

Jak **mě** fraktály a grafy a vzorce a texty **učily** létat

Pavel Stríž, OSSConf 2019, Žilina



Jevišťe

Motto a vize

def. Létat = nehladá sazba, např. sazba mimo účaří
= sazba čehokoliv kamkoliv (neodborně)
= skákání a poletování radostí (u lidí)
= rotování zoufalstvím (u mne)

Vrstva 1. Pozadí videa: fraktály

Vrstva 2. Zlatý střed: podklady T_EXové

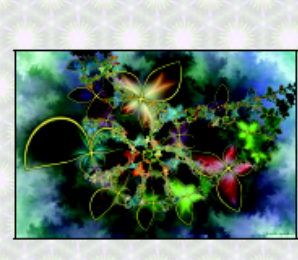
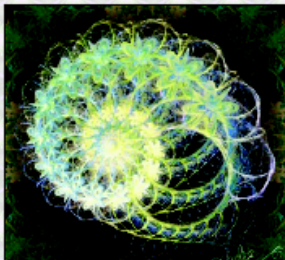
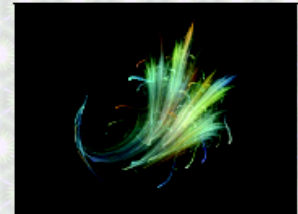
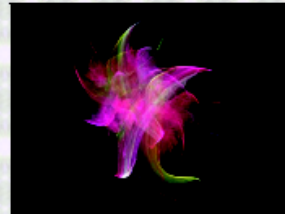
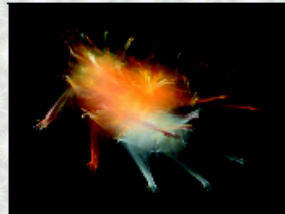
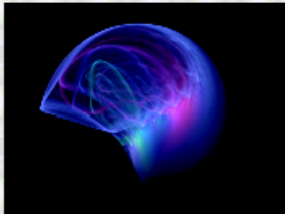
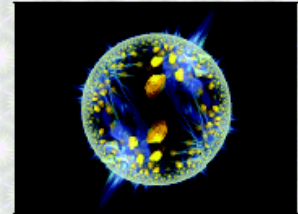
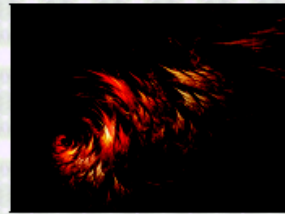
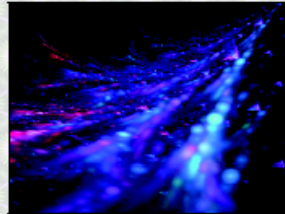
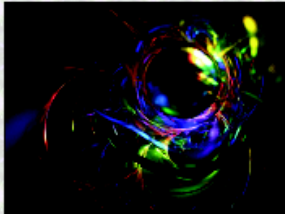
Vrstva 3. Popředí videa: něco pohyblivého ve 3D

Fraktálové začátky

- Tu a tam zmíním české přínosy. Fraktály se u nás zabýval profesor Drahoš, kandidát na prezidenta. Skončil v pořadí druhý.
- Jsem odkojen na programu POV-Ray.
- Nejdřív jsem testoval program ContextFree.
- Přešel jsem na programy Apophysis (3D), ApoMap, Incendia, HexQuilts a Geometrica.
- Zaujal mě web <http://bridgesmathart.org>
- Pro OSSConf 2012 vznikla výstava, 5 stran.

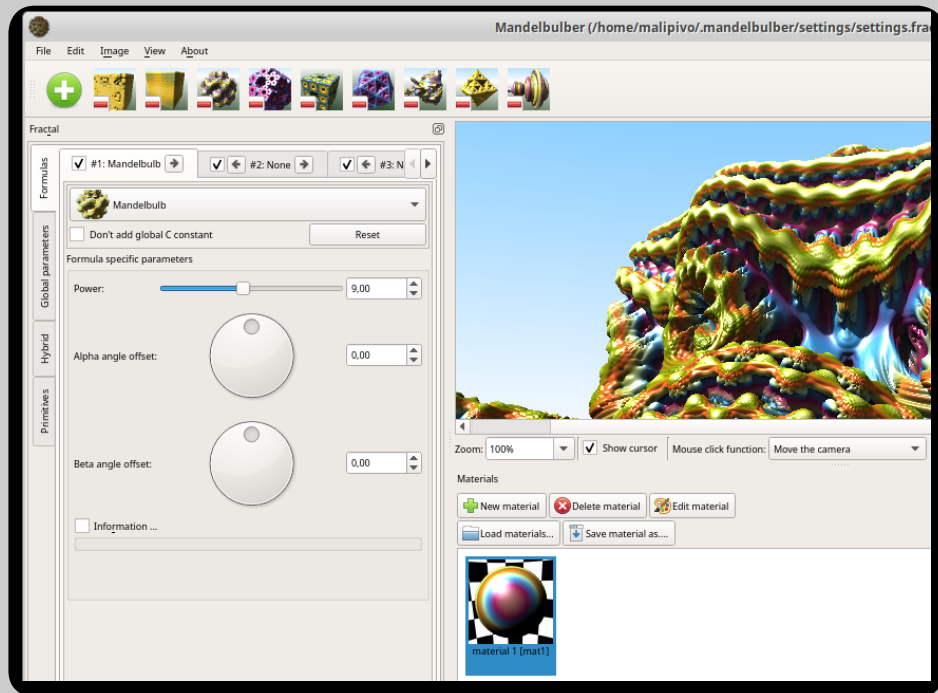
Apophysis &

Rendered by www.strain.cc, updated by Pavel Štíhl for CGSConf2011



Rodina Mandelbulb*

Z novějších mě zaujaly Mandelbulber(2) (včetně Linuxu) a Mandelbulb 3D. Občas se tam video urodí, spíš běh variacemi jednoho fraktálu. Zkoumal jsem dál.

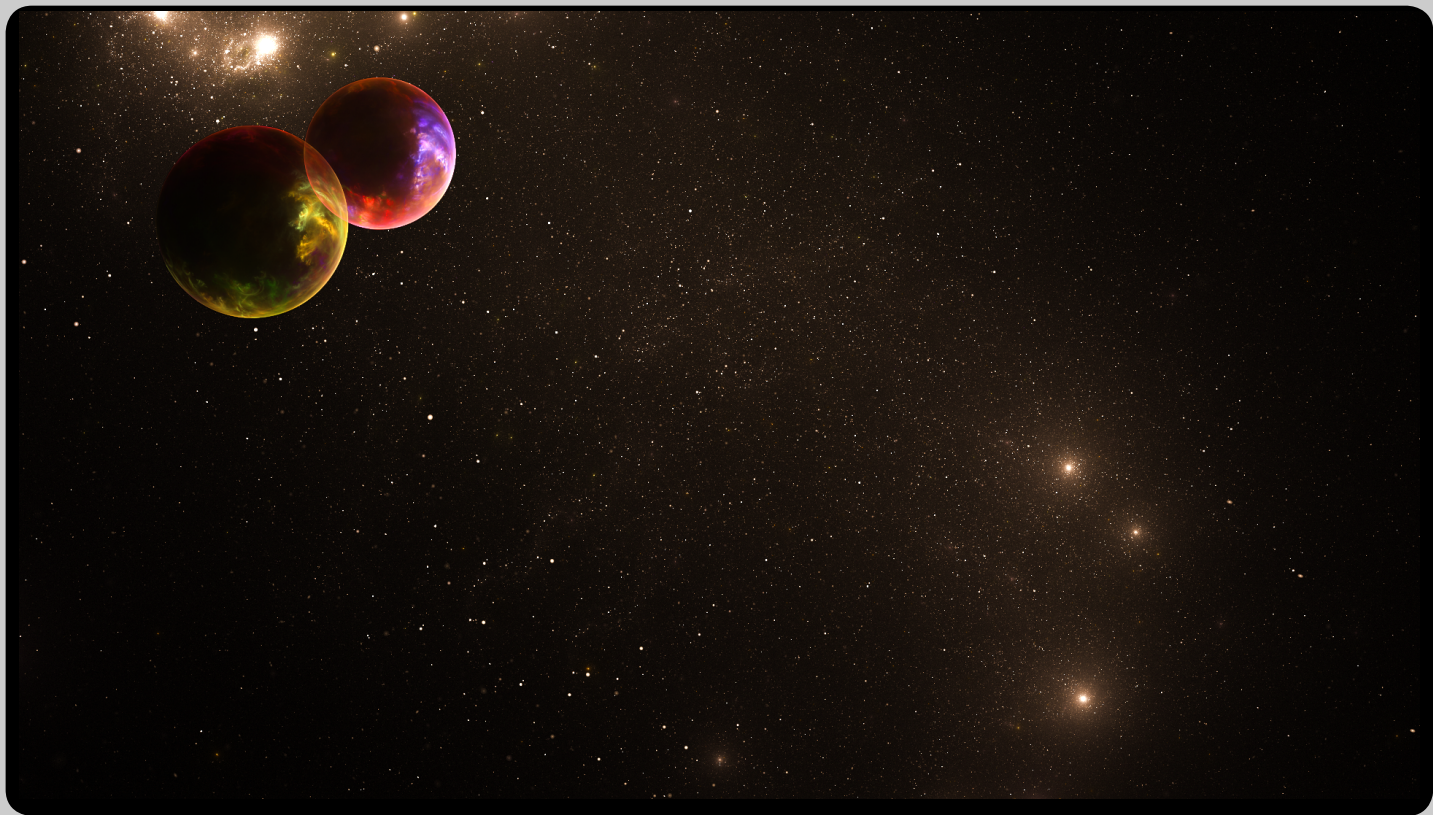


Současné pokusy

- Čistě na 2D animace jsem zkusil SynFig. **Ukázka**
- Vyzkoušel jsem program Qosmic.
- Následoval pokus s programem Chaotica.
- Prozkoumal jsem současný stav starších programů a objevil Apophysis-j-28.jar.
- U diskuzí jsem zahlédl název jWildfire.

jWildfire

- Rozšiřuje transformační možnosti `flam3` (má příponu souborů `flame` a `jwfmovie`).
- Možnost statického fraktálu, animování (zapíná se 3D) i přechodů (Movies).
- Lze použít texty (`text_wf`), načíst si SVG (`svg_wf`), IFLAME (`iflames_wf`) či OBJ (`obj_mesh_wf`). Vše je řízeno přes Transformations – Nonlinear.
- Nastavitelné rozlišení, výstup lze i do HDR.
- Komunita rozšiřuje možnosti skripty v Javě. Po pár kliknutích jsem získal pozadí do Linuxu (HB 2 planetoid smooth). Nový obrázek přes tlačítko Run.



Příprava podkladu

```
#!/bin/bash
echo "LaTeXuji..."
lualatex texty.tex 1>/dev/null 2>/dev/null
echo "Konvertuji..."
gm convert -density 100 -fuzz 100%
    -transparent white texty.pdf texty.png
inkscape --export-plain-svg=texty.svg
    -T texty.pdf
```

Pozn.: -T Convert text object to paths on export (PS, EPS, PDF, SVG)

První vzorky

- jWildfire+Kdenlive. Ukázka
- jWildfire+Flowblade. Ukázka
- Ikonka rozpohybování parametrů:
Create/edit a motion curve. 
- Klíčový problém je přechod mezi fraktály. Ukázka

Přechod mezi fraktály

Ten na základě úprav parametrů řeší `flam3`. Ale funguje to jen, když jsou geneticky zpřízněné, jinak vzniká neostrý obraz, de facto šum. **Ukázka**

Co s tím?

Flam3

- High Fidelity Demo, místy napojené, místy nikoliv.
- Ukázky, historie: <https://flam3.com/>
- Matematické pozadí:
http://flam3.com/flame_draves.pdf
- Pro žáky: představa přechodu z úsečky na část paraboly.
- Představa sítě bodů: v Blenderu přechod z krychle na kouli a obráceně.
- Github: github.com/scottdraves/flam3/wiki

Flam3-genome

- Soubor s příponou flam3 je textový XML soubor s parametry. Přípona flame (program jWildfire) je de facto to samé.
- Klíčový je flam3-genome: generuje z aktuálního fraktálu nový (Genetic Algorithm). Lze se dostat do původního fraktálu.

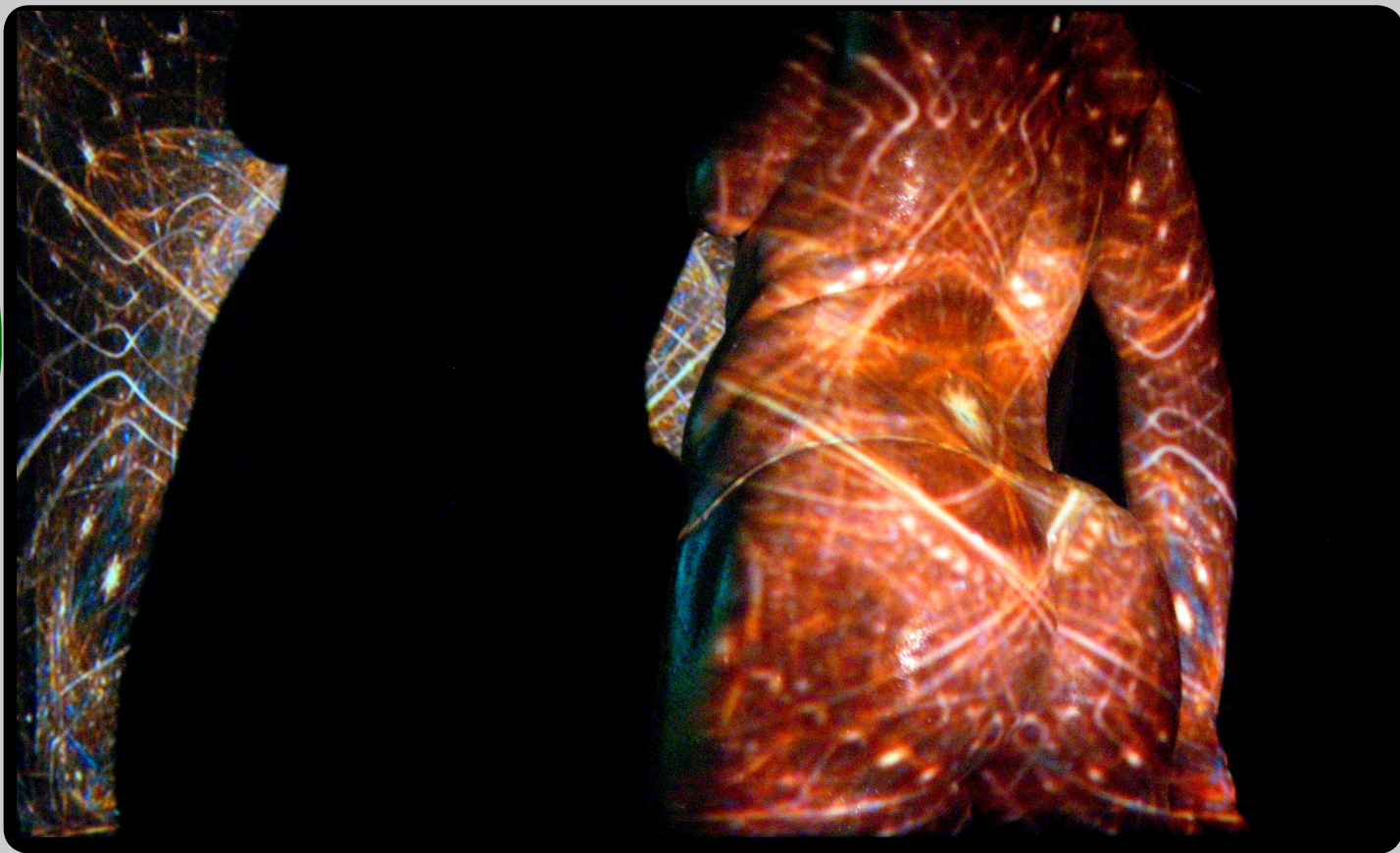
```
env sequence=ABC.flame nframes=30  
flam3-genome > ABC_anim.flame
```

Flam3-animate

- Poté flam3-animate vezme dva fraktály a vytvoří sekvenci. Navíc hlídá ostrost obrazu (Density Estimation) a přechod mezi snímky (Motion Blur).

```
env prefix=ABC format=png  
flam3-animate < ABC_anim.flame
```

- Následuje spojení obrázků do videa čímkoliv (ffmpeg, mencoder).

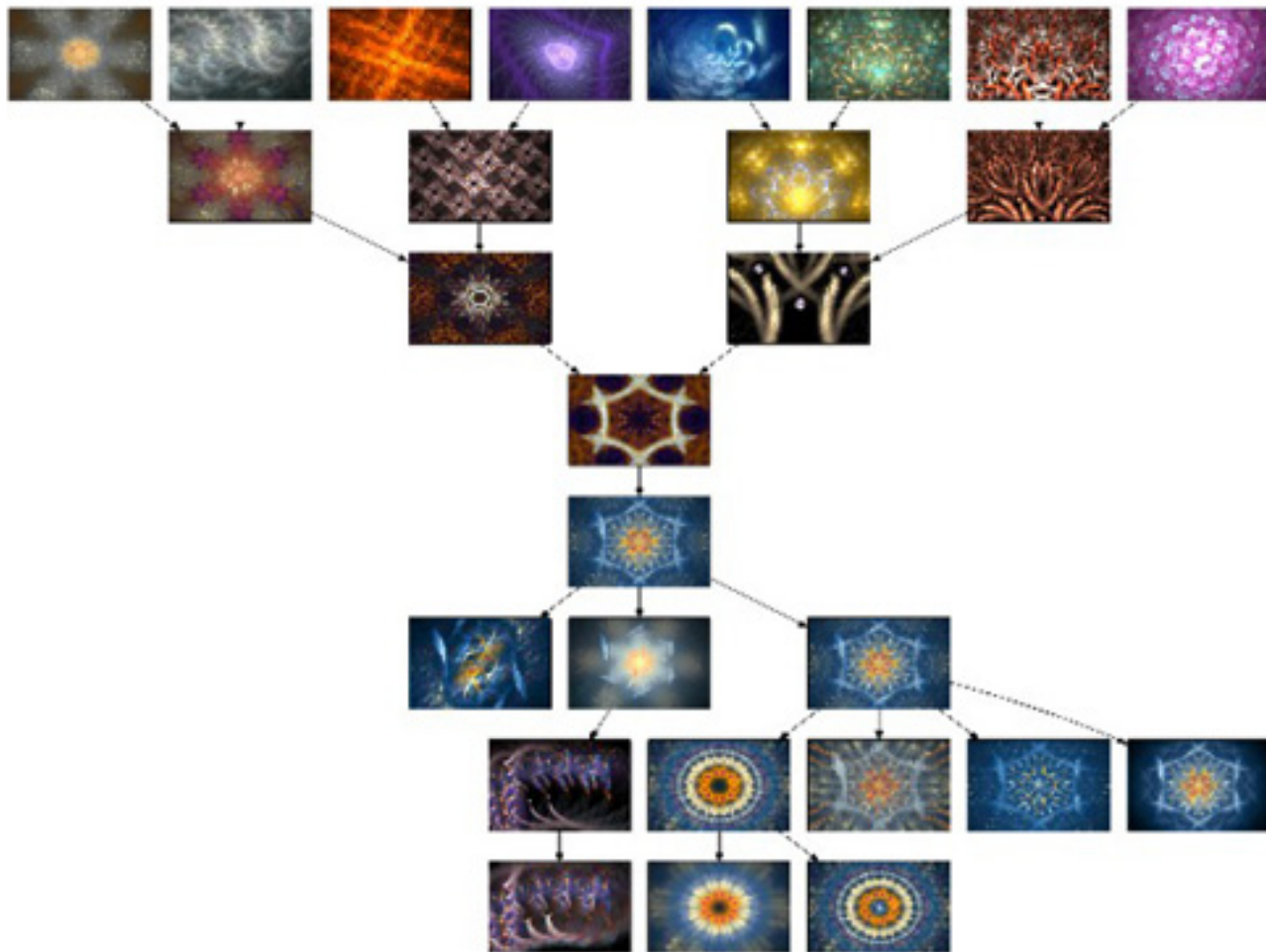


Electric Sheep (ES)

- Výpočetních prostředí je hodně: IBM, Google, Amazon.
- Tohle generuje snímky jako spořič obrazovky.
- Bohužel na pevný disk se stahuje jen poslední vypočtené.
- Problém při kompilaci (Xubuntu) jsem řešil přes:
`make CXXFLAGS+=-DGL_GLEXT_PROTOTYPES`
místo `make`, vyřešeno však v Arch Linuxu, Manjaru aj.

Aktuální generace: 00247

- <http://v3d0.sheepserver.net/cgi/status.cgi>
- Zde si lze stáhnout fraktály (icons, faces), přechod do sebe sama (loops) a přechod mezi dvěma fraktály (edges).
- Pod konkrétním fraktálem v sekci Genome si lze stáhnout zdrojový soubor flam3.
- O pokračování výpočtů rozhoduje hlasování. Některé větve se dál nevyvíjí, vzniká jistý strom z párů.



Free vs Gold

Na Electric Sheep se můžete stát zlatým členem (150 USD na celý život). Rozdíly jsou následující:

Ukazatel	Free	Gold
Resolution	800×598	1280×720
Frames/Sheep	160	320
Bandwidth	4.7mbps	12.3mbps
Server	archive.org	Amazon Cloudfront

Délka jednoho videa (typ loop či edge) je 6,7 vteřin.

Počítat to?

- Nereálné!
- Přidal jsem se do FB skupiny **ElectricSHE3P**.
- Jeden z členů (John Schuepbach) ta spojení stahoval víc jak 10 let. Dal své sbírky na archive.org. Cca 200 GB.
- Lze užít jen metadata (názvy souborů), není tam však délka. Mám rád metadata (poprvé v životě)!
- Pozn. Když už, tak je lepší zjišťovat počet snímků (`ffmpeg -i`) než čas videa.

Stahování

- `wget --limit-rate ...`
- Ideál ze sf.net: `curl -C -O -L --limit-rate ...`
- `youtube-dl --limit-rate ...`
- Bez CLI v Java: FreeRapid Downloader (od Čecha, Vity)
- `transmission-cli --downlimit ...`
- Konfigurovatelný program `aria2c`

Vstupní doktorská prohlídka

Flock No.	Souborů	ZIP (GB)	Částí	H×W (px)
00004	1342	8,9	2	1280×720
00202	12846	55,6	10	640×480
00243	2930	7,5	1	800×592
00244	15662	36,9	8	800×592
00245	2853	5,8	1	800×592
00247	5356	10,6	2	800×592

Název souboru

Přes metadata z archive.org či přes výpis souborů po stažení (ls), získáváme tisíce souborů tohoto typu:

00202=09556=09503=08806.avi

- 00202 je generace, flock, generation, sheep,
- 09556 je pořadové číslo výpočtu,
- 09503 je číslo/ID počátečního fraktálu a
- 08806 je číslo/ID konečného fraktálu.

Tvorba grafu

- Poslední dvě pětímístná čísla mohou být stejná: video začíná a končí ve stejném fraktálu.
- Pokud přehrajeme video pozpátku, virtuálně prohazujeme tato dvě pětímístná čísla.
- Každé video utváří orientovanou hranu. Jednu reálnou a jednu potenciální.
- Na graf se díváme jako na neorientovaný, ale hlídáme-li (ne)možnost cesty zpět, graf se stává orientovaným.

Soutěžní programování

- Následují algoritmy a úvahy spadající pod soutěžní programování.
- Nepočítáme-li **TAOCP**, tak komplexní kniha je **Competitive Programming**, autoři bratři Steven a Felix Halimovi, 3. vydání (4. vydání se čeká v květnu 2020). Problémy odkazují na servery UVa a Kattisu.
- Bioinformatické problémy: <http://rosalind.info>
- Bioinformatická soutěž: <https://bioinf.me/en/contest>
- Praktické problémy: <https://www.topcoder.com>

Rekurze

- Vnořování voláním funkce sebe sama.
- Občas lze rekurzi přepsat na **for** cyklus (faktoriál).
- Výhodné je využití paměti, např. Fibonnacciho řada.
- Využívá se alfa-ořez u her typu bridž či šachy, hlouběji M. L. Ginsberg o GIBu a Artificial Intelligence od autorů S. Russella a P. Norviga.
- Google DeepMind porazil světovou jedničku ve hře Go (kombinace metod).

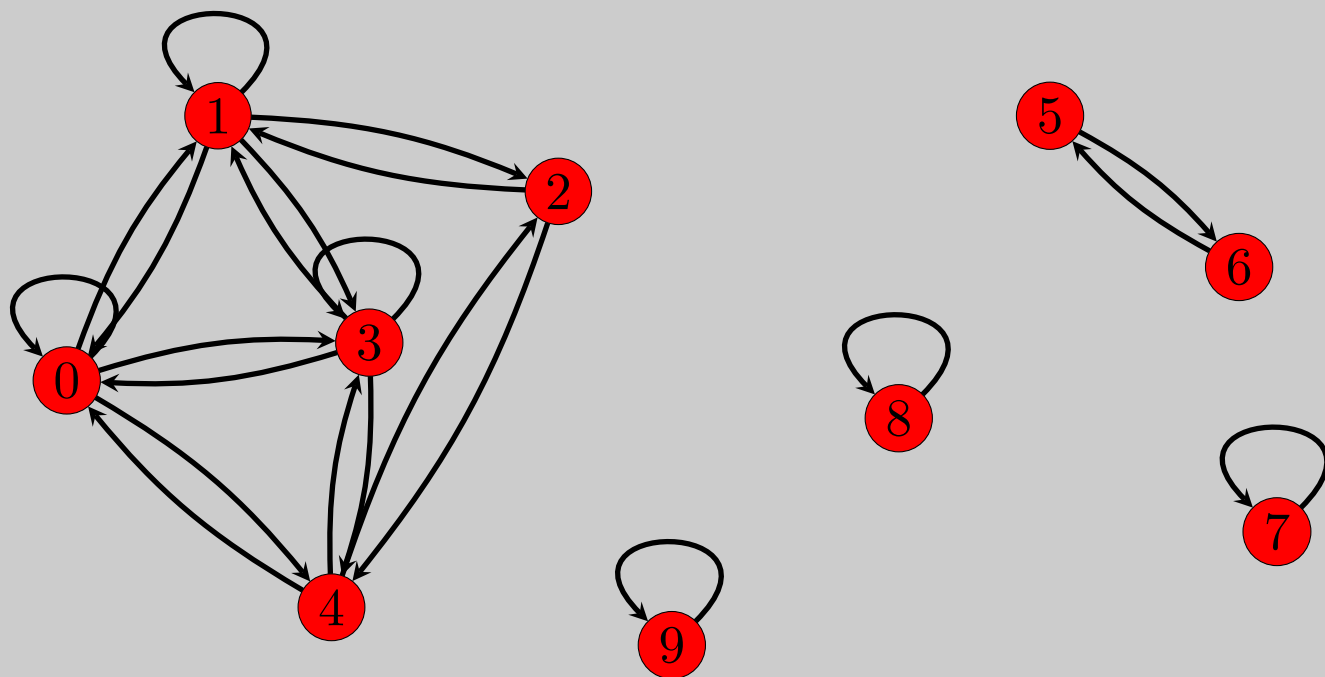
Rekurze (pokr.)

- V Pythonu jsem zvyšovat hloubku přes:

```
import sys  
sys.setrecursionlimit(50000)
```

- Zastavení programu:
 - kompletní projití možností,
 - dosažení určité délky cesty,
 - nastavením limitu běhu programu (knihovna `time`), či
 - do stisknutí klávesy (knihovna `pynpu.keyboard`).
- **OpenProblem:** Pozastavení programu s rekurzí a jeho znovuspuštění.

Nalézt největší graf

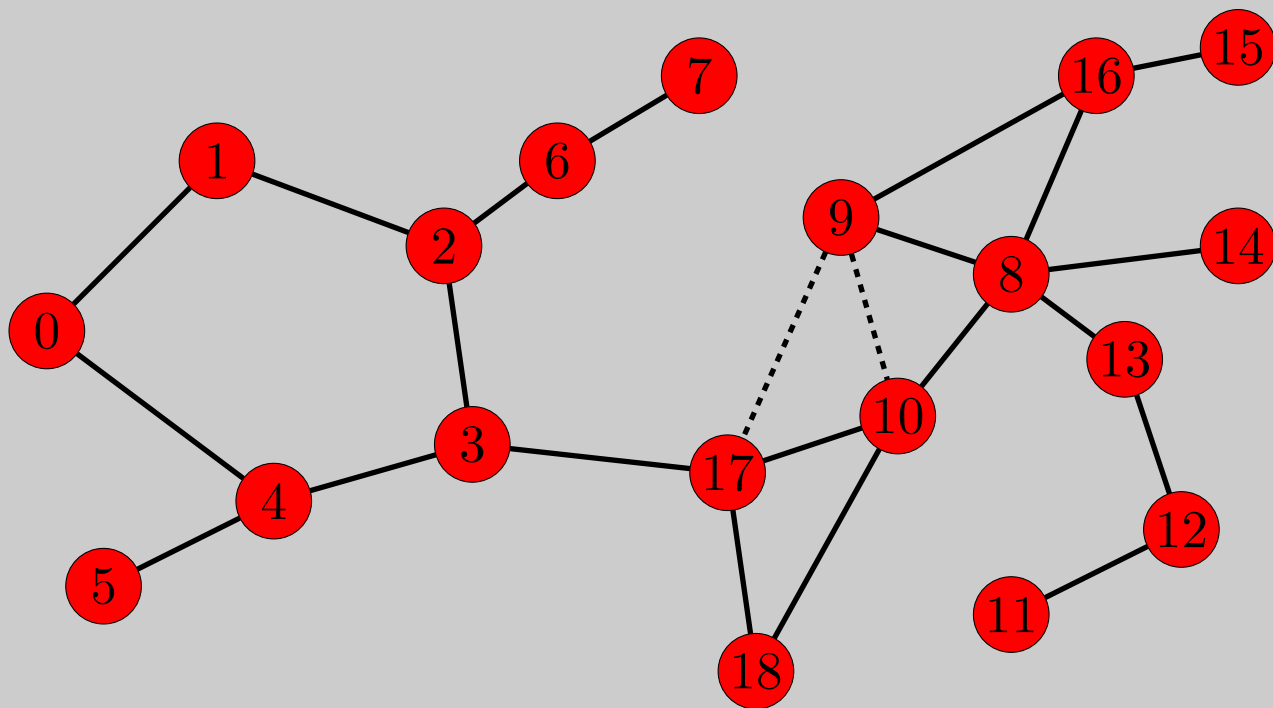


Pracovní zápis by byl 5(1), 2(1), 1(3).

Největší grafy

Flock No.	Skupiny: počet fraktálů (opakování)
00004	246(1), 9(1), 5(1), 4(1), 3(4), 2(8), 1(185)
00202	3194(1), 222(1), ..., 3(156), 2(181), 1(3394)
00243	377(1), 184(1), ..., 3(24), 2(58), 1(717)
00244	4089(1), 329(1), ..., 3(6), 2(17), 1(3949)
00245	806(1), 78(1), 4(1), 3(2), 2(1), 1(738)
00247	1648(1), 46(1), 3(3), 2(4), 1(1651)

Nalezení cyklů



Cykly by byly 0-1-2-3-4, 10-17-18 a 8-9-16.

OpenProblem: Situace se komplikuje při přidání hran.

Nalezení cyklů (pokr.)

- Startovní bod viz <https://www.geeksforgeeks.org/print-all-the-cycles-in-an-undirected-graph>
- Proveden pokus o ořez (s možností všech možných ořezů), ale přichází se o to nejužitečnější v grafu, tedy možnost si snadno navázat cestu dál.
- Není-li v grafu cyklus, algoritmy jsou známe (DFS z každého uzlu – prý to však také není ideál; Floyd-Warshall; Streaming). Víc Algorithms (4. vyd.), Wayne, Sedgewick. <https://stackoverflow.com/questions/12367801/finding-all-cycles-in-undirected-graphs>, <https://www.codeproject.com/Articles/1158232/Enumerating-All-Cycles-in-an-Undirected-Graph> a další zdroje.

Superrychlé řešení

- Krok 1: Užít DFS (Depth-First Search).
- Krok 2: Navázat nalezením nejdelší cesty v orientovaném grafu (DAG).
- Rychlé na naprogramování, ale mizerné najít nejdelší cestu.
- V Pythonu např. přes knihovnu `networkx`.

Nejdelší cesta

- Máme před sebou neorientovaný graf (původní video a přehrané pozpátku).
- Klasická rekurze se zacyklí. Je nutnost použít proměnnou na navštívené cesty.
- Nahlédneme na randomized depth-first traversal/search z <https://bost.ocks.org/mike/algorithms/>, abych objasnil svou úvahu.

Video algoritmu (bez)

resp.

Video algoritmu (s)

D3.js: replikace dat/grafiky

- **OpenProblem:** Vytažení dat / png / svg.
- Aplikace na stahování získávají tak 5 snímků/vteřinu.
- Nahrávání obrazovky jako video.
- Občas se podaří animaci zpomalit.
- `console.log`, nenašel jsem to však v `__transitions__`, jak to na fórech radí.
- NodeJS + knihovna FS.
- Uložení pozice myši: `xev`, `xsel`, `xdotool`, např.
`watch -ptn 0 "xdotool getmouselocation"`
- **OpenProblem:** Uložit si i stav tlačítek myši.

D3.js (řešení 1, pseudo)

```
<div id="pajovo">foo</div>
<script>
var x = new XMLSerializer();
function outputToConsole() {
  console.log(x.serializeToString(document.
    getElementById("pajovo")));
  console.log(document.getElementsByTagName
    ("div")[0].getAttribute("style"));
  if(n++ >= 100) return; // ulož 100 snímků a přestaň
  setTimeout(outputToConsole,33);
  // 30 snímků za vteřinu
}
outputToConsole();
```

D3.js (řešení 2, pseudo)

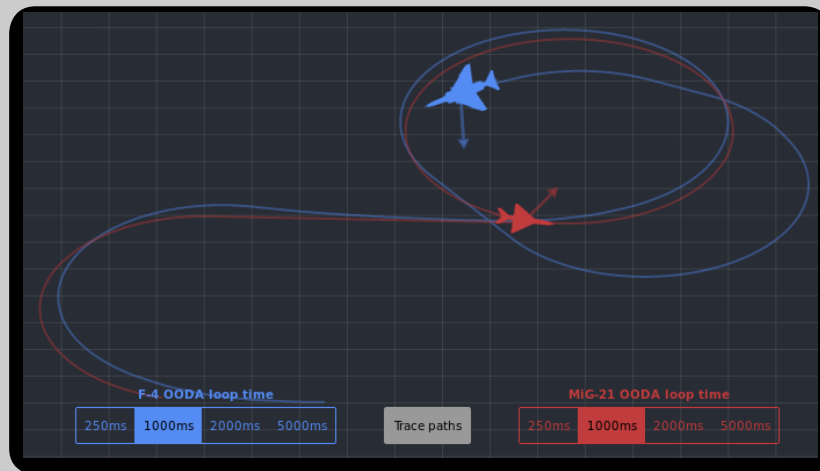
```
<canvas id="moje" width="60" height="30"></canvas>
<script>
var posledni="";
posledni=document.getElementById("moje")
    .toDataURL("image/png");
console.log(posledni);
```

Uložení do cele.txt, po řádcích do posledni.txt a Python:

```
import base64
newpng = open("posledni.txt","rb").read()
kam = open("posledni.png","wb")
kam.write(base64.decodestring(newpng))
kam.close()
```

Nefunguje vždy

- **OpenProblem:** Kde to funguje? Kde ne? Nefunguje mi to plně např. v OODA Loop od Bena Eatera, neuloží se letadla. Užity vrstvy přes react.js?
- Instalace: `npm install`, spuštění: `npm start`, produkční verze: `npm run build`.
- Zdroj: <https://github.com/beneater/ooda-loop>



uDnAG

- Výmysl: unDirected nonAcyclic Graph, hledání nejdelší cesty v neorientovaném grafu s cykly.
- Výmysl: a trimmed stack.
- Uzel, ze kterého jsem již šel, tak při cestě zpět jej už nikdy nenavštěvuj.
- Výhody: rychlá cesta zpět při velkém grafu; snadnější možnost pozastavení programu při užití typu stack (máme původní graf a tuto proměnnou); rychlá možnost si ukládat víc řešení.
- **OpenProblem:** Mohli bychom optimalizovat po částech/cyklech? Graf tak zjednodušit a pak optimalizovat celek?

Srovnání s rekurzí s pamětí

Nastaveny vždy 2 minuty.

Flock No.	Rekurze	Stack s ořezem	Rozdíl
00004	830	646	+184
00202	4868	4983	−115
00243	590	607	−17
00244	7853	8084	−231
00245	1313	1380	−67
00247	2398	2543	−145

Spojení AVI

- Nelze vzít soubory jak jsou; nelze mpv pouštět pozpátku a dopředu či po sobě; extended playlist EDL nepomohl.
- Lepší bude CLI než kdenlive, flowblade, ...
- ffmpeg (ideál libx265) přes -vf reverse.
- ffmpeg (-c copy) nezabralo; mencoder (neumí avi); mkvmerge z balíku mkvtoolnix (v34) max. 1024 souborů? Sekané.
- Do png a pak spojovat? Šílenství.
- Přes filtr huffyuv, ffv1 či qtrle? Obří velikost jako při volbě -crf 0 či -qp 0.
- Zabrala mi konverze do mpegu (-lossless 1), parametr je z formátu webm a -bufsize 4M -maxrate 30M.

Spojení AVI (pokr.)

- Plus `-loglevel error -hide_banner -n`.
- Spojování mpegů do jednoho pak přes `-c copy` (mkv).
- Plus `-fflags +genpts -y -safe 0`.
- Vizuální kontrola `mpv --pause`, případně tečka a čárka na posun po snímcích. Past: při malém rozlišení to zlobí. Zkusit `-codec png` (mov).
- U malých souborů ideál zobrazit s `--scale=nearest`. Viz github.com/mpv-player/mpv/wiki/Upscaling
- **ClosedProblem**: Srovnej dva obrázky. Např. OpenCV. Tip www.pyimagesearch.com/ a vznikající třídílná kniha Raspberry Pi for Computer Vision, osnova viz <https://www.pyimagesearch.com/2019/04/05/table-of-contents-raspberry-pi-for-computer-vision/>

Externí výpočetní síla

- `ssh`; `ssh -X`; `x11vnc`, `teamviewer`
- `sshpass`, ideál pro připojení bez hesla `ssh-copy-id` (profil bez passphrase, jinak řve keyring)
- `nohup <skript> &`
- Nebo po spuštění a zachování běhu po odlogování: `Ctrl+z`; `bg`; `disown -a`.
- Výpis pak přes `cat nohup.out | tail -n 3`. Zabití procesu obvykle přes `kill -9 (ps -es)`.
- Uspání notebooku přes `pm-suspend-hybrid`.
- Probuzení přes ethernet: `wakeonline/etherwake`. Stroj mám přes wifi+ethernet. Lze přes appky poslat signál.
- Užití `telnetd-ssl` nedoporučeno.

Stínové sshd

```
sudo cp /etc/systemd/system/sshd.service  
      /lib/systemd/system/sshd_paja.service  
sudo cp /usr/sbin/sshd /usr/sbin/sshd_paja  
sudo cp /etc/pam.d/sshd /etc/pam.d/sshd_paja  
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_paja
```

```
# Zásahy v sshd_paja.service  
# Description # ExecStartPre # ExecStart  
#   + v ExecStart přidat: -f /etc/ssh/sshd_config_paja  
# ExecReload # RuntimeDirectory # Alias
```

```
# Zásahy v sshd_config_paja  
# Port # GatewayPorts yes # ClientAliveInterval 60 # PidFile
```

```
sudo systemctl enable sshd_paja  
sudo systemctl start sshd_paja
```

Kopírování souborů

- `scp` (lze limit přes `-l`); `sshfs` s parametry `-o reconnect, ServerAliveInterval=1, ServerAliveCountMax=2`
- Příprava vlastní služby v Linuxu `systemd` vs. `init.d`.
- Bez omezení jsem si schodil domácí wifi.
- Hlídání sítě: `sudo iftop -i <wifi síť>`
- Test nakopírování souboru: `rsync -P -e ssh <co> <kam>;` omezení přes `--bwlimit`; umí rozbalování přes `Tab`.
- Obarvil jsem si `PROMPT` na různých strojích přes `PS1`.
- Po nakopírování: `md5`, `sha{1,224,256,384,512}`, `crc32`.
- Kompletnost videa: `ffmpeg s/bez -sseof -f null - 2>error.log`; `ffprobe` nebo `mediainfo`.
- Zaujal mě parametr `-err_detect ignore_err`, možnost spojit soubory i při poškození.

Spouštění

- mpv, vlc, ...
- Alternativa. Na spuštění nelze použít mpv *, potřebujeme `--playlist`, nevíme však celkovou spouštěnou délku, nelze snadno skákat v čase. Artefakt při přechodu videí tam však není.
- Na RPi3B mi nejel mpv ani vlc, jen omxplayer.
- **OpenProblem:** Lze rozjet mpv na RPi? Vypadá to, že ano.
<https://nwgat.ninja/installing-mpv-with-hardware-decoding-on-raspberry-pi-1-2-3-zero/>
- RPi4B ještě nemám, ale je už víc jak týden v prodeji. :-)

Zveřejnění videí

- Na uloz.to, viz <https://uloz.to/m/malipivo/fractals>. Mají manažer, lze přes Wine. Max. 19900 MB na soubor; 300 KB/s stahování bez poplatků. Soubory jsem rozdělil přes `split -b`; spojení přes `cat * > <video>`. Teoreticky lze klikání zvažovat přes `pyautogui` či `xdotool`.
- Dostal jsem to na archive.org. 86 GB mkv. Zkontrolovali na malware. Přidali mi během několika dní náhledy (jpg; thumbs), mp4, ogv, metadata (xml, sqlite), torrent a také kontrolní manifest (seznam souborů).
- Nahrnul jsem to přes webových prohlížeč, ale mají manažer `ia /ajej` nebo česky `íá/`. Viz archive.org/services/docs/api/internetarchive/
- Sledovatelné přímo na archive.org (JW Player 8.8.2).

Zveřejnění videí (pokr.)

- Zveřejněno 12. 6. 2019. Mám první lajk (~ 28. 6. 2019).
- **OpenProblem:** Pokud stahuji soubor po bajtech (nikoliv torrent), některé MP4 lze spustit, některé nespustím. Je důvod keyframe?
- Poděkování ve FB skupině The Official Electric Sheep / Scott Draves Group:
Bonus (Seamless "Flock" Videos from Paul Shearing)
(These are to be played with a video player,
not the ES client):
<https://archive.org/download/mega-collection-1>

Výstupní doktorská prohlídka

Flock No.	Velikost (GB)	Doba (h:m:s)	Počet AVI
00004	10,9	2:52:22	1544
00202	20,1	11:43:49	9746
00243	2,6	1:43:44	1167
00244	36,8	23:26:56	15839
00245	5,2	4:00:15	2703
00247	10,6	7:33:55	5046

OpenProblem: Lze získat 24 hodin v generaci číslo 00244? Nepodařilo se mi to, nejvíc jsem získal 23,5 hodiny a pár minut k tomu, ale už jsem to na webu neměnil.

Ukázka z konferenční videopozvánky
přechody

Konferenční video

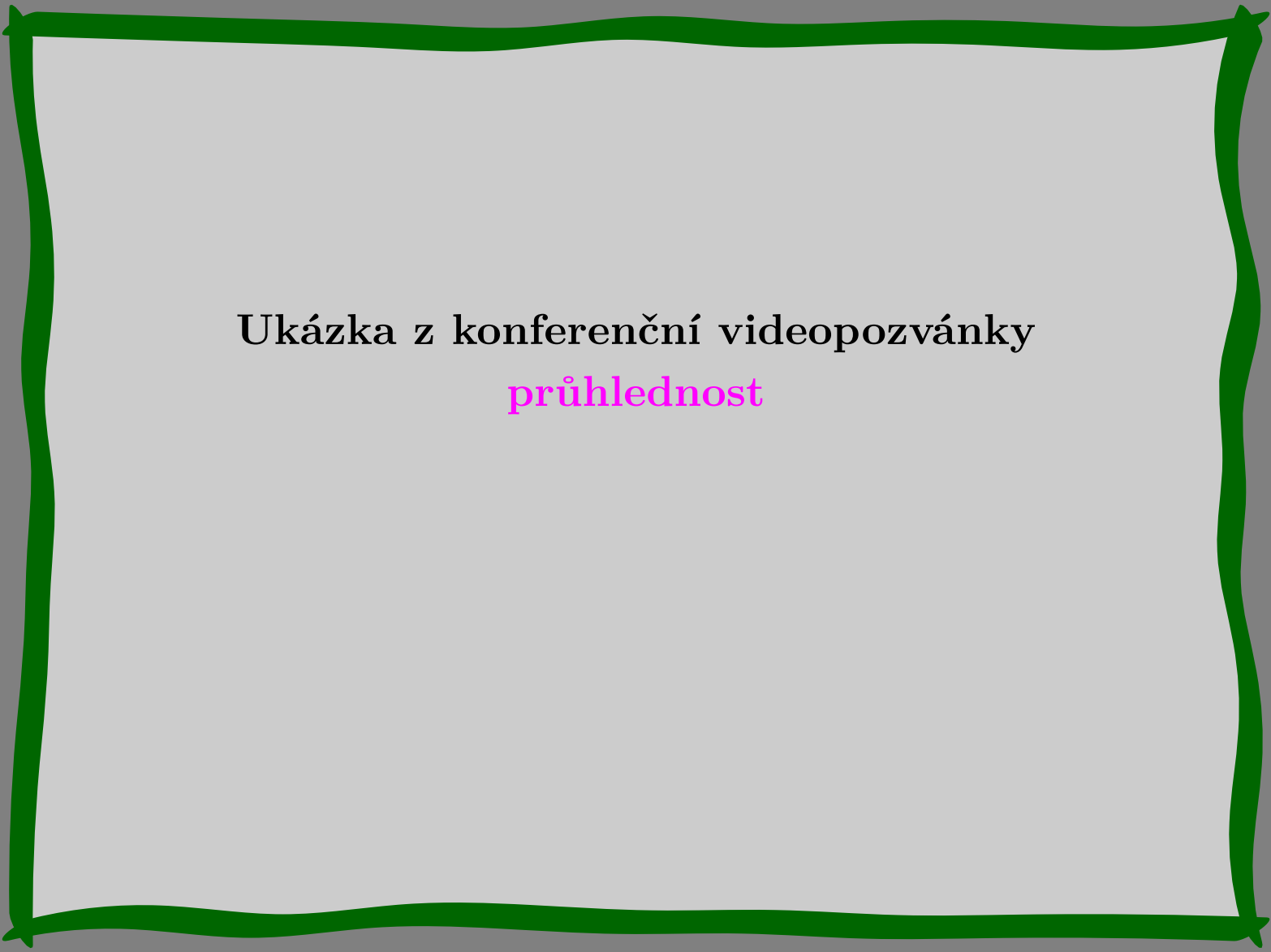
- **ClosedProblem:** Vrstvení. Umí `ffmpeg` přes filtry, ale nepoužil jsem to. Nápad byl odletěn do Koše.
- Tak jsme to spojili, teď to rozpojíme...
- `ffmpeg -y -i <video> snimek-%05d.png`, lze použít `-vframes`.
- Použil jsem `TEX` na slajdy, které jsem na úrovni snímků spojil s fraktálovým pozadím.

```
gs -r400x400 -dSAFER -dNOPROMPT -dNOPAUSE  
-dBATCH -sDEVICE=pngalpha -sOutputFile=  
vystupy0/vysledek-%02d.png ../podklady.pdf
```

Konferenční video (pokr.)

- Přechody jsem připravoval ze dvou obrázků a `convert -morph <počet snímků>`.
- Klíčové je, že zachovává průhlednost.
- Lze využít na přechod barevné stránky do černobílé ap.
- Častokrát je potřeba barvu dostat na průhlednost (ImageMagick, GraphicsMagick, GIMP).
- Na prodloužení časovač či obrázek sám do sebe.
- Na začátku a konci videa je přechod z a do plně černého obrázku.
- Vytažení konkrétní stránky z pdf lze přes `gs` (GhostScript) a hranatou závorku za názvem souboru.

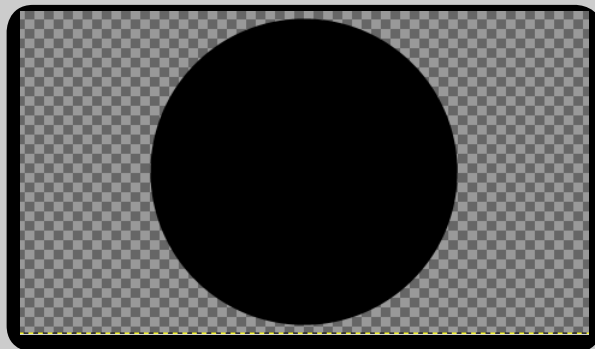
- Bashem jsem řídil nekonečný cyklus (rotace tučňáka) a zařazení snímků.
- Na kompozici jsem vrstvil png (opakovaně):
`composite <horní vrstva> <pozadí>`
`-gravity northwest -compose over <výstup>`
- Spojení přes `ffmpeg -f concat -i <seznam souborů>`.
- Struktura seznamu souborů je `file 'soubor'` na každém řádku. Je tak dané pořadí a soubory se mohou opakovat. Uvozovky tam být nemusí, ale je to jistota na soubory s mezerou v názvu.



Ukázka z konferenční videopozvánky
průhlednost

Průhlednost: příprava masky

- **Ukázka:** video Měsíce, vedle sebe s/bez průhlednosti.
- U takové ukázky může být problém synchronizace videí.
- Maska vytvořena v GIMPu. Načten libovolný snímek z videa a odměřen kruh z toho.
- Užití masky: `composite -compose dst-in <maska> <vstup> -matte <výsledek>`
- Test videa: `mpv <video> --alpha=yes --no-border --geometry=209x216+100%+100%`



Kresba průhledností

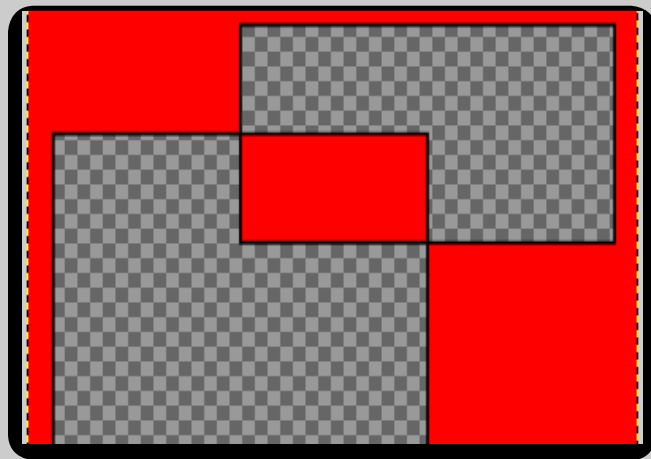
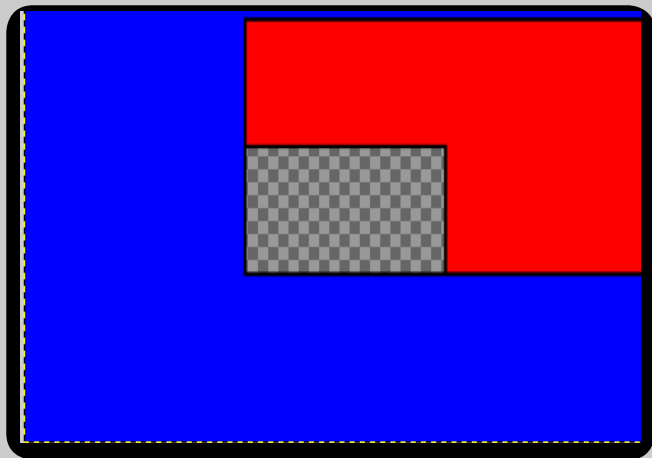
- Lze například v Metapostu, příkaz `unfill`.
- Já hvězdičky řešil v GIMPu (png+alfa kanál) a výsledek načten zpět do T_EXu.
- Lze užít jinak nepoužitou barvu a GraphicsMagick.
- Pokus přes TikZ (preambule dokumentu):

```
\documentclass[a4paper]{article}
\pagestyle{empty}
\usepackage{tikz}
\usetikzlibrary{shapes.geometric}
\tikzstyle{reverseclip}=[insert path=
  {(current page.north east) -- (current page.south east) --
    (current page.south west) -- (current page.north west) --
    (current page.north east)} ]
```

Kresba průhledností (pokr.)

Tělo dokumentu:

```
\begin{document}
\begin{tikzpicture}[remember picture, overlay]
\path[fill=red, draw, reverseclip]
  (0,0)--(2,0)--(2,1.5)--(0,1.5)--cycle;
\path[fill=blue, draw, reverseclip]
  (1,1)--(3,1)--(3,2)--(1,2)--cycle;
\end{tikzpicture}
\newpage
\begin{tikzpicture}[remember picture, overlay]
\path[fill=red, draw, reverseclip]
  (0,0)--(2,0)--(2,1.5)--(0,1.5)--cycle
  (1,1)--(3,1)--(3,2)--(1,2)--cycle;
\end{tikzpicture}
\end{document}
```



Rezervy

- Na úrovni znaků toto řešení v TikZu nefunguje. Musíme na to jinak či přes Metapost (na tohle zlobí `luamplib`).
- **OpenProblem:** Umíme v $\text{\TeX}$ u mazat vysázené? Funkce guma chybí. Vyřešilo by mnohé.
- **OpenProblem:** Jak dostat kresbu písma do TikZu? Nabízí se konverze (FontForge) a načtení přes svg. První pokusy proběhly. Hans Hagen dále prohlubuje Metapost s $\text{\TeX}$ em.





Můj divný dokument

Vysázel jsem bílou stranu, nikoliv přes `\pagecolor{white}`.
Následně do ní vyřezával objekty. Je to dobře vidět?

Můj divný dokument

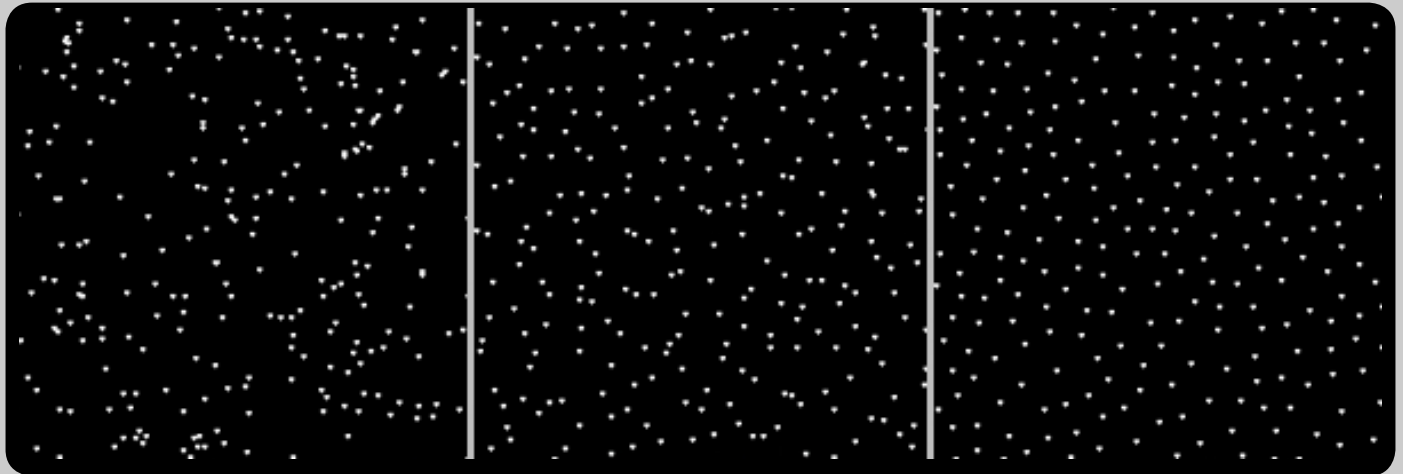
Vysázel jsem bílou stranu, nikoliv přes `\pagecolor{white}`.
Následně do ní vyřezával objekty. Je to dobře vidět?



Ukázka z konferenční videopozvánky
náhodnost

Pseudonáhodná čísla

- Nejsou vždy ideální pro přípravu podkladu, kdy nám jde o zachování proporcí.
- Ukazuje se, že lepší je přístup fyzikální (styl Voronoi) či z počítačové grafiky (procedurální textury).
- Zleva: pseudonáhodná čísla (random), náhodné umístění do mřížky (jitter) a Poisson Disc Sampling:



Poisson Disc Sampling

- Zkoumá vztahy, podobně jako algoritmus SOM z umělých neuronových sítí.
- Čtivě na <http://devmag.org.za/2009/05/03/poisson-disk-sampling/>
- Mé pokusy o sazbu do spirály přes Lua shrnuji v odpovědi na tex.stackexchange.com/questions/197389/how-to-fill-random-spaces-with-random-circles-in-tikz
- Zájemci nechtě hledají termíny Voronoi / Cellular Noise; Perlin / Simple Noise / Moiré Patterns.
- Poisson Disk Sampling byl pro T_EX zpracován, viz tex.stackexchange.com/questions/185419/poisson-disk-sampling-implemented-in-luatex/

Vzorek ze svatby

- Ukázka z nedávné svatby. Písmo je cíleně vyplněno kolečky.
- Odhaduji, že je to řešeno na úrovni písma jako u předchozí ukázky.
- Pravá nožička písmene R je podezřelá (je tam méně koleček a další jim tam už nevešlo).
- Je tam zajímavé zaoblení všech ostrých úhlů. Vymazlené!





Létání znaků: nahoru dolů

OpenProblem: Sazba na horní linku (styl hindštiny), znaky vizuálně padají (zavěšené prádlo). Jak na to technicky, jaké písmo, jak řešit háčky a čárky, jak verzálky? Možné použití?



TEXOVISE
2019

Létání znaků: doleva doprava

OpenProblem: Algoritmus na automatický kerning: FontForge umí, ale stále je co zlepšovat. Zde je ruční napojování znaků BDSM black latex ze serveru www.dreamstime.com.



Ukázka z konferenční videopozvánky
zakrývání

Rozhození log distribucí

- Velikost stránky jsem si dal poměrně k budoucímu videu, např. u 1920×1080 pixelů, bych si dal $19,2 \times 10,8$ cm.
- Přiskakují a pak ubývají. Sázím věci z minulé strany + nové logo. Při ubývání odebírám. Dvojitý cyklus `\foreach`.
- Rozdělil jsem si obrazovku a rozházel loga do mřížky. Je to předvolené, ale poměrně náhodné.
- Postupné mizení jsem proházel řádky v TikZu:

```
\foreach \obrazek/\x/\y in {%  
  redhat/7/0,scientific/5/2,steamos/2/0,kali/1/1,  
    linuxcnc/3/2,mageia/4/0,manjaro/6/1,%  
  opensuse/4/2,parrot/6/0,poseidon/3/0,osgeo/5/1,  
    mint/2/1,ubuntu/7/2,elementary/1/2,%  
  gentoo/7/1,fedora/4/1,arch/5/0,biolinux/1/0,  
    cae/3/1,centos/2/2,debian/6/2} {...}
```

Jak na to jinak?

- Zde je **inspirace** z kanadského seriálu Mayday.
- **Ukázka** z Blenderu.
- Scale na 0 (nepoužíval jsem) není násobení a zmizení.
- Nastavení průhlednosti: Render; v Output nastavit RGBA a v záložce Film nastavit Alpha na Transparent.
- Lze renderovat přes **příkazový řádek**.

```
blender -b tux.blend -f 10 -F PNG
```

```
blender -b tux.blend -o /tmp/tux/tux_### -a -t 4
```

- Renderovací farmy, např. **SheepIt**. V červnu 2019 ohlásili 100 milionů obrázků odpovídající tisíc rokům výpočetního času. **<https://www.sheepit-renderfarm.com/>**

3D Tux na Facebook

- 3D tučňák/tux je ze serveru **www.sketchfab.com**.
Zahlédl jsem poprvé formát GLTF. Binární formě odpovídá přípona GLB.
- Nahodil jsem Blender 2.80 vedle 2.79b.
- **Podrobný návod**
- Import GLTF na první dobrou v pořádku, export nikoliv. Blender si bral Pythona ode mne. U mne visel balíček numpy v Python2, odinstaloval jsem jej.
- Zůstaly tam artefakty u očí a zobáku, ale mělo by to být řešitelné. První vzorek zveřejněn!

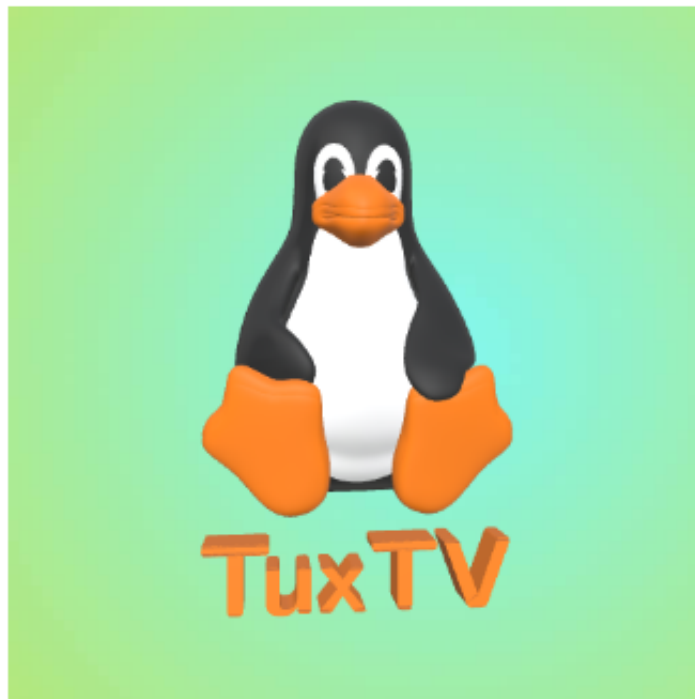


Paul Shearing přidal 3D příspěvek.

...

21. června v 13:06 · 🌐

Pokus s GLB/GLTF... Jsou tam nějaké artefakty, ale Čert je vem... 😊



To se mi líbí



Okomentovat



Sdílet

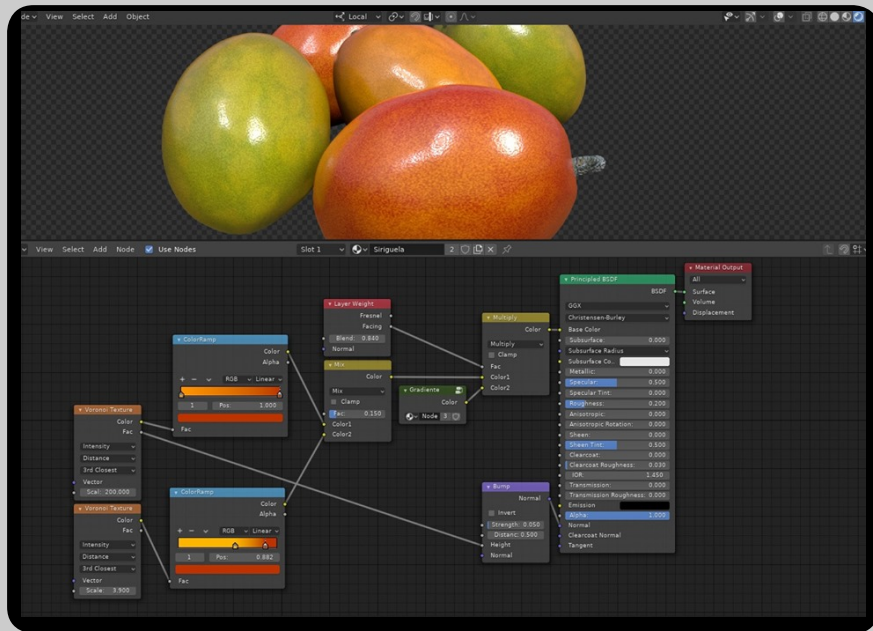


Napište komentář...



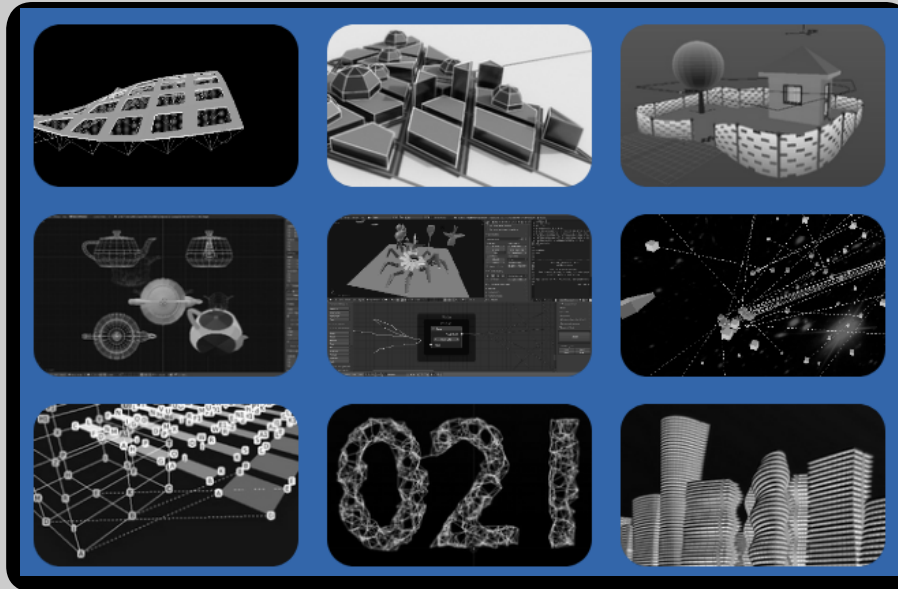
Postřehy

- Add-on FX Text na 2D animování.
<http://codeofart.com/text-fx/>
- Vizuální programování (Nodes). Procedurální textury. Lze uložit strukturu do JSON a tu editovat mimo Blender.



Add-on Sverchok (SV)

- /Svječok/ = MATH + MESH + SPLINES + NURBS
- Tvorba modelů parametricky, založeno na Nodes.
- <https://github.com/nortikin/sverchok>
- http://nikitron.cc.ua/sverchok_en.html



Add-on Animation Nodes (AN)

- Mezník pro animování v Blenderu.
- Například texty lze měnit a animovat je. Býval to problém.
- github.com/JacquesLucke/animation_nodes
- <https://animation-nodes-manual.readthedocs.io/>
- Základní tutoriály, Zach Hixson na **YouTube**.
- Pokročilé tutoriály, Chris Prenninger na **YouTube**.
- Tzv. **showreel**. Tento je z roku 2017.

Klasici/YouTubeři Blenderu

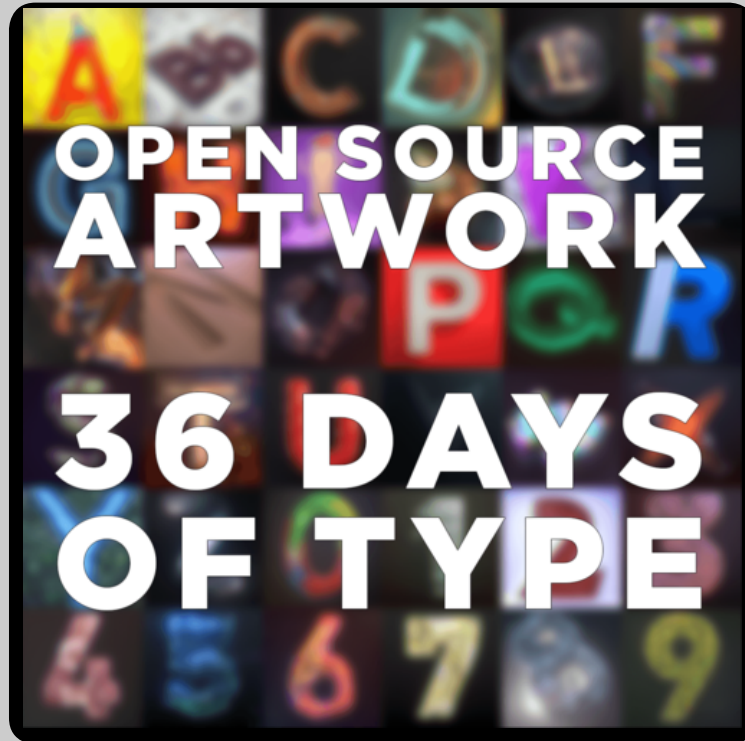
- Australan Andrew Price a.k.a. **Blender Guru**: The Cycles **Shader** a **Modifier** a **Input** Encyclopedia, videotutoriály (viz **thepiratebay.org**), rozhovory s Blenderisty a tvůrci z 3D grafiky, vyhlašovatel soutěží, blog, podcasty a další. Kniha **The Wow Factor** z něj udělala milionáře (v USD).
- Bělorus Gleb Alexandrov a.k.a. **CreativeShrimp**. Tvorba videotutoriálů typu realistické osvětlení, vesmírné VFX; výuková DVD pro pokročilé.
- Lotyš Alan Melikdjanian a.k.a. **Captain Disillusion**. Užití Blenderu v podobě VFX efektů. 1,5+ milionů odběratelů, 100+ milionů shlédnutí. Na Patreonu má cíl 10 tisíc USD za měsíc. Plní. Na **Shorty Awards** vyhrává v kategorii Best in Weird (volně: Nejlepší podivnost) za rok 2019.

Midge „Mantissa“ Sinnaeve

- Pokročilá práce s modifikátory v Blenderu.
- Na **YouTube** má návody v podobě seriálu Weird Sh!t, Workflow Logs a Live Streams. Nenechte se odradit ne zrovna slušnými popisky.
- Na **<https://mantissa.xyz>** má ke stažení VJ Loops, 8K textury, 3D modely (stromy, ovoce), RandoMesh add-on a tento druh materiálu.
- Jeho tzv. **demo reel**. Tento je z roku 2018.

Midge Sinnaeve (pokr.)

- 36 Days of Type (2016, 2017, 2018). Vše stáhnutelné.
- Ukázky znaků A až Z a cifer 0 až 9. Jakoby portfolio.





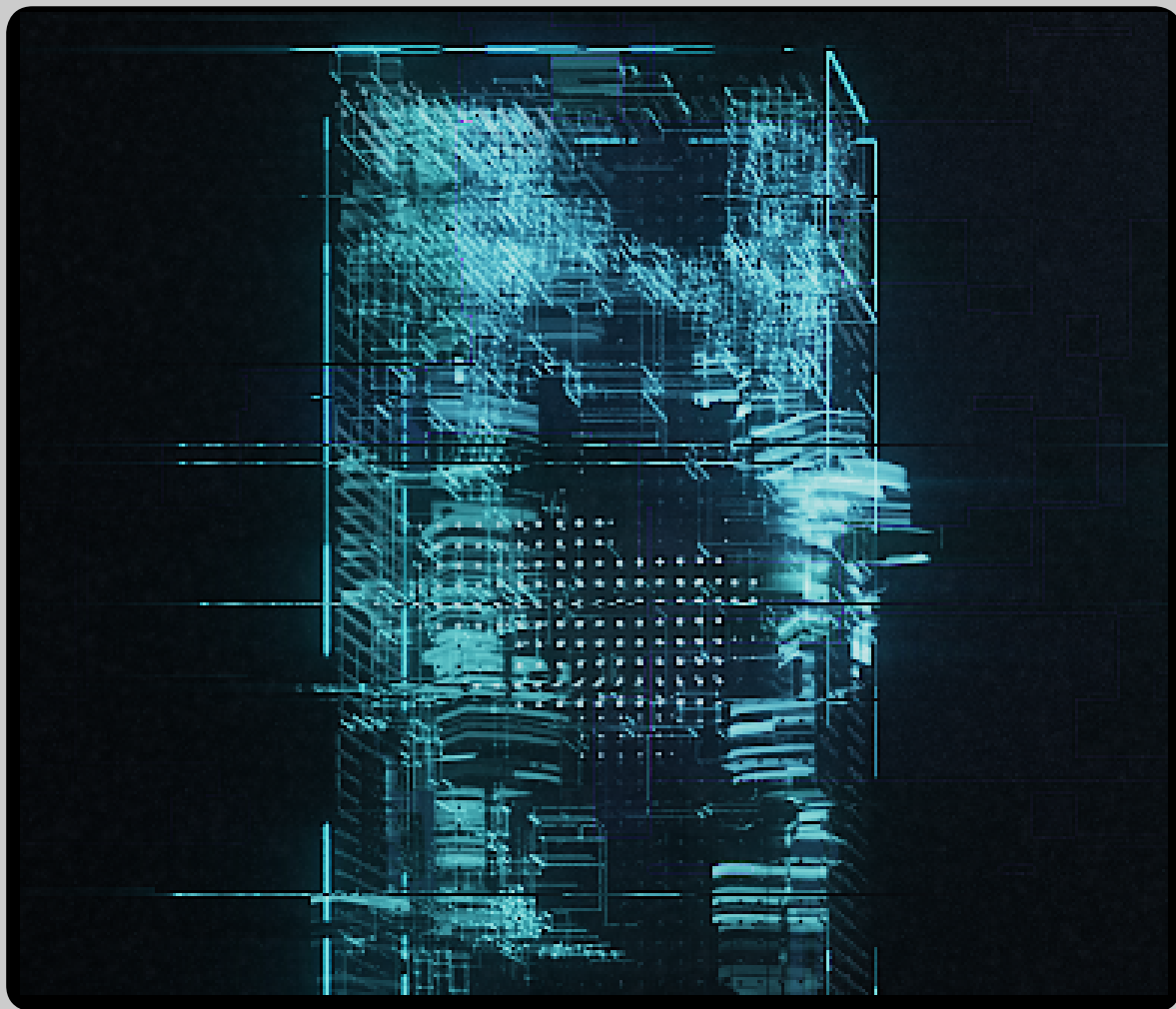
36 Days of Type 2017
by Midge Sinnaeve

All files available as a free
download.

themantissa.net

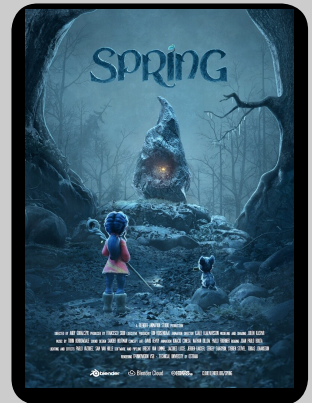






Zaznamenal jsem...

- Starší kurz **Blenderella** (2011): Aneta Šmrhová a.k.a. Anetta Keys, česká pornoherečka, jim stála modelem. Na DVD je zdokumentováno. 35 let / 165 cm / 50 kg / 88-59-88.
- Film **Next Gen** a.k.a. 未来机器城 (Čína/Kanada, 2018) je první komerční film vytvořený jen v Blenderu.
- U krátkého filmu **Spring** (2019) je v závěrečných titulcích zmíněna renderovací podpora VŠB-TU Ostrava.

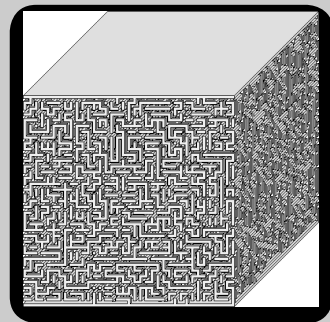
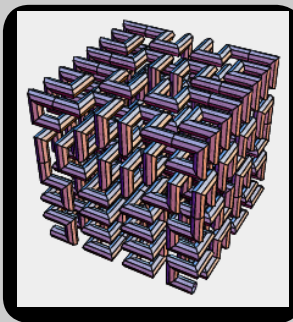
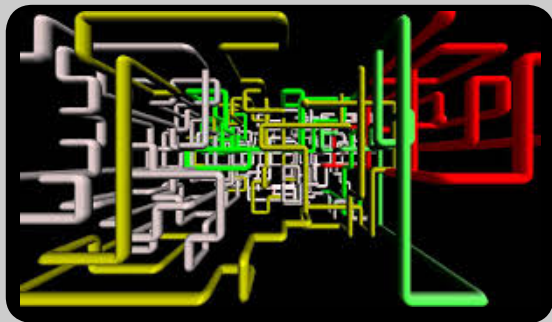


BlenderCloud

- <https://cloud.blender.org/>
- Členský přístup za 15 eur/měsíc až po 109 eur/rok.
- Stáhnutelné produkční věci ke krátkým filmům, obrázky v HDR, textury, workshopy, kurzy ap.
- Přístup v Blenderu přes Blender Cloud add-on.
- Zaujal mě Blender Benchmark (obdoba **torture testu** v T_EXu). Vyrenderoval jsem první obrázek (cca 2,5 hodiny) a testy na své šunce ukončil.
- Mám zakoupen přístup do BlenderCloudu, kdyby někdo něco potřeboval stáhnout (po dobu OSSConf 2019, možná i poté: to je však bez slibu)... Dejte prosím vědět!

OpenProblem: Running Pipe

- Jedna trubka, zaplnit viděný prostor, **bez průniků**.
- Vy proti Linuxu nic nemáte, že? Tak bych ze spořiče obrazovky xscreensaver užil **Pipes** (obr. vlevo).
- `xscreensaver-command -demo 39`
- Je to trochu podobné Hilbertově křivce (obr. prostřední).
- 2D/3D; rovná/křivící se; ne moc náhodná, ne moc bludiště (obr. vpravo; na generování a řešení bludišť **Daedalus**).
- Při křížení: uhne nová část či se posune již položená?



ClosedProblem: Running Pipe

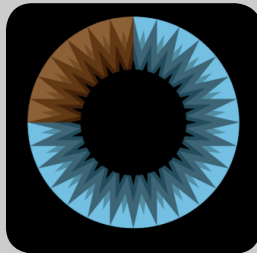
- Vyřešil jsem v Blenderu. Vzdálenost nového bodu nesmí být blíží k úsečkám než poloměr kolena/potrubic. Zároveň musí být dál než tento poloměr. Podobně nová úsečka nesmí být blíží k již položeným.
- Dostudovával jsem kvaterniony (prvně je popsal W.R. Hamilton, 1843), v té době v manuálu Blenderu byly vysvětleny, ale hledal jsem detaily.
- Je to výhodné pro transformace 3D objektů.
- Jedná se o rozšíření komplexních čísel:

$$i^2 = j^2 = k^2 = ijk = -1,$$

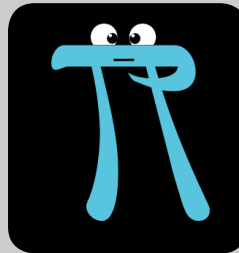
$$ij = k, \quad ji = -k, \quad jk = i, \quad kj = -i, \quad ki = j, \quad ik = -j.$$

Přednášky

- **Zaujala** mě youtube.com/watch?v=d4EgbgTm0Bg
- A zkoumal jsem i další videa od tvůrců. **Who cares about topology?** (Inscribed rectangle problem)
- **Sneaky Topology: The Borsuk-Ulam theorem and stolen necklaces**
- **Solving 2D equations using color, a story of winding numbers and composition**
- **Divergence and curl: The language of Maxwell's equations, fluid flow, and more**



Logo tvůrce 3b1b



Píthon = π + Python

3Blue1Brown or 3b1b

- Videa tvoří Grant Sanderson, **3blue1brown.com**.
- Technicky zajišťuje Ben Eater, **<https://eater.net>**.
- Detaily jsou na **<https://eater.net/quaternions>**.
Na tomto webu mě zaujal tento dotaz s odpovědí.



Axel • 8 months ago

Those explorable videos are revolutionary! What software / library was used to do this?

26 ^ | v • Reply • Share >



Ben Eater **Mod** ➔ Axel • 8 months ago

Since there wasn't really anything like this yet, we built it all ourselves from a variety of existing web tools: WebGL (using threejs plus some custom shaders) for 3d stuff, raw canvas for the 2d stuff, howlerjs for handling the audio playback, and lots of React for the UI and to glue it all together. It's very much a bespoke app. In the future, I hope we'll build more of these and as we do so, the tech will evolve to something more easily generalized.

59 ^ | v • Reply • Share >

Diskuze tamtéž

- Klíčová slova: WebGL, three.js, canvas, howler.js, react.js.
- Výčet by se dal obohatit o d3.js a babylon.js, který má zajímavý sandbox na prohlížení GLTF.

A hned pod tím bylo napsáno: „[...] Will/has [REDACTED] (Grants home grown Math Animation tool (available on github)) [...]“

Diskuze tamtéž

- Klíčová slova: WebGL, three.js, canvas, howler.js, react.js.
- Výčet by se dal obohatit o d3.js a babylon.js, který má zajímavý sandbox na prohlížení GLTF.

A hned pod tím bylo napsáno: „[...] Will/has [REDACTED] (Grants home grown Math Animation tool (available on github)) [...]“

Název nástroje je...

Fanfáry prosím!



manim

Manim

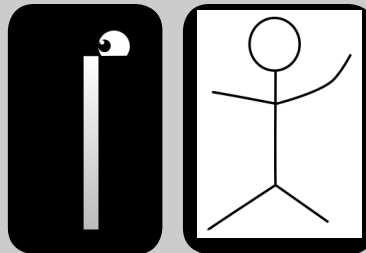
- Malý v názvu, ale velký v potenciálu.
- Velké spojení Pythonu a T_EXu v možnostech.
- Matematické animace.
- Popisky v duchu Asymptote.
- V pip hledejte pod manimlib.
- Poslední verze na <https://github.com/3b1b/manim>
- Našel jsem dřívější kurz za 2 kredity u **Mirka Olšáka**.
- Umí živé promítání přes **twitch.tv**.
- Existuje **manim.js**.
- Autoři rozšířili možnosti přes **react.js**. **Ukázka**.

Manim: Užítí

- Nejelo mi ze začátku nic! Vůbec nic! Kolikrát se vám to v životě natrefí? Ale odhodlání to rozjet bylo veliké.
- Viselo mi na stroji `pycairo`, obešel jsem `sudo pip3 install manimlib --ignore-installed pycairo`
- `python3 <muj.py>` nejel, musí se přes řídicí soubor, `python3 -m manim <muj.py>` nebo `python3 ./manim.py <muj.py>`.
- Náповěda přes `python3 -m manim --help`.
- Ukázky jsou přes třídy, řešil jsem přes `cat <soubor.py> | grep <třída či class>`. Obvykle vidíme `Scene`, `GraphScene`, `ThreeDScene`, `SVGMOject` ad.
- **Hello World!** (zpomaleno 4×) je `python3 ./manim.py example_scenes.py SquareToCircle -pl`

Manim: Spuštění

- `python3 -m manim <muj.py> <třídy...> <parametry>`
- Třídy oddělujeme mezerami.
- Bez zadání třídy vyběhne nabídka a zadáváme čísla oddělená čárkou. Ne všechny třídy jsou takto spustitelné.
- Hodila se mi rychlovka na test víc věcí `echo {1..10} | sed "s/ /,/g"`
- Můj nejoblíbenější příkaz byl `killall python3`, poněvadž občas jsou ukázky s nekonečným cyklem.
- Vedle PiCreature má autor k ruce ještě Linuse a StickMana.



Manim: Vybrané parametry

- `-p` = preview, otevření souboru po dokončení
- `-l` | `-m` | bez parametru = low | medium | high
- `-t` = transparent, pozadí bude průhledné
- `-c` = color, pozadí bude mít specifickou barvu
- `-r` = resolution, rozlišení v pixelech, výška čárka délka
- `-s` = save the last frame, uloží jen poslední snímek
- `--livestream` | `--to-twitch` = živé vysílání
- `-h` | `--help` = nápověda

Výsledky se ukládají do adresáře `media/video`, pak do složky dle názvu skriptu, následuje složka výška p rychlost, např. `480p15` znamená 480 pixelů je výška obrazu v rychlosti 15 snímků/vteřinu.

Manim: Inspirativní zdroje

- Proklepnutí manimu: **Getting Started Animating with manim and Python 3.7** + podpůrné soubory na https://github.com/zimmermant/manim_tutorial
- Autor Solara5570: **Demo-solara** Nerozjel jsem skoro nic, o to víc láká se k tomu vrátit.
- Autor Theorem of Beethoven: **Animation course with Manim**, podpořeno **videi na YouTube**:
 - Návod na vlastní figurku do videí, řekněme, Omegáčka, „Number Creature“,
 - vychytávky ve složce **extras**: leave progress bars by default, tex formulas, a hlavně,
 - **render old projects**. Spouštím `python3 -m manim.py old_projects/<soubor.py> <další parametry>`

Poklad vedle active_projects

- Autor skvěle zachází s Pythonem, když už se projekt nepodaří rozjet, stojí za projití zdrojového kódu.

```
malipivo@hp:~/.../manim/old_projects$ ls
256.py                fourier.py            pi_day.py
alt_calc.py           fractal_charm.py      playground_counting_in_binary.py
basel                 fractal_dimension.py  putnam.py
bell.py               generate_logo.py      pythagorean_proof.py
borsuk_addition.py    gradient.py           qa_round_two.py
borsuk.py             hanoi.py             quat3d.py
brachistochrone       highD.py              quaternions.py
cba.py                hilbert               sphere_area.py
clacks                inventing_math_images.py tattoo.py
complex_multiplication_article.py inventing_math.py     tau_poem.py
counting_in_binary.py leibniz.py            three_dimensions.py
crypto.py             lost_lecture.py       triangle_of_power
dandelin.py           matrix_as_transform_2d.py triples.py
div_curl.py           moser_intro.py        turbulence.py
domino_play.py        moser_main.py         uncertainty.py
dominos              mug.py               wallis.py
efvgt.py              music_and_measure.py  waves.py
eoc                  mvcr.py              wcat.py
eola                  name_animation.py     WindingNumber_G.py
eulers_characteristic_formula.py nn                    WindingNumber.py
fcl.py               number_line_scene.py  zeta.py
for_flammy.py        patreon.py
malipivo@hp:~/.../manim/old_projects$
```

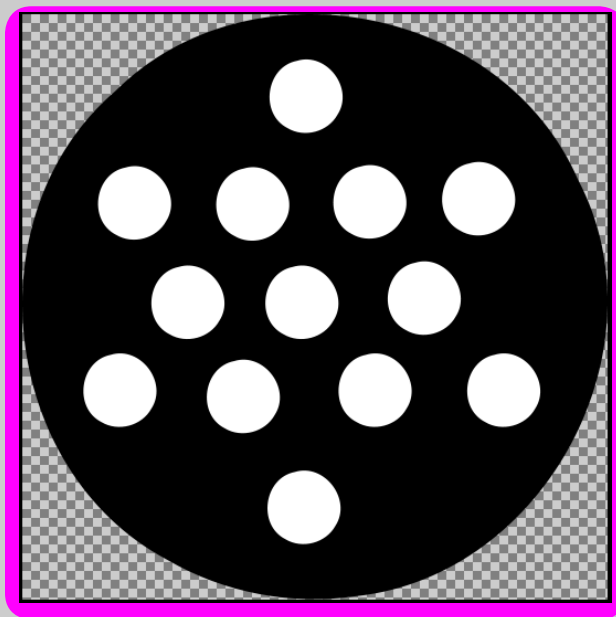
Manim: Proč projekty nejedou?

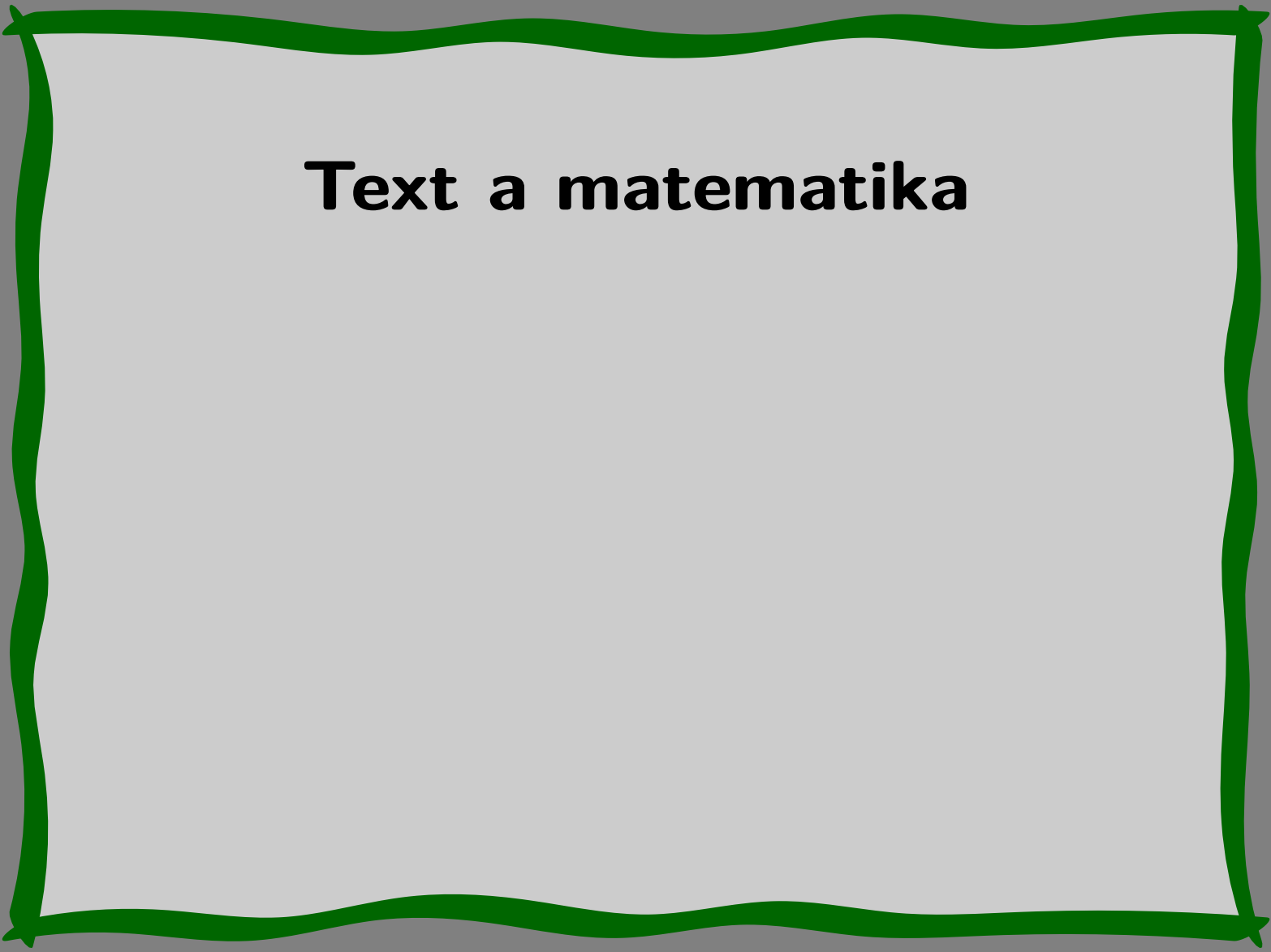
- Manim hodně vyvíjí (2018/2019). Některé projekty, hlavně skrz přepracovaný kód u animací, neběží.
- V kódu `from big_ol_pile_of_manim_imports import *` je potřeba přepsat na `from manimlib.imports import *`, tipuji pozůstatek po přechodu z Python2 na Python3.
- `manim.py` je řídicí soubor, spouštění přímo z `old_projects` nejede, tam takový soubor není.
- Některé funkce doznaly změn, přepisoval jsem parametry `("()()")` na `("(",")", "(","")`. Např. v `quat3d.py`, scéna `ComplexNumberRotation`.
- Občas kód proběhl, ale nespojil dílčí videa do jednoho souboru. Použil jsem `mpv *` na rychlé přehrání či `ffmpeg` na trvalé spojení. Srovnat s jiným kódem a přepsat zdroják?

Python se má rád, je to tam jedno selfie za druhým.
Hemží se to tam samým `self`, `*self`, `**self`, ...

Manim: Vzorkovník alias Manimovník

Pozn. Tohle je ovoce Manimovníku (chutná po tuši,
ideální nestandardní značka na zakončení důkazu)...



A thick, dark green border with a wavy, hand-drawn appearance surrounds the central text. The border is composed of four segments, one for each side of the page, meeting at the corners.

Text a matematika

Grafika ve 2D

Základní 3D grafika

Pokročilá 3D grafika

Pájo, zpátky ke stroji!

Manim: Vzorkovník

- Za programátorskou sekci proběhly Hanojské věže.
- Za GIS sekci proběhly EulerAngleAndGimbal, ...
- Za učitelskou sekci SolveGeneralQuadraticEquation, ...
- Za typografii texty ve 3D. A co se mi připletlo do cesty...
- Nyní lze na dálku videa spouštět a časovat (`bash`, `kill`).
- **OpenProblem:** **ClosedProblem:** Nepřekrývající se okna/obdélníky/části (popisky v Měsíci). Algoritmus kontroluje obdélník nad/pod a vlevo/vpravo od druhého. Více viz **Find if two rectangles overlap.**



Otázky sám sobě

- Nebude zlobit diakritika, lze zasáhnout do preambule?
- Lze přepnout pdf $\text{\TeX}$ na Lua $\text{\TeX}$ či jiný formát?
- Pojede japonština a podobné východní jazyky?
- Lze získat vektorový výstup daného snímku (SVG)?
- Je možnost (de)aktivovat antialiasing?
- Půjde zařadit grafiku z TikZu, Metapostu či PSTricks?
- Můžeme užít grafiku z matplotlib (např. `networkx`)?
- Lze zařadit video z jiného zdroje?
- Dá se pohybovat po spirále?

Některé odpovědi sám sobě

- Dle návodů jsem do videa postupně dostal text, rovnice, 2D i 3D animace, zvuk a svg obrázek.
- **Ukaž!** Zkusil jsem si vlastní text a transformace.
- **Ukaž!** Po odklonění `manimlib/tex_template.tex` se aktivovala šablona `manimlib/ctex_template.tex` se XeTeXem. Zásahem do `utils/tex_file_writing.py` se lze přepnout kamkoliv. Vyzkoušel jsem takto CJKV.
- **Ukaž!** TikZ se mi podařil do animace dostat, ale jen křivky, nikoliv sílu linek a barvy se vyply. Nastavil jsem až v Manimu. Možná zkusit cestu přes pdf do svg (`pstoedit`, `inkscape`)? Manim užívá `dvisvgm`.
- **Ukaž!** Na závěr jsem zkusil transformaci rovnice. Něco pro novou generaci. Následuje zdrojový kód.

Ukázka kódu v Manimu

```
from manimlib.imports import *
class Rovnice(Scene):
    def construct(self):
        sum1=TextMobject(r"$$\sum_{i=0}^n{n\choose i}
            a^{n-i}b^i = (a+b)^n$$")
        sum1.scale(2)
        sum1.to_edge(UP)
        sum1.set_color(BLACK)
        self.play(Write(sum1),run_time=3)
        self.wait(1)
        sum2=TextMobject(r"$$\mathit{witch*}_{i=0}^n{n\choose i}
            \mathit{rightghost}^{n-i}\mathit{leftghost}^i\mathit{rightwitchonbroom*}\{
            (\mathit{rightghost}+\mathit{leftghost})^n$$")
        sum2.scale(1.75)
        sum2.to_edge(DOWN)
        sum2.set_color(BLACK)
        self.play(ReplacementTransform(sum1.copy(),sum2),run_time=6)
        self.wait(4)
```

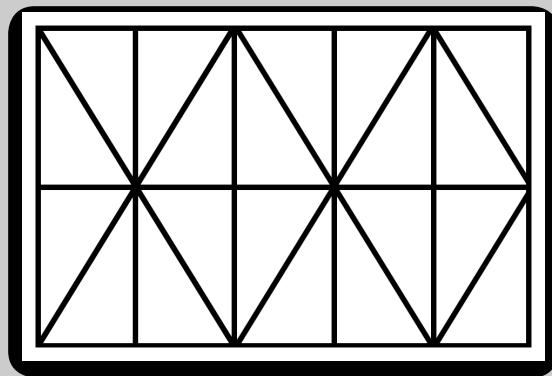
Na pokusy

- **OpenProblem:** Na jedné straně obrazovky by to chtělo při spuštění videa vidět zdrojový kód bloku v Pythonu.
- **OpenProblem:** Jak připravit aktuální čas či zbývající čas přednášky (digitální hodiny)? JavaScript a v okně bez rámu? Lze zobrazit webovou stránku s průhledným pozadím (občas je to vidět u terminálu)? Užít Manim, C++ či Python s Qt5?
- **OpenProblem:** Rád bych analogové hodiny 150×150 pixelů, bez rámu okna, s průhledností. Nápad je zkusit video, kdy jeden snímek bude jedna vteřina (obří soubor). Nebo lépe 4 vrstvy: základ hodin (1 snímek) a ručičky pro hodiny (12 snímků), minuty a vteřiny (60 snímků).
- **OpenProblem:** Je možné mít v Linuxu neobdélníkové okno? Např. kruhové, eliptické, ale obecně jakékoliv dle masky?

Závěrem odpočinková matematika

ClosedProblem: Ve významu tečky/trojúhelníku na závěr.

V rádiovém bloku Audio, Math Education & Programming Animation Software hovoří Grant Sanderson na téma Math Education & Programming Animation Software, slyšte na www.youtube.com/watch?v=pHwzy7Y3Qa0, je zmínka o YouTuberovi Numberphileovi. Ten si půjčil problém (Rug Puzzle, autor Andris Bērziņš). Problém zní: Kolik je možné napočítat trojúhelníků? Dodávám: Lze je všechny vykreslit?



Na řešení se neřívejte...



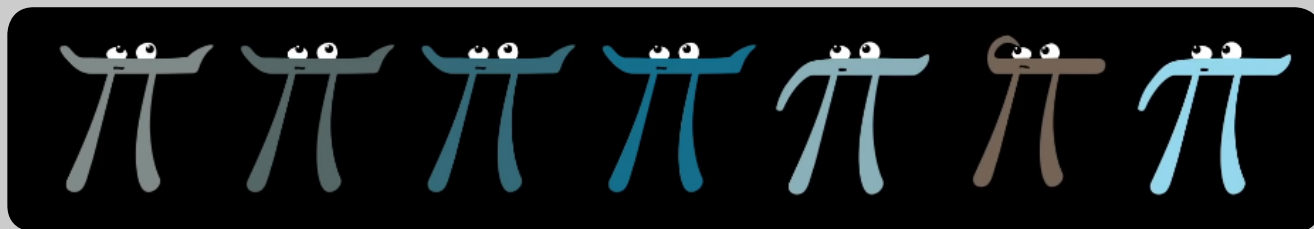
Řešení Bena Eatera (2018) přes JavaScript:

<https://github.com/beneater/rug-puzzle>

Řešení Klebera Lopese da Silvy (2018), JavaScript; jen výpočet: <https://github.com/kleber-swf/rug-puzzle>

Děkuji za pozornost! Hezký den!

pavel@striz.cz



Už mi říkají, abych odletěl... Tak letím!