počtu znaků. *Dwitter* umožňuje zobrazování výtvarů komunity, *Twigl.app* je spíše editor umožňující export videa. Stránky mají integrovaný editor pro vytváření příspěvků.

<sup>18</sup> つぶやき znamená japonsky *tweet*.

<sup>19</sup> <https://www.decombatch.com/2020/01/what-is-processing.html>

<sup>20</sup> V roce 2022 to je 280 znaků.

<sup>21</sup> Zdroj: <https://twitter.com/yuruyurau/status/1485987035207041024>.

## 2.5 Genuary, Nodevember

Některé komunitní výzvy se konají každoročně v určitý měsíc. Příkladem těch momentálně aktivních je Nodevember konající se v listopadu a Genuary v lednu<sup>22</sup>. Rámcová zadání pro daný den v měsíci – *prompts* – se odkrývají postupně, nebo všechna najednou. Vývojáři a umělci participující se těchto událostí poté sdílejí svoje výtvory na sociálních sítích s *hashtagy* události.

### Nodevember

Při události Nodevember lze použít libovolný procedurální či node-based software, jakými jsou například Blender, Substance, Houdini, či Unity. Podle pravidel není možné například používat obrázkové textury, načítat externí data, či manuální modelování a scriptování. Na stránkách (<https://nodevember.io>) jsou dostupné statistiky pro roky 2019, 2020, 2021 ukazující počet účastníků, vstupů, používaných programů a další. Nejúspěšnější rokem je 2020 s 3840 vstupy od 496 účastníků.

### Genuary

Genuary (<https://genuary.art>) nemá striktní pravidla a je možné vytvářet díla pomocí jakéhokoliv frameworku. Zadání na další roky je možné navrhnut prostřednictvím stránky Github. Zadání z roku 2022 zahrnují výzvy k inspiraci se malířem *Sol LeWitt Wall*, použití *ditheringu*, letištění koberec, nebo pouze barevnou paletu.

## 2.6 Webové technologie

Zvýšení kreativity omezením možností využívají i nadšenci používající webové technologie. Jednou takovou umělkyní je *Diana Smith*, která vytváří „olejomalby“ pouze pomocí HTML a CSS, jak ukazuje 2.3a<sup>23</sup>.

Existují i příklady jednoduchých mini her, opět využívající pouze technologie HTML a CSS, tentokrát i pro implementaci jednoduché logiky. Příkladem je hra 2.3b<sup>24</sup>, jejíž cílem je projít pomocí kurzoru myši postupně se odkrývajícím labryntem.

Lze se domnívat, že ačkoliv se pravděpodobně jedná především o výzvu, může být tento přístup vhodný v situacích, kdy není možné použít *JavaScript* (například když prohlížeč zakazuje provádění *scriptů*).

---

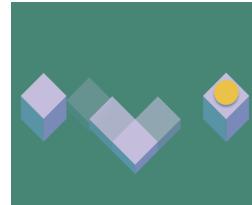
<sup>22</sup>Poslední běh *Codevember.xyz* byl roku 2019.

<sup>23</sup>Zdroj: <https://diana-adrienne.com>.

<sup>24</sup>Zdroj: <https://codepen.io/nathantaylor/pen/KaLvXw>.



(a) HTML a CSS olejomalba.



(b) HTML a CSS minihra.

## 2.7 The Future Sketches group

Příkladem uskupení zajímající se o využití kódu kreativním způsobem je *The Future Sketches group* na MIT. Jejich slovy se zabývají zkoumáním *software* jako média pro umění a design, a zároveň otázce, jak mohou nástroje a pedagogické přístupy zasvětit novou generaci do „výpočetního řemesla“. Kromě používání existujících nástrojů se soustředí i na vytvýření nástrojů nových, neb „nástroje dneška pomáhají utvářet umění zítřka“ [23]. Polem jejich současného výzkumu je generativní forma, strojové učení a rozšířená realita s důrazem na to, jak pochopit esenci těchto technologií a použít je nečekanými, poetickými způsoby.

## 2.8 Generativní script

Současným trendem je tvoření generativních *scriptů*, do nichž je nutné pro získání výsledku vložit *seed*, unikátní náhodný řetězec, parametrizující výsledný artefakt. Jedná se o jiný přístup k tvorbě a umění, kdy autor místo jednoho díla parametrizuje program tak, aby bylo možné získat různé (ideálně použitelné) variace. Tímto způsobem se dají napsat programy pro získání generativních stromů, domů, či celých světů<sup>25</sup>.

### NFT

Generativní scripty mají využití například ve světě obchodu s Non-fungible token (NFT). Platformy, jakou je například *Art Blocks*, umožňují digitálním umělcům nahrát generativní *script* na *Etherum blockchain* a následně specifikovat kolik z tohoto *scriptu* lze vygenerovat iterací. Sběratel poté například iteraci zakoupí a tím se mu vygeneruje unikátní artefakt, tudíž nikdo předem neví jakou variaci dostane.

---

<sup>25</sup>Jako ve hře *Valheim*.

# 3

## KAPITOLA

# Technologie pro kreativní programování

V této kapitole budou představeny typické technologie pro kreativní programování jakou je například *Processing*. Další programy jsou známé pro použití v *new media art*, jako například *Touchdesigner*. Dalším zohledněným hlediskem bude dostupnost webového editoru. Bude zde snaha prezentovat i kontext, ve kterém by mohly být použité nástroje jako *Blender* a *Desmos*.<sup>26</sup>

## 3.1 Processing

*Processing* je jedním z prvních nástrojů pro kreativní programování a pochází z roku 2001. Nyní je dostupný nejen v původní formě programu s vlastním vývojovým prostředím, ale i jako knihovna, kterou lze použít ve vlastním projektu:

**Processing** Program je možné stáhnout pro operační systémy Linux, Windows a Mac. Obsahuje editor pro psaní kódu, display pro zobrazení výsledku a manažer, pomocí kterého lze stáhnout přídavné knihovny, příklady, módy a nástroje. Nativním programovacím jazykem je Java, ale je zde možnost stáhnout módy pro Python, R, *p5.js* (JavaScript), GLSL a další. Módy mohou a nemusí být udržovány pro aktuální verzi Processingu.

**Processing.py** Dalším způsobem jak vyvíjet v jazyce Python je instalace knihovny *p5* pro Python. Prerekvizitou je instalace knihovny *GLFW*.

---

<sup>26</sup>Více nástrojů lze nalézt například na adrese <https://github.com/terkelg/awesome-creative-coding>.

### **3. TECHNOLOGIE PRO KREATIVNÍ PROGRAMOVÁNÍ**

---

**p5.js** Pro webový vývoj existuje knihovna *p5.js*, kterou lze přidat pomocí HTML tagu *script*, či nainstalovat do projektu pomocí balíčkovacího systému (*npm*, *yarn*)

Díky Processingu je možné s psaním relativně málo řádků kódu (do souborů zvaných *sketche*) vytvořit zajímavý vizuál. V Každém *sketchi* je funkce *setup* určená pro instrukce které se vykonávají před začátkem animace a funkce *draw* určená pro instrukce, které se provádí v nekonečné kreslící smyčce.

Mimo kreslení primitiv je knihovna nativně vybavena například funkcemi šumu a různými matematickými funkcemi. Lze načítat soubory, pracovat s pixely, detektovat vstupy z klávesnice a myši, pracovat s křivkami a maticemi. Je možné pracovat ve 2D i ve 3D.

Lze se domnívat, že vzhledem k množství materiálů a referencí je *Processing* (potažmo *p5.js*) momentálně (v roce 2022) nejpopulárnější knihovnou pro začátky v kreativním programování. Projekt *Processing* sponzoruje nezisková organizace *Processing Foundation*.

#### **Online editory**

Pravděpodobně nejpopulárnější volbou pro rychlý začátek bez nutnosti instalace žádného programu je online editor.

**editor.p5js.org** Jednoduchý způsob, jakým zkusit vyvíjení pomocí p5.js. Na chází se zde intuitivní editor a po přihlášení je možné své *sketche* ukládat a sdílet ostatním. Z hlediska výkonu může být lepší lokální vývoj.

**openprocessing.org** JavaScriptová verze Processingu, tedy p5.js. Na stránce je možné prohlížet výtvory uživatelů.

**teddavis.org/p5live/** Stránka Theodora Davise pro živé kódování, například pro VJing (živé vizualizace) i výuku. Výchozí nastavení zobrazuje průsvitný editor, za kterým je vidět výsledek renderu. Mezi funkce patří například i zobrazení tabule. Stránka umožňuje kolaborativní kódování, různé možnosti nastavení editoru a zobrazení, ukázkové kódy i vestavěnou dokumentaci p5.js.

<https://www.reddit.com/r/creativecoding/>

openFrameworks

Cinder

Houdini

MAX/MSP

Pure Data

## 3.2 Technologie pro webový vývoj

Výhodou webových technologií je například jistá nezávislost na operačním systému. Typické je pro webový vývoj použití *JavaScriptu*, či některého z jazyků, které se do něj komplilují, jako například *TypeScript*, *Dart* a *CoffeeScript*. Zmiňované technologie je možné použít v projektu pomocí *CDN*, instalovat pomocí balíčkovacích systémů, nebo jsou oficiálně nabízené obě tyto možnosti.

Další výhodu poskytují stránky umožňující vývoj přímo v prohlížeči, například codesandbox.io JSFiddle

### Technologie s 2D *enginem*

Tyto webové technologie se dají odlišit například nejčastějším počtem dimenzí vykreslovacího *enginu*. Příkladem jsou uvedeny p5.js, PixiJS a Paper.js.

**p5.js** O p5.js již bylo psáno v sekci .

**PixiJS** PixiJS je lightweight multiplatformní 2D knihovna používající WebGL pro hardwarovou akceleraci. Knihovnu je mimo jiné možné použít pro vývoj her.

**Phaser** Framework pro vývoj her.

**Paper.js**

### Technologie s 3D *enginem*

Další skupina knihoven, které lze použít při kreativních projektech využívá, JavaScriptové API pro renderování 2D a 3D grafiky přes kompatibilní webové prohlížeče. WebGL umožňuje akceleraci přes grafickou kartu a tudíž knihovny, které ho využívají lze použít v projektech větších rozsahů.

**Three.js** Již zmíněná kn

**Babylon.js**

stack.gl

### Pomocné knihovny

**D3** JavaScriptová knihovna pro vytváření dynamických, interaktivních vizualizací dat ve webových prohlížečích

**Quicksettings**

## Z hlediska výkonu

Většina prezentovaných webových technologií využívá pro akceleraci výkonu přes grafickou kartu standart *WebGL* (který staví na základech *OpenGL* a *DirectX*), některé používají modernější a rychlejší *WebGPU* (využívající *Vulkan*) – například *Babylon.js*. Další alternativou je využití *WebAssembly*, což je cílem například projektu *GLAS*<sup>27</sup>.

### 3.3 Vizuální programování

Vizuální programovací jazyk je druh programovacího jazyka, ve kterém nejsou příkazy znázorněny textem, nýbrž nějakým grafickým znázorněním. Většinou mají příkazy podobu takzvaných bloků, které dohromady tvoří program. Bloky vypadají jako barevné kostičky, kde každý druh bloků plní jinou funkci. Wikipedia

#### TouchDesigner

TouchDesigner je uzlový vizuální programovací jazyk pro interaktivní multimediální obsah v reálném čase, vyvinutý torontskou společností Derivative. Používali jej umělci, programátoři, tvůrci programátoři, softwaroví designéři a umělci k vytváření představení, instalací a děl pevných médií. Wikipedia (angličtina)

(free) for personal and non-commercial use image resolution is limited to 1280x1280.

#### Vvvv

Vvvv je sada nástrojů pro všeobecné použití se zvláštním zaměřením na syntézu videa v reálném čase a programování prostředí velkých médií s fyzickými rozhraními, pohybovou grafikou v reálném čase, zvukem a videem. Vvvv používá přístup k toku dat a vizuální programovací rozhraní pro rychlé prototypování a vývoj. Wikipedia (angličtina)

Vvvv is free for evaluation and non-commercial use. For all commercial applications, you need to buy a license. Commercial use means that the work done with Vvvv is primarily intended for or directed toward commercial advantage or private monetary compensation.

We are well aware of the fact that the distinction between commercial and non-commercial has a somehow blurry boundary. We trust the fair judgement of our users.

---

<sup>27</sup>Odkaz na projekt: <https://github.com/lume/glas>.

### 3.4 Herní enginy

Herní engine (někdy také herní motor) je softwarový framework, který soustřeďuje obecné funkce používané v počítačových hrách, díky čemuž dovoluje zrychlit a zlevnit vývoj nových her.[1] Rozsah funkcí se u různých enginů liší a lze tak nalézt jak jednoduché knihovny omezující se na vykreslování, tak rozsáhlé enginy i s vlastní sadou vývojových nástrojů. Některé herní firmy si vyvíjejí vlastní engine, jiné si licencují cizí engine.

#### Unreal

Unreal Engine je herní engine, který byl vytvořen firmou Epic Games. Jeho první verze z roku 1998 byla použita ve hře Unreal. Od té doby byl Unreal Engine několikrát vylepšen a doplněn, aby mohl být použit v několika desítkách novějších herních titulů. Původně byl tento engine určen pouze pro střílečky z pohledu první osoby, ale našel využití i v některých MMORPG, RPG a adventurách.

#### Unity

Unity je multiplatformní herní engine vyvinutý společností Unity Technologies. Byl použit pro vývoj her pro PC, konzole, mobily a web. První verze podporovala pouze OS X a byla představena na celosvětové konferenci Apple v roce 2005. Od té doby byl rozvinut o více než patnáct dalších platform.

#### Godot

Godot je 2D a 3D multiplatformní open source herní engine pod licencí MIT vyvinutý komunitou, byl interně používán v několika společnostech v Latinské Americe, předtím než byl uvolněn jako open-source a zpřístupněn veřejnosti.[2] Vývojové prostředí běží na Windows, macOS a Linux (oba 32 a 64 bit) a může vytvářet hry cílené na PC, konzole, mobily a webu.

## 3.5 Kreativní programování mimo digitální umění

Kreativní programování mimo digitální umění

#### Hudba

**SuperCollider** Platform for audio synthesis and algorithmic composition.

**Sonic Pi** The live coding music synth for everyone.

### Plotter art

<https://asmen.icopy.site/awesome/awesome-plotters/>

**AxiDraw** The AxiDraw is controlled from within Inkscape by means of the standard WaterColorBot (WCB) extensions which were developed by EvilMadScientist.

### 3D tisk

<https://github.com/ad-si/awesome-3d-printing>

**an-embroidery-machine-dst-export** <https://entagma.com/python-in-houdini-controlling-an-embroidery-machine-dst-export/>

### Vyšívací stroj

**an-embroidery-machine-dst-export** <https://entagma.com/python-in-houdini-controlling-an-embroidery-machine-dst-export/>

# KAPITOLA 4

## Vybrané projekty

Cílem kapitoly je poskytnout příklady známých i méně známých projektů, přibližující rozsah a možnosti umění nových médií, kreativního programování a vizualizací.

### 4.1 Signal

Signal je pražský festival probíhající pravidelně čtyři dny v říjnu a to každoročně již od roku 2013. Program je tvořen renomovanými zahraničními i českými umělci z oblasti light designu, vizuálního a digitálního umění, umělé inteligence, ale i konceptuálního umění. Slovy oficiální stránky projektu[24] – „Festival rád vzdělává sebe i návštěvníky“. Příklady z ročníku 2021 jsou na obrázku 4.1<sup>28</sup>.



Obrázek 4.1: Signal festival 2021

### 4.2 Expo

<https://www.czexpo.com/en/photogallery/?id=24>

<sup>28</sup>Zdroj: <https://www.signalfestival.com/rocnik/2021/>.

#### 4. VYBRANÉ PROJEKTY

---



(a) Linky na FEL ČVUT



(b) SAGE lab



(c) Tunel mezi Národními muzei



(d) CAMP, 25metrová stěna

Obrázek 4.2: Atypické vizualizační plochy

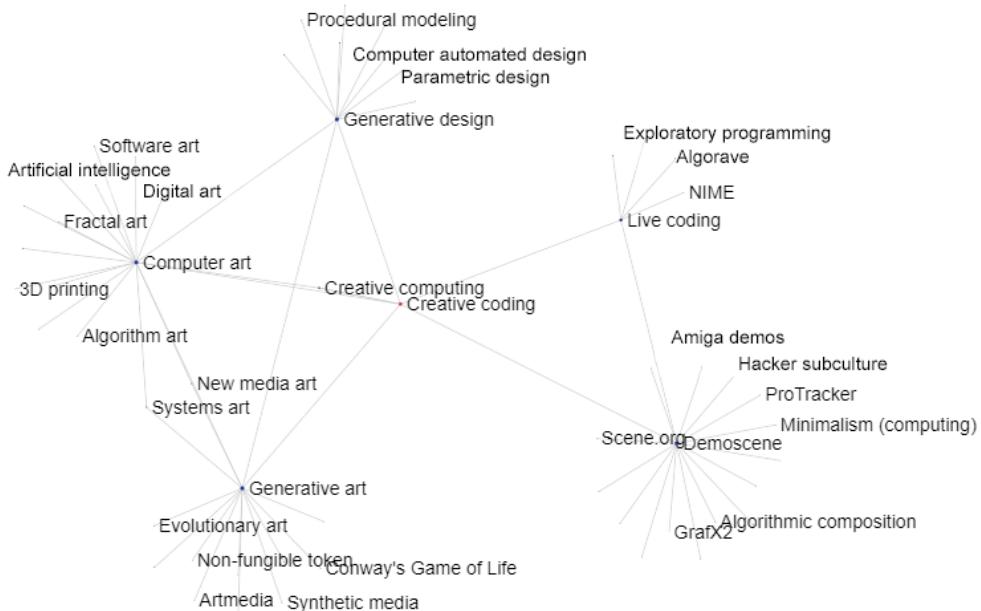
### 4.3 Příklady atypických vizualizačních ploch v Praze

**Linky** Světelná instalace na budově FEL ČVUT je zhodnocením vizuálního podoby celého dejvického kampusu. Oživuje prostor nejen pro studenty a zaměstnance ČVUT, ale i pro širokou veřejnost. Budova, kterou do posud kolemjdoucí míjeli bez povšimnutí, nabízí nyní díky proměnlivé světelné instalaci chodcům na Dejvickém náměstí možnost dialogu s architekturou a v širším kontextu oživuje i celou městskou čtvrt.<sup>4.2a</sup>

#### SAGE lab

#### CAMP

#### Tunel



Obrázek 4.3: Příklad výstupu z projektu *Seealsology*.

## 4.4 Grafové vizualizace

### Obsidian

#### Seealsology

Jako zástupce ze oblasti informačních vizualizací byl vybrán projekt Seealsology<sup>29</sup>, od *DensityDesign Research Lab* z Milána, dostupný i se zdrojovým kódem na stránce *Github*<sup>30</sup>.

Vstupem vizualizace jsou příspěvky na stránce *wikipedia.org* a nastavení, jaké odkazy a do jaké hloubky se mají procházet. Výstupem je graf propojení různých stránek na Wikipedii. Tento nástroj byl použit i pro hledání pojmu související s tématem této diplomové práce.

Aplikaci je vhodné používat interaktivně, kromě odkazů na vyhledané stránky je možné například pomocí klávesové zkratky přidat další vyhledávací vrchol v reálném čase. Příklad výstupu<sup>31</sup> pro stránku z Wikipedie na téma *creative coding* ukazuje obrázek 4.3.

<sup>29</sup>Zdroj: <https://densitydesign.github.io/strumentalia-seealsology/>

<sup>30</sup><https://github.com/densitydesign/strumentalia-seealsology>

<sup>31</sup>Mírně editovaný s lepší viditelností popisků.

### *Six degrees of Wikipedia*

Stránka *Six degrees of Wikipedia*<sup>32</sup> vyhledá nejkratší cesty mezi dvěma zadanými stránkami na Wikipedii a následně zobrazí orientovaný graf vybraných nejkratších cest. Jméno referuje hypotézu „Šest stupňů odloučení“, tedy myšlenku, že lidé jsou v průměru šest nebo méně sociálních kontaktů od sebe.

## 4.5 Electric Sheep

Jak je uvedeno na stránce projektu<sup>33</sup>, Electric Sheep je kolaborativní abstraktní umělecké dílo. Aplikaci je možné nainstalovat na mnoho různých zařízení a použít například jako spořič obrazovky.

Autorem projektu Electric Sheep<sup>34</sup> je Scott Draves. Využívá druh fraktálů *Fractal flame*, který Draves napsal roku 1992 a následně zpřístupnil pod GNU General Public License (GPL) licencí.

Oblíbenější animace – ty, pro které více lidí hlasovalo v aplikaci – žijí v systému déle a reprodukují se genetickým algoritmem pomocí mutace a křížení.

<https://github.com/mame/quine-relay>

---

<sup>32</sup>[sixdegreesofwikipedia.com](http://sixdegreesofwikipedia.com)

<sup>33</sup><https://electricsheep.org>

<sup>34</sup>Název je narážkou na román Philipa K. Dicka – *Do Androids Dream of Electric Sheep?* z roku 1968.

# Výukové kanály a materiály

Pod pojmem *creative coding* lze na internetu nalézt<sup>35</sup> mnoho knih, učebních materiálů a návodů, z nichž některé jsou i placené. Materiály jsou dostupné pro různé technologie i různé úrovně programovacích schopností. Některí se kromě výukových videí zaměřují i na budování komunity, ve které je možné se například vzájemně učit, povzbuzovat v tvorbě.

## 5.1 Processing, p5.js, Three.js, webové technologie

### Dan Shiffman

Dan Shiffman je členem představentstva *Processing Foundation* a profesorem Interactive Telecommunications Program (ITP) na New York University Tisch School of the Arts (Tish). Napsal dvě knihy, *The Nature of Code* a *Learning Processing*. Provozuje populární kanál na platformě *YouTube* jménem *The Coding Train*. Začátkem roku 2022 je počet jeho odběratelů 1,37 milionů.

Jeho YouTubový kanál *The Coding Train* se zaměřuje především na kreativní kódování v *Processingu* a *p5.js*. Kromě základů používání těchto technologií se jeho videa věnují i na kódování algoritmů. Mezi rozebíranými tématy jsou například různé funkce šumu, celulární automaty, fraktály, ale i interakce pomocí snímaných dat přes *Kinect*, využití strojového učení a datových Application Programming Interface (API).

Součástí kanálu je například seznam videí doplňují výukové materiály, později vydané i jako kniha *The Nature of Code*, jež je dostupná zadarmo. Ukázky kódu z nikdy i mnohých dalších videí jsou dostupné na stránce *Github*.

<sup>35</sup>Rozcestníkem pro mnohé materiály je například tento komunitou vytvořený seznam: <https://github.com/terkelg/awesome-creative-coding>.

## 5. VÝUKOVÉ KANÁLY A MATERIÁLY

---

### **newandnewermedia**

Za zástupce živých vizualizací pomocí knihovny *p5.js* se nabízí kanál *newandnewermedia*.

### **codingmath**

<https://www.bit-101.com/blog/2021/11/thoughts-on-art/> <http://www.codingmath.com>

### **Franks laboratory**

Čech <https://www.youtube.com/c/Frankslaboratory>

## 5.2 Unreal, Blender, Touchdesigner

### **Freya Holmér**

Acegikmo  
<https://www.youtube.com/c/Acegikmo>

### **Erindale**

Erindale se zaměřuje na procedurální tvorbu v programu *Bledner*.

### **Noto The Talking Ball**

## 5.3 GLSL

### **The Book of Shaders**

<https://thebookofshaders.com>

### **Inigo Quilez**

Inigo Quilez, pocházející z krásného městečka<sup>36</sup> ve Španělsku, je softwarový vývojář, technický umělec, produktový manager a učitel s více jako 20 letou praxí, jež pracoval například pro Pixar, či Oculus. O svém hobby, počítačové grafice, natáčí video návody a píše články.

Je autorem mnohých dem<sup>37</sup>. Na svých stránkách<sup>38</sup> má pod MIT licencí užitečné funkce a části kódu. Na svém YouTube kanále popisuje tvorbu scén, které i přestože vypadají být vytvářeny v modelovacím softwaru, jsou vytvořeny pouze procedurálně pouze pomocí jazyka *GLSL*. Příklad zobrazuje orbázek 5.1.

---

<sup>36</sup>[urlhttps://iquilezles.org/personal/donostia/donostia.htm](https://iquilezles.org/personal/donostia/donostia.htm)

<sup>37</sup>Zmiňovaných zde 2.3

<sup>38</sup><https://iquilezles.org/www/index.htm>



Obrázek 5.1: 3D grafika vytvořena pouze pomocí matematiky.

### The Art of Code

Návody specializující se GLSL shadery



# 6 KAPITOLA

## Osobní portfolia

Je mnoho artistů, kteří zasluhují obdiv. Pro účely této práce zde budou jmenováni některí z těch, kteří algoritmy a postupy tvorby popisují a sdílejí prostřednictvím internetu, například formou blogových článků, či videí. Na rozdíl od předchozí kapitoly, jsou popsané algoritmy vysvětleny v kontextu vlastních uměleckých projektů.

### 6.1 Sage Jenson

V této kapitole bu

Physarum Characteristics of pattern formation and evolution in approximations of physarum transport networks <https://uwe-repository.worktribe.com/output/980579>

Organic 3D Mesh Creation through Particle-Based Physics Simulation  
<https://archive.bridgesmathart.org/2016/bridges2016-411.pdf>

<https://cargocollective.com/sagejenson/particle-studies-2016>

### 6.2 Robert Hodgin

<https://roberthodgin.com/project/akari-lanterns>

Akari Lanterns procedural lantern generation

Meander procedural map generator for meandering rivers

Embroidery Procedural embroidery system

Murmuration Simulation of bird flocks

Fish Tornado VR Simulation of collective behavior

Sol Realtime interactive simulation of the Sun

### 6.3 Webové a plotter projekty

Následující umělci svoje díla prezentují převážně pomocí webových stránek, či fyzicky vytiskněné pomocí plotterů. Přínosem této sekce je inspirace a zdroj odkazů úspěšných, populárních, či technicky popsaných a propracovaných projektů.

#### Tyler Hobbs

Tyler Hobbs ve svých esejích<sup>39</sup> píše o generativním umění, kterému se věnuje profesionálně. Jeho slovy „používá způsob přemýšlení naučený z programování pro analýzu vizuálního světa kolem nás“. Některé jeho inspirativní přednášky (například z události *Strange Loop Conference*) a rozhovory jsou na platformě *YouTube*.

Tyler Hobbs ve svých projektech již od roku 2016 věnoval vektorovým polím (*flow fields*) a tuto techniku použil například ve svém díle *Fidenza*. *Fidenza* je příkladem digitální umění, které vzniká generativním skriptem (vytvořeným pomocí knihovny *p5.js*) a je prodáváno přes platformu *Art Blocks*. Výdělky z prodeje následně přispěl organizacím *Processing Foundation*, *Girls Who Code* či *AGE of Central Texas*. Příklad výstupu díla *Fidenza* zobrazuje obrázek 6.1.

#### Baptiste Crespy

Baptiste Crespy (známý pod přezdívkou *ciphrd*) je generativní umělec a WebGL vývojář. Zajímá se například o autonomní systémy, tvořící život připomínající chování. Na svém blogu<sup>40</sup> dokumentuje jakým způsobem simuloval známé reakce a procesy, jakými je například *Diffusion-Limited Aggregation* či *Reaction-diffusion*. Obrázek 6.2 je příkladem jeho práce.

Je autorem projektu *Seed Simul*, interaktivní webové stránky, kde uživatelé mohou zasadit virtuální semínka ve 3D prostředí. Tyto vstupy od uživatelů poté pomalu časem rostou a tvoří se tak unikátní kolaborativní struktura. Momentálně *ciphrd* zakládá a vyvíjí platformu FxHash, sloužící pro tvorbu a sběr generativních NFT na *tezos blockchainu*.

#### Anders Hoff

Další inspirativní generativní artista Anders Hoff sdílí svoje nadšení pro simulaci vzorů pomocí svého blogu *Inconvergent*<sup>41</sup>. Příklad jeho práce ukazuje obrázek 6.3. Kolem let 2015 a 2016 napsal generativní eseje inspirující se růstem vláken hub a stromů, či tvorbou různých umělců. K těmto esejům je dostupný kód. Pozdější články se inspirují písečnými dunami (*Grains of sand*),

---

<sup>39</sup><https://tylerxhobbs.com/essays>

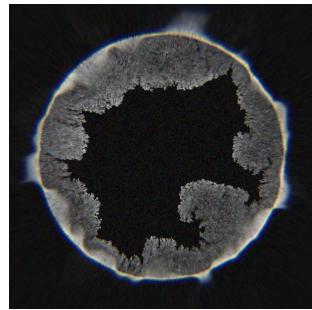
<sup>40</sup><https://ciphrd.com>

<sup>41</sup><https://inconvergent.net/generative/>

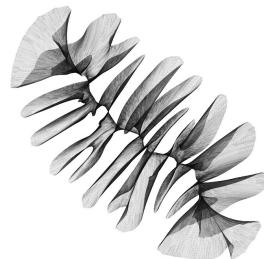
či využívají grafovou strukturou (*Tangle of webs*). Je autorem framework pro generativní systémy jménem *Weir*<sup>42</sup>, využívající programovací jazyk *Lisp*.



Obrázek 6.1: Fidenza



Obrázek 6.2: DLA



Obrázek 6.3: inconvrgent

## 6.4 Herní vývoj

Následující sekce se opět zaměřuje na vývojáře sdílející svůj kreativní postup s veřejností, ovšem zaměřuje se na ty, kteří tvoří převážně komplexnější interaktivní světy, či přímo hry. Uvedení autoři používají pro sdílení informací videa a platformu *YouTube*.

### Sebastian Lague

#### CodeParade

Jedním z tvůrců

<https://www.youtube.com/c/CodeParade>

prosli jsme to (a vlastne uz to, ze jsem to celkem zbezne prochazeli 4 hodiny rika neco o tom, ze to není jen trocha nudnyho a zbytecnyho textu), plno veci okecali a rekli, ale jen ti jeste sem musim nekam napsat ze je to velka a krasna tuna prace. klobouk dolu. opravdu. není to jeste samozrejme hotove, ale uz ted je v tom videt za me skvely prehled a obri halda inspirace.

<sup>42</sup><https://github.com/inconvrgent/weir>



# **Část II**

## **Praktická část**



# KAPITOLA

# 7

## Analýza

### 7.1 Rešerše podobných aplikací

Před návratem podoby aplikace Metaviz byla provedena rešerše podobných webových aplikací, které sdružují několik vizualizací. V této sekci budou popsány příklady těchto aplikací a jejich funkcí.

#### Cineshader, IFS, Shadertoy, Threejs

Příspěvky komunity, konkrétní technologie – GLSL, či Three.js. Je možné zobrazit kód, některé jsou interaktivní.

#### devart

Devart<sup>43</sup> je kolekce kreativních projektů různých výstav, kde vývojáři – umělci popisují své projekty po technické a technologické stránce.

Mezi funkce webové aplikace patří mimo vyhledávání filtrace programovacích jazyků, frameworků a knihoven. U konkrétního projektu je poté možné jej sdílet.

#### generativeartistry, jurasic

Hugo Další skupinou aplikací jsou interaktivní návody, které například postupným scrollováním po stránce rozkrývají části kódu, který produkuje dané vizualizace. Příkladem takových interaktivních tutoriálů je *GenerativeArtistry*<sup>44</sup>, kde autoři zároveň provozují podcast. Stránky jsou vygenerovány pomocí

<sup>43</sup>[devart.withgoogle.com](http://devart.withgoogle.com)

<sup>44</sup><https://generativeartistry.com/tutorials/>

## 7. ANALÝZA

---

knihovny pro statické generování stránek *Hugo*. Další interaktivní stránkou s vysvětlením algoritmů je RedblobGames<sup>45</sup> a jurasic<sup>46</sup>.

### Výsledky rešerše podobných aplikací

Mnohá řešení jsou interaktivní, buď tím způsobem, že uživatelé mohou přidávat své artefakty, či návštěvníci ovládat vizualizace přednastavenými parametry. Pro potřeby některých stránek jsou dostatečná statická webová aplikace.

## 7.2 Požadavky na aplikaci

Aplikace *Metaviz* má za cíl představit konkrétní vizualizace, čili požadavek na přidávání dalších artefaktů komunitou není ambicí. Aplikaci je však nutné přizpůsobit možnosti použití různých technologií.

### Funkční požadavky

čím aplikace bude disponovat

- **FP01 Zobrazení vizualizací**

nejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popis

– **Priorita:** 5

– **Složitost:** 4

### FP02 Zobrazení kódu

nejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popis

- – **Priorita:** 4

– **Složitost:** 2

- **FP03 Zobrazení popisu vizualizací**

nejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popisnejaky popis

– **Priorita:** 3

– **Složitost:** 2

---

<sup>45</sup><https://www.redblobgames.com>

<sup>46</sup><https://www.jurasic.dev>

## Nefunkční požadavky

Technické požadavky.

- **NFP01 Webová aplikace**

Aplikace bude dostupná výhradně přes web.

- **NFP02 HTTPS**

Aplikace bude komunikovat za pomoci protokolu HTTPS.

- **NFP03 Spustitelnost**

Aplikace bude spustitelná v minimálně následujících prohlížečích.

- **NFP04 Optimalizovanost vizualizací**

Minimální hardwarové parametry pro spuštění vizualizací:

- GPU –
- RAM –

### **NFP04 Optimalizovanost vizualizací**

V případě, kdy není možné dostát hardwarovým požadavkům, bude dostupný záznam fungování vizualizace.

## Případy užití

Případy užití rozvíjí funkční požadavky a dále je specifikují a člení na jednotlivé současné případy užití.

### **1. FP01 Zobrazení vizualizací**

- UC01 Zobrazení přehledu vizualizací
- UC02 Zobrazení konkrétní vizualizace

### **FP02 Zobrazení kódu**

2.
  - UC03 Zobrazení kódu konkrétní vizualizace

## **7. ANALÝZA**

---

### **3. FP03 Zobrazení popisu vizualizací**

- UC04 Zobrazení popisu konkrétní vizualizace

Tabulka pokrytí funkčních požadavků případy užití

Diagram aktivit

Nepřihlášený uživatel

Diagram případů užití

# 8

## KAPITOLA

# Návrh

## 8.1 Doménový model

UML diagram doménového modelu

- Vizualizace
- Popis
- Odkaz na kód
- Video

## 8.2 Struktura a architektura

Statická stránka

## **8. NÁVRH**

---

Serverová aplikace

Klientská aplikace

### **8.3 Technologie**

Serverová aplikace

Klientská aplikace

### **8.4 Uživatelské rozhraní**

Použitelnost

Úkoly aplikace

Lo-fi prototyp

Bávrh lo-fi prototypu

Heuristická analýza lo-fi prototypu

# 9

## KAPITOLA

# Implementace

## 9.1 Vizualizace 1 – periodicia videa

Cílem tohoto projektu je synchronizovat libovolnou hudbu s videi ve standar-dizovaném stylu *rubber hose* animací, které byly populární kolem roku 1920. Budou zde porovnány různé způsoby, jakým byla snaha tohoto úkolu docílit.

### Videotextury

Video textury je typ média na pomezí fotografií a videem. Plynulý a nekonečně se opakující proud obrázků. Analýzou videa se obdrží jeho struktura, již synté-zou vznikne nové video, podobné tomu původnímu, o *libovolné délce*. Použití video textur zahrnuje zobrazení dynamických scén na webových stránkách, tvorbu dynamických kulis pro scény například v počítačových hrách nebo interaktivní ovládání videem - poháněných animací. Způsob, jakým by se daly využít v tomto projektu je ukázán na stránce [25].

### Popis algoritmu

Začátek video textur je článek[26] z roku 2020, od té doby byly provedeny optimalizace a rozšíření, například [27] či [28]. Algoritmus je popsán například i v jedné z implementací [29].

### Analýza videa

První částí je analýza videa a uložení dat do struktur, z nichž bude možné vycházet v následující části sestavení videotextury.

**Černobílé hodnoty** V matici  $\mathcal{I}_i$  o rozměrech  $M \times N$  (šířka  $x$  výška videa) jsou pro každý snímek  $i$  uloženy hodnoty od 0 do 255 (černobílý jas pixelu získaný z červené, zelené a modré složky).

**Podobnost snímků** Pro každou dvojici snímků se zjistí jejich vzdálenost (Euklidovská norma) a uloží do matice  $D_{ij} = \|\mathcal{I}_i - \mathcal{I}_j\|_2$ .

**Způsoby zachování dynamiky** Pokud se objekt hýbe například zleva doprava, algoritmus by mohl mylně měnit směr. Zachování dynamiky se může zajišťovat například optickým tokem, či váženým oknem, kdy se v úvahu berou i příspěvky sousedících snímků (podle binomického rozdělení).

**Vážené okénko** Dalším krokem je přičtení příspěvků okolních  $m+m$  snímků přenásobených o váhy  $[w_{-m}, \dots, w_{m-1}]$ . Přepis matice je tedy

$$D'_{ij} = \sum_{k=-m}^{m-1} w_k D_{i+k, j+k}, \quad (9.1)$$

kde hodnota pohyblivého okénka  $m$  je typicky 1 či 2. Výsledkem je matice s rozměry  $(N - 2m) \times (N - 2m)$ , jelikož na každém kraji bude chybět  $m$  snímků pro porovnání.

**Očekávaná budoucí cena** Pro generování potenciálně nekonečného videa by bylo velmi nevýhodné vybrat přechod, ze kterého je málo přechodů, případně po kterém už video končí. Toto se optimalizuje výpočtem ceny přechodů. Definuje se matice  $D''_{ij}$  očekávané ceny přechodu ze snímku  $i-1$  na snímek  $j$  součtem přes všechny očekávané ceny:

$$D''_{ij} = (D'_{ij})^p + \alpha \sum_k P''_{jk} D''_{jk}, \quad (9.2)$$

kde

$$P''_{ij} \propto \exp(-D''_{i+1,j}/\sigma). \quad (9.3)$$

Konstanta  $p$  parametruje jestli má být výsledkem spíše málo (ale kvalitních), nebo více (méně kvalitních) přechodů. Konstanta  $0 < a < 1$  reprezentuje relativní váhy budoucích přechodů. Systém rychle konverguje pokud  $0.99 < a < 0.999$ .

Jedním ze způsobů řešení je simultánní iterativní vyhodnocování  $D''_{ij}$  a  $P''_{ij}$  přivádějící systém blíže do konvergence a přerušení, další výpočty neovlivňují matici. Bohužel tento způsob konverguje pomalu.

Jelikož při  $\sigma \rightarrow 0$  půjde hodnota  $P''_{ij}$  k 1 pro nejlepší přechod a k 0 pro nejhorší, lze přepsat rovnici 9.2 na

$$D''_{ij} = (D'_{ij})^p + \alpha \min_k D''_{jk}. \quad (9.4)$$

Tato rovnice je známá jako Q-Learning (metoda model-free reinforcement learningu). Díky tomu se matice bude procházet jako graf, kde vrcholy jsou vážené přechody. Je zde možnost selektivního přepočítávání hodnot řádků v  $D''_{ij}$  pro každý krok. Nejnižší cena často zahrnuje přechod ze snímku, který je blízko konci, což lze dále propagovat dopředu. Dále se inicializuje  $D''_{ij} = (D''_{ij})^p$  a definuje

$$m_j = \min D''_{jk} \quad (9.5)$$

Od posledního řádku k prvnímu a naopak se počítá

$$D''_{ij} = (D''_{ij})^p + \alpha m_j \quad (9.6)$$

a aktualizuje se hodnota  $m_j$  použitím rovnice 9.5.

**Pravděpodobnosti z ceny přechodů** Získaná cena přechodů se vyjádří jako pravděpodobnost tak, že se namapuje použitím exponenciální funkce

$$P''_{ij} = \exp(-D_{i+1,j}/\sigma). \quad (9.7)$$

Tímto budou vyšší ceny přechodu namapovány na menší pravděpodobnost. Nižší hodnoty konstanty  $\sigma$  upřednostňují méně lepších přechodů, vyšší hodnoty více horších přechodů.

**Výběr nejlepších přechodů** jsou dva způsoby, jak zredukovat celkový počet stavů (uzlů/přechodů) v matici:

1. braní v potaz pouze lokální maxima, nebo
2. nastavení pravděpodobnost přechodů pod určitou hranici na 0.

### Syntéza textur

V práci [29] jsou jmenovány 4 způsoby syntézy textur ze získané matice přechodů.

tedy tedy zbyva rici ze syntéza není provedena ALE je vizualizovana matice (+ukazka, obrazek, barvicky), povidani o tom jak moc to je, ci není z matice videt atd. plus bych prihodil nejakou trivialitu - napr. zvyrazneni nejdelsiho cyklu (ty dva bodu budou cervene, jasne, tmave, ...). tim rekneme, ze mame zcela a uplne vedecka data a princip, a i tato vedecka data se daji vizualizovat hezky

**Náhodné přehrávání** Náhodné přehrávání znamená, že není použito předchozí analýzy a v každém snímku  $i$  se vybere snímek  $j$  tak, že  $j > i$ . Původní článek uvádí náhodné přehrávání pro kontrast s informovaným náhodným přehráváním.



Obrázek 9.1: Crown shyness

**Nejdelší cyklus** Nejdelší cyklus jednoduše identifikuje nejdelší možný cyklus z matice přechodů. Ve výsledku tohoto přístupu lehce pozorovatel pozná opakování.

**Informované náhodné přehrávání** Po přehrání snímku  $i$  je vybrán následující snímek  $j$  podle pravděpodobnostní matice  $P_{ij}$ .

### Video loop play

Mezi dostupné implementace videotextur patří například ještě [30], [31], [32], [33], [34] a [35]. Repozitář [35] funguje, má komentáře a testy. Pro použití by možná šel optimalizovat. Repozitář [29] má výborné video i text k videotexturám a kód se povedl po přeložení do Pythonu 3 a troše snahy autorce této práce spustit, avšak výpočet se po několika hodinách nedokončil na systému Windows ani Linux.

### Popis vytvořeného repozitáře

## 9.2 Vizualizace 2 - Crown Shyness

zaprve - strasne se mi to libi. zadruhe - nezapomen i tady ukazat klidne vyvoj, kterym to proslo.

*Crown shyness* je přírodní fenomén, který se vyskytuje u některých druhů stromů, kdy se koruny stromů navzájem nedotýkají, ale tvoří se mezi nimi prázdná místa. Hypotézy pro tento jev jsou debatovány například zde – [36].

Tento jev společně s prvním pokusem o implementaci tohoto jevu zobrazuje obrázek 9.1<sup>47</sup>. Pro implementaci byla použita knihovna *p5.js*, *Voronoi noise* a *L-systémy*. Výstup má náhodné vstupní parametry a tudíž je po načtení mírně odlišný.

<sup>47</sup>Zdroj: <https://www.demilked.com/crown-shyness-phenomenon-trees-avoid-touching/>

### 9.3. Vizualizace 3 - Kensuke Koike

#### **9.3 Vizualizace 3 - Kensuke Koike**

Kensuke Koike je umělec původem z Japonska \*1980 June 28, born in Nagoya, Japan

2004-2007

IUAV University, Faculty of Arts and Design, Venice, Italy 1999-2004

Accademia di Belle Arti, Venice, Italy

<https://www.kensukekoike.com>

**Rešerše prací inspirovaných tímto umělcem**

#### **9.4 Vizualizace 4 - Convex hull**

desmos convex hull



## KAPITOLA

# 10 ■

## Testování a reflexe nad projektem

10.1 Persony

10.2 Scénáře testování

10.3 Výsledky testování

10.4 Aplikace kreativního programování

Výuka programování

Řemeslo

Na pomezí mezi řemeslem a koní

Průmysl

Prezentace výsledků výzkumu

10.5 Pokračování projektu



# Závěr

Kreativní programování může mít využití pro lidi bez zkušeností s programováním, i pro zkušené programátory. Může to být způsob, jak se v programování zdokonalit,

Cílem práce bylo ...

Důležitá a nejpracnější je myšlenka a nápad.



# Literatura

- [1] Feynman, R. P.: *The pleasure of finding things out: The best short works of Richard P. Feynman*. Basic Books, 2005.
- [2] Taylor, G. D.: *When the machine made art: the troubled history of computer art*. Bloomsbury Publishing USA, 2014.
- [3] Fry, B. J.: *Computational information design*. Dizertační práce, Massachusetts Institute of Technology, 2004.
- [4] Fry, B.: *Visualizing data: Exploring and explaining data with the processing environment*. "O'Reilly Media, Inc.", 2008.
- [5] Monoskop. [online]. Dostupné z: <https://monoskop.org/Monoskop>
- [6] Digital art timeline. [online]. Dostupné z: [https://wikieducator.org/Digital\\_art\\_timeline](https://wikieducator.org/Digital_art_timeline)
- [7] Creative Coding: Perspectives & Case Studies. [online]. Dostupné z: <https://javascript.plainenglish.io/all-about-creative-coding-e79268d944e8>
- [8] The Pioneers (1950-1970). [online]. Dostupné z: <http://www.vam.ac.uk/content/articles/a/computer-art-history/>
- [9] Computer Art History, Characteristics of Digital Imagery. [online]. Dostupné z: <http://www.visual-arts-cork.com/computer-art.htm>
- [10] Nake, F.: PARAGRAPHS ON COMPUTER ART, PAST AND PRESENT. 01 2010.
- [11] Nedvedová, M.; Marek, J.: Pioneering Works in the Application of Random Numbers to Digital Art and Linear Programs of Zdeněk SýKora. In *Proceedings of the 32nd Spring Conference on Computer Graphics, SCCG '16*, New York, NY, USA: Association for Computing Machinery, 2016, ISBN 9781450344364, str. 61–66, doi:10.1145/2948628.2948637. Dostupné z: <https://doi.org/10.1145/2948628.2948637>

## LITERATURA

---

- [12] Wands, B.: Creating continuity between computer art history and contemporary art. 2010.
- [13] Ceric, V.: Algorithms, Mathematics and Art. *Simul. Notes Eur.*, ročník 21, 2011: s. 1–10.
- [14] Dietrich, F.: Visual Intelligence: The First Decade of Computer Art (1965–1975). *IEEE Computer Graphics and Applications*, ročník 5, 1985: s. 33–45.
- [15] Shiffman, D.; Fry, S.; Marsh, Z.: *The nature of code*. D. Shiffman, 2012.
- [16] Lagae, A.; Lefebvre, S.; Cook, R.; aj.: A survey of procedural noise functions. In *Computer Graphics Forum*, ročník 29, Wiley Online Library, 2010, s. 2579–2600.
- [17] Vojtěch, T.: *Vizualizace 3-rozměrných dat slunečního povrchu*. B.S. thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2019.
- [18] Jones, J.: Characteristics of pattern formation and evolution in approximations of physarum transport networks. *Artificial Life*, ročník 16, 2010: s. 127–153, ISSN 1064-5462, doi:10.1162/artl.2010.16.2.16202. Dostupné z: <https://uwe-repository.worktribe.com/output/980579>
- [19] Turing, A. M.: The chemical basis of morphogenesis. *Bulletin of mathematical biology*, ročník 52, č. 1, 1952: s. 153–197.
- [20] Nationwide Inventory of Intangible Cultural Heritage. [online]. Dostupné z: <https://www.unesco.de/en/culture-and-nature/intangible-cultural-heritage/demoscene-culture-digital-real-time-animations>
- [21] Demoscene. [online]. Dostupné z: <https://en.wikipedia.org/wiki/Demoscene>
- [22] Kudra, A.: AoC | Art of Coding –The Demoscene as Intangible World Cultural Heritage. 2020.
- [23] Group Overview' Future Sketches. [online]. Dostupné z: <https://www.media.mit.edu/groups/future-sketches/overview/>
- [24] Signal Festival. [online]. Dostupné z: <https://www.signalfestival.com/o-festivalu/>
- [25] Xiaofeng Tao : Video Textures. Dostupné z: <http://cs.brown.edu/courses/cs129/results/final/taox/>

- [26] Schödl, A.; Szeliski, R.; Salesin, D.; aj.: Video textures. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.
- [27] Sevilla-Lara, L.; Wulff, J.; Sunkavalli, K.; aj.: Smooth loops from unconstrained video. In *Computer Graphics Forum*, ročník 34, Wiley Online Library, 2015, s. 99–107.
- [28] Panchal, K.: Improved Algorithm for Seamlessly Creating Infinite Loops from a Video Clip, while Preserving Variety in Textures. *arXiv preprint arXiv:2011.02579*, 2020.
- [29] Struan McDonough: VideoTextures. Dostupné z: <https://github.com/VirtualVirtuoso/VideoTextures>
- [30] Maxim Kazakov: CP\_video\_textures. Dostupné z: [https://github.com/max-kazak/CP\\_video\\_textures](https://github.com/max-kazak/CP_video_textures)
- [31] sudeepgbits: video textures using python. Dostupné z: <https://github.com/sudeepgbits/video-textures-using-python>
- [32] lucaslefaucher: Video-Textures. Dostupné z: <https://github.com/lucaslefaucher/Video-Textures>
- [33] J. Mecom, A. Wang: video-textures. Dostupné z: <https://github.com/microaeris/video-texture>
- [34] omnimeta: Video Texture Generator. Dostupné z: <https://github.com/omnimeta/video-texture-generator>
- [35] Patrick Violette: video-texture. Dostupné z: <https://github.com/pviolette3/video-texture>
- [36] Hattimare, R.: Crown shyness in various tree species. *Int. J. Sci. Dev. Res*, ročník 3, 2018: s. 322–324.



## Seznam použitých zkratek

**1D** 1-dimenziální.

**2D** 2-dimenziální.

**3D** 3-dimenziální.

**API** Application Programming Interface.

**CA** celulární automat.

**GLSL** OpenGL Shading Language.

**GPL** GNU General Public License.

**HCI** Human – Computer Interaction.

**ITP** Interactive Telecommunications Program.

**MIT** Massachusetts Institute of Technology.

**NFT** Non-fungible token.

**SIGGRAPH** Special Interest Group on Computer Graphics and Interactive Techniques.

**Tish** New York University Tisch School of the Arts.



# B

## PŘÍLOHA

### Obsah přiloženého média

```
└── readme.txt ..... stručný popis obsahu média
    └── text ..... text práce
        ├── thesis.pdf ..... text práce ve formátu PDF
        └── thesis.ps ..... text práce ve formátu PS
```