



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

DOKUMENTACE

Samuel Koribanič

Interaktívna hypergeometria

Programovanie I

Praha 2022

1. Popis programu

Program interaktívna geometria slúži na vizualizáciu objektov v rôznych dimenziách. Je v ňom možné vytvoriť si ľubovoľný objekt a následne ho zobraziť na obrazovku. Pri vizualizácii je možné tento objekt otáčať voči ľubovoľnej rovine vytvorenej dvoma osami. Taktiež je tam možnosť vizualizovať rôzne podobjektov, či skupiny podobjektov. Je tam nastaviteľná rýchlosť jednotlivých rotácií, počiatočná odchilka otočenia, veľkosť objektu a farby, ktoré program využíva.

2. Potrebné informácie pre spustenie

Tento zápočtový program je napísaný v programovacom jazyku Python, okrem neho su taktiež potrebné externé knižnice. To menovite knižnica pygame, ktorá slúži na vizualizáciu a taktiež knižnica numpy, ktorá slúži na rýchlejšie výpočty v programe. Pre spustenie programu je potrebné spustiť súbor main.py

3. Ovládanie

Pred spustením samotnej vizualizácie je nutné vytvoriť si objekt, ktorý chceme zobrazovať (Niekoľko základných objektov je v zložke presets). Ten vytvoríme ako textový dokument, v ktorom budeme musieť špecifikovať dimenziu hľadaného objektu a jednotlivé vrcholy tohto objektu. Pre lepšiu vizualizáciu je možné taktiež tam pridať hrany, ktoré sa budú vizualizovať a taktiež rôzne skupiny podobjektov ktoré budeme chcieť zobraziť.

Tento textový dokument by mal mať nasledujúci formát (príklad napr v súbore presets):

```
dimensions:
<počet dimenzcií>

points:
<počet bodov>
<jednotlivé body, ktoré majú súradnice oddelne medzerami>
.
.
.

edges:
<počet hrán>
<jednotlivé hrany, ktoré maju indexy bodov oddelne medzerami>
.
.
.

subobject_manager:
subobject <meno skupiny podobjektov>:
<počet objektov v danej skupine podobjektov>
<jednotlivé skupiny indexov vrcholov podobjektov, oddelne medzerami>
.
.
.
```

Subobjekty fungujú následovne, vytvoríme si skupinu objektov, ktoré majú nejakú spoločnú vlastnosť, napríklad 3d kocky v 4d hyperkocke, na vymedzenie týchto subobjektov potrebujeme, pre každý jeden subobjekt vypísať indexy vrcholov, ktoré sú v danom subobjekte (indexujeme v poradií akom sú definované v časti points). Následne vieme prepínať medzi jednotlivými subobjektami v danej skupine. Takisto vieme pri jednom objekte vytvoriť viacero takýchto skupín (každú z nich začneme ako subobject <meno>).

Po vytvorení objektu, je potrebné taktiež vytvoriť, vizualizér (Niekoľko základných ich je v zložke visualisers). Ten slúži na to, aby sa nejakým konkrétnym spôsobom daný objekt vyzobrazil. Jeho súčasťou je typ projekcie, a to konkrétne buď ortografická alebo perspektívna. Následujú 3 nie nutné časti, a to počiatočne odchylky rotácie, potom rotácie, ktoré prebehnú v každom snímku, a farby, ktoré bude program využívať.

Vizualizer je taktiež vo forme textového dokumentu a mal by mať nasledujúci formát:

```
projection_type:
<orthographic/perspective>

offsets:
<počet počiatočných otočení>
<jednotlivé offsety vo forme <prvá osa hrany> <druhá osa hrany> <uhol v radiánoch>,
kde osi sú značené ako čísla od 0 do dimenzia - 1>
.
.
.

rotations:
<počet rotácií, ktoré prebehnú v každom snímku>
<jednotlivé rotácie vo forme <prvá osa hrany> <druhá osa hrany> <uhol za snímok>>
.
.
.

colors:
<farba pozadia>
<farba hrán daného objektu>
<farba hrán zvýrazneného podobjektu>
```

V bode, ak máme pripravené oba súbory, tak môžeme spustiť aplikáciu, automaticky na začiatku (alebo stlačením klávesy <i>) vyskočí okno na voľbu objektu, ktorý chceme vyzobraziť (čiže nášho prvého textového súboru). Následne vyskočí opäť možnosť výberu súboru, kde musíme vybrať vizualizér. Objekt sa nám následne zobrazí na obrazovke.

Ďalšie pomocné klávesy:

```
<w> prepína objekty v danom type subobjektov v poradií akom boli zadané
<s> prepína objekty v danom type subobjektov v opačnom poradií
<a> prepína skupiny subobjektov v poradií akom boli zadané
<d> prepína skupiny subobjektov v opačnom poradií
<medzerník> pozastaví animácie
<e> posunie objekt do ďalšieho snímku (hodí sa pri pozastavenom objekte)
```

<q> posunie objekt do predchádzajúceho snímku
<esc> ukončí program
<posun kolečkom myše hore> zväčší objekt na obrazovke
<posun kolečkom myše dole> zmenší objekt na obrazovke
<m> vloží do vybraného súboru predlohu prázdneho vizualizéru
<n> vloží do vybraného súboru predlohu prázdneho objektu

4. Programátorská príloha

Program je napísaný vo viacerých logicky oddelených súboroch, v hlavnom súbore main.py sa inicializuje hlavný cyklus a importujú sa zvyšné dôležité časti. To konkrétne zahŕňa súbor input.py, kde je vytvorená funkcia na načítavanie súboru input_config.json, ktorá zvládne spracovať rôzne typy vstupov a je aplikovateľná na rôzne typy aplikácií. Následuje súbor vrt_space.py, v ktorom je hlavná logická časť môjho programu, v ňom sú uložené funkcie na načítanie objektov/vizualizérov a taktiež funkcia na otáčanie a následnú projekciu bodov (matematika za projekciou a rotáciou je v súbore matrix_utils.py), tu využívam knižnicu numpy, vzhľadom na to, že rotácie a projekciu vykonávam pomocou rôznych manipulácií s maticou s bodmi, tak táto knižnica zabezpečuje rýchlejší beh programu. (Tieto manipulácie sú vysvetlené v zdrojoch, ale v podstate spočívajú v tom, že maticu s bodmi prenášobím nejakými inými správne zvolenými maticami). Následne súbor renderer.py slúži na vyzobrazenie všetkého na plochu a súbor window.py slúži na vytvorenie okna, pri týchto posledných dvoch súboroch využívam na grafické zobrazenie knižnicu pygame.

Celý program je štruktúrovaný objektovo. Jednotlivé časti kódu sú oddelené vo svojich vlastných funkciách.

Pri tvorbe objektov, ktoré sú v zložke presets som postupoval pomocou wikipédie, napríklad ako posledný link v zdrojoch, pri každom objekte je na wikipédii niečo ako časť Coordinates, kde je spísané ako vygenerovať dané body objektu, to už sa dá triválne aplikovať do programu. (Ja som využil python, kde mi dané body vygenerovalo)

Pri tvorbe projekcií a rotácií som využil nasledujúce zdroje:

https://en.wikipedia.org/wiki/Orthographic_projection

https://en.wikipedia.org/wiki/3D_projection#Perspective_projection

https://www.academia.edu/43646395/A_Computer_Technique_for_Displaying_n_Dimensional_Hyperobjects

https://en.wikipedia.org/wiki/Rotation_matrix

<https://en.wikipedia.org/wiki/600-cell>